

SIEMENS

SIMATIC

S7 Программируемый контроллер S7-1200

Системное руководство

Предисловие	
Обзор продукта	1
Новые функции	2
Программный пакет STEP 7	3
Монтаж	4
Основы ПЛК	5
Конфигурирование устройств	6
Основы программирования	7
Базовые инструкции	8
Расширенные инструкции	9
Технологические инструкции	10
Коммуникации	11
Веб-сервер	12
Коммуникационный процессор и Modbus TCP	13
TeleService коммуникации (SMTP email)	14
Инструменты режима он-лайн и диагностики	15
Технические данные	A
Расчет баланса мощностей	B
Номера для заказа	C
Замена устройств и совместимость запасных частей	D

Правовая справочная информация

Система предупреждений

Настоящее руководство содержит указания, которые Вы должны соблюдать для Вашей личной безопасности и для предотвращения материального ущерба. Указания по Вашей личной безопасности выделены предупреждающим треугольником, общие указания по предотвращению материального ущерба не имеют этого треугольника. В зависимости от степени опасности, предупреждающие указания представляются в убывающей последовательности следующим образом:

 ОПАСНОСТЬ
означает, что непринятие соответствующих мер предосторожности приводит к смерти или получению тяжелых телесных повреждений.

 ПРЕДУПРЕЖДЕНИЕ
означает, что непринятие соответствующих мер предосторожности может привести к смерти или получению тяжелых телесных повреждений

 ВНИМАНИЕ
означает, что непринятие соответствующих мер предосторожности может привести к получению незначительных телесных повреждений.

ЗАМЕТКА
означает, что несоблюдение соответствующего указания может привести к материальному ущербу.

При возникновении нескольких степеней опасности всегда используется предупреждающее указание, относящееся к наивысшей степени. Если в предупреждении с предупреждающим треугольником речь идет о предупреждении ущерба, причиняемого людям, то в этом же предупреждении дополнительно могут иметься указания о предупреждении материального ущерба.

Квалифицированный персонал

Работать с изделием или системой, описываемой в данной документации, должен только **квалифицированный персонал**, допущенный для выполнения поставленных задач и соблюдающий соответствующие указания документации, в частности, указания и предупреждения по технике безопасности. Квалифицированный персонал в силу своих знаний и опыта в состоянии распознать риски при обращении с данными изделиями или системами и избежать возникающих угроз.

Использование изделий Siemens по назначению

Соблюдайте следующее:

 ПРЕДУПРЕЖДЕНИЕ
Изделия Сименс разрешается использовать только для целей, указанных в каталоге и в соответствующей технической документации. Если предполагается использовать изделия и компоненты других производителей, то обязательным является получение рекомендации и/или разрешения на это от фирмы Сименс. Исходными условиями для безупречной и надежной работы изделий являются надлежащая транспортировка, хранение, размещение, монтаж, оснащение, ввод в эксплуатацию, обслуживание и поддержание в исправном состоянии. Необходимо соблюдать допустимые условия окружающей среды. Обязательно учитывайте указания в соответствующей документации.

Товарные знаки

Все наименования, обозначенные символом защищенных авторских прав ®, являются зарегистрированными товарными знаками компании Siemens AG. Другие наименования в данной документации могут быть товарными знаками, использование которых третьими лицами для их целей может нарушить права владельцев.

Исключение ответственности

Мы проверили содержимое документации на соответствие описанному аппаратному и программному обеспечению. Тем не менее, отклонения не могут быть исключены, в связи с чем мы не гарантируем полное соответствие. Данные в этой документации регулярно проверяются и соответствующие коррективы вносятся в последующие редакции.

Предисловие

Назначение данного руководства

Семейство S7-1200 представляет собой серию программируемых логических контроллеров (ПЛК), с помощью которых можно решать широкий спектр задач автоматизации. Компактная конструкция, низкая стоимость и мощный набор инструкций делают S7-1200 в высшей степени пригодным для множества приложений в области управления. Различные модели S7-1200 и программный пакет STEP 7 на основе Windows обеспечивают гибкость, необходимую Вам для решения задач автоматизации.

Это руководство содержит информацию о монтаже и программировании ПЛК S7-1200, и оно ориентировано на инженеров, программистов и обслуживающий персонал, имеющий общие знания о программируемых логических контроллерах.

Необходимые базовые знания

Для понимания этого руководства необходимы общие знания об автоматизации и программируемых логических контроллерах.

Область применения данного руководства

В настоящем руководстве представлены следующие продукты

- STEP 7 V13 SP1 Basic и Professional (стр. 35)
- S7-1200 CPU версия встроенного ПО V4.1

Полный список продуктов S7-1200, описанных в этом руководстве, вы найдете в технических данных (стр. 1099).

Сертификация, метка CE, C-Tick и другие стандарты

Подробную информацию вы найдете в технических данных (стр. 1099).

Обслуживание и поддержка

В дополнение к нашей документации мы предлагаем наши технические знания в Интернете по адресу: (<http://www.siemens.com/tiaportal>)

Если у вас есть технические вопросы, вам нужно обучение, или вы хотите заказать продукты S7, обратитесь в свое представительство фирмы Сименс. Так как торговые представители фирмы Сименс технически хорошо подготовлены и имеют специальные знания о возможностях использования и процессах, а также о различных продуктах фирмы Сименс, то они могут быстрее всего предложить наиболее эффективные решения любых проблем, с которыми вы можете встретиться.

Документация и информационные материалы

S7-1200 и STEP 7 предлагают разнообразную документацию и другие ресурсы для нахождения требуемой технической информации.

- Системное руководство для программируемого контроллера S7-1200 предоставляет специализированную информацию по эксплуатации, программированию и техническим характеристикам полного семейства продуктов S7-1200. В дополнение к системному руководству Краткий справочник по S7-1200 обеспечивает более общий обзор возможностей семейства S7-1200.

Как системное руководство, так и Краткий справочник доступны как электронные (PDF) руководства. Электронные руководства могут быть загружены с веб-сайта поддержки клиентов и могут также быть найдены на диске документации, который поставляется с каждым ЦПУ S7-1200.

- Он-лайн информационная система для STEP 7 обеспечивает немедленный доступ к концептуальной информации и конкретным инструкциям, которые описывают эксплуатацию и функциональность пакета программирования и базовые режимы работы ЦПУ SIMATIC.
- My Documentation Manager получает доступ к комплекту электронных (PDF) версий документации по SIMATIC, включая Системное руководство, Краткий справочник и Информационную систему STEP 7. С помощью My Documentation Manager Вы можете перетаскивать темы из различных документов, чтобы создать свое собственное руководство.

Входной портал поддержки клиентов (<http://support.automation.siemens.com>) обеспечивает ссылку на My Documentation Manager и mySupport.

- Веб-сайт поддержки клиентов также обеспечивает подкасты, часто задаваемые вопросы и другие полезные документы для S7-1200 и STEP 7. Подкасты используют короткие образовательные видео-презентации, которые фокусируются на определенных функциях или сценариях, чтобы продемонстрировать взаимодействия, удобство и эффективность, предлагаемые STEP 7. Посетите следующие веб-сайты, чтобы получить доступ к коллекции подкастов:
 - Веб-страница для STEP 7 Basic (<http://www.automation.siemens.com/mcms/simaticcontroller-software/en/step7/step7-basic/Pages/Default.aspx>)
 - Веб-страница для STEP 7 Professional (<http://www.automation.siemens.com/mcms/simaticcontroller-software/en/step7/step7-professional/Pages/Default.aspx>)
- Вы можете также ознакомиться или присоединиться к обсуждениям на техническом форуме Service & Support (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid=34612486>) Эти форумы позволят Вам общаться с различными экспертами по продукту.
 - Форум для S7-1200 (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=258&Language=en&onlyInternet=False>).
 - Форум для STEP 7 Basic (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=265&Language=en&onlyInternet=False>).

Информация о безопасности

Siemens предлагает продукты и решения с промышленными функциями безопасности, которые поддерживают безопасную работу производств, решений, машин, оборудования и/или сетей. Они являются важными компонентами в целостной промышленной концепции безопасности. Учитывая это, продукты и решения Siemens подвергаются непрерывным усовершенствованиям. Siemens настоятельно рекомендует Вам регулярно проверять наличие обновлений для продукта.

Для безопасной работы продуктов и решений от Siemens, необходимо принимать подходящие превентивные меры (например, концепция защиты ячейки) и интегрировать каждый компонент в целостную, передовую промышленную концепцию безопасности. Сторонние продукты, которые могут использоваться, нужно также принять во внимание. Вы можете найти больше информации о промышленной безопасности в Интернете (<http://www.siemens.com/industrialsecurity>).

Чтобы получать информацию о вышедших обновлениях продукта, подпишитесь на специфичную для продукта новостную рассылку. Вы можете найти больше информации в Интернете (<http://support.automation.siemens.com>).

Содержание

	Предисловие	3
1	Обзор продукта.....	25
1.1	Введение в ПЛК S7-1200.....	25
1.2	Расширение возможностей ЦПУ.....	28
1.3	HMI панели серии Basic	30
2	Новые функции.....	31
3	Программный пакет STEP 7	35
3.1	Системные требования	36
3.2	Различные представления, чтобы сделать работу проще.....	37
3.3	Простые в использовании инструменты.....	38
3.3.1	Вставка инструкций в Вашу пользовательскую программу.....	38
3.3.2	Доступ к инструкциям из панели инструментов "Избранное".....	38
3.3.3	Создание сложного уравнения с помощью простой инструкции	39
3.3.4	Добавление входов или выходов в LAD или FBD инструкцию.....	41
3.3.5	Расширяемые инструкции	41
3.3.6	Выбор версии для инструкции	42
3.3.7	Изменение визуального представления и конфигурирование STEP 7.....	42
3.3.8	Перетаскивание между редакторами	43
3.3.9	Изменение рабочего режима ЦПУ	44
3.3.10	Изменение типа вызова DB.....	45
3.3.11	Временное отключение устройств от сети	46
3.3.12	Виртуальное отключение устройств из конфигурации.....	47
3.4	Совместимость с предыдущими версиями.....	48
4	Монтаж.....	49
4.1	Рекомендации по монтажу S7-1200 устройств	49
4.2	Баланс мощности	52
4.3	Процедуры монтажа и демонтажа	54
4.3.1.	Монтажные размеры для S7-1200 устройств	54
4.3.2.	Монтаж и демонтаж ЦПУ.....	58
4.3.3.	Монтаж и демонтаж сигнальной, коммуникационной или батарейной плат.....	60
4.3.4.	Монтаж и демонтаж сигнального модуля	62
4.3.5.	Установка и снятие коммуникационного модуля или коммуникационного процессора	64
4.3.6.	Снятие и повторная установка соединителя клеммной колодки в S7-1200	65
4.3.7.	Установка и снятие кабеля расширения.....	66
4.3.8.	TS (TeleService) адаптер	68
4.3.8.1	Подключение адаптера удаленного управления	68
4.3.8.2	Установка SIM-карты	70
4.3.8.3	Установка узла TS адаптера на DIN рейке.....	71
4.3.8.4	Установка TS адаптера на щите.....	72

4.4	Рекомендации по проводному монтажу.....	73
5	Основы ПЛК.....	79
5.1	Обработка программы пользователя	79
5.1.1.	Рабочие режимы ЦПУ	83
5.1.2.	Процедура цикла сканирования в режиме RUN	87
5.1.3.	Организационные блоки (ОВ)	88
5.1.3.1.	ОВ обработки программного цикла	88
5.1.3.2.	ОВ обработки запуска.....	89
5.1.3.3.	ОВ обработки прерывания с задержкой по времени	89
5.1.3.4.	ОВ обработки циклического прерывания	90
5.1.3.5.	ОВ обработки аппаратных прерываний	90
5.1.3.6.	ОВ обработки прерываний ошибки времени.....	91
5.1.3.7.	ОВ обработки диагностического прерывания	92
5.1.3.8.	ОВ обработки удаления или вставки модулей.....	95
5.1.3.9.	ОВ обработки отказа стойки или станции	96
5.1.3.10.	ОВ обработки прерывания по времени суток	97
5.1.3.11.	ОВ обработки изменения состояния	97
5.1.3.12.	ОВ обработки обновления.....	98
5.1.3.13.	ОВ обработки специфического для профиля прерывания	98
5.1.3.14.	Приоритеты обработки событий и организации очереди	98
5.1.4.	Контроль и конфигурирование времени цикла.....	101
5.1.5.	Память ЦПУ	103
5.1.5.1.	Системная и тактовая память	105
5.1.6.	Диагностический буфер	107
5.1.7.	Часы времени суток	107
5.1.8.	Конфигурирование выходов для перехода из RUN в STOP	108
5.2	Хранение данных, области памяти, ввод/вывод и адресация	109
5.2.1.	Доступ к данным в S7-1200	109
5.3	Обработка аналоговых значений.....	115
5.4	Типы данных.....	117
5.4.1.	Типы данных Bool, Byte, Word и DWord	118
5.4.2.	Целочисленные типы данных	119
5.4.3.	Типы данных для действительных чисел с плавающей точкой	119
5.4.4.	Данные даты и времени.....	120
5.4.5.	Символьные и строковые типы данных.....	122
5.4.6.	Тип данных Array	124
5.4.7.	Структурированные типы данных	125
5.4.8.	ПЛК тип данных.....	125
5.4.9.	Типы данных указателей.....	125
5.4.9.1.	Тип данных указателя "Pointer"	126
5.4.9.2.	Тип указателя " ANY".....	127
5.4.9.3.	Тип указателя "Variant"	128
5.4.10.	Обращение к "срезу" тегированного типа данных.....	129
5.4.11.	Обращение к тегу с использованием АТ наложения.....	130
5.5	Использование карты памяти	132
5.5.1.	Вставка карты памяти в ЦПУ	133
5.5.2.	Конфигурирование параметра запуска ЦПУ прежде, чем копировать проект на карту памяти.....	135
5.5.3.	Карта передачи	135

5.5.4.	Карта программирования	138
5.6	Восстановление из состояния потерянного пароля	144
6	Конфигурирование устройств	145
6.1	Вставка ЦПУ	146
6.2	Выгрузка конфигурации подключенного ЦПУ	148
6.3	Добавление модулей в конфигурацию	150
6.4	Управление конфигурацией	151
6.4.1.	Преимущества и практическое использование управления конфигурацией	151
6.4.2.	Конфигурирование центральной установки и дополнительных модулей	151
6.5	Замена устройства	161
6.6	Конфигурирование работы ЦПУ	162
6.6.1.	Обзор	162
6.6.2.	Конфигурирование времен фильтра для цифрового входа	164
6.6.3.	Захват импульсов	166
6.7	Конфигурирование параметров модулей	167
6.8	Конфигурирование ЦПУ для коммуникаций	169
7	Основы программирования	171
7.1	Рекомендации по разработке ПЛК системы	171
7.2	Структурирование Вашей пользовательской программы	173
7.3	Использование блоков для структурирования Вашей программы	175
7.3.1.	Организационный блок (OB)	176
7.3.2.	Функция (FC)	178
7.3.3.	Функциональный блок (FB)	178
7.3.4.	Блок данных (DB)	180
7.3.5.	Создание повторно используемого кодовых блоков	181
7.3.6.	Передача параметров в блоки	182
7.4	Понимание согласованности данных	185
7.5	Язык программирования	186
7.5.1.	Лестничная логика (LAD)	186
7.5.2.	Функционально-блочная диаграмма (FBD)	187
7.5.3.	SCL	188
7.5.3.1.	Программный редактор SCL	188
7.5.3.2.	SCL выражения и операции	189
7.5.3.3.	Индексная адресация с помощью PEEK и POKE инструкций	193
7.5.3.4.	EN и ENO для LAD, FBD и SCL	195
7.6	Защита	197
7.6.1.	Защита доступа для ЦПУ	197
7.6.2.	Защита "ноу-хау"	200
7.6.3.	Защита от копирования	201
7.7	Загрузка элементов Вашей программы	203
7.8	Выгрузка из он-лайн ЦПУ	204
7.8.1.	Сравнение он-лайн ЦПУ и офф-лайн ЦПУ	204

7.9	Отладка и тестирование программы	205
7.9.1.	Наблюдение и изменение данных в ЦПУ	205
7.9.2.	Таблицы наблюдения и таблицы форсирования	206
7.9.3.	Перекрестная ссылка, чтобы показать использование	206
7.9.4.	Структура вызовов, чтобы исследовать иерархию вызовов	208
8	Базовые инструкции	209
8.1	Операции бинарной логики	209
8.1.1.	Инструкции бинарной логики	209
8.1.2.	Инструкции установки и сброса	212
8.1.3.	Инструкции выделения положительного и отрицательного фронта	215
8.2	Таймеры	218
8.3	Операции счета	226
8.4	Операции сравнения	232
8.4.1.	Инструкции сравнения значений	232
8.4.2.	Инструкции IN_Range (Значение в диапазоне) и OUT_Range (Значение вне диапазона)	233
8.4.3.	Инструкции OK (Проверить на достоверность) и NOT_OK (Проверить на недостоверность)	234
8.4.4.	Инструкции сравнения типов данных Variant и Array	235
8.4.4.1	Инструкции проверки равенства и неравенства	235
8.4.4.2	Функции сравнения с нулевым значением	236
8.4.4.3	IS_ARRAY (Проверка на массив)	236
8.5	Арифметические функции	237
8.5.1	Инструкция CALCULATE (вычисление)	237
8.5.2	Инструкции сложения, вычитания, умножения и деления	238
8.5.3	Инструкция MOD (возвращает остаток от деления)	240
8.5.4	Инструкция NEG (Создать двоичное дополнение)	241
8.5.5	Инструкции INC (Инкремент) и DEC (Декремент)	242
8.5.6	Инструкция ABS (Получить абсолютное значение)	243
8.5.7	Инструкции MIN (Вычислить минимум) и MAX (Вычислить максимум)	244
8.5.8	Инструкция LIMIT (Задать предел)	245
8.5.9	Экспоненциальные, логарифмические и тригонометрические инструкции	246
8.6	Операции пересылки	248
8.6.1.	Инструкции MOVE (переслать значение), MOVE_BLK (переслать блок), UMOVE_BLK (переслать блок без прерываний), и MOVE_BLK_VARIANT (переслать блок)	248
8.6.2.	Инструкция Deserialize	251
8.6.3.	Инструкция Serialize	255
8.6.4.	Инструкции FILL_BLK (Заполнить блок) и UFILL_BLK (Заполнить блок без прерываний)	258
8.6.5.	Инструкция SWAP (Обмен байтов)	259
8.6.6.	Инструкции чтения / записи памяти	260
8.6.6.1	Инструкции PEEK и POKE (только SCL)	260
8.6.6.2	Инструкции чтения и записи big endian и little endian (SCL)	262
8.6.7.	Инструкции для типа данных Variant	264
8.6.7.1	VariantGet (Чтение значения тега VARIANT)	264
8.6.7.2	Инструкция VariantPut (Запись значения тега VARIANT)	265
8.6.7.3	Инструкция CountOfElements (Количество элементов массива)	266
8.6.8.	Унаследованные инструкции	267
8.6.8.1	Инструкции FieldRead (Чтение поля) и FieldWrite (Запись поля)	267

8.7	Операции преобразования.....	269
8.7.1.	Инструкция CONV (Преобразовать значение)	269
8.7.2.	Инструкции преобразования для SCL	270
8.7.3.	Инструкции ROUND (Округлить численное значение) и TRUNC (Отбросить дробную часть численного значения)	273
8.7.4.	Инструкции CEIL и FLOOR (Округление до следующего большего или меньшего целого числа).....	274
8.7.5.	Инструкции SCALE_X (Масштабирование) и NORM_X (Нормализация).....	275
8.7.6.	Инструкции преобразования типа Variant.....	278
8.7.6.1	Инструкция VARIANT_TO_DB_ANY (Преобразование из VARIANT в DB_ANY)	278
8.7.6.2	Инструкция DB_ANY_TO_VARIANT (Преобразование из DB_ANY в VARIANT)	279
8.8	Операции управления обработкой программы	281
8.8.1.	Инструкции JMP (Переход, если RLO = 1), JMPN (Переход, если RLO = 0), и Label (Метка перехода)	281
8.8.2.	Инструкция JMP_LIST (Определение списка переходов)	282
8.8.3.	Инструкция SWITCH (Распределение переходов)	283
8.8.4.	Инструкция RET (Завершение блока).....	285
8.8.5.	Инструкция ENDIS_PW (Активировать/деактивировать пароли ЦПУ).....	286
8.8.6.	Инструкция RE_TRIGR (Рестарт времени контроля цикла)	288
8.8.7.	Инструкция STP (Окончание работы программы)	289
8.8.8.	Инструкции GET_ERROR и GET_ERROR_ID (Локальное получение ошибки и ее ID).....	290
8.8.9.	Инструкция RUNTIME (Оценка работы программы).....	293
8.8.10.	Операторы управления программой в SCL	295
8.9	Логические операции	305
8.9.1.	Логические операции AND, OR, и XOR.....	305
8.9.2.	Инструкция INV (Инверсия).....	306
8.9.3.	Инструкции DECO (Расшифровать) и ENCO (Закодировать)	306
8.9.4.	Инструкции SEL (Выбор), MUX (Мультиплексирование), и DEMUX (Демультимплексирование).....	308
8.10	Сдвиг и циклический сдвиг	312
8.10.1.	Инструкции SHR (Сдвиг вправо) и SHL (Сдвиг влево)	312
8.10.2.	Инструкции ROR (Циклический сдвиг вправо) и ROL (Циклический сдвиг влево)	313
9	Расширенные инструкции.....	315
9.1	Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)	315
9.1.1.	Инструкции "Date" (Дата) и "time-of-day" (Время дня)	315
9.1.2.	Функции часов (Clock)	318
9.1.3.	Структура данных TimeTransformationRule.....	322
9.1.4.	Инструкция SET_TIMEZONE (установка часового пояса)	323
9.1.5.	Инструкция RTM (счетчик рабочего времени).....	324
9.2	Строка и символ	326
9.2.1.	Обзор строковых данных.....	326
9.2.2.	Инструкция S_MOVE (Перемещение символов строки).....	326
9.2.3.	Инструкции преобразования строки	327
9.2.3.1.	Инструкции S_CONV, STRG_VAL и VAL_STRG (Конвертирование в(из) строку символов или числовые значения).....	327
9.2.3.2.	Инструкции Strg_TO_Chars и Chars_TO_Strg (Преобразование в(из) строку символов и массив CHAR)	337
9.2.3.3.	Инструкции ATH и HTA (Преобразование в(из) ASCII-строку и шестнадцатеричное число)	339

9.2.4.	Строковые инструкции	342
9.2.4.1.	Инструкция MAX_LEN (Максимальная длина символьной строки).....	342
9.2.4.2.	Инструкция LEN (Определение фактической длины строки символов).....	343
9.2.4.3.	Инструкция CONCAT (Объединение строк символов)	343
9.2.4.4.	Инструкции LEFT, RIGHT и MID (Чтение подстрок в строке символов)	344
9.2.4.5.	Инструкция DELETE (Удаление символов из строки символов).....	346
9.2.4.6.	Инструкция INSERT (Вставка символов в строку символов)	347
9.2.4.7.	Инструкция REPLACE (Замена символов в строке символов)	348
9.2.4.8.	Инструкция FIND (Поиск символов в строке символов).....	349
9.3	Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс).....	350
9.3.1.	Инструкции распределенного ввода/вывода.....	350
9.3.2.	Инструкции RDREC и WRREC (чтение/запись записей данных)	351
9.3.3.	Инструкция RALRM (Получение прерывания).....	354
9.3.4.	Параметр STATUS для RDREC, WRREC и RALRM	358
9.3.5.	Инструкции DPRD_DAT и DPWR_DAT (Чтение/запись консистентных данных ведомых DP-устройств).....	362
9.3.6.	Инструкция DPNRM_DG (чтение диагностических данных ведомого DP-устройства)	365
9.4	Прерывания	368
9.4.1.	Инструкции ATTACH и DETACH (подключение и отключение ОБ и прерывания)	368
9.4.2.	Циклические прерывания.....	372
9.4.2.1.	Инструкция SET_CINT (Установка параметров циклических прерываний).....	372
9.4.2.2.	Инструкция QRY_CINT (Параметры запроса циклического прерывания).....	374
9.4.3.	Прерывания по времени суток.....	375
9.4.3.1.	SET_TINTL (установка параметров прерывания "time of day")	376
9.4.3.2.	Инструкция CAN_TINT (Отмена прерывания "time of day").....	377
9.4.3.3.	Инструкция ACT_TINT (Активация прерывания по времени дня).....	378
9.4.3.4.	QRY_TINT (Запрос состояния прерывания по времени дня).....	379
9.4.4.	Время задержки прерываний	380
9.4.5.	Инструкции DIS_AIRT и EN_AIRT (Задержка/включение выполнения прерываний с высоким приоритетом и события, вызывающие асинхронные ошибки)	383
9.5	Диагностика (PROFINET или PROFIBUS).....	384
9.5.1.	Диагностические инструкции	384
9.5.2.	Диагностические события для распределенного ввода/вывода	384
9.5.3.	Инструкция LED (Считывание состояния LED индикаторов)	385
9.5.4.	Инструкция DeviceStates	386
9.5.4.1.	Примеры конфигураций DeviceStates	388
9.5.5.	Инструкция ModuleStates	392
9.5.5.1.	Примеры конфигурации ModuleStates	394
9.5.6.	Инструкция GET_DIAG (Чтение диагностической информации)	398
9.5.7.	Инструкция Get_IM_Data (Чтение идентификационных и эксплуатационных данных)	404
9.6	Импульсы.....	406
9.6.1.	Инструкция CTRL_PWM (Широтно-импульсная модуляция)	406
9.6.2.	Работа импульсных выходов	407
9.6.3.	Конфигурирование импульсного канала для PWM	409
9.7	Рецептуры и записи данных.....	411
9.7.1.	Рецептуры	411
9.7.1.1.	Обзор рецептов	411
9.7.1.2.	Примеры рецептов	412
9.7.1.3.	Инструкция RecipeExport (Экспорт рецептуры) для передачи рецептурных данных	415
9.7.1.4.	Пример программы для работы с рецептурами.....	419

9.7.2.	Записи данных (Data log).....	422
9.7.2.1	Структура журнала записей данных	422
9.7.2.2	Программные инструкции для управления записями данных	423
9.7.2.3	Работа с журналами записей данных	434
9.7.2.4	Ограничение размера файлов журналов записей данных	436
9.7.2.5	Пример программы записи данных	439
9.8	Управление блоком данных	443
9.8.1	Инструкции READ_DBL и WRIT_DBL (Чтение запись блока данных в загрузочной области памяти)	443
9.9	Обработка адресов	447
9.9.1.	Инструкция GEO2LOG (Определение аппаратного идентификатора слота)	447
9.9.2.	Инструкция LOG2GEO (Определение слота по аппаратному идентификатору).....	449
9.9.3.	Инструкция IO2MOD (Определение аппаратного идентификатора по адресу ввода/вывода).....	451
9.9.4.	Инструкция RD_ADDR (Определение адреса ввода/вывода по аппаратному идентификатору)	452
9.9.5.	Тип системных данных GEOADDR	453
9.10	Основные коды ошибок для "Расширенных" инструкций.....	455
10	Технологические инструкции	457
10.1	Высокоскоростной счетчик.....	457
10.1.1.	Инструкция CTRL_HSC (Управление высокоскоростным счетчиком)	457
10.1.2.	Инструкция CTRL_HSC_EXT (Управление высокоскоростным счетчиком (расширенное)) ...	460
10.1.3.	Работа высокоскоростного счетчика	463
10.1.4.	Конфигурирование высокоскоростного счетчика	470
10.2	ПИД-регулирование	472
10.2.1.	Вставка ПИД-инструкции и технологического объекта.....	474
10.2.2.	Инструкция PID_Compact	476
10.2.3.	Параметр ErrorBit инструкции PID_Compact	480
10.2.4.	Параметр Warning инструкции PID_Compact	482
10.2.5.	Инструкция PID_3Step	483
10.2.6.	Параметр ErrorBit инструкции PID_3Step	490
10.2.7.	Параметр Warning инструкции PID_3Step	492
10.2.8.	Инструкция PID_Temp	493
10.2.8.1.	Работа PID_Temp регулятора	497
10.2.8.2.	Каскадное включение регуляторов	499
10.2.9.	Параметр ErrorBit инструкции PID_Temp	502
10.2.10.	Параметр Warning инструкции PID_Temp	504
10.2.11.	Конфигурирование регуляторов PID_Compact и PID_3Step	505
10.2.12.	Конфигурирование регулятора PID_Temp	508
10.2.13.	Ввод в эксплуатацию регуляторов PID_Compact и PID_3Step	522
10.2.14.	Ввод в эксплуатацию регулятора PID_Temp	524
10.3	Управление перемещением.....	533
10.3.1.	Фазирование.....	538
10.3.2.	Конфигурирование импульсного генератора	540
10.3.3.	Управление перемещением с разомкнутой обратной связью	541
10.3.3.1.	Конфигурирование оси.....	541
10.3.3.2.	Ввод в эксплуатацию.....	545
10.3.4.	Управление перемещением с замкнутой обратной связью	550
10.3.4.1.	Конфигурирование оси.....	550

10.3.5.	Конфигурирование TO_CommandTable_PTO.....	557
10.3.6.	Работа управления перемещением для S7-1200.....	560
10.3.6.1.	Выходы ЦПУ, используемые для управления перемещением.....	560
10.3.6.2.	Аппаратные и программные ограничители для управления перемещением.....	562
10.3.6.3.	Перемещение в начальное положение.....	565
10.3.6.4.	Ограничение рывков.....	570
10.3.7.	Инструкции управления перемещением.....	571
10.3.7.1.	Обзор MC-инструкций.....	571
10.3.7.2.	Инструкция MC_Power (Освободить/блокировать ось).....	572
10.3.7.3.	Инструкция MC_Reset (Подтвердить ошибку).....	575
10.3.7.4.	Инструкция MC_Home (Перевести ось в начальное положение).....	576
10.3.7.5.	Инструкция MC_Halt (Приостановить ось).....	578
10.3.7.6.	Инструкция MC_MoveAbsolute (Позиционировать ось абсолютно).....	580
10.3.7.7.	Инструкция MC_MoveRelative (Позиционировать относительно).....	582
10.3.7.8.	Инструкция MC_MoveVelocity (Переместить ось с заданной скоростью)у.....	584
10.3.7.9.	Инструкция MC_MoveJog (Перемещать ось в старт-стопном режиме).....	587
10.3.7.10.	Инструкция MC_CommandTable (Выполнение команд оси как последовательности перемещений).....	589
10.3.7.11.	Инструкция MC_ChangeDynamic (Изменить динамические настройки для оси).....	592
10.3.7.12.	Инструкция MC_WriteParam (Записать параметры технологического объекта).....	594
10.3.7.13.	Инструкция MC_ReadParam (считать параметры технологического объекта).....	596
10.3.8.	Контроль активных команд.....	598
10.3.8.1.	Контроль MC-инструкций с помощью выходного параметра "Done".....	598
10.3.8.2.	Контроль инструкции MC_Velocity.....	602
10.3.8.3.	Контроль инструкции MC_MoveJog.....	606
11	Коммуникации.....	611
11.1	Соединения асинхронной передачи данных V4.1.....	613
11.2	PROFINET.....	616
11.2.1.	Создание сетевого соединения.....	618
11.2.2.	Конфигурирование локального/партнерского пути соединения.....	619
11.2.3.	Назначение адресов Интернет протокола (IP).....	622
11.2.3.1.	Назначение IP адресов программатору и сетевым устройствам.....	622
11.2.3.2.	Проверка IP-адреса Вашего программатора.....	624
11.2.3.3.	Назначение IP-адреса ЦПУ он-лайн.....	625
11.2.3.4.	Конфигурирование IP-адреса для ЦПУ в проекте.....	626
11.2.4.	Тестирование сети PROFINET.....	629
11.2.5.	Определение Ethernet (MAC) адреса в ЦПУ.....	630
11.2.6.	Конфигурирование синхронизации протокола сетевого времени.....	632
11.2.7.	Время запуска, присвоение имени и назначение адреса PROFINET- устройству.....	633
11.2.8.	Открытые пользовательские коммуникации.....	634
11.2.8.1.	Протоколы.....	634
11.2.8.2.	TCP и ISO on TCP.....	635
11.2.8.3.	Коммуникационные услуги и используемые номера портов.....	636
11.2.8.4.	Ситуативный режим.....	637
11.2.8.5.	ID соединений для инструкций открытых пользовательских коммуникаций.....	637
11.2.8.6.	Параметры для PROFINET-соединения.....	641
11.2.8.7.	Инструкции TSEND_C и TRCV_C.....	644
11.2.8.8.	Унаследованные инструкции TSEND_C и TRCV_C.....	655
11.2.8.9.	Инструкции TCON, TDISCON, TSEND и TRCV.....	663
11.2.8.10.	Унаследованные инструкции TCON, TDISCON, TSEND и TRCV.....	672
11.2.8.11.	Инструкция T_RESET (Разорвать и восстановить существующее соединение).....	681

11.2.8.12.	Инструкция T_DIAG (Проверяет состояние соединения и считывает информацию) ...	683
11.2.8.13.	Инструкция TMAIL_C (Послать электронное письмо, используя Ethernet-интерфейс ЦПУ).....	688
11.2.8.14.	UDP	697
11.2.8.15.	TUSEND и TURCV	698
11.2.8.16.	T_CONFIG	704
11.2.8.17.	Общие параметры для инструкций.....	712
11.2.9.	Коммуникации с программатором.....	714
11.2.9.1.	Установка аппаратного коммуникационного соединения	714
11.2.9.2.	Конфигурирование устройств.....	715
11.2.9.3.	Назначение адресов интернет протокола (IP)	715
11.2.9.4.	Тестирование Вашей PROFINET-сети	715
11.2.10.	Коммуникации HMI-ПЛК	716
11.2.10.1.	Конфигурирование логических сетевых соединений между двумя устройствами	717
11.2.11.	Коммуникации ПЛК-ПЛК.....	717
11.2.11.1.	Конфигурирование логических сетевых соединений между двумя устройствами	718
11.2.11.2.	Конфигурирование локального/партнерского пути соединения между двумя устройствами	719
11.2.11.3.	Конфигурирование параметров передачи (отправки) и приема	719
11.2.12.	Конфигурирование ЦПУ и PROFINET IO устройства	722
11.2.12.1.	Добавление PROFINET IO устройства	722
11.2.12.2.	Конфигурирование логического сетевого соединения между ЦПУ и PROFINET IO устройством	723
11.2.12.3.	Назначение имен ЦПУ и устройству.....	723
11.2.12.4.	Назначение адресов интернет протокола (IP).....	724
11.2.12.5.	Конфигурирование длительности IO цикла	724
11.2.13.	Конфигурирование ЦПУ и PROFINET I-устройства.....	726
11.2.13.1.	Функциональность I-устройства.....	726
11.2.13.2.	Свойства и преимущества I-устройств	727
11.2.13.3.	Характеристики I-устройства	728
11.2.13.4.	Обмен данными между IO-системами верхнего и нижнего уровней.....	730
11.2.13.5.	Конфигурирование I-устройства	732
11.2.14.	Устройства совместного использования	734
11.2.14.1.	Функциональные возможности устройства совместного использования.....	734
11.2.14.2.	Пример: Конфигурирование совместно используемого устройства (GSD конфигурация)	737
11.2.14.3.	Пример: Конфигурирование I-устройства, как совместно используемого устройства	743
11.2.15.	Диагностика	751
11.2.16.	Инструкции для распределенного ввода-вывода.....	751
11.2.17.	Диагностические инструкции.....	751
11.2.18.	Диагностические события для распределенного ввода-вывода	751
11.3	PROFIBUS.....	752
11.3.1.	Коммуникационные службы PROFIBUS-CM.....	754
11.3.2.	Ссылка на руководства пользователя для PROFIBUS-CM.....	755
11.3.3.	Конфигурирование ведущего и ведомого DP устройств.....	755
11.3.3.1.	Добавление CM 1243-5 (ведущее устройство DP) модуля и ведомого устройства DP.....	755
11.3.3.2.	Конфигурирование логических сетевых соединений между двумя PROFIBUS устройствами	756
11.3.3.3.	Назначение PROFIBUS адресов для CM 1243-5 модуля и ведомого устройства DP	756
11.3.4.	Инструкции для распределенного ввода-вывода.....	758
11.3.5.	Диагностические инструкции.....	758
11.3.6.	Диагностические события для распределенного ввода-вывода	758

11.4	AS-i.....	759
11.4.1	Конфигурирование ведущего и ведомого AS-i устройств.....	760
11.4.1.1	Добавление ведущего AS-i устройства CM 1243-2 и ведомого AS-i устройства.....	760
11.4.1.2	Конфигурирование логических сетевых соединений между двумя AS-i устройствами.....	761
11.4.1.3	Конфигурирование свойств ведущего AS-i устройства CM1243-2.....	761
11.4.1.4	Назначение AS-i адреса ведомому AS-i устройству.....	762
11.4.2	Обмен данными между пользовательской программой и ведомыми AS-i устройствами.....	765
11.4.2.1	Базовая конфигурация в STEP 7.....	765
11.4.2.2	Конфигурирование ведомых устройств с помощью STEP 7.....	766
11.4.3	Инструкции для распределенного ввода-вывода.....	768
11.4.4	Работа с AS-i он-лайн инструментами.....	768
11.5	S7 коммуникации.....	770
11.5.1	Инструкции GET и PUT (чтение и запись из удаленного ЦПУ).....	770
11.5.2	Создание S7-соединения.....	774
11.5.3	Конфигурирование пути соединения локальный/партнерский между двумя устройствами.....	775
11.5.4	Назначение параметра соединения GET/PUT.....	776
11.5.4.1	Параметры соединения.....	777
11.5.4.2	Конфигурирование S7-соединения ЦПУ-ЦПУ.....	779
12	Веб-сервер.....	785
12.1	Активация веб-сервера.....	787
12.2	Конфигурирование пользователей веб-сервера.....	789
12.3	Доступ к веб-страницам от ПК.....	791
12.4	Доступ к веб-страницам из мобильного устройства.....	792
12.5	Использование CP модуля для доступа к веб-страницам.....	793
12.6	Стандартные веб-страницы.....	794
12.6.1.	Компоновка стандартных веб-страниц.....	794
12.6.2.	Авторизация и привилегии пользователей.....	796
12.6.3.	Страница Introduction.....	800
12.6.4.	Страница Start.....	801
12.6.5.	Страница Identification.....	802
12.6.6.	Страница Diagnostic Buffer.....	803
12.6.7.	Страница Module Information.....	804
12.6.8.	Страница Communication.....	807
12.6.9.	Страница Variable Status.....	808
12.6.10.	Страница File Browser.....	810
12.7	Пользовательские веб-страницы.....	813
12.7.1.	Создание HTML-страниц.....	814
12.7.2.	AWP-команды, поддерживаемые веб-сервером S7-1200.....	815
12.7.2.1.	Чтение переменных.....	817
12.7.2.2.	Запись переменных.....	818
12.7.2.3.	Чтение специальных переменных.....	820
12.7.2.4.	Запись специальных переменных.....	821
12.7.2.5.	Использование псевдонимов для ссылки на переменные.....	823
12.7.2.6.	Определение типов перечислений.....	823
12.7.2.7.	Ссылка на переменные ЦПУ с типом перечислений.....	824
12.7.2.8.	Создание фрагментов.....	826

12.7.2.9.	Импортирование фрагментов.....	827
12.7.2.10.	Объединение определений	827
12.7.2.11.	Обработка имен тегов, содержащих специальные символы	828
12.7.3.	Конфигурирование использования пользовательских веб-страниц.....	830
12.7.4.	Программирование инструкции WWW для пользовательских веб-страниц.....	831
12.7.5.	Загрузка программных блоков в ЦПУ	832
12.7.6.	Доступ к пользовательским веб-страницам.....	833
12.7.7.	Ограничения, связанные с пользовательскими веб-страницами	833
12.7.8.	Пример пользовательской веб-страницы	835
12.7.8.1.	Веб-страница для контроля и управления ветряной турбиной.....	835
12.7.8.2.	Чтение и отображение данных контроллера.....	837
12.7.8.3.	Использование типа перечисления	838
12.7.8.4.	Запись введенных пользователем данных в контроллер	839
12.7.8.5.	Запись специальной переменной.....	840
12.7.8.6.	Справка: Распечатка HTML-кода веб-страницы удаленного контроля ветряной турбины.....	841
12.7.8.7.	Пример конфигурирования веб-страницы в STEP 7	846
12.7.9.	Создание пользовательских веб-страниц на нескольких языках.....	847
12.7.9.1.	Создание структуры папок.....	847
12.7.9.2.	Программирование переключателя языка	847
12.7.9.3.	Конфигурирование STEP 7 для использования многоязыковой структуры страницы	851
12.7.10.	Расширенное управление пользовательской веб-страницей	851
12.8	Ограничения	856
12.8.1.	Ограничения функциональности, когда интернет-опции отключают JavaScript.....	857
12.8.2.	Ограничения функциональности, когда интернет-опции отключают куки.....	858
12.8.3.	Импортирование сертификата безопасности Сименс	858
12.8.4.	Импорт журналов данных в CSV формате в non-USA/UK версии Microsoft Excel	860
13	Коммуникационный процессор и Modbus TCP	861
13.1	Применение последовательных коммуникационных интерфейсов.....	861
13.2	Смещение и концевая нагрузка в сетевом RS485-разъеме.....	862
13.3	Коммуникации точка-к-точке (PtP)	864
13.3.1.	Конфигурирование коммуникационных портов	865
13.3.1.1.	Организация управления потоком.....	867
13.3.2.	Конфигурирование параметров передачи (отправки) и приема.....	868
13.3.2.1.	Конфигурирование параметров передачи (отправки).....	868
13.3.2.2.	Конфигурирование параметров приема.....	869
13.3.3.	Инструкции точка-к-точке.....	877
13.3.3.1.	Общие параметры для инструкций точка-к-точке	877
13.3.3.2.	Инструкция Port_Config (динамическое конфигурирование коммуникационных параметров).....	879
13.3.3.3.	Инструкция Send_Config (динамическое конфигурирование параметров последовательной передачи).....	882
13.3.3.4.	Инструкция Receive_Config (динамическое конфигурирование параметров последовательного приема).....	884
13.3.3.5.	Инструкция Send_P2P (отправить данные из приемного буфера)	889
13.3.3.6.	Инструкция Receive_P2P (разрешить прием сообщений).....	893
13.3.3.7.	Инструкция Receive_Reset (удалить приемный буфер)	895
13.3.3.8.	Инструкция Signal_Get (запросить RS-232 сигналы).....	895
13.3.3.9.	Инструкция Signal_Set (установить RS-232 сигналы).....	897
13.3.3.10.	Get_Features	898
13.3.3.11.	Set_Features	899

13.3.4.	Программирование PtP-коммуникаций	900
13.3.4.1.	Архитектура опроса	901
13.3.5.	Пример: Коммуникации точка-к-точке	902
13.3.5.1.	Конфигурирование коммуникационного модуля	903
13.3.5.2.	Рабочие режимы RS422 и RS485	905
13.3.5.3.	Создание STEP 7 программы	908
13.3.5.4.	Конфигурирование эмулятора терминала	909
13.3.5.5.	Обработка примера программы	910
13.4	Коммуникация через универсальный последовательный интерфейс (USS)	911
13.4.1.	Обзор	911
13.4.2.	Выбор версии USS-инструкций	914
13.4.3.	Требования к использованию USS-протокола	915
13.4.4.	USS-инструкции	918
13.4.5.	Коды состояния USS	925
13.4.6.	Общие требования USS по настройке привода	927
13.4.7.	Пример: Общее USS-подключение и настройка привода	927
13.5	Коммуникации по Modbus	931
13.5.1	Обзор Modbus RTU и TCP коммуникационных инструкций V13	931
13.5.2	Modbus TCP	934
13.5.2.1.	Обзор	934
13.5.2.2.	Выбор версии Modbus TCP инструкций	935
13.5.2.3.	Modbus TCP инструкции	935
13.5.2.4.	Примеры Modbus TCP	950
13.5.3	Modbus RTU	955
13.5.3.1.	Обзор	955
13.5.3.2.	Выбор версии инструкций Modbus RTU	957
13.5.3.3.	Modbus RTU инструкции	958
13.5.3.4.	Примеры Modbus RTU	973
13.6	Унаследованные PtP коммуникации (только CM/CB 1241)	977
13.6.1.	Унаследованные инструкции "точка-к-точке"	977
13.6.1.1.	Инструкция PORT_CFG (Конфигурировать коммуникационные параметры динамически)	977
13.6.1.2.	Инструкция SEND_CFG (динамическое конфигурирование параметров последовательной передачи)	979
13.6.1.3.	Инструкция RCV_CFG (динамическое конфигурирование параметров последовательного приема)	981
13.6.1.4.	Инструкция SEND_PTP (отправить данные из приемного буфера)	985
13.6.1.5.	Инструкция RCV_PTP (разрешить прием сообщений)	987
13.6.1.6.	Инструкция RCV_RST (Удалить приемный буфер)	989
13.6.1.7.	Инструкция SGN_GET (Запросить RS-232 сигналы)	990
13.6.1.8.	Инструкция SGN_SET (Установить RS-232 сигналы)	991
13.7	Унаследованные USS коммуникации (только CM/CB 1241)	993
13.7.1.	Выбор версии USS инструкций	994
13.7.2.	Требования к использованию USS протокола	995
13.7.3.	Унаследованные USS инструкции	997
13.7.3.1.	Инструкция USS_PORT (редактировать обмен, использующий USS-сеть)	997
13.7.3.2.	Инструкция USS_DRV (обмениваться данными с приводом)	999
13.7.3.3.	Инструкция USS_RPM (Считывание параметров из привода)	1002
13.7.3.3	Инструкция USS_WPM (Изменить параметры в приводе)	1003
13.7.4.	Унаследованные коды состояний USS	1005

13.7.5.	Общие требования унаследованных конфигураций USS по настройке привода.....	1007
13.8	Унаследованные Modbus TCP коммуникации	1008
13.8.1.	Обзор	1008
13.8.2.	Выбор версии Modbus TCP инструкций	1008
13.8.3.	Унаследованные Modbus TCP инструкции.....	1009
13.8.3.1.	Инструкция MB_CLIENT (Выполнять обмен, используя PROFINET, как Modbus TCP клиент)	1009
13.8.3.2.	Инструкция MB_SERVER (выполнять обмен, используя PROFINET, как Modbus TCP сервер).....	1015
13.8.4.	Пример: Несколько TCP соединений для унаследованных MB_SERVER	1021
13.8.5.	Пример: Унаследованный MB_CLIENT 1: Несколько запросов с общим TCP соединением	1022
13.8.6.	Пример: Унаследованный MB_CLIENT 2: Несколько запросов с различными TCP соединениями.....	1023
13.8.7.	Пример: Унаследованный MB_CLIENT 3: Запрос записи образа по выводу	1024
13.8.8.	Пример: Унаследованный MB_CLIENT 4: Координация нескольких запросов	1024
13.9	Унаследованные Modbus RTU коммуникации (только CM/CB 1241)	1026
13.9.1.	Обзор	1026
13.9.2.	Выбор версии Modbus RTU инструкций	1026
13.9.3.	Унаследованные Modbus RTU инструкции.....	1027
13.9.3.1.	Инструкция MB_COMM_LOAD (Сконфигурировать порт на PtP модуле для Modbus RTU).....	1027
13.9.3.2.	Инструкция MB_MASTER (Выполнять обмен, используя PtP порт в качестве ведущего устройства Modbus RTU).....	1030
13.9.3.3.	Инструкция MB_SLAVE (Выполнять обмен, используя PtP порт в качестве ведомого устройства Modbus RTU)	1035
13.9.3.4.	Пример: Программа унаследованного ведущего устройства Modbus RTU.....	1041
13.9.3.5.	Пример: Программа унаследованного ведомого устройства Modbus RTU	1043
13.10	Удаленное управление и обслуживание с помощью CP 1242-7	1044
13.10.1.	Обзор коммуникационных процессоров для телеуправления	1044
13.10.2.	Подключение к GSM-сети	1046
13.10.3.	Приложения для CP 1242-7	1047
13.10.4.	Другие функции CP-1242-7	1048
13.10.5.	Конфигурация и электрические соединения	1048
13.10.6.	Дополнительная информация	1049
13.10.7.	Принадлежности.....	1050
13.10.8.	Ссылка на руководство для GSM-антенны	1051
13.10.9.	Примеры конфигурации для телеуправления.....	1051
14	TeleService коммуникации (SMTP email).....	1057
14.1	Инструкция TM_Mail (послать электронное сообщение).....	1057
15	Инструменты режима он-лайн и диагностики	1065
15.1	Индикаторы состояния	1065
15.2	Переход в режим он-лайн и подключение к ЦПУ	1069
15.3	Присвоение имени PROFINET IO устройству онлайн	1070
15.4	Установка IP адреса и времени суток.....	1072
15.5	Сброс к заводским настройкам	1073

15.6	Обновление встроенного ПО	1074
15.7	Операторская панель он-лайн ЦПУ	1075
15.8	Контроль времени цикла и использование памяти	1075
15.9	Отображение диагностических событий в ЦПУ	1076
15.10	Сравнение офф-лайн и он-лайн ЦПУ	1077
15.11	Контроль и изменение значений в ЦПУ	1078
15.11.1.	Переход в он-лайн, чтобы наблюдать значения в ЦПУ	1079
15.11.2.	Отображение состояния в программном редакторе	1080
15.11.3.	Захват он-лайн значений DB, чтобы переустановить стартовые значения	1080
15.11.4.	Использование таблицы наблюдения, чтобы контролировать и изменять значения в ЦПУ	1081
15.11.4.1.	Использование триггера при наблюдении или изменении ПЛК тегов	1082
15.11.4.2.	Разрешение выходов в режиме STOP	1083
15.11.5.	Форсирование значений в ЦПУ	1084
15.11.5.1.	Использование таблицы форсирования	1084
15.11.5.2.	Действие функции Force	1085
15.12	Загрузка в режиме RUN	1087
15.12.1.	Предпосылки для "загрузки в режиме RUN"	1088
15.12.2.	Изменение Вашей программы в режиме RUN	1089
15.12.3.	Загрузка выбранных блоков	1090
15.12.4.	Загрузка одиночного выбранного блока с ошибкой компиляции в другом блоке	1091
15.12.5.	Изменение и загрузка существующих блоков в режиме RUN	1092
15.12.6.	Реакция системы, если процесс загрузки окончился неудачей	1095
15.12.7.	Факторы для рассмотрения при загрузке в режиме RUN	1095
15.13	Трассировка и запись данных ЦПУ по условиям запуска	1097
A	Технические данные	1099
A.1.	Общие технические данные	1099
A.2.	CPU 1211C	1109
A.2.1.	Общие технические данные и характеристики	1109
A.2.2.	Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1211C	1110
A.2.3.	Цифровые входы и выходы	1113
A.2.4.	Аналоговые входы	1114
A.2.4.1.	Реакция на скачок встроенных аналоговых входов ЦПУ	1115
A.2.4.2.	Время выборки для встроенных аналоговых входов ЦПУ	1115
A.2.4.3.	Диапазоны измерений для аналоговых входов напряжения (ЦПУ)	1115
A.2.5.	Схемы электрических соединений ЦПУ 1211	1116
A.3.	CPU 1212C	1119
A.3.1.	Общие технические данные и характеристики	1119
A.3.2.	Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1212C	1120
A.3.3.	Цифровые входы и выходы	1122
A.3.4.	Аналоговые входы	1124
A.3.4.1.	Реакция на скачок встроенных аналоговых входов ЦПУ	1124
A.3.4.2.	Время выборки для встроенных аналоговых входов ЦПУ	1125
A.3.4.3.	Диапазоны измерений для аналоговых входов напряжения (ЦПУ)	1125
A.3.5.	Схемы электрических соединений ЦПУ 1212	1126
A.4.	CPU 1214C	1129

A.4.1.	Общие технические данные и характеристики.....	1129
A.4.2.	Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1214C.....	1130
A.4.3.	Цифровые входы и выходы	1133
A.4.4.	Аналоговые входы.....	1135
A.4.4.1.	Реакция на скачок встроенных аналоговых входов ЦПУ	1135
A.4.4.2.	Время выборки для встроенных аналоговых входов ЦПУ	1136
A.4.4.3.	Диапазоны измерений для аналоговых входов напряжения (ЦПУ).....	1136
A.4.5.	Схемы электрических соединений ЦПУ 1214.....	1137
A.5.	CPU 1215C.....	1141
A.5.1.	Общие технические данные и характеристики.....	1141
A.5.2.	Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1215C.....	1142
A.5.3.	Цифровые входы и выходы	1145
A.5.4.	Аналоговые входы и выходы	1146
A.5.4.1.	Реакция на скачок встроенных аналоговых входов ЦПУ	1147
A.5.4.2.	Время выборки для встроенных аналоговых входов ЦПУ	1147
A.5.4.3.	Диапазоны измерений для аналоговых входов напряжения (ЦПУ).....	1148
A.5.4.4.	Характеристики аналоговых выходов	1148
A.5.5.	Схемы электрических соединений ЦПУ 1215.....	1150
A.6.	CPU 1217C.....	1155
A.6.1.	Общие технические данные и характеристики.....	1155
A.6.2.	Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1217C.....	1156
A.6.3.	Цифровые входы и выходы	1158
A.6.4.	Аналоговые входы и выходы	1162
A.6.4.1.	Характеристики аналоговых входов	1162
A.6.4.2.	Реакция на скачок встроенных аналоговых входов ЦПУ	1162
A.6.4.3.	Время выборки для встроенных аналоговых входов ЦПУ	1163
A.6.4.4.	Диапазоны измерений для аналоговых входов напряжения (ЦПУ).....	1163
A.6.4.5.	Характеристики аналоговых выходов	1163
A.6.5.	Схемы электрических соединений ЦПУ 1217.....	1165
A.6.6.	CPU 1217C. Подробные сведения и примеры использования дифференциального входа (DI)	1167
A.6.7.	CPU 1217C. Подробные сведения и примеры использования дифференциального выхода (DQ).....	1168
A.7.	Цифровые сигнальные модули (SM)	1169
A.7.1.	Технические данные цифровых входов SM 1221.....	1169
A.7.2.	Технические данные 8-канальных цифровых выходов SM 1222.....	1171
A.7.3.	Технические данные 16-канальных цифровых выходов SM 1222.....	1172
A.7.4.	Технические данные цифровых входов/выходов постоянного тока SM 1223.....	1177
A.7.5.	Технические данные цифровых входов/выходов переменного тока SM 1223.....	1182
A.8.	Аналоговые сигнальные модули (SM)	1185
A.8.1.	Технические данные модулей аналоговых входов SM 1231.....	1185
A.8.2.	Технические данные модулей аналоговых выходов SM 1232.....	1188
A.8.3.	Технические данные модулей аналоговых входов/выходов SM 1234.....	1191
A.8.4.	Реакция на скачок аналоговых входов	1194
A.8.5.	Время выборки и время обновления для аналоговых входов	1194
A.8.6.	Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM)	1194
A.8.7.	Диапазоны аналоговых выходов для напряжения и тока (SB и SM)	1195
A.9.	Сигнальные модули для термопар и термосопротивлений (SM).....	1197
A.9.1.	SM 1231 Thermocouple	1197
A.9.1.1.	Базовый режим работы для термопар.....	1199

A.9.1.2. Таблицы выбора для SM 1231 с подключением термопар	1200
A.9.2. SM 1231 RTD	1202
A.9.2.1. Таблицы выбора для SM 1231 RTD	1205
A.10. Технологические модули.....	1208
A.10.1. SM 1278 4xIO-Link Master	1208
A.10.1.1. Обзор модуля SM 1278 4xIO-Link Master	1211
A.10.1.2. Подключение	1214
A.10.1.3. Параметры / адресное пространство	1216
A.10.1.4. Прерывание, ошибка и системные тревоги.....	1218
A.11. Цифровые сигнальные платы (SB).....	1221
A.11.1. Технические характеристики платы цифровых входов SB 1221 200 kHz.....	1221
A.11.2. Технические характеристики платы цифровых выходов SB 1222 200 kHz	1223
A.11.3. Технические характеристики платы цифровых входов / выходов SB 1223 200 kHz.....	1226
A.11.4. Технические характеристики платы 1223 с двумя входами 24 В пост.т. / двумя выходами 24 В пост.т.	1229
A.12. Аналоговые сигнальные платы (SB).....	1232
A.12.1. Технические характеристики SB 1231 с 1 аналоговым входом	1232
A.12.2. Технические характеристики SB 1231 с 1 аналоговым выходом.....	1234
A.12.3. Диапазоны аналоговых входов и выходов	1236
A.12.3.1. Реакция на скачок аналоговых входов	1236
A.12.3.2. Время выборки и обновления для аналоговых входов	1236
A.12.3.3. Диапазоны измерения аналоговых входов для напряжения и тока (SB и SM)	1236
A.12.3.4. Диапазоны аналоговых выходов для напряжения и тока (SB и SM)	1237
A.12.4. Сигнальные платы для термопар (SB)	1239
A.12.4.1. Технические характеристики SB 1231 с 1 аналоговым входом для термопар	1239
A.12.5. Сигнальные платы для подключения терморезисторов (SB)	1243
A.12.5.1. Технические характеристики SB 1231 с 1 аналоговым входом для терморезисторов ...	1243
A.12.5.2. Таблицы выбора для SB 1231 с подключением терморезисторов	1246
A.13. Батарейная плата BB 1297	1248
A.14. Коммуникационные интерфейсы	1250
A.14.1. PROFIBUS	1250
A.14.1.1. CM 1242-5 ведомое устройство PROFIBUS DP.....	1250
A.14.1.2. Привязка контактов D-sub гнезда модуля CM 1242-5	1251
A.14.1.3. CM 1243-5 ведущее устройство PROFIBUS DP	1252
A.14.1.4. Привязка контактов D-sub гнезда модуля CM 1243-5	1253
A.14.2. CP 1242-7.....	1254
A.14.2.1. CP 1242-7 GPRS.....	1254
A.14.2.2. GSM/GPRS антенна ANT794-4MR	1256
A.14.2.3. Плоская антенна ANT794-3M	1257
A.14.3. Ведущее устройство CM 1243-2 AS-i шины.....	1257
A.14.3.1. Технические характеристики ведущего устройства AS-i шины CM 1243-2	1257
A.14.3.2. Электрические соединения ведущего устройства AS-i шины.....	1259
A.14.4. RS232, RS422 и RS485.....	1260
A.14.4.1. Технические характеристики коммуникационной платы CB 1241 RS485.....	1260
A.14.4.2. Технические характеристики коммуникационного модуля CM 1241 RS232	1263
A.14.4.3. Технические характеристики коммуникационного модуля CM 1241 RS422/485	1264
A.15. Удаленное управление (TS адаптер и модульный TS адаптер).....	1265
A.16. Карта памяти SIMATIC	1266

A.17.	Имитатор входов	1266
A.18.	Модуль потенциометра для S7-1200	1268
A.19.	Кабель для расширения ввода/вывода	1269
A.20.	Сопутствующие продукты	1270
A.20.1.	Модуль питания PM 1207	1270
A.20.2.	Компактный коммутационный модуль CSM 1277	1270
A.20.3.	Коммуникационный модуль CANopen	1271
A.20.4.	Коммуникационный модуль RF120C	1271
B	Расчет баланса мощностей	1273
C	Номера для заказа	1277
C.1	Модули ЦПУ	1277
C.2	Сигнальные модули (SM), сигнальные платы (SB) и батарейные платы (BB)	1278
C.3	Коммуникации	1280
C.4	Отказоустойчивые ЦПУ и сигнальные модули	1281
C.5	Прочие модули	1281
C.6	Карты памяти	1282
C.7	Устройства человеко-машинного интерфейса серии Basic	1282
C.8	Запасные части и другое аппаратное обеспечение	1282
C.9	Средства программирования	1285
D	Замена устройств и совместимость запасных частей	1287
D.1	Комплекты запасных клеммных колодок S7-1200 V3.0 и V4.0	1292

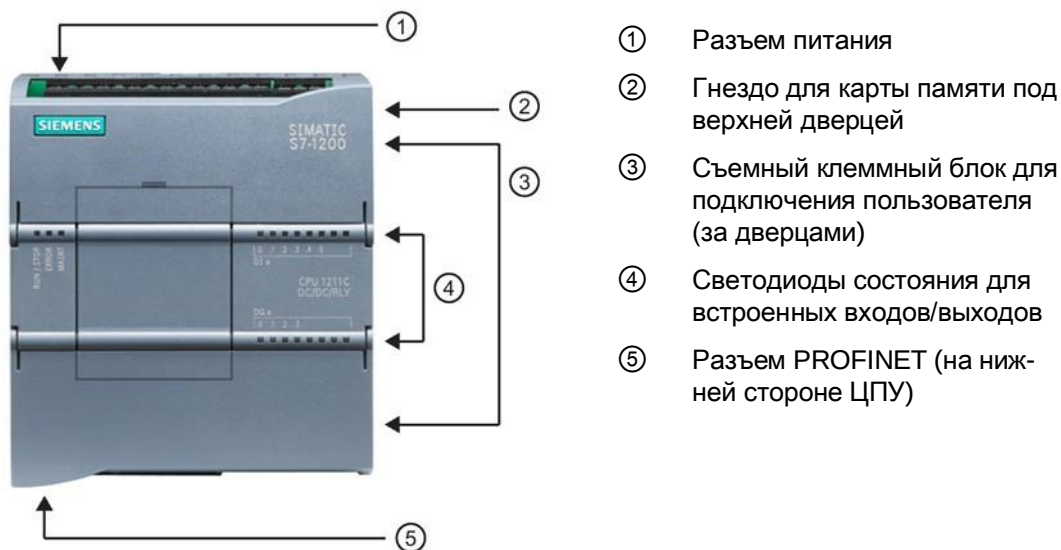
Обзор продукта

1.1 Введение в ПЛК S7-1200

Контроллер S7-1200 обеспечивает гибкость и производительность для управления широким разнообразием устройств в поддержку Ваших потребностей по автоматизации. Компактный дизайн, гибкая конфигурация и мощная система команд объединяются, чтобы сделать S7-1200 совершенным решением для управления широким спектром приложений.

ЦПУ объединяет в себе микропроцессор, интегрированный источник питания, схемы ввода и вывода, встроенный PROFINET, высокоскоростной ввод-вывод управления движением и встроенные аналоговые входы в компактном корпусе для создания мощного контроллера. После того, как Вы загрузите свою программу, ЦПУ содержит логику, требуемую для контроля и управления устройств в Вашем приложении. ЦПУ контролирует входы и изменяет выходы согласно логике Вашей пользовательской программы, которая может включать двоичную логику, подсчет, выдержку по времени, сложные математические операции и коммуникации с другими интеллектуальными устройствами.

ЦПУ обеспечивает порт для коммуникаций по сети PROFINET. Дополнительные модули доступны для коммуникаций по PROFIBUS, GPRS, RS485, RS232, IEC, DNP3 и WDC сетям.



Ряд функций обеспечения безопасности помогают защитить доступ как к ЦПУ, так и к управляющей программе:

- Каждый ЦПУ обеспечивает защиту паролем (стр. 197), которая позволяет Вам конфигурировать доступ к функциям ЦПУ.
- Вы можете использовать "защиту ноу-хау" (стр. 200), чтобы скрыть код внутри определенного блока.
- Вы можете воспользоваться защитой от копирования (стр. 201), чтобы привязать Вашу программу к определенной карте памяти или ЦПУ.

Таблица 1- 1 Сравнение моделей ЦПУ

Свойство		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Физические размеры (мм)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Пользовательская память	Рабочая	50 кБайт	75 кБайт	100 кБайт	125 кБайт	150 кБайт
	Загрузочн.	1 МБайт		4 МБайт		
	Сохраняем.	10 кБайт				
Встроенные входы/выходы	Цифровые	6 входов/ 4 выхода	8 входа/ 6 выходов	14 входов/ 10 выходов		
	Аналоговые	2 входа			2 входа/ 2 выхода	
Размер области отображения	Входы (I)	1024 Байт				
	Выходы (Q)	1024 Байт				
Битовая память (M)		4096 Байт		8192 Байт		
Дополнительные сигнальные модули (SM)		Нет	2	8		
Сигнальная плата (SB), батареинная плата (BB) или коммуни- кационная плата (CB)		1				
Коммуникационный модуль (CM) (стыковка слева)		3				
Высоко- скоростные счетчики	Всего	До 6 сконфигурированных, с использованием любых встроенных или SB входов				
	1 МГц					с Ib.2 по Ib.5
	100/180 кГц	с Ia.0 по Ia.5				
	30/120 кГц	--	с Ia.6 по Ia.7	с Ia.6 по Ib.5		с Ia.6 по Ib.1
	200 кГц³					
Импульсные выходы²	Всего	До 4 сконфигурированных, с использованием любых встроенных или SB входов				
	1 МГц					с Qa.0 по Qa.3
	100 кГц	с Qa.0 по Qa.3				с Qa.4 по Qb.1
	20 кГц	--	с Qa.4 по Qa.5	с Qa.4 по Qb.		--
Карта памяти		Карта памяти SIMATIC (дополнительно)				
Время хранения часов реального времени		20 дней, типовое/12 дней мин. при 40 градусах C (необслуживаемый супер- конденсатор)				
PROFINET Ethernet коммуникационный порт		1			2	
Скорость обработки выражения с действительными числами		2.3 мкс/инструкция				
Скорость обработки операций цифровой логики		0.08 мкс/инструкция				

¹ Более медленная скорость применима, когда счетчик сконфигурирован для квадратурного режима работы.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровой сигнал (SB), чтобы использовать импульсные выходы.

³ Частоты до 200 кГц доступны при использовании SB 1221 DI x 24 VDC 200 kHz и SB 1221 DI 4 x 5 VDC 200 kHz.

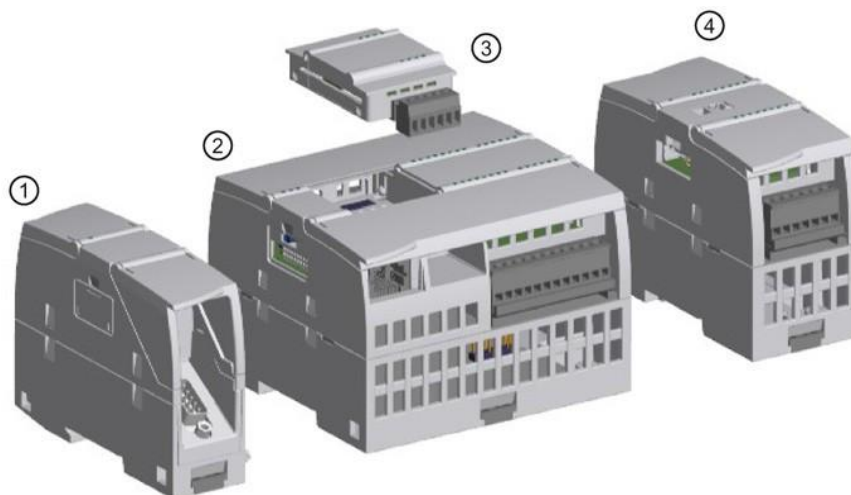
Различные модели ЦПУ обеспечивают разнообразие функций и возможностей, которые помогают Вам создавать эффективные решения для Ваших различных приложений. Для получения дальнейшей информации об определенном ЦПУ, смотри технические характеристики (стр. 1099).

Таблица 1- 2 Блоки, таймеры и счетчики, поддерживаемые S7-1200

Элемент		Описание
Блоки	Тип	ОВ, FB, FC, DB
	Размер	50 кБайт (CPU 1211C) 75 кБайт (CPU 1212C) 100 кБайт (CPU 1214C) 125 кБайт (CPU 1215C) 150 кБайт (CPU 1217C)
	Количество	До 1024 блоков в сумме (OBs + FBs + FCs + DBs)
	Глубина вложения	16, начиная от циклического программного или стартового ОВ; 6, начиная от ОВ прерывания по любому из событий
	Мониторинг	Состояние 2 кодовых блоков может контролироваться одновременно
Блоки ОВ	Цикл программы	Произвольное количество
	Стартовые	Произвольное количество
	Прерывания с задержкой	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени исполнения	1
	Диагностические прерывания	1
	Снятие / установка модулей	1
	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета - SInt, USInt: 3 Байт - Int, UInt: 6 Байт - DInt, UDInt: 12 Байт

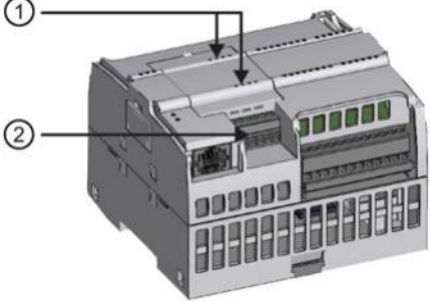
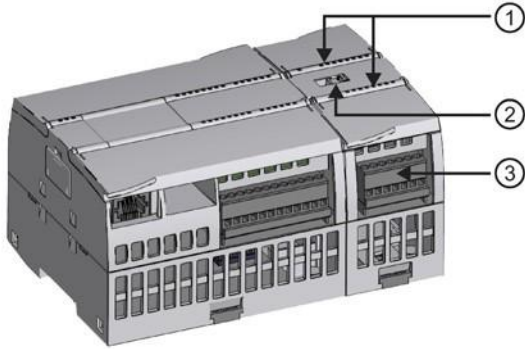
1.2 Расширение возможностей ЦПУ

Семейство S7-1200 предлагает множество модулей и сменных плат для расширения возможностей ЦПУ при помощи дополнительных входов-выходов или других коммуникационных протоколов. Для получения подробной информации об определенном модуле см. технические характеристики (стр.1099).



- ① Коммуникационный модуль (CM) или коммуникационный процессор (CP) (стр. 1250)
- ② ЦПУ (CPU 1211C (стр. 1109), CPU 1212C (стр.1119), CPU 1214C (стр.1129), CPU 1215C (стр.1141), CPU 1217C (стр.1155))
- ③ Сигнальная плата (SB) (цифровая SB (стр.1221), аналоговая SB (стр.1232)), коммуникационная плата (CB) (стр.1260) или батареяная плата (BB) CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C) (стр.1248)
- ④ Сигнальный модуль (SM) (цифровой SM (стр.1169), аналоговый SM (стр.1185), SM для подключения термодатчиков (стр. 1197), RTD SM (стр.1202), технологический SM (стр.1208)

Таблица 1- 3 Модули расширения для S7-1200

Тип модуля	Описание
ЦПУ поддерживает одну съемную плату расширения: • Сигнальная плата (SB) обеспечивает дополнительные входы-выходы для Вашего ЦПУ. SB подключается на передней стороне ЦПУ. • Коммуникационная плата (CB) позволяет Вам добавлять другой коммуникационный порт к Вашему ЦПУ. • Батарейная плата (BB) позволяет Вам обеспечить долговременное резервирование часов реального времени.	 ① Светодиоды состояния на SB ② Съемный соединитель для подключений пользователя
Сигнальные модули (SM) и дополнительная функциональность для ЦПУ. SM подключаются справа от ЦПУ. • Цифровой ввод-вывод • Аналоговый ввод-вывод • Терморезисторы и термодпары • SM 1278 IO-Link Master	 ① Светодиоды состояния ② Передвижной флажок подключения шины ③ Съемный соединитель для подключений пользователя
Коммуникационные модули (CM) и коммуникационные процессоры (CP) добавляют ЦПУ возможности для коммуникаций, например, подключение к PROFIBUS или RS232/RS485 (для PtP, Modbus или USS), а также возможность стать ведущим устройством на шине ASI. CP обеспечивают возможности для других типов коммуникаций, таких как подключение к ЦПУ по GPRS, IEC, DNP3 или WDC сети. • ЦПУ поддерживает до трех CM или CP • Каждый CM или CP подключается слева от ЦПУ (или слева от другого CM или CP)	 ① Светодиоды состояния ② Соединитель для коммуникаций

1.3 HMI панели серии Basic

HMI-панели SIMATIC серии Basic представляют собой устройства с сенсорным экраном для базовых задач контроля и управления оператором. Все панели обладают степенью защиты для IP65 и сертифицированы по CE, UL, cULus и NEMA 4x.

Доступные базовые HMI панели (стр. 1282) описаны ниже:

- KTP400 Basic: 4" Сенсорный экран с 4 конфигурируемыми клавишами, разрешением 480 x 272 и 800 тэгами
- KTP700 Basic: 7" Сенсорный экран с 8 конфигурируемыми клавишами, разрешением 800 x 480 и 800 тэгами
- KTP700 Basic DP: 7" Сенсорный экран с 8 конфигурируемыми клавишами, разрешением 800 x 480 и 800 тэгами
- KTP900 Basic: 9" Сенсорный экран с 8 конфигурируемыми клавишами, разрешением 800 x 480 и 800 тэгами
- KTP1200 Basic: 12" Сенсорный экран с 10 конфигурируемыми клавишами, разрешением 800 x 480 и 800 тэгами
- KTP 1200 Basic DP: 12 Сенсорный экран с 10 конфигурируемыми клавишами, разрешением 800 x 480 и 800 тэгами

Смотри также

Поддержка пользователя (<http://www.siemens.com/automation/>).

Новые функции

Перечисленные ниже функции являются новыми в этой версии:

- Теперь Вы можете реализовать функциональную безопасность, используя аппаратную часть и встроенное ПО отказоустойчивых ЦПУ S7-1200 и сигнальных модулей (SM) в сочетании с программой безопасности, загруженной при помощи программного пакета(ES). Для получения более подробной информации обратитесь к Руководству по функциональной безопасности для S7-1200 (<http://support.automation.siemens.com/WW/view/en/104547552>).
- Симуляция ЦПУ S7-1200 с версией ПО V4.0 и выше: S7-PLCSIM V13 SP1 позволяет Вам протестировать свои программы на симулируемом ПЛК, не используя реальные аппаратные модули. S7-PLCSIM - отдельно установленное приложение, которое работает в комплексе со STEP 7 из TIA Portal. Вы можете сконфигурировать свой ПЛК и любые, относящиеся к нему модули, в STEP 7, создать Вашу прикладную логику и затем загрузить аппаратную конфигурацию и программу в S7-PLCSIM. Вы можете затем использовать инструменты S7-PLCSIM для симуляции и тестирования Вашей программы. Обратитесь к онлайн-справке для получения полной документации для S7-PLCSIM. Обратите внимание на то, что Вы не можете симулировать отказоустойчивые ЦПУ.
- Управление конфигурацией (управление опциями) (стр.151): Вы можете сконфигурировать аппаратные средства для максимальной конфигурации машины включая модули, которые Вы могли бы фактически не использовать во время работы. Конфигурация и назначение этих гибких модулей вводятся с этой версией STEP 7 и S7-1200. Модули, которые Вы назначаете таким образом, не вызовут состояния ошибки, если будут отсутствовать.
- Веб-сервер (стр.785) теперь поддерживает доступ через IP-адрес выбранных (коммуникационный процессор) модулей в локальной стойке, а также через IP-адрес ЦПУ S7-1200.
- Расширенная функциональность управления движением:
 - аналоговое и PROFIdrive подключения;
 - расширенные параметры по модулю и для контура регулирования
- Измерение периода с помощью высоко-скоростных счетчиков (HSC) (стр. 457)
- Повышение производительности компилятора SCL
- Динамическая защита от копирования (стр. 201) привязка программных блоков к обязательному паролю
- Расширенная функциональность PROFINET, включая поддержку совместно используемых устройств (стр. 734).

- Новые программные инструкции:
 - EQ_Type, NE_Type, EQ_ElemType, NE_ElemType (стр.235)
 - IS_NULL, NOT_NULL (стр.236)
 - IS_ARRAY (стр.236)
 - Deserialize (стр.251), Serialize (стр.255)
 - VariantGet (стр.264), VariantPut (стр.265), CountOfElements (стр.266)
 - Variant_to_DB_Any (стр.278), DB_Any_To_Variant (стр.279)
 - GET_IM_DATA (стр.404)
 - RUNTIME (стр.293)
 - GEO2LOG (стр.447), IO2MOD (стр.451)
 - ReadLittle, WriteLittle, ReadBig, WriteBig (только SCL) (стр.262)
 - T_RESET (стр.681), T_DIAG (стр.683) и TMAIL_C (стр.688)
 - PID_Temp (стр.493)
 - Новые Modbus инструкции (стр.931)
 - Новые Point-to-point (PtP) инструкции (стр.864)
 - Новые USS инструкции (стр.911)

Новые модули для S7-1200

Новые модули расширяют возможности ЦПУ S7-1200 и обеспечивают гибкость в удовлетворении Ваших потребностей по автоматизации:

- Промышленные коммуникационные модули дистанционного управления (стр. 1280): Вы можете использовать их в качестве коммуникационных модулей с ЦПУ S7-1200 V4.1.
- Отказоустойчивые ЦПУ и ввод-вывод: есть четыре отказоустойчивых ЦПУ и три отказоустойчивых сигнальных модуля (SM) для S7-1200 версии V4.1 или позднее:
 - CPU 1214FC DC/DC/DC (6ES7 214-1AF40-0XB0)
 - CPU 1214FC DC/DC/RLY (6ES7 214-1HF40-0XB0)
 - CPU 1215FC DC/DC/DC (6ES7 215-1AF40-0XB0)
 - CPU 1215FC DC/DC/RLY (6ES7 215-1HF40-0XB0)
 - SM 1226 F-DI 16 x 24 VDC (6ES7 226-6BA32-0XB0)
 - SM 1226 F-DQ 4 x 24 VDC (6ES7 226-6DA32-0XB0)
 - SM 1226 F-DQ 2 x Relay (6ES7 226-6RA32-0XB0)

Вы можете использовать стандартные сигнальные модули (SM) S7-1200, коммуникационные модули (CM) и сигнальные платы (SB) в одной системе с отказоустойчивыми SM, чтобы добавить прикладные функции управления, не классифицированные по функциональной безопасности. Стандартные SM, которые поддерживаются при совместном использовании с отказоустойчивым SM, имеют заказные номера (6ES7-----32 0XB0) или более поздние.

Замена Вашего ЦПУ V3.0 на ЦПУ V4.1

Если Вы заменяете ЦПУ S7-1200 V3.0 на ЦПУ S7-1200 V4.1, обратите внимание на задокументированные различия (стр. 1287) в версиях и требуемые пользовательские действия.

Программный пакет STEP 7

STEP 7 предлагает удобную для пользователя среду, чтобы разрабатывать, редактировать и осуществлять наблюдение за логикой, необходимой для управления Вашим приложением, включая инструменты для управления и конфигурирования всех устройств в Вашем проекте, таких как устройства HMI и контроллеры. Чтобы помочь Вам найти нужную информацию, STEP 7 предлагает обширную он-лайн справочную систему.

STEP 7 предлагает стандартные языки программирования для удобства и эффективности при разработке управляющей программы для Вашего приложения.

- LAD (многозвенная логическая схема) (стр. 186) является графическим языком программирования. Представление основывается на принципиальных схемах.
- FBD (функционально-блочная диаграмма) (стр. 187) является языком программирования, который основывается на графических логических символах, используемых в Булевой алгебре.
- SCL (структурированный язык управления) (стр. 188) является основанным на тексте, высокоуровневым языком программирования.

При создании кодового блока, Вы выбираете язык программирования, который будет использован в этом блоке. Ваша пользовательская программа может использовать кодовые блоки, созданные на любом из языков программирования.

Примечание

STEP 7 является программным компонентом TIA Portal, предназначенным для программирования и конфигурирования. TIA Portal, в дополнение к STEP 7, включает в себя WinCC для разработки и исполнения в режиме реального времени проекта визуализации процесса, а также он-лайн справку как для WinCC, так и для STEP 7.

3.1 Системные требования

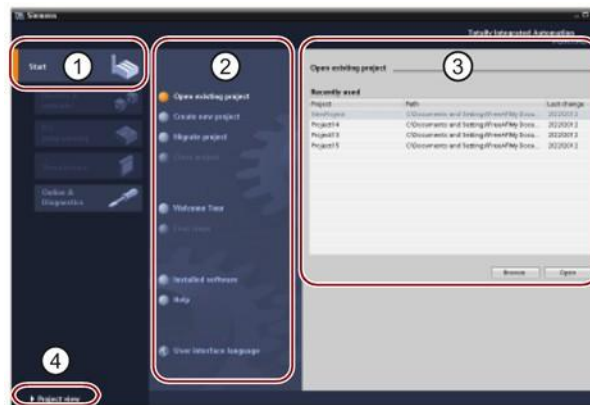
Вы должны устанавливать STEP7 с администраторскими правами.

Таблица 3- 1 Системные требования

Аппаратные средства/ПО	Требования
Тип процессора	Intel® Core™ i5-3320M 3.3 GHz или лучше
ОЗУ	8 GB
Доступное пространство на жестком диске	2 ГБ на системном диске C:\
Операционные системы	Вы можете использовать STEP 7 со следующими операционными системами (как 64-битная, так и 32-битная Windows 7) • Microsoft Windows 7 Home Premium SP1 или выше (только STEP 7 Basic, не поддерживается для STEP 7 Professional) • Microsoft Windows 7 или выше (Professional SP1, Enterprise SP1, Ultimate SP1) • Microsoft Windows 8.1 (только STEP 7 Basic, не поддерживается для STEP 7 Professional) • Microsoft Windows 8.1 (Professional, Enterprise) • Microsoft Server 2008 R2 StdE SP1 (только STEP 7 Professional) • Microsoft Server 2012 R2 StdE
Графический адаптер	32 МБ ОЗУ 24-бита глубина цвета
Разрешение экрана	1920 x 1080 (рекомендуется)
Сеть	20 МБ/с Ethernet или быстрее
Привод оптических дисков	DVD-ROM

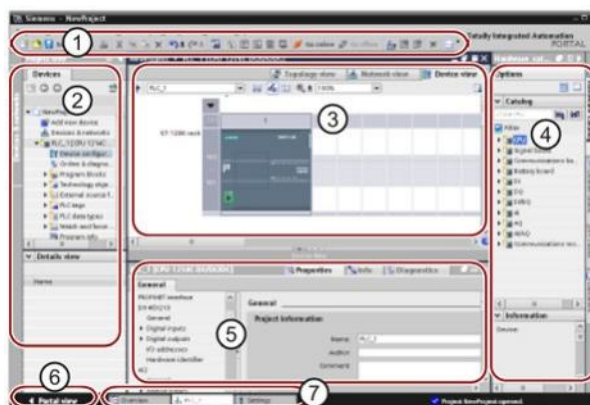
3.2 Различные представления, чтобы сделать работу проще

STEP 7 предлагает удобную для пользователя среду, чтобы разработать логику контроллера, сконфигурировать HMI визуализацию и настроить сетевые коммуникации. Чтобы помочь Вам повысить производительность, STEP 7 предлагает два различных представления проекта: ориентированный на задачу набор порталов, которые организованы на функциональности инструментов (Portal view) или ориентированное на проект представление элементов в проекте (Project view). Выбирайте представление, которое помогает Вам работать наиболее эффективно. Одним щелчком Вы можете переключиться между представлениями Portal view и Project view.



Portal view

- ① Порталы для разных задач
- ② Задачи для выбранного портала
- ③ Панель выбора для выбранного действия
- ④ Переход в Project view



Project view

- ① Меню и панель инструментов
- ② Навигатор проекта
- ③ Рабочая область
- ④ Карты задач
- ⑤ Окно инспектора
- ⑥ Переход в Portal view
- ⑦ Панель редактора

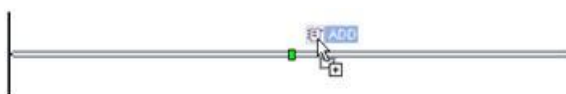
Благодаря наличию всех этих компонентов в одном месте Вы обладаете простым доступом к каждому аспекту Вашего проекта. Например, окно инспектора показывает свойства и информацию об объекте, который Вы выбрали в рабочей области. Поскольку Вы выбираете различные объекты, окно инспектора выводит на экран свойства, которые Вы можете сконфигурировать. Окно инспектора содержит вкладки, которые позволяют Вам видеть диагностическую информацию и другие сообщения.

Отображая все открытые редакторы, панель редактора помогает Вам работать более быстро и эффективно. Чтобы переключиться между открытыми редакторами, просто щелкните по другому редактору. Вы можете также расположить два редактора вместе вертикально или горизонтально. Эта функция позволяет Вам перетаскивать объекты между редакторами.

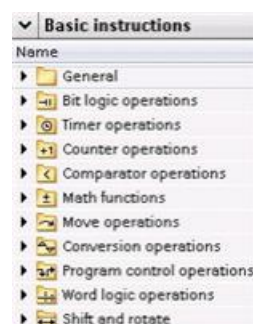
3.3 Простые в использовании инструменты

3.3.1 Вставка инструкций в Вашу пользовательскую программу

STEP 7 предлагает карты задач, которые содержат инструкции для Вашей программы. Инструкции сгруппированы согласно функциям.



Чтобы создать Вашу программу, Вы перетаскиваете инструкции из карты задач в сегмент программы.

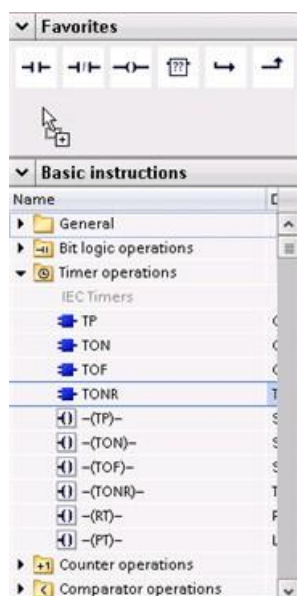


3.3.2 Доступ к инструкциям из панели инструментов "Избранное"

STEP 7 предлагает панель инструментов "Избранное", чтобы предоставить Вам быстрый доступ к инструкциям, которые Вы часто используете. Просто щелкните по значку инструкции для вставки ее в Ваш сегмент!



(Для "Избранного" в дереве инструкций дважды щелкните по значку.)



Вы можете легко настроить "Избранное" путем добавления новых инструкций.

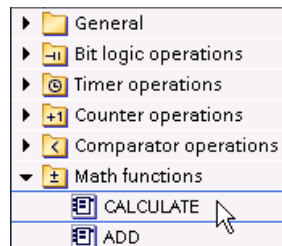
Просто перетащите инструкцию в "Избранное".

После этого инструкция доступна по щелчку!

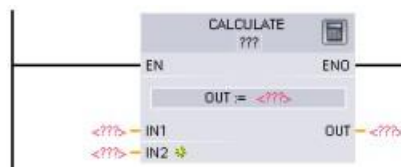


3.3.3 Создание сложного уравнения с помощью простой инструкции

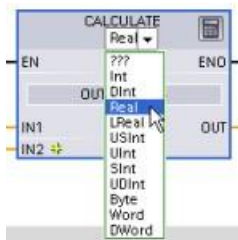
Инструкция Calculate (стр. 237) позволяет Вам создать математическую функцию, которая производит действие над несколькими входными параметрами для получения результата согласно уравнению, которое Вы определяете.



В разделе базовых операций раскройте папку Math functions. Выполните двойной щелчок левой клавишей мыши по инструкции Calculate для вставки ее в Вашу пользовательскую программу.



Несконфигурированная инструкция Calculate предлагает два входных и один выходной параметры.

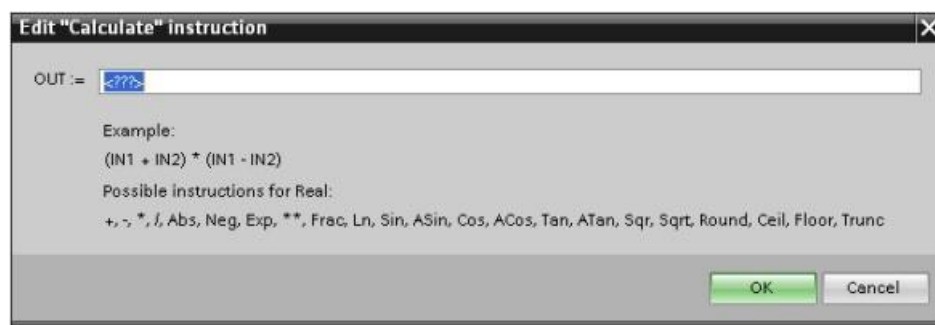


Щелкните по "???" и выберите типы данных для входных и выходных параметров. (Все входные и выходные параметры должны иметь один и тот же тип данных.)

Например, выберите тип данных "Real".



Щелкните по значку "Редактировать выражение" для редактирования.



Для этого примера войдите в следующее уравнение для масштабирования необработанного аналогового значения. ("In" и "Out" обозначения соответствуют параметрам инструкции Calculate.)

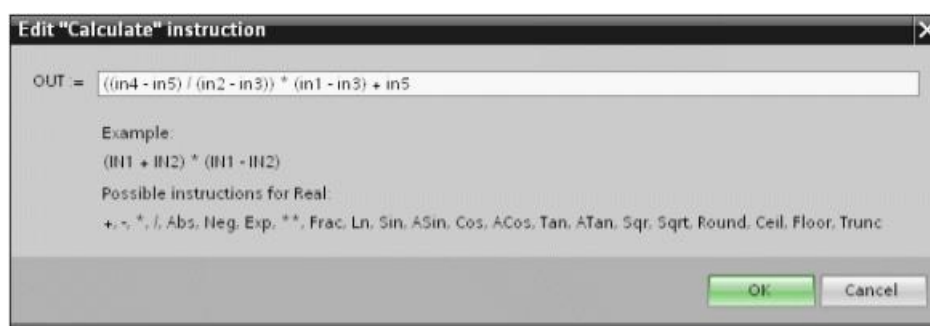
$$\text{Out_value} = ((\text{Out_high} - \text{Out_low}) / (\text{In_high} - \text{In_low})) * (\text{In_value} - \text{In_low}) + \text{Out_low}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

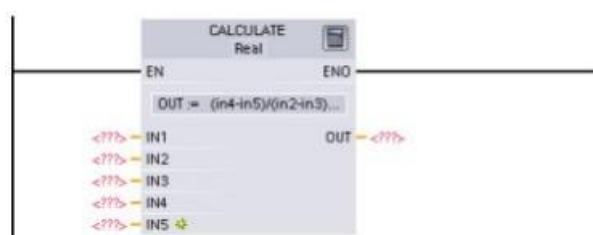
Где:	Out_value	(Out)	Масштабированное выходное значение
	In_value	(in1)	Аналоговое входное значение
	In_high	(in2)	Верхняя граница для масштабированного входного значения
	In_low	(in3)	Нижняя граница для масштабированного входного значения
	Out_high	(in4)	Верхняя граница для масштабированного выходного значения
	Out_low	(in5)	Нижняя граница для масштабированного выходного значения

В окне "Edit Calculate" введите уравнение с именами параметров:

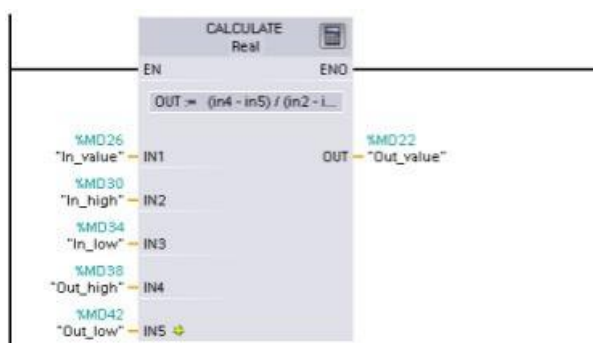
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



Когда Вы щелкаете левой клавишей мыши по "ОК", инструкция Calculate создает входы требуемые для инструкции.



Введите имена тегов для значений, которые соответствуют параметрам.



программируемый контроллер S7-1200

3.3.4 Добавление входов или выходов в LAD или FBD инструкцию

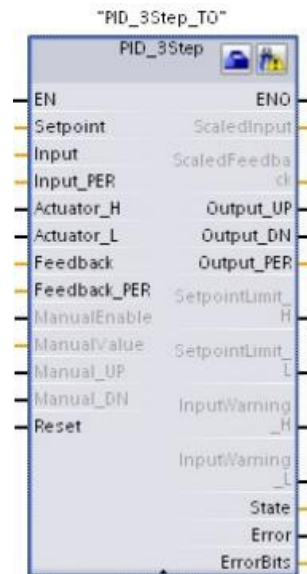
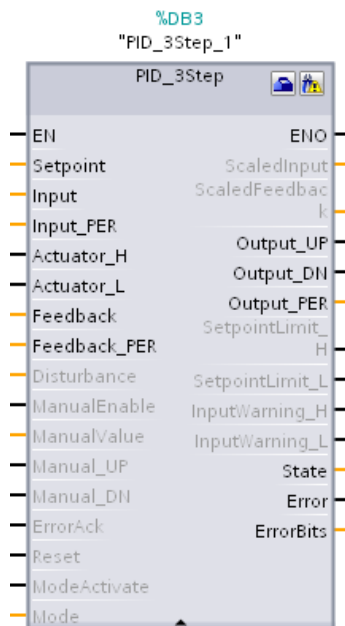


Некоторые инструкции позволяют Вам создавать дополнительные входы или выходы.

- Чтобы добавить вход или выход, щелкните левой клавишей мыши по символу "Создать" или щелкните правой кнопкой мыши по выводу для одного из существующих входных или выходных параметров и выберите команду " Insert input".
- Чтобы удалить вход или выход, щелкните правой кнопкой мыши по выводу для одного из существующих входных или выходных параметров (когда есть больше, чем оригинальные два контакта), и выберите команду " Delete".

3.3.5 Расширяемые инструкции

Некоторые из более сложных инструкций являются расширяемыми, отображая только ключевые входы и выходы. Чтобы показать все входы и выходы, щелкните по стрелке у основания инструкции.

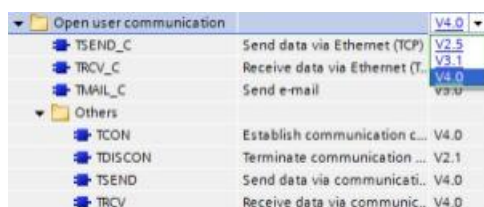


3.3.6 Выбор версии для инструкции

Циклы разработки и выпуска для определенных наборов инструкций (таких как Modbus, ПИД и движение) привели к созданию нескольких рабочих версий для этих инструкций. Чтобы помочь в обеспечении совместимости и миграции старых проектов, STEP 7 позволяет Вам выбирать версию инструкции, вставляемой в Вашу пользовательскую программу.



Щелкните по значку на карте задач с деревом инструкций, чтобы разрешить заголовки и колонки дерева инструкций.

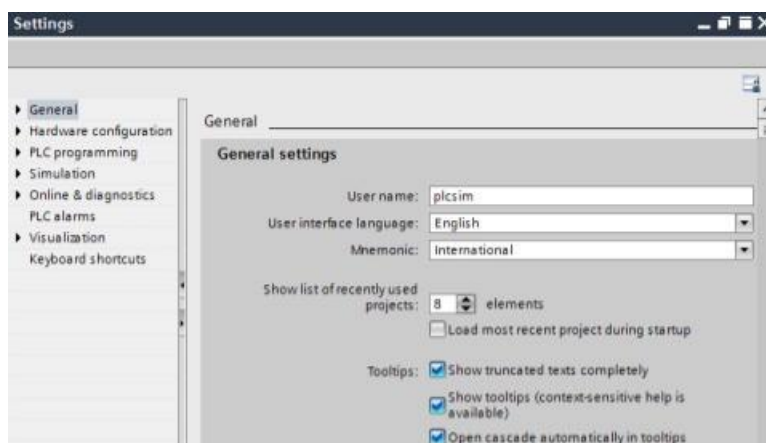


Чтобы изменить версию инструкции, выберите соответствующую версию из раскрывающегося списка.

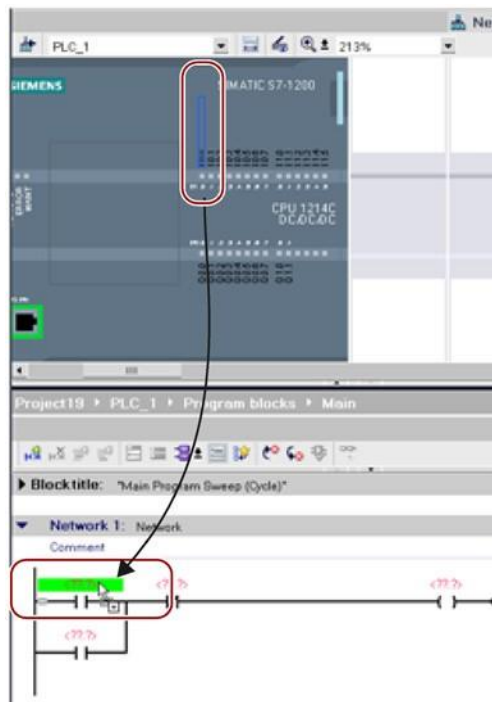
3.3.7 Изменение визуального представления и конфигурирование STEP 7

Вы можете выбрать множество настроек, таких визуальное представление интерфейса, язык или папки для сохранения Вашей работы.

Выберите команду "Settings" в меню "Options", чтобы изменить эти настройки.



3.3.8 Перетаскивание между редакторами

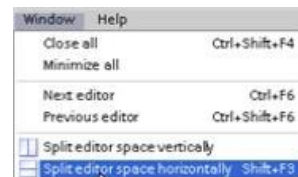


Чтобы помочь Вам выполнить задачи быстро и легко, STEP 7 позволяет Вам перетаскивать элементы из одного редактора в другой. Например, Вы можете перетянуть вход от ЦПУ на адрес инструкции в Вашей пользовательской программе.

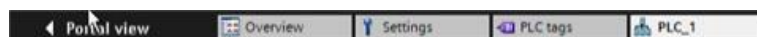
Вы должны увеличить масштаб по крайней мере до 200%, чтобы выбрать входы или выходы ЦПУ.

Заметьте, что имена тегов отображаются не только в таблице ПЛК тегов, но также отображаются на ЦПУ.

Чтобы показать два редактора одновременно используйте команды меню "Split editor" или кнопки на панели инструментов.



Чтобы переключиться между редакторами, которые были открыты, щелкайте левой клавишей мыши по символам в панели редактора.



3.3.9 Изменение рабочего режима ЦПУ

У ЦПУ нет физического переключателя для изменения рабочего режима (STOP или RUN).

Используйте кнопки панели инструментов "Start CPU" и "Stop CPU", чтобы изменить рабочий режим ЦПУ.

Когда Вы настраиваете ЦПУ в конфигурации устройства, Вы определяете в свойствах ЦПУ его поведение при запуске (стр. 162).

Портал "Online and diagnostics" также предлагает пульт управления для изменения рабочего режима ЦПУ в режиме онлайн. Чтобы использовать пульт управления ЦПУ, у Вас должно быть подключение онлайн с ЦПУ. Карта задач "Online tools" отображает пульт управления оператора, где показан рабочий режим ЦПУ онлайн. Пульт управления также позволяет Вам изменять рабочий режим ЦПУ онлайн.

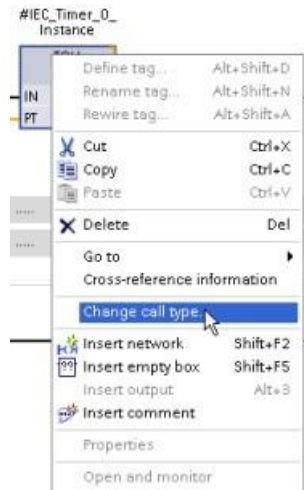


Используйте кнопку на пульте управления, чтобы изменить рабочий режим (STOP или RUN). Пульт управления также имеет кнопку MRES для сброса памяти.

Цвет индикатора RUN/STOP показывает текущий рабочий режим ЦПУ. Желтый цвет указывает на режим STOP, а зеленый - на режим RUN.

Находясь в конфигурации устройства в STEP 7 Вы можете также настроить рабочий режим по умолчанию при подаче питания на ЦПУ (стр. 83).

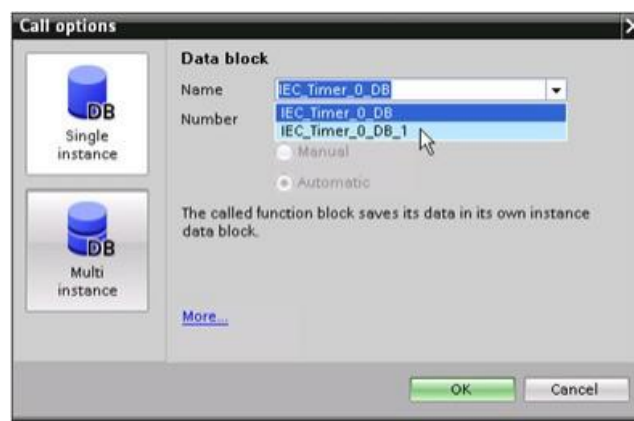
3.3.10 Изменение типа вызова DB



STEP 7 позволяет Вам легко создавать или изменять связь DB с инструкцией или FB, который вызывается в другом FB.

- Вы можете выбирать связь с различным DB.
- Вы можете выбирать связь либо с экземплярным, либо мультиэкземплярным DB.
- Вы можете создавать экземплярный DB (если экземплярный DB отсутствует или недоступен).

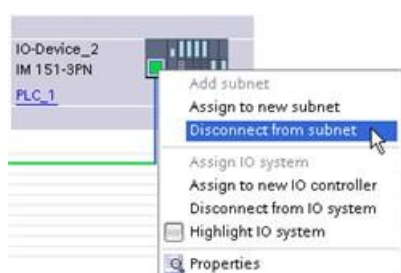
Вы можете получить доступ к команде "Change call type" либо щелкнув правой кнопкой мыши по инструкции или FB в программной редакторе, либо выбирая команду "Block call" из меню "Options".



Диалог "Call options" позволяет Вам выбирать экземплярный или мульти-экземплярный DB. Вы можете также выбрать определенные DB из раскрывающегося списка доступных DB.

3.3.11 Временное отключение устройств от сети

Вы можете отключить отдельные сетевые устройства от подсети. Поскольку конфигурация устройства не удалена из проекта, Вы можете легко восстановить соединение с устройством.



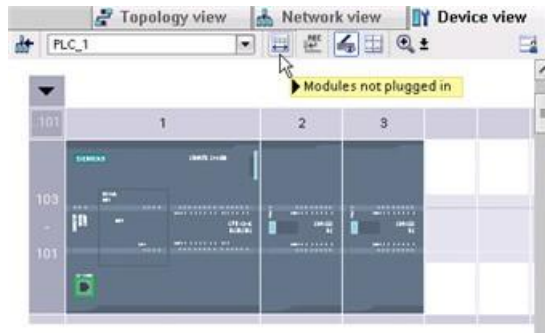
Щелкните правой кнопкой мыши по интерфейсному порту сетевого устройства и выберите команду "Disconnect from subnet" из контекстного меню.

STEP 7 повторно конфигурирует сетевые соединения, но не удаляет отключенное устройство из проекта. Несмотря на то, что сетевое соединение удалено, адреса интерфейсов не изменяются.



Когда Вы загружаете новые сетевые соединения, ЦПУ должен быть переведен в режим STOP. Чтобы выполнить повторное подключение устройства, просто создайте новое сетевое соединение с портом устройства.

3.3.12 Виртуальное отключение устройств из конфигурации



STEP 7 предлагает место для хранения "отключенных" модулей. Вы можете перетянуть модуль со стойки, чтобы сохранить его конфигурацию. Эти отключенные модули сохраняются в Вашем проекте, позволяя Вам повторно вставить модуль в будущем без необходимости повторно конфигурировать параметры.

Одним из вариантов использования этой функции является временное обслуживание. Представьте случай, когда Вы ждете модуль для замены и планируете временно использовать другой модуль. Вы могли бы перетянуть сконфигурированный модуль из стойки в "Unplugged modules" и затем вставить временный модуль.

3.4 Совместимость с предыдущими версиями

STEP 7 V13 SP1 поддерживает конфигурирование и программирование ЦПУ S7-1200 V4.1. Вы можете, однако, загрузить конфигурацию и программы для S7-1200 V4.0 из STEP 7 V13 в ЦПУ S7-1200 V4.1. Ваша конфигурация и программа будут ограничены набором функций и инструкций, которые поддерживаются STEP 7 V13 и S7-1200 V4.0.

Эта совместимость с предыдущими версиями позволяет Вам запускать программы на новых версиях S7-1200 V4.1 ЦПУ, которые Вы ранее разрабатывали и программировали для более старых версий.

4.1 Рекомендации по монтажу S7-1200 устройств

Оборудование S7-1200 разработано для упрощенного монтажа. Вы можете смонтировать S7-1200 на панели или на стандартной рейке, также Вы можете ориентировать S7-1200 горизонтально или вертикально. Небольшой размер S7-1200 позволяет Вам эффективно использовать пространство.

Отказоустойчивые ЦПУ S7-1200 не поддерживают распределенный отказоустойчивый ввод/вывод, подключаемый по PROFIBUS или PROFINET.

Стандарты на электрооборудование классифицируют систему SIMATIC S7-1200 как Открытое Оборудование. Вы должны установить S7-1200 в корпусе, шкафу или электрощитовой. Вы должны разрешить доступ к корпусу, шкафу или электрощитовой только уполномоченному персоналу.

Установка должна обеспечить сухую окружающую среду для S7-1200. Схемы SELV / PELV, как полагают, обеспечивают защиту от поражения током в сухих помещениях.

Установка должна обеспечить механическую защиту и защиту от воздействия окружающей среды, которая одобрена для открытого оборудования в Вашей особой категории местоположения согласно применимым электрическим и строительным нормам и правилам.

Проводящее загрязнение вследствие пыли, влаги и загрязнения воздуха, может вызвать эксплуатационные и электрические отказы в ПЛК.

Если Вы размещаете ПЛК в обстановке, где может присутствовать проводящее загрязнение, ПЛК должен быть защищен корпусом с соответствующей степенью защиты. IP54 - это та степень защиты, которая обычно используется для корпусов электронного оборудования в загрязненной окружающей среде и может подойти для Вашего случая применения.



ПРЕДУПРЕЖДЕНИЕ

Неправильная установка S7-1200 может привести к электрическим отказам или непредсказуемой работе оборудования.

Электрические отказы или непредсказуемая работа машины могут привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Необходимо следовать Всем инструкциям по установке и обслуживанию надлежащей рабочей среды, чтобы обеспечить безопасную эксплуатацию оборудования.

Изолируйте S7-1200 устройства от высокой температуры, высокого напряжения и электрических помех

В качестве основного правила для компоновки устройств Вашей системы, всегда изолируйте устройства, которые вырабатывают высокое напряжение и сильные электрические помехи от низковольтных устройств логического типа, таких как S7-1200.

При разработке компоновки S7-1200 внутри Вашего распределительного щита примите во внимание теплогенерирующие устройства и расположите устройства электронного типа в зонах охлаждения Вашего шкафа. Снижение воздействия высокотемпературной окружающей среды продлит срок службы любого электронного устройства.

Обратите также внимание на прокладку проводки для устройств в распределительном щите. Избегайте размещать низковольтные сигнальные провода и коммуникационные кабели в одном и том же лотке с силовой проводкой переменного тока и мощной, быстро коммутируемой проводкой постоянного тока.

Обеспечьте достаточные зазоры для охлаждения и прокладки проводки

S7-1200 устройства разработаны для естественного охлаждения путем конвекции. Для надлежащего охлаждения Вы должны обеспечить зазор не менее 25 мм сверху и снизу устройств. Кроме того, обеспечьте по крайней мере 25 мм глубины между лицевой панелью модулей и внутренней поверхностью корпуса.



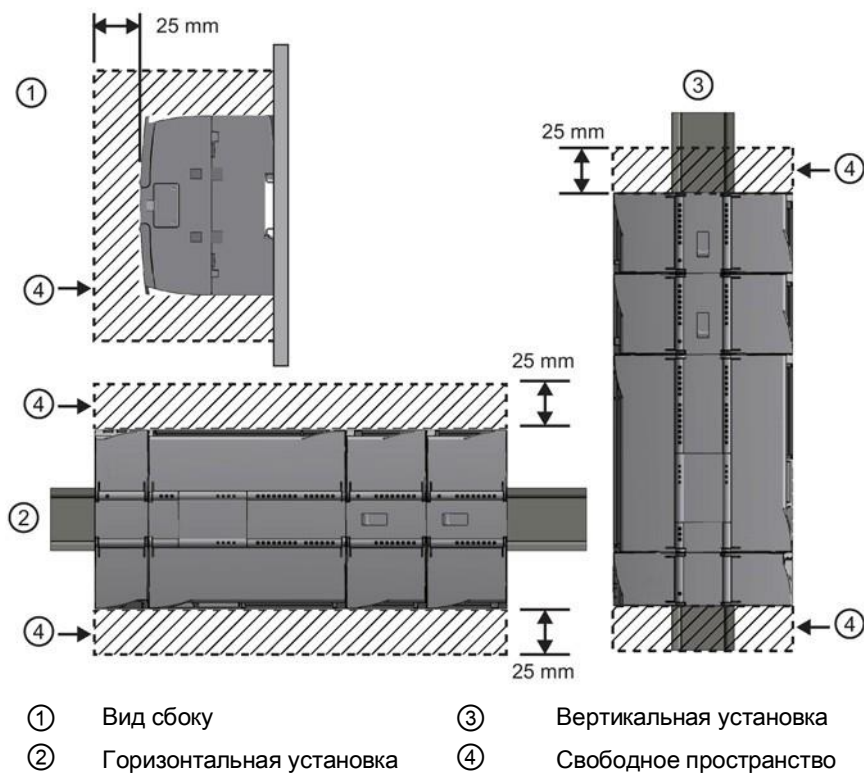
ВНИМАНИЕ

Для вертикальной установки максимальная допустимая температура окружающей среды уменьшена на 10 градусов по Цельсию.

Сориентируйте вертикально установленную систему S7-1200, как показано на следующем рисунке. Обеспечьте правильную установку системы S7-1200.

4.1 Рекомендации по монтажу S7-1200 устройств

Разрабатывая Вашу компоновку системы S7-1200, оставьте достаточные зазоры для коммуникационных кабельных соединений и проводки.



4.2 Баланс мощности

Ваш ЦПУ имеет внутренний источник, который обеспечивает питание ЦПУ, сигнальных модулей, сигнальной платы и коммуникационных модулей, а также других пользовательских потребителей с напряжением 24 В постоянного тока.

Обратитесь к техническим характеристикам (стр. 1099) для получения информации относительно ресурса мощности источника питания логики напряжением 5 В пост.т., поставляемого ЦПУ и потребляемой от источника 5 В пост.т. мощности для сигнальных модулей, сигнальных плат и коммуникационных модулей. Обратитесь к разделу "Расчет баланса мощностей" (стр. 1273), какую мощность (или ток) ЦПУ может поставить для вашей конфигурации.

ЦПУ обеспечивает напряжение питания датчиков 24 В пост.т., которое может использоваться для питания входных каналов, катушек реле в сигнальных модулях или для других потребностей. Если Ваши требования к питанию 24 В пост.т. превышают ресурс источника питания датчиков, то Вы должны добавить внешний источник питания 24 В пост.т. в Вашу систему. Обратитесь к техническим характеристикам (стр. 1099) для получения информации о ресурсе источника питания датчиков 24 В пост.т. для Вашего определенного ЦПУ.

Примечание

Коммуникационному модулю CM 1243-5 (модуль ведущего устройства PROFIBUS) требуется питание от источника питания датчиков 24 В пост.тока в ЦПУ.

Если Вам требуется внешний источник питания 24 В пост.т., убедитесь, что питание не подается параллельно источнику питания датчиков в ЦПУ. Для улучшения защиты от электрических помех рекомендуется, чтобы общие проводники (М) различных источников были соединены.

ПРЕДУПРЕЖДЕНИЕ
Подключение внешнего источника питания 24 В пост.т. параллельно источнику питания датчиков 24 В пост.т. может привести к конфликту между ними, поскольку каждый стремится установить свой собственный предпочтительный уровень выходного напряжения Результатом этого конфликта может стать сокращение срока эксплуатации или немедленный отказ одного или обоих источников с последующей непредсказуемой работой ПЛК системы. Непредсказуемая работа может привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу. Источник пост.т. для питания датчиков и любой внешний источник питания должны подключаться к различным потребителям.

Некоторые входные порты питания 24 В пост.т. в системе S7-1200 связаны с общей логической схемой, соединяющей несколько терминалы М. Например, следующие схемы соединены, когда обозначаются как "не изолированные" в технических характеристиках: источник питания 24 В пост.т. ЦПУ, входное питание для катушек реле сигнального модуля или питание для неизолированного аналогового входа. Все неизолированные клеммы М, должны быть соединены с одним и тем же внешним опорным потенциалом.

 ПРЕДУПРЕЖДЕНИЕ

Подключение неизолированных клемм М к различным опорным потенциалам, вызовет непреднамеренные электрические токи, которые могут привести к ущербу или непредсказуемой работе ПЛК и любом подключенном оборудовании.

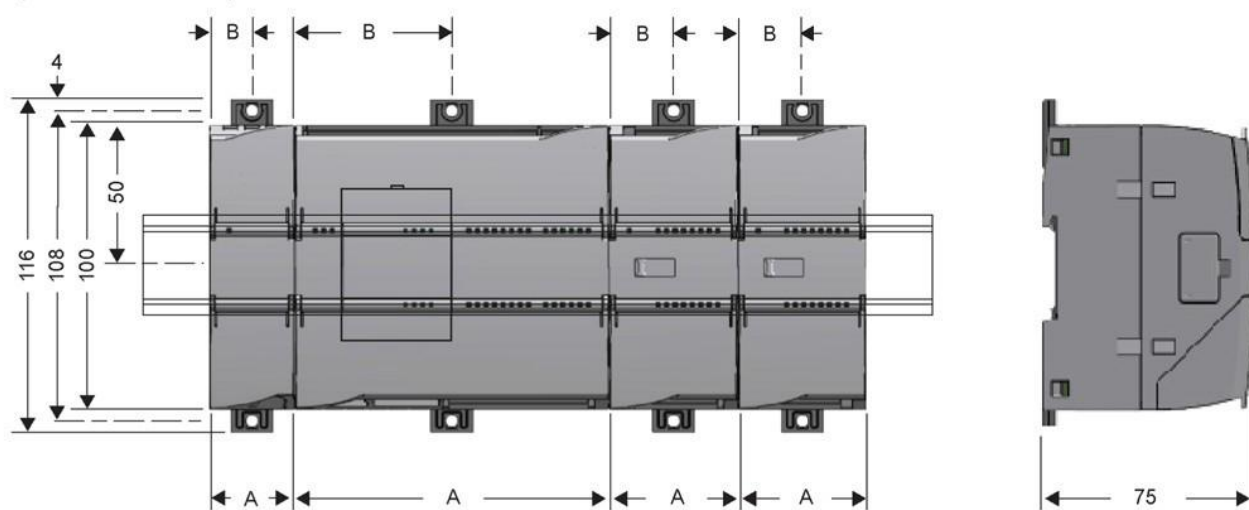
Отказ выполнить эти рекомендации может привести к ущербу или непредсказуемой работе, что могло бы стать причиной смерти, тяжелых телесных повреждений и/или материального ущерба.

Всегда гарантируйте, что все неизолированные клеммы М в системе S7-1200 связаны с одним и тем же опорным потенциалом.

4.3 Процедуры монтажа и демонтажа

4.3.1. Монтажные размеры для S7-1200 устройств

CPU 1211C, CPU 1212C, CPU 1214C
(measurements in mm)



CPU 1215C, CPU 1217C

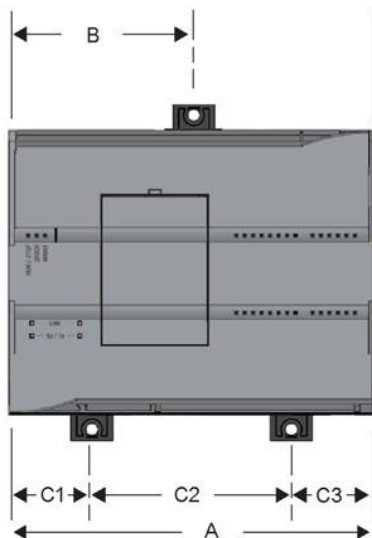


Таблица 4- 1 Монтажные размеры (мм)

S7-1200 устройства		Ширина А (мм)	Ширина В (мм)	Ширина С (мм)
ЦПУ	CPU 1211C и CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (верх)	Основание: C1: 32.5 C2: 65 C3: 32.5
	CPU 1217C	150	75	Основание: C1: 37.5 C2: 75 C3: 37.5
Сигнальные модули	Цифровые 8 и 16 каналов Аналоговые 2, 4 и 8 каналов Для термодпар 4 и 8 каналов Для терморезисторов 4 канала SM 1278 IO Link-Master	45	22.5	--
	Цифровые выходы 8 х реле (с переключ. контактом)	70	35	--
	Аналоговые 16 каналов Для терморезисторов 8 каналов	70	35	--
Коммуникационные интерфейсы	CM 1241 RS232 и CM 1241 RS422/485 CM 1243-5 PROFIBUS master и CM 1242-5 PROFIBUS slave CM 1242-2 AS-i Master CP 1242-7 GPRS V2 CP 1243-7 LTE-EU CP 1243-1 DNP3 CP 1243-1 IEC CP 1243-1 CP1243-1 PCC CP 1243-8 ST7 RF120C	30	15	--
	TS (TeleService) адаптер IE Advanced ¹ TS (Teleservice) адаптер IE Basic ¹ TS адаптер TS модуль	30 30	15 15	-- --

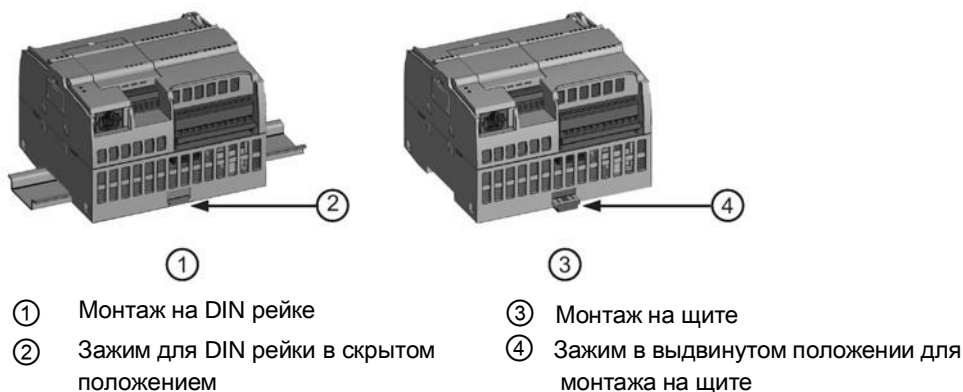
¹ Прежде, чем установить TS (TeleService) адаптер IE Advanced или IE Basic, Вы должны сначала соединить TS адаптер и TS модуль. Полная ширина ("ширина А") составляет 60 мм.

Каждый ЦПУ, сигнальный модуль, коммуникационный модуль и коммуникационный процессор предполагает установку на DIN рейке или в щите. Используйте зажимы для DIN рейки на модуле, чтобы зафиксировать устройство на рейке. Эти зажимы также фиксируются в выдвинутом положении, чтобы обеспечить установку блока непосредственно на щите. Внутренний размер отверстия для DIN зажимов в устройстве составляет 4.3 мм.

Необходимо обеспечить 25-миллиметровую тепловую зону выше и ниже блока для свободного воздушного обмена.

Монтаж и демонтаж S7-1200 устройств

ЦПУ может быть легко установлен на стандартной DIN рейке или на щите. Имеются зажимы для фиксации устройства на DIN рейке. Зажимы также фиксируются в выдвинутом положении, чтобы обеспечить установку блока на щите.



Прежде чем Вы установите или демонтируете любое электрическое устройство, обеспечьте, чтобы питание, подаваемое на оборудованию было выключено. Кроме того убедитесь, что было выключено питание любого связанному с ним оборудования.

⚠ ПРЕДУПРЕЖДЕНИЕ

Установка или демонтаж S7-1200 или связанного с ним оборудования с поданным питанием могло бы вызвать поражение током или непредсказуемую работу оборудования.

Отказ отключить любое питание для S7-1200 и связанного с ним оборудования во время процедур установки или демонтажа могло бы привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу из-за поражения током или непредсказуемой работы оборудования.

Всегда следуйте надлежащим мерам безопасности и убедитесь, что питание S7-1200 отключено прежде, чем попытаться установить или демонтировать S7-1200 ЦПУ или связанное с ними оборудование.

Каждый раз при замене или установке S7-1200 устройства убедитесь, что Вы используете правильный модуль или эквивалентное устройство.

**ПРЕДУПРЕЖДЕНИЕ**

Неправильная установка модуля S7-1200 может вызвать непредвиденную реакцию программы в S7-1200.

Отказ заменить устройство S7-1200 устройством той же самой модели или заказного номера, соблюсти ориентацию мог бы привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу из-за непредсказуемой работы оборудования.

Замените S7-1200 устройство той же самой моделью и убедитесь, что сориентировали

**ПРЕДУПРЕЖДЕНИЕ**

Не отключайте оборудование в огнеопасной или легковоспламеняемой атмосфере.

Отключение оборудования при наличии огнеопасной или легковоспламеняемой атмосферы, может вызвать пожар или взрыв, которые могли бы привести к смерти, серьезным травмам и/или материальному ущербу.

Всегда следуйте надлежащим мерам безопасности, когда присутствует огнеопасная или легковоспламеняемая атмосфера.

Примечание

Электростатический разряд может повредить устройство или соединитель на ЦПУ.

Прикоснитесь к заземленной проводящей площадке и/или наденьте заземленный браслет каждый раз, когда Вы обращаетесь с устройством.

4.3.2. Монтаж и демонтаж ЦПУ

Вы можете установить ЦПУ на щите или на DIN рейке.

Примечание

Пристыковывайте любые коммуникационные модули к ЦПУ и монтируйте сборку как узел. Устанавливайте сигнальные модули по-отдельности после того, как ЦПУ будет установлен.

Примите во внимание следующую информацию, когда монтируете блоки на DIN рейку или в щит:

- Для монтажа на DIN рейку удостоверьтесь, что верхний зажим для DIN рейки находится в скрытом (внутреннем) положении и что нижний зажим для DIN рейки находится в выдвинутом положении для ЦПУ и пристыкованных коммуникационных модулей.
- После монтажа устройств на DIN рейку переведите нижние зажимы для DIN рейки в скрытое положение, чтобы закрепить устройство на DIN рейке.
- Для установки в щите убедитесь, что зажимы для DIN рейки переведены в выдвинутое положение. Чтобы установить ЦПУ в щите, выполните следующие действия:
 1. Разместите, просверлите и нарежьте внутреннюю резьбу (M4), используя размеры представленные в таблице, Монтажные размеры (мм) (стр. 54).
 2. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания.
 3. Выдвиньте монтажные зажимы из модуля. Удостоверьтесь, что зажимы для DIN рейки сверху и снизу ЦПУ находятся в выдвинутом положении.
 4. Зафиксируйте модуль на щите, используя винт M4 с плоскоконической головкой с пружинной и плоской шайбой. Не используйте винт с плоской головкой.

Примечание

Тип винта будет определяться материалом, в котором он закрепляется. Вы должны применить соответствующий вращающий момент, пока пружинная шайба не станет плоской. Избегайте применять чрезмерный вращающий момент к монтажным винтам. Не используйте винт с плоской головкой

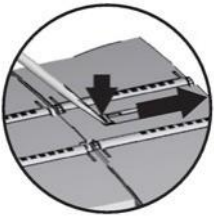
Примечание

Использование стопоров для DIN рейки могло бы быть полезными, если Ваш ЦПУ находится в окружающей среде с сильной вибрацией или если ЦПУ был установлен вертикально. Используйте концевую скобу (8WA1808 или 8WA1805) на DIN рейке, чтобы гарантировать, чтобы модули остаются соединенными. Если Ваша система находится в окружающей среде с сильной вибрацией, то монтаж ЦПУ на щите обеспечит больший уровень защиты от вибрации.

Таблица 4- 2 Установка ЦПУ на DIN рейке

Задача	Процедура
	1. Смонтируйте DIN рейку. Прикрепите рейку к монтажной панели каждые 75 мм. 2. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 3. Навесьте ЦПУ на верхнюю часть DIN рейки. 4. Выдвиньте зажим для DIN рейки в нижней части ЦПУ, чтобы позволить ЦПУ опуститься на рейку. 5. Поверните ЦПУ вниз до соприкосновения с рейкой. 6. Сдвиньте зажимы внутрь, чтобы зафиксировать ЦПУ на рейке.
	

Таблица 4- 3 Снятие ЦПУ с DIN рейки

Задача	Процедура
 	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Отключите соединители ввода/вывода, проводку и кабели от ЦПУ (стр. 65). 3. Снимите ЦПУ и любые пристыкованные коммуникационные модули как один узел. Все сигнальные модули должны остаться установленными. 4. Если сигнальный модуль соединяется с ЦПУ, отведите шинный соединитель: – Разместите отвертку рядом с ползунком в верхней части сигнального модуля. – Нажмите вниз, чтобы отвести соединитель от ЦПУ. – Сдвиньте ползунок до конца вправо. 5. Снимите ЦПУ: – Выдвиньте наружу зажим для DIN рейки, чтобы освободить ЦПУ от рейки. – Поверните ЦПУ и от рейки и удалите его из системы.
	

4.3.3. Монтаж и демонтаж сигнальной, коммуникационной или батарейной плат

Таблица 4- 4 Установка сигнальной, коммуникационной или батарейной платы 1297

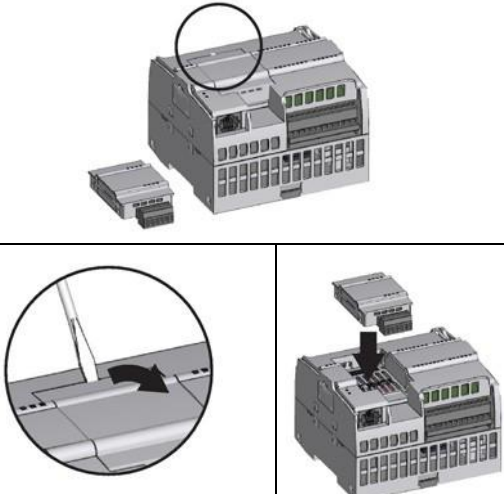
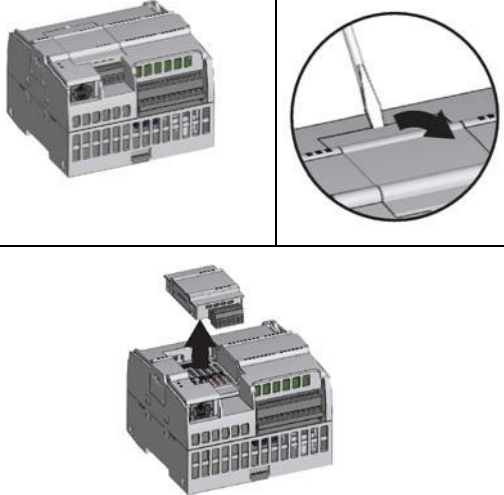
Задача	Процедура
	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Снимите верхнюю и нижнюю крышки клеммных колодок в ЦПУ. 3. Вставьте отвертку в щель сверху ЦПУ с задней стороны крышки. 4. Мягко подденьте крышку и снимите ее с ЦПУ. 5. Расположите модуль вертикально вниз над его монтажным положением в верхней части ЦПУ. 6. Плотно посадите модуль, пока он не будет зафиксирован. 7. Установите на место крышки клеммных колодок.

Таблица 4- 5 Снятие сигнальной, коммуникационной или батарейной платы 1297

Задача	Процедура
	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Снимите верхнюю и нижнюю крышки клеммных колодок в ЦПУ. 3. Вставьте отвертку в щель над верхним краем платы. 4. Мягко подденьте плату вверх, чтобы отсоединить ее от ЦПУ. 5. Выньте плату вертикально вверх из ее монтажного положения в верхней части ЦПУ. 6. Установите на место крышку ЦПУ. 7. Установите на место крышки клеммных колодок.

Установка или замена батареи в батарейной плате BB 1297

Плата BB 1297 требует батарея типа CR1025. Батарея не включена в поставку для BB 1297 и должна приобретаться отдельно. Чтобы установить или заменить батарею, выполните следующие шаги:

1. В плату BB 1297 установите новую батарею положительным полюсом вверх и отрицательным полюсом в сторону печатного монтажа.
2. BB 1297 готова для установки в ЦПУ. Следуйте инструкциям по монтажу, приведенным выше, чтобы установить BB 1297.

Чтобы заменить батарею в плате BB 1297:

1. Снимите плату BB 1297 с ЦПУ, следуя инструкциям по демонтажу, приведенным выше.
2. Аккуратно демонтируйте старую батарею, используя маленькую отвертку. Выньте батарею из зажима под установочным фиксатором BB 1297.
3. Установите новую сменную батарею CR1025 положительным полюсом вверх и отрицательным полюсом в сторону печатного монтажа.
4. Повторно установите батарейную плату BB 1297 положительным полюсом вверх и отрицательным полюсом в сторону печатного монтажа.



ПРЕДУПРЕЖДЕНИЕ

Установка непригодной батареи в плату BB 1297, или иначе подключение непригодной батареи к схеме могут привести к возгоранию или повреждению компонентов и непредсказуемой работе оборудования.

Возгорание или непредсказуемая работа оборудования могут привести к смертельному случаю, тяжелым телесным повреждениям или материальному ущербу.

Используйте только специальную батарею CR1025 для резервирования часов реального времени.

4.3.4. Монтаж и демонтаж сигнального модуля

Таблица 4- 6 Установка сигнального модуля

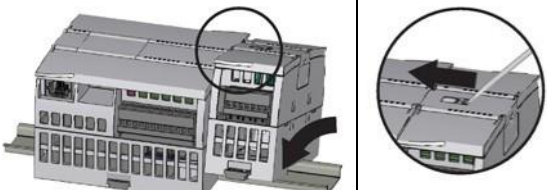
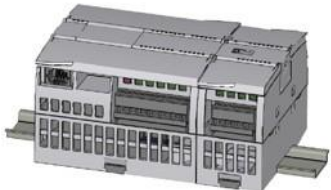
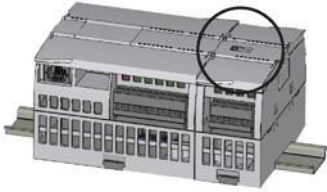
Задача	Процедура
	Устанавливайте Ваш сигнальный модуль после установки ЦПУ. 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Снимите крышку соединителя на правой стороне ЦПУ: – Вставьте отвертку в щель над крышкой. – Мягко подденьте крышку в ее верхней части и снимите крышку. 3. Сохраните крышку для ее повторного использования.
	Подключите сигнальный модуль к ЦПУ: 1. Разместите сигнальный модуль рядом с ЦПУ. 2. Навесьте сигнальный модуль на верхний край DIN рейки. 3. Выдвиньте нижний зажим для DIN рейки, чтобы позволить сигнальному модулю опуститься на рейку. 4. Поверните сигнальный модуль вниз до контакта с рейкой и сдвиньте зажим внутрь, чтобы зафиксировать модуль на рейке.
	Перемещение шинного соединителя выполняет, как механические так и электрические соединения сигнального модуля. 1. Разместите отвертку рядом с ползунком в верхней части сигнального модуля. 2. Переведите ползунк влево, чтобы ввести шинный соединитель в ЦПУ. Выполните ту же самую процедуру, чтобы пристыковать следующий сигнальный модуль.

Таблица 4- 7 Снятие сигнального модуля

Задача	Процедура
	Вы можете снять любой коммуникационный модуль, не снимая ЦПУ или другой сигнальный модуль. 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Снимите соединители ввода / вывода и проводку с сигнального модуля (стр. 65). 3. Отведите шинный соединитель. – Разместите отвертку рядом с ползунком в верхней части сигнального модуля. – Нажмите вниз, чтобы отвести соединитель от ЦПУ. – Сдвиньте ползунок до конца вправо Если справа находится другой сигнальный модуль, повторите данную процедуру для этого сигнального модуля.
	Снимите сигнальный модуль: 1. Выдвиньте нижний зажим для DIN рейки, чтобы освободить сигнальный модуль от рейки. 2. Поверните сигнальный модуль в сторону от рейки. Удалите сигнальный модуль из системы. 3. Если требуется, закройте шинный соединитель ЦПУ крышкой, чтобы избежать загрязнения. Выполните ту же самую процедуру, чтобы снять сигнальный модуль с другого сигнального модуля.

4.3.5. Установка и снятие коммуникационного модуля или коммуникационного процессора

Пристыкуйте любые коммуникационные модули к ЦПУ и установите сборку как узел, как показано в разделе Установке и снятие ЦПУ (стр. 58).

Таблица 4- 8 Установка коммуникационного модуля или коммуникационного процессора

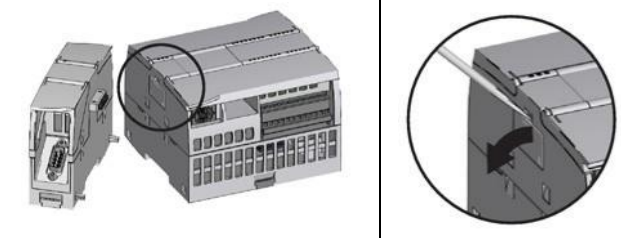
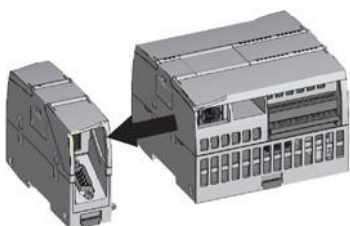
Задача	Процедура
	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Присоедините коммуникационный модуль к ЦПУ прежде, чем установить сборку в качестве узла на DIN рейку или щит. 3. Снимите крышку шины с левой стороны ЦПУ: – Вставьте отвертку в щель над крышкой шины. – Мягко подденьте крышку в ее верхней части. 4. Удалите крышку шины. Сохраните ее для повторного использования. 5. Присоедините коммуникационный модуль или коммуникационный процессор к ЦПУ. – Выровняйте шинный соединитель и штыри коммуникационного модуля с отверстиями в ЦПУ – Плотно прижмите блоки друг к другу пока штыри не встанут на место. 6. Установите ЦПУ и коммуникационный процессор на DIN рейку или щит.
	

Таблица 4- 9 Снятие коммуникационного модуля или коммуникационного процессора

Задача	Процедура
	Снимите ЦПУ и коммуникационные модули, как единый узел с DIN рейки или щита. 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Снимите соединители ввода / вывода, всю проводку и кабели с ЦПУ и коммуникационных модулей. 3. Для монтажа на DIN рейке переведите нижние зажимы для DIN рейки на ЦПУ и коммуникационных модулях в выдвинутое положение. 4. Снимите ЦПУ и коммуникационные модули с DIN рейки или щита. 5. Плотно возьмитесь за ЦПУ и коммуникационные модули и разделите их.
	

ЗАМЕТКА

Разделяйте модули без использования инструмента.

Не используйте инструмент, чтобы разделить модули, так как это может повредить их.

4.3.6. Снятие и повторная установка соединителя клеммной колодки в S7-1200

ЦПУ, сигнальная плата и сигнальные модули обладают съемными соединителями, чтобы облегчить подключение проводки.

Таблица 4- 10 Снятие соединителя

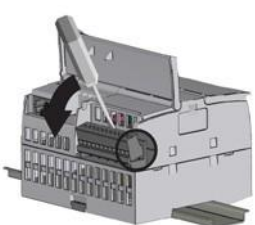
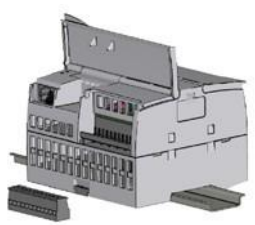
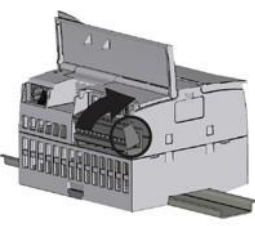
Задача	Процедура
	Подготовьте систему к снятию соединителя клеммной колодки, отключив питание ЦПУ и открыв крышку над соединителем. 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Осмотрите верхнюю часть соединителя и найдите прорезь для наконечника отвертки. 3. Вставьте отвертку в прорезь. 4. Мягко подденьте верхнюю часть соединителя в сторону от ЦПУ. Соединитель освободится со щелчком. 5. Возьмитесь за соединитель и снимите его с ЦПУ.
	

Таблица 4- 11 Установка соединителя

Задача	Процедура
	Подготовьте компоненты для установки клеммной колодки, отключив питание ЦПУ и открыв крышку над соединителем. 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Выровняйте соединитель по контактам модуля. 3. Выровняйте притупленный край соединителя с внутренним обрамлением основания соединителя. 4. Плотнo нажмите вниз и покачивайте соединитель, пока он со щелчком не встанет на место. Тщательно проверьте соединитель на предмет надлежащего выравнивания и фиксации.
	

4.3.7. Установка и снятие кабеля расширения

Кабель расширения S7-1200 обеспечивает дополнительную гибкость в конфигурировании компоновки Вашей системы S7-1200. Только один кабель расширения допустимо использовать в системе ЦПУ. Вы устанавливаете кабель расширения или между ЦПУ и первым сигнальным модулем, или между любыми двумя сигнальными модулями.

Таблица 4- 12 Установка и снятие штеккерного соединителя кабеля расширения

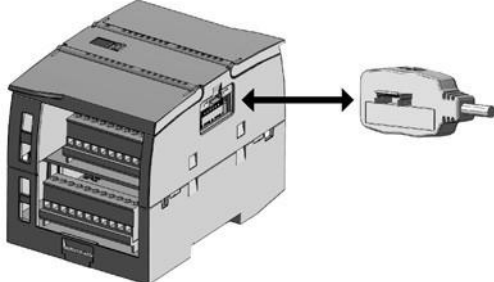
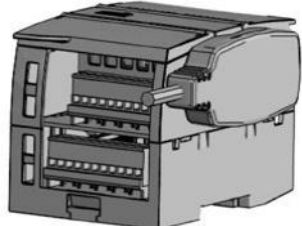
Задача	Процедура
	Чтобы установить штеккерный соединитель: 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Вставьте штеккер в шинный соединитель на правой стороне сигнального модуля или ЦПУ. Чтобы снять штеккерный соединитель: 1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Потяните за штеккер, чтобы отключить его от сигнального модуля или ЦПУ.
	

Таблица 4- 13 Установка гнездового соединителя кабеля расширения

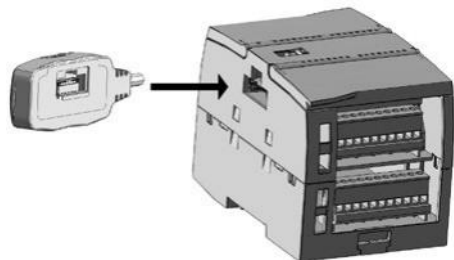
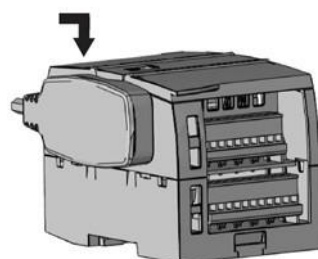
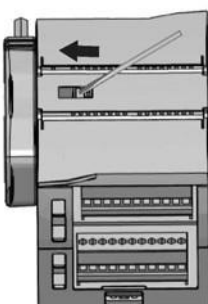
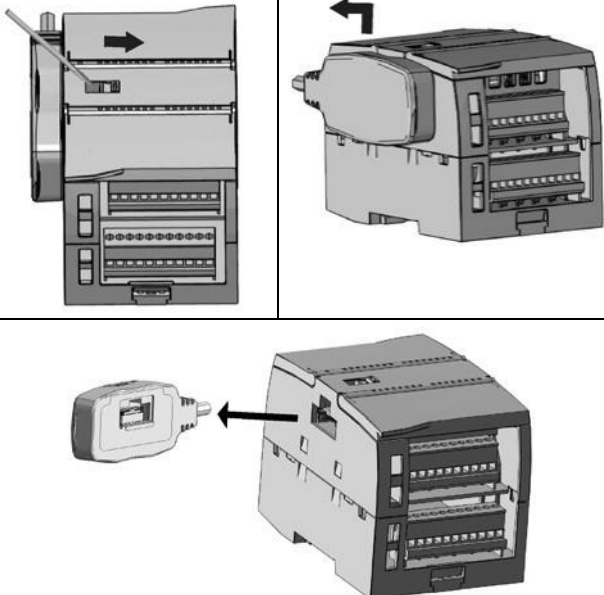
Задача	Процедура
	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Разместите гнездо рядом с шинным соединителем на левой стороне сигнального модуля. 3. Вставьте расширение гнездового соединителя в виде крюка в выемку возле шинного соединителя и слегка надавите, чтобы ввести крюк в зацепление. 4. Зафиксируйте соединитель на месте: – Разместите отвертку рядом с ползунком в верхней части сигнального модуля. – Передвиньте ползунок до конца влево. Чтобы ввести соединитель в зацепление, Вы должны передвинуть ползунок соединителя полностью влево. Ползунок соединителя должен зафиксироваться на месте.
 	

Таблица 4- 14 Снятие гнездового соединителя кабеля расширения

Задача	Процедура
	1. Убедитесь, что ЦПУ и все оборудование S7-1200 отключено от электропитания. 2. Расцепите соединитель: – Разместите отвертку рядом с ползунком в верхней части сигнального модуля. – Слегка нажмите вниз и передвиньте ползунок до конца вправо. 3. Немного приподнимите соединитель, чтобы освободить расширение в виде крюка. 4. Снимите гнездовой соединитель.

Примечание**Установка кабеля расширения в условиях воздействия вибрации**

Если кабель расширения связан с модулями, которые перемещаются или непрочны зафиксированы, то штеккерный конец кабельного соединения с фиксацией может постепенно ослабнуть.

Используйте кабельный хомут для закрепления штеккерного конца кабеля на DIN рейке (или другом месте), чтобы обеспечить дополнительную разгрузку натяжения.

Избегайте применения чрезмерной силы, когда Вы протягиваете кабель во время монтажа. Убедитесь, что соединение кабель - модуль находится в правильном положении после завершения установки

4.3.8. TS (TeleService) адаптер

4.3.8.1 Подключение адаптера удаленного управления

Прежде, чем установить TS адаптер (TeleService) IE Basic или TS адаптер (TeleService) IE Advanced, Вы должны сначала соединить TS адаптер и TS модуль.

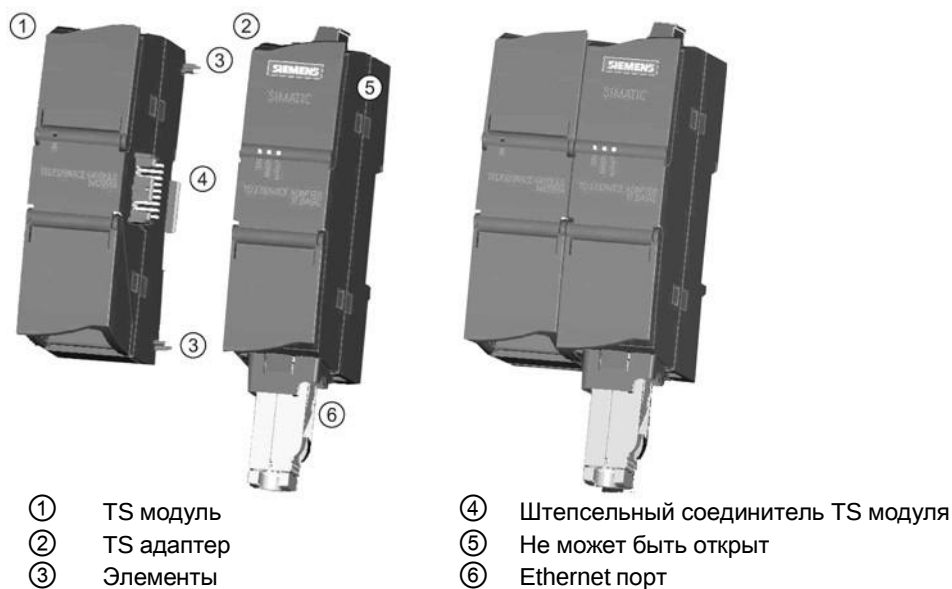
Доступные TS модули:

- TS модуль RS232
- TS модуль Modem
- TS модуль GSM
- TS модуль ISDN

Примечание

TS модуль может быть поврежден, если Вы касаетесь контактов штепсельного соединителя ④ TS модуля

Следуйте рекомендациям по работе в условиях электростатического разряда, чтобы избежать повреждения TS модуля. Прежде, чем соединить TS модуль и TS адаптер, убедитесь, что оба находятся в нерабочем состоянии.



Примечание

Прежде, чем соединить TS модуль и базовый блок TS адаптера, убедитесь что контактные штырьки ④ не согнуты.

При выполнении соединения убедитесь, что штепсельный соединитель и направляющие штырьки расположены правильно.

Подключите только TS модуль к TS адаптеру. Не форсируйте соединение TS адаптера с другим устройством, таким, как ЦПУ S7-1200. Не меняйте механическую конструкцию соединителя, не удаляйте и не повреждайте направляющие штырьки.

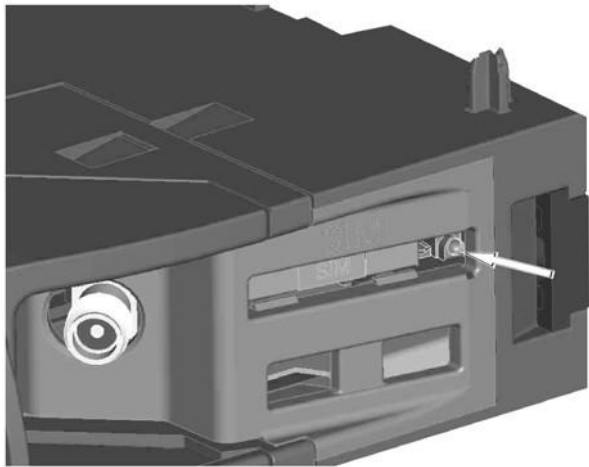
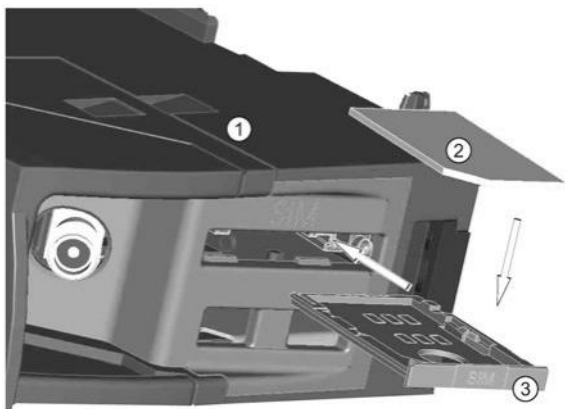
4.3.8.2 Установка SIM-карты

Определите положение SIM-карты на нижней стороне TS модуля GSM.

Примечание

SIM-карта может быть удалена или вставлена только тогда, когда TS модуль GSM обесточен.

Таблица 4- 15 Установка SIM-карты

Задача	Процедура
	Используйте острый предмет, чтобы нажать кнопку извлечения лотка SIM-карты (в направлении стрелки) и вынуть лоток SIM-карты.
	Поместите SIM-карту в лоток, как показано на рисунке, и вставьте лоток SIM-карты в гнездо. ① TS модуль GSM ② SIM-карта ③ Лоток SIM-карты

Примечание

Убедитесь, что SIM-карта правильно ориентирована в лотке. В противном случае, SIM-карта не соединится с модулем, и кнопка выброса может не извлекать лоток карты.

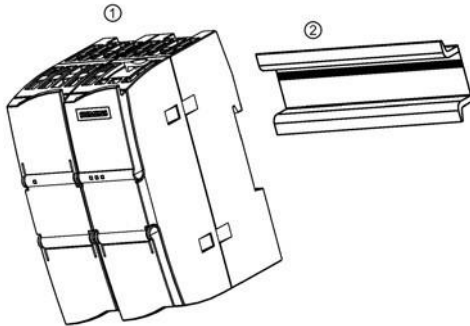
4.3.8.3 Установка узла TS адаптера на DIN рейке

Предпосылки: Вы, должно были соединить TS адаптер и TS модуль вместе, и DIN рейка должна быть установлена.

Примечание

Если Вы устанавливаете TS узел вертикально или в среде с повышенной вибрацией, то TS модуль может отсоединиться от TS адаптера. Используйте концевую скобу 8WA1808 на DIN рейке, чтобы гарантировать, чтобы модули остались соединенными.

Таблица 4- 16 Установка и снятие TS адаптера

Задача	Процедура
	Установка: 1. Навесьте TS адаптер с прикрепленным TS модулем ① на DIN рейку ②. 2. Поверните узел назад до контакта с рейкой. 3. Передвиньте внутрь зажимы для DIN рейки на каждом из модулей, чтобы закрепить каждый из модулей на рейке.
	Снятие: 1. Отключите аналоговый кабель и Ethernet – кабель в нижней части TS адаптера. 2. Выключите питание TS адаптера. 3. Используйте отвертку, чтобы освободить зажимы для рейки на обоих модулях. 4. Поверните узел вверх, чтобы снять его с DIN рейки.



ПРЕДУПРЕЖДЕНИЕ

Правила безопасности для установки или снятия TS адаптера.

Прежде чем Вы выключите питание узла, отсоедините заземление TS адаптера, отключая аналоговый кабель и Ethernet-кабель. Отказ в соблюдении этой предосторожности может привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу из-за непредсказуемой работы оборудования.

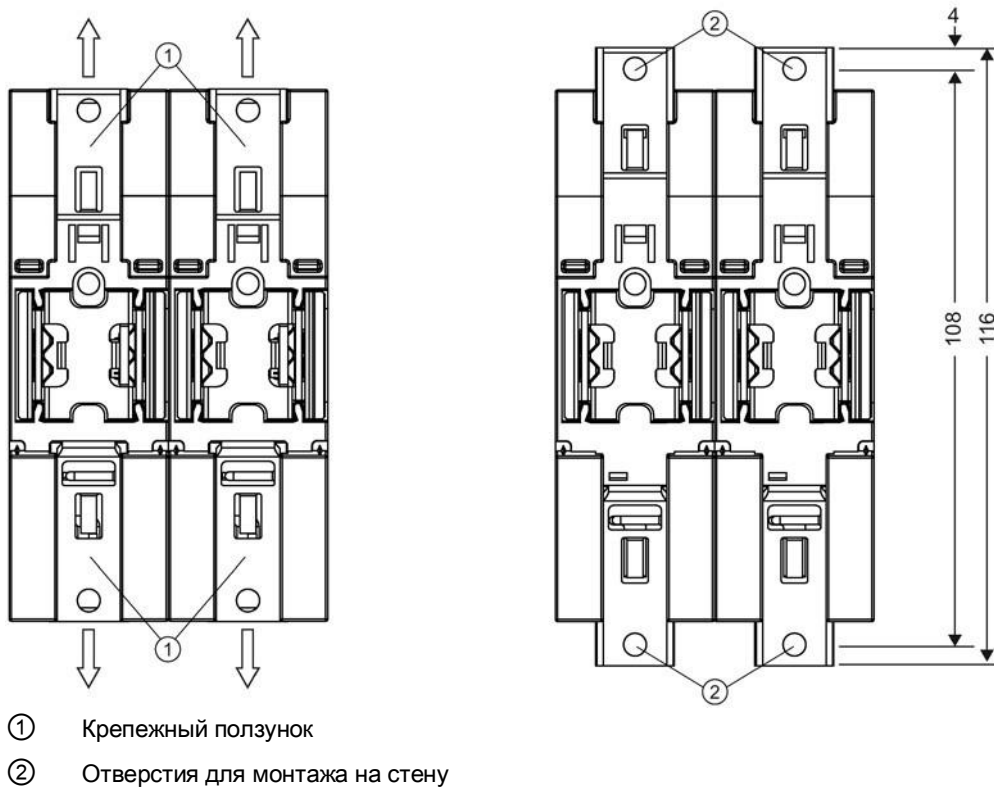
Всегда следуйте этим требованиям, устанавливая или демонтируя TS адаптер.

4.3.8.4 Установка TS адаптера на щите

Предпосылки: Вы, должно были соединить TS адаптер и TS модуль вместе.

1. Переместите крепежный ползунок ① в сторону от TS адаптера и TS модуля в направлении по стрелке, пока он не зафиксируется.
2. Закрепите TS адаптер и TS модуль винтами в положении, отмеченном ② к предназначенной для монтажа стене.

Следующая иллюстрация показывает TS адаптер сзади с крепежными ползунками ① в обоих положениях:



4.4 Рекомендации по проводному монтажу

Надлежащее заземление и проводной монтаж всего электрооборудования важны, чтобы помочь обеспечить оптимальную работу Вашей системы и дополнительную защиту от электрических помех для Вашего приложения и S7-1200. Обратитесь к техническим характеристикам (стр. 1099) для схем электрических соединений S7-1200.

Предпосылки

Прежде чем Вы заземлите или выполните проводной монтаж любого электрического устройства, убедитесь, что питание оборудования выключено. Кроме того обеспечьте отключение питания любого смежного оборудования.

Убедитесь, чтобы Вы следовали всем применимым электрическим правилам и нормам, выполняя проводной монтаж S7-1200 и смежного оборудования. Установите и эксплуатируйте все оборудование согласно всем применимым национальным и местным стандартам. Свяжитесь с местными властями, чтобы определить, какие правила и стандарты относятся к Вашему конкретному случаю.



ПРЕДУПРЕЖДЕНИЕ

Монтаж или подключение S7-1200 или относящегося к нему оборудования во включенном состоянии может привести к поражению электрическим током или к непредсказуемой работе оборудования.

Отказ отключить питание S7-1200 и относящегося к нему оборудования во время процедур монтажа или демонтажа может привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу вследствие поражения электрическим током или непредсказуемой работы оборудования.

Всегда следуйте надлежащим мерам техники безопасности и обеспечьте отключение питания S7-1200 прежде, чем пытаться установить или снять S7-1200 или относящееся к нему оборудование.

Всегда принимайте во внимание вопросы безопасности при проектировании заземления и выполнении проводного монтажа Вашей системы S7-1200. Электронные устройства управления, например S7-1200, могут выйти из строя и вызвать непредсказуемое поведение контролируемого или управляемого оборудования. Поэтому вам следует внедрить предохранительные устройства, независимые от S7-1200, чтобы защитить персонал от возможных травм, а оборудование от повреждения



ПРЕДУПРЕЖДЕНИЕ

Устройства управления могут выйти из строя в небезопасных рабочих состояниях и вызвать вследствие этого неконтролируемое поведение управляемого оборудования

Такое непредсказуемое поведение могло бы привести к смерти, тяжелым травмам и / или материальному ущербу.

Используйте функцию аварийного останова, электромеханические блокировки или другие блокирующие устройства, которые не зависят от S7-1200.

Указания для потенциальной развязки

Граничные значения напряжения для источника питания переменного тока S7-1200 и для входов и выходов цепей переменного тока рассчитаны и одобрены, чтобы обеспечить надежную электрическую развязку между напряжениями линий переменного тока и низковольтными цепями. В зависимости от стандарта, эти границы требуют двойной, усиленной или основной плюс дополнительной изоляции. Компоненты, находящиеся вне этих границ, например, оптопары, конденсаторы, трансформаторы и реле, допущены к эксплуатации как устройства, обеспечивающие надежную электрическую развязку. Только электрические цепи, классифицированные для сетевого напряжения переменного тока, включают в себя надежную изоляцию по отношению к другим цепям. Границы развязки между цепями 24 В постоянного тока связаны исключительно с функциональностью, и Вы не должны рассчитывать на эти границы в плане безопасности.

Выход питания датчиков, коммуникационные цепи и электрические цепи внутренней логики S7-1200 со встроенным источником питания переменного тока в соответствии с EN 61131-2 получают питание как цепи безопасного сверхнизкого напряжения (SELV, safety extra-low voltage).

Для поддержания безопасного характера низковольтных цепей S7-1200 внешние соединения с коммуникационными портами, аналоговыми цепями и всеми источниками питания с номинальным напряжением 24 В пост.т., а также с цепями ввода / вывода должны получать питание от сертифицированных согласно различным стандартам источников, удовлетворяющих требованиям SELV, PELV, класс 2, с ограничением напряжения или с ограничением мощности.



ПРЕДУПРЕЖДЕНИЕ

Использование для питания низковольтных цепей источников, не имеющих развязки с линией переменного тока или имеющих одиночную изоляцию, может привести к появлению опасных напряжений в цепях, которые считаются безопасными для прикосновения, таких как, цепи связи и в проводке низковольтных датчиков

Такие неожиданно высокие напряжения могли бы вызвать поражение электрическим током, приводящее к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Используйте высокое напряжение только для преобразователей низкого напряжения, которые одобрены как безопасные для прикосновения источники с ограничением напряжения.

Указания по заземлению S7-1200

Заземлять ваше приложение лучше всего, подключив все общие клеммы и клеммы заземления вашего S7-1200 и всех подключенных к нему устройств к одной точке. Эта точка должна быть непосредственно соединена с грунтовым заземлением.

Провода заземления должны быть как можно короче и иметь жилы с большим поперечным сечением, например, 2 мм² (14 AWG).

При выборе точек заземления примите во внимание предписания по безопасному заземлению и обеспечьте надлежащее функционирование защитных отключающих устройств.

Указания по выполнению проводных соединений S7-1200

При проектировании проводных соединений для Вашего S7-1200 предусмотрите единое устройство отключения, которое одновременно снимает напряжение с блока питания CPU S7-1200, всех входных и выходных цепей. Предусмотрите защиту от превышения тока, например, предохранитель или автоматический выключатель, чтобы ограничить аварийный ток в питающей проводке. Продумайте дополнительную защиту с помощью предохранителей или других ограничителей тока в каждой из выходных цепей.

Установите соответствующие устройства подавления скачков напряжения для любой проводки, которая может подвергнуться удару молнии. Для получения дополнительной информации см. Устойчивость к скачкам напряжения (стр. 1099) в разделе Общие технические характеристики.

Избегайте располагать провода низковольтных сигналов и кабели связи в одном кабельном канале с проводами питания переменного тока, проводами с высоким напряжением, проводам с быстро коммутируемым постоянным током. Всегда прокладывайте провода парами: нейтраль или нулевой провод вместе с фазой или сигнальным проводом.

Используйте наиболее короткие из возможных проводов и выбирайте сечение соответствующее току. Соединители ЦПУ и коммуникационных модулей позволяют использовать сечение провода от 2 мм² до 0.3 мм² (14 AWG ... 22 AWG). Длина зачистки составляет 6.4 мм. Соединитель сигнальной платы рассчитан на применение провода от 1.3 мм² до 0.3 мм² (16 AWG ... 22 AWG). Длина зачистки составляет от 6.3 до 7 мм.

Провод и кабель должны иметь предельную температуру эксплуатации на 30 °C выше, чем температура среды вокруг S7-1200 (например, минимум 85 °C для проводников для 55 °C температуры окружающей среды). Вам следует определить другой тип проводки и требования к материалу на основе определенных классов электрических схем и Вашей инсталляционной среды.

Используйте экранированные провода для оптимальной защиты от электрических помех. Как правило, заземление экрана у S7-1200 дает лучшие результаты. Вам следует заземлить экраны коммуникационных кабелей, соединив их с корпусом коммуникационного соединителя S7-1200, используя соединители, которые фиксируют экран кабеля, или соединяя экраны коммуникационных кабелей с отдельной землей. Вам следует заземлить экраны других кабелей, используя зажимы или медную ленту вокруг экрана, чтобы обеспечить соединение с заземляющей точкой по большой площади.

При выполнении проводных соединений входных цепей, которые запитаны от внешнего источника питания, используйте защиту от перегрузки по току в этой цепи. Во внешней защите нет необходимости для цепей, которые запитаны от источника 24 В пост.т. для питания датчиков в S7-1200, потому что источник питания датчиков уже имеет схему ограничения тока.

Все модули S7-1200 обладают съемным соединителем для подключения пользовательской проводки. Чтобы предотвратить слабые контакты, убедитесь, что соединитель вставлен плотно, и что провод надежно закреплен в соединителе. Siemens рекомендует, чтобы изоляция проводов была снята с провода приблизительно на 6 мм, чтобы обеспечить надежное соединение. Чтобы избежать повреждения соединителя, будьте осторожны и не затягивайте чрезмерно винты. Максимальный момент затяжки для винтов соединителя ЦПУ и сигнального модуля - 0.56 Нм (5 дюймов на фунт). Максимальный момент затяжки для винтов соединителя сигнальной платы, симулятора и модуля потенциометра составляет 0.33 Нм (3 дюйма на фунт).

Чтобы помочь предотвратить нежелательные электрические токи в Вашей установке, S7-1200 обеспечивает границы электрической развязки в определенных точках. Когда Вы планируете проводной монтаж своей системы, Вы должны рассмотреть эти границы электрической развязки. Обратитесь к техническим характеристикам (стр. 1099) для получения сведений об объеме обеспеченной развязки и положении границ развязки. -

Электрические цепи, классифицированные для сетевого напряжения переменного тока, включают в себя надежную изоляцию по отношению к другим цепям. Границы развязки между цепями 24 В постоянного тока связаны исключительно с функциональностью, и Вы не должны рассчитывать на эти границы в плане безопасности.

Смотри также

Общие технические характеристики и особенности (стр. 1155)

Указания для ламповых нагрузок

Ламповые нагрузки повреждают контакты реле из-за больших бросков тока при включении. Этот бросок тока обычно в 10 – 15 выше стационарного тока лампы накаливания. Для часто включаемых в течение срока службы приложения ламповых нагрузок рекомендуется использовать сменное промежуточное реле или ограничитель бросков тока.

Указания для индуктивных нагрузок

Используйте цепи подавления с индуктивными нагрузками, чтобы ограничить рост напряжения, когда управляющий выход выключается. Цепи подавления защищают Ваши выходы от преждевременного выхода из строя, вызванного переходным процессом высокого напряжения, который происходит, когда прерывается электрический ток через индуктивную нагрузку.

Кроме того, цепи подавления ограничивают электрические помехи, произведенные при коммутации индуктивных нагрузок. Высокочастотные помехи от плохо подавленных индуктивных нагрузок могут нарушить работу ПЛК. Размещение внешней схемы подавителя так, чтобы она была электрически перпендикулярно нагрузке и физически располагалась около нагрузки, является наиболее эффективным способом уменьшения электрических помех.

Выходы пост.т. в S7-1200 имеют внутренние схемы подавления, которые достаточны для индуктивных нагрузок в большинстве приложений. Так как релейные контакты выходов S7-1200 могут использоваться для коммутации нагрузок как постоянного так и переменного тока, внутренняя защита не предусмотрена.

Хорошее решение для подавителя состоит в том, чтобы использовать контакторы и другие индуктивные нагрузки, для которых изготовитель предусматривает цепи подавления, интегрированные в нагрузочное устройство или как дополнительную опцию. Однако цепи подавления некоторых изготовителей могут быть неподходящими для Вашего приложения. Дополнительная цепь подавления может оказаться необходимой для уменьшения помех и оптимального срока службы контактов.

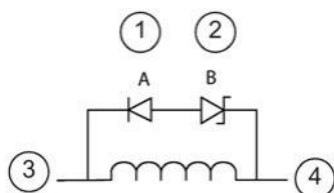
Для нагрузок переменного тока металло-оксидный варистор (MOV) или другое устройство стабилизации напряжения могут использоваться с параллельной RC-цепочкой, но не столь эффективны, когда используются без них. MOV подавитель без параллельной RC-цепочки часто приводит к значительным высокочастотным помехам амплитудой до напряжения стабилизации.

Строго контролируемый переходный процесс выключения будет иметь частоту звона не больше 10 кГц, предпочтительно менее 1 кГц. Пиковое напряжение для линий переменного тока должно быть в пределах ± 1200 В относительно земли. Отрицательное пиковое напряжение для нагрузок постоянного тока, используя внутреннее подавление ПЛК, будет на ~40 В ниже напряжения питания 24 В пост.т. Внешнее подавление должно ограничить переходный процесс в пределах 36 В относительно питания, чтобы разгрузить внутреннюю цепь подавления.

Примечание

Эффективность схемы подавителя зависит от приложения и должна быть проверена для Вашего конкретного случая. Обеспечьте правильный подбор компонентов и используйте осциллограф, чтобы наблюдать переходный процесс выключения

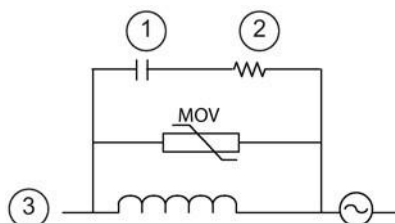
Типовая схема подавителя для выходов пост.т. или релейных выходов, переключающих индуктивные нагрузки пост.т.



- ① Диод 1N4001 или аналог
- ② 8.2 В стабилитрон
(выходы пост.т.),
36 В стабилитрон
(релейные выходы)
- ③ Выходной канал
- ④ М, 24 В опорный потенциал

В большинстве приложений добавление диода (А) параллельно индуктивной нагрузке пост.т. достаточно, но если Ваше приложение требует более быстрых времен выключения, то рекомендуется добавление диода Зенера (В). Обязательно подберите Ваш диод Зенера должным образом по суммарному току в Вашей выходной цепи.

Типовая схема подавителя для релейных выходов, переключающих индуктивные нагрузки пер.т.



- ① Смотри таблицу для величины С
- ② Смотри таблицу для величины R
- ③ Выходной канал

Обеспечьте, чтобы рабочее напряжение металлооксидного варистора (MOV) было по крайней мере на 20% больше, чем номинальное сетевое напряжение.

Выберите неиндуктивные резисторы для импульсных схем и конденсаторы, рекомендуемые для импульсных приложений (обычно металлопленочные). Убедитесь, что компоненты удовлетворяют требованиям средней мощности, пиковой мощности и пикового напряжения.

Если Вы проектируете свою собственную схему подавителя, следующая таблица предлагает значения резистора и конденсатора для диапазона нагрузок пер.т. Эти значения основываются на расчетах с параметрами идеальных компонентов. I rms в таблице относится к установившемуся току нагрузки при полном включении.

Table 4- 17 AC suppressor circuit resistor and capacitor values

Индуктивная нагрузка			Величины для подавителя		
I rms	230 В пер.т.	120 В пер.т.	Резистор		Конденсатор
Амперы	ВА	ВА	Ω	Вт (мощность)	нФ
0.02	4.6	2.4	15000	0.1	15
0.05	11.5	6	5600	0.25	470
0.1	23	12	2700	0.5	100
0.2	46	24	1500	1	150
0.05	115	60	560	2.5	470
1	230	120	270	5	1000
2	460	240	150	10	1500

Условия, приемлемые для табличных значений:

Maximum turn-off transition step < 500 V

Пиковое напряжение резистора < 500 В

Пиковое напряжение конденсатора < 1250 В

Ток подавителя < 8% от тока нагрузки (50 Гц)

Ток подавителя < 11% от тока нагрузки (60 Гц)

Конденсатор: $dV/dt < 2 \text{ В/мкс}$ Импульсное рассеяние конденсатора : $\int (dv/dt)^2 dt < 10000 \text{ В}^2/\text{мкс}$

Частота резонанса < 300 Гц

Мощность резистора для макс. частоты переключения 2 Гц

Коэффициент мощности 0.3 принят для типовой индуктивной нагрузки

Указания для дифференциальных входов и выходов

Дифференциальные входы и выходы ведут себя иначе, чем стандартные входы и выходы. Существует по два контакта на дифференциальный вход и выход. Определение, включен или выключен дифференциальный вход или выход требует, чтобы Вы измерили разность потенциалов между этими двумя контактами.

См. подробные спецификации для CPU 1217C в Приложении А (стр. 1155).

5.1 Обработка программы пользователя

ЦПУ поддерживает следующие типы кодовых блоков, которые позволяют Вам создавать эффективную структуру для своей пользовательской программы:

- Организационные блоки (OBs) определяют структуру программы. Некоторые OB имеют заранее определенное поведение и стартовые события, но Вы можете также создать OB с пользовательскими стартовыми событиями.
- Функции (FC) и функциональные блоки (FB) содержат программный код, который соответствует определенным задачам или комбинациям параметров. Каждый FC или FB обеспечивают набор входных и выходных параметров для того, чтобы обменяться данными с вызывающим блоком. FB также использует связанный блок данных (называемый DB экземпляром), чтобы хранить значения данных для вызова FB с этим экземпляром. Вы можете вызывать FB многократно, каждый раз с уникальным DB экземпляром. Вызов того же самого FB с различными экземплярами DB не повлияет на значения данных ни в одном другом экземпляре DB.
- Блоки данных (DB) хранят данные, которые могут использоваться программными блоками.

Обработка пользовательской программы начинается с одного или более дополнительных стартовых организационных блоков (OB), которые исполняются однократно после перехода в режим RUN, сопровождаемый одним или более OB программного цикла, которые исполняются циклически. Вы можете также связать OB с событием, вызывающим прерывание, которое может быть стандартным событием или событием, связанным с ошибкой. Эти OB исполняются каждый раз, когда имеет место соответствующее стандартное или связанное с ошибкой событие.

Функция (FC) или функциональный блок (FB) являются блоком программного кода, который можно вызвать из OB или от другого FC или FB, вниз к следующим уровням вложения:

- 16 от программного цикла или стартового OB
- 6 от любого OB обработки события, вызывающего прерывания

FC не связаны ни с каким особым блоком данных (DB). FB связаны непосредственно с DB и используют DB для передачи параметров и хранения промежуточных значений и результатов.

Размер пользовательской программы, данных и конфигурации ограничен доступной загрузочной и рабочей памятью в ЦПУ. Не существует определенного ограничения на количество каждой отдельной категории блоков OB, FC, FB и DB. Однако общее количество блоков ограничено числом 1024.

Каждый цикл включает в себя запись выходов, чтение входов, выполнение пользовательских инструкций по программе и выполнение фоновой обработки. Цикл упоминается как цикл сканирования или скан.

Ваше решение для автоматизации на базе S7-1200 может состоять из центральной стойки с ЦПУ S7-1200 и дополнительными модулями. Термин "центральная стойка" относится к сборке из ЦПУ и связанных с ним модулей на рейке или щите. Модули (SM, SB, BB, CB, CM или CP) обнаруживаются и регистрируются только при включении питания.

- Вставка или удаление модуля в центральной стойке при наличии питания (на горячую) не поддерживается. Никогда не вставляйте и не удаляйте модуль из центральной стойки, когда на ЦПУ подано питание.

ПРЕДУПРЕЖДЕНИЕ

Меры безопасности для вставки или удаления модулей

Вставка или удаление модуля (SM, SB, BB, CD, CM или CP) из центральной стойки, когда на ЦПУ подано питание, может вызвать непредсказуемую работу, приводящую к повреждению оборудования и/или травмам персонала.

Всегда отключайте питание ЦПУ и центральной стойки и следуйте соответствующим мерам безопасности прежде, чем вставить или удалить модуль из центральной стойки.

- Вы можете вставить или удалить карту памяти SIMATIC, в то время как ЦПУ находится под питанием. Однако вставка или удаление карты памяти, когда ЦПУ находится в режиме RUN, заставляют центральный процессор перейти в режим STOP.

ЗАМЕТКА

Риски, связанные со вставкой или удалением модулей, когда ЦПУ находится в режиме RUN.

Вставка или удаление карты памяти, когда ЦПУ находится в режиме RUN заставляют ЦПУ перейти в режим STOP, что могло бы нанести вред оборудованию или управляемому процессу.

Каждый раз, когда Вы вставляете или удаляете карту памяти, ЦПУ незамедлительно переходит в режим STOP. Прежде, чем вставить или удалить карту памяти, всегда убеждайтесь, что ЦПУ не находится в фазе активного управления машиной или процессом. Всегда используйте цепь аварийного останова для Вашего приложения или процесса.

- Если Вы вставляете или удаляете модуль на стойке распределенного ввода/вывода (AS-i, PROFINET или PROFIBUS), когда ЦПУ находится в режиме RUN, ЦПУ генерирует запись в диагностический буфер, исполняет ОВ вставки или удаления модулей, если он существует, и, по умолчанию, остается в режиме RUN.

Обновление образа процесса и разделы образа процесса

ЦПУ обновляет локальные цифровые и аналоговые каналы ввода/вывода синхронно с циклом сканирования, используя внутреннюю область памяти, называемую образом процесса. Образ процесса содержит снимок физических входов и выходов (физические каналы ввода/вывода в ЦПУ, сигнальных платах и сигнальных модулях).

Вы можете конфигурировать обновление каналов ввода/вывода в образе процесса в каждом цикле просмотра или при наличии прерывания по определенному событию. Вы можете также конфигурировать исключение канала ввода/вывода из обновления образа процесса. Например, Вашему процессу, возможно, понадобились бы только определенные значения данных, когда происходит событие, такое как аппаратное прерывание. Конфигурируя обновление образа процесса для этих каналов ввода / вывода, которые будут связаны с разделом, которому Вы назначаете ОБ аппаратного прерывания, Вы избегаете излишнего обновления значения данных центральным процессором в каждом цикле сканирования, если Ваш процесс не требует непрерывного обновления.

Для ввода/вывода, который обновляется в каждом цикле сканирования, ЦПУ выполняет следующие задачи во время каждого цикла сканирования:

- ЦПУ переносит состояния из области образа процесса по выходу в физические выходы.
- ЦПУ читает физические входы непосредственно перед выполнением пользовательской программы и хранит значения входов в области образа процесса по входу. Эти значения, таким образом, остаются неизменными в течение выполнения пользовательских инструкций.
- Центральный процессор обрабатывает логику пользовательских инструкций и обновляет значения выходов в области образа процесса по выходу вместо того, чтобы записывать в фактические физические выходы.
Этот процесс обеспечивает последовательную логику посредством выполнения пользовательских инструкций для данного цикла и предотвращает мерцание физических выходных каналов, которые могли бы многократно менять состояние в области образа процесса по выходу.

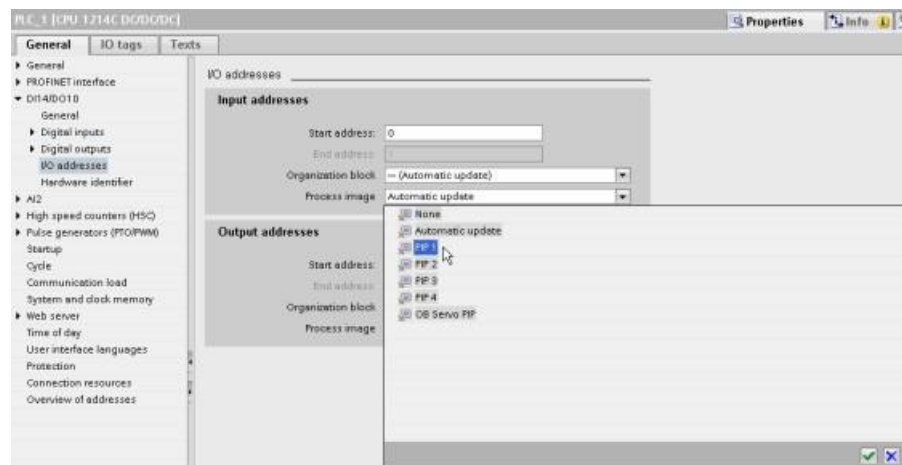
Для управления, несмотря на то обновляет ли Ваш процесс каналы ввода/вывода автоматически на каждом цикле сканирования или при возникновении событий, S7-1200 предлагает пять разделов образа процесса. Первый раздел образа процесса PIP0 предназначен для ввода/вывода, который должен быть автоматически обновлен в каждом цикле сканирования и является назначением по умолчанию. Вы можете использовать остальные четыре раздела, PIP1, PIP2, PIP3 и PIP4 для назначения обновлений образа процесса ввода/вывода различным прерываниям по событию.

Вы ставите в соответствие ввод/вывод разделам образа процесса в Конфигурации Устройства, и Вы назначаете разделам образа процесса прерывания по событию, когда Вы создаете ОБ обработки прерывания (стр. 176) или редактируете свойства ОБ (стр. 176).

По умолчанию, когда Вы вставляете модуль в представлении устройства, STEP 7 устанавливает обновление образа процесса ввода/вывода на "Автоматическое обновление". Для ввода/вывода, сконфигурированного для "Автоматического обновления", ЦПУ управляет обменом данными между модулем и областью образа процесса автоматически во время каждого цикла сканирования.

Чтобы привязать цифровые или аналоговые каналы к разделу образа процесса или исключить каналы ввода/вывода из обновления образа процесса, выполните следующие шаги:

1. Рассмотрите вкладку Properties для соответствующего устройства в Конфигурации устройств.
2. Раскройте при необходимости пункты под "General", чтобы найти нужные каналы ввода/вывода.
3. Выберите "I/O addresses".
4. Дополнительно выберите определенный ОБ из выпадающего списка "Organization block".
5. В выпадающем списке "Process image" перейдите от "Automatic update" к "PIP1", "PIP2", "PIP3", "PIP4" или "None". Выбор "None" означает, что Вы можете только читать и записывать в этот ввод/вывод, используя инструкции прямого доступа. Чтобы вернуться назад к автоматическому обновлению каналов с образом процесса, измените этот выбор на "Automatic update".



Вы можете напрямую прочитать физические входные значения и напрямую записать физические выходные значения при выполнении инструкции. Непосредственное чтение обращается к текущему состоянию физического входа и не обновляет область образа процесса по входам, независимо от того, сконфигурирован ли канал для сохранения в образе процесса. Непосредственная запись в физический выход обновляет как область образа процесса по выходу (если канал сконфигурирован для сохранения в образе процесса), так и физический выходной канал. Добавьте суффикс ":P" к адресу ввода/вывода, если Вы хотите, чтобы программа получила прямой доступ к данным о вводе/выводе непосредственно от физического канала вместо того, чтобы использовать образ процесса.

Примечание

Использование разделов образа процесса

Если Вы привязываете ввод/вывод к одному из разделов образа процесса PIP1 - PIP4 и не ставите в соответствие этому разделу определенный ОБ, то ЦПУ никогда не производит передачу данных ввода/вывода в образ или из образа процесса.

Привязка ввода/вывода к PIP, у которого нет соответствующего назначенного ОБ, равнозначна выбору настройки "None". Вы можете прочитать состояние физического ввода/вывода с помощью инструкции непосредственного чтения или записать в физический ввод / вывод с помощью инструкции непосредственной записи. ЦПУ не обновляет образ процесса.

ЦПУ поддерживает распределенный ввод/вывод для обеих сетей PROFINET и PROFIBUS (стр. 611).

5.1.1. Рабочие режимы ЦПУ

У ЦПУ есть три режима работы: STOP, STARTUP и RUN. Светодиоды состояния на передней стороне ЦПУ указывают на текущий режим работы.

- В режиме STOP ЦПУ не обрабатывает программу. Вы можете загружать проект.
- В режиме STARTUP стартовые ОВ (если есть) обрабатываются однократно. ЦПУ не обрабатывает прерывания по событию во время режима запуска.
- В режиме RUN с постоянным повторением обрабатываются ОВ программного цикла. Прерывания по событию могут иметь место, и соответствующий прерыванию ОВ может быть обработан в любой момент при нахождении в режиме RUN. Вы можете загружать некоторые части проекта в режиме RUN (стр. 1087).

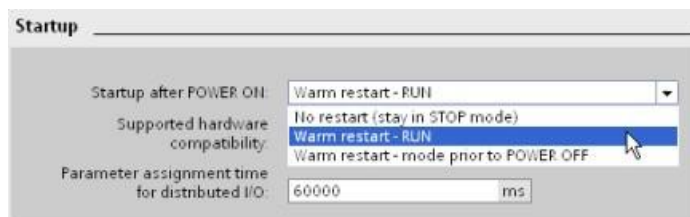
ЦПУ поддерживает теплый перезапуск для перехода в режим RUN. Теплый перезапуск не включает в себя сброс памяти. ЦПУ инициализирует все несохраняемые системные и пользовательские данные при теплом перезапуске и сохраняет значение всех сохраняемых пользовательских данных.

Сброс памяти очищает всю рабочую память, очищает сохраняемую и несохраняемую области памяти, копирует загрузочную память в рабочую память и устанавливает выходы в сконфигурированное состояние "Reaction to CPU STOP". Сброс памяти не очищает диагностический буфер или постоянно сохраненные настройки IP-адреса.

Вы можете сконфигурировать параметр ЦПУ "startup after POWER ON". Этот пункт конфигурации появляется под "Конфигурацией устройства" для ЦПУ под пунктом "Startup". После включения ЦПУ выполняет последовательность диагностических проверок и инициализацию системы.

Во время инициализации системы ЦПУ очищает всю несохраняемую битовую память (M) и замещает все несохраняемое содержимое DB начальными значениям из загрузочной памяти. ЦПУ оставляет неизменной сохраняемую битовую память (M) и сохраняемое содержимое DB и затем переходит в соответствующий рабочий режим. Определенные обнаруженные ошибки препятствуют тому, чтобы ЦПУ перешел в режим RUN. ЦПУ поддерживает следующий выбор конфигурации:

- No restart (оставаться в режиме STOP)
- Warm restart - RUN
- Warm restart - mode prior to POWER OFF



ЗАМЕТКА

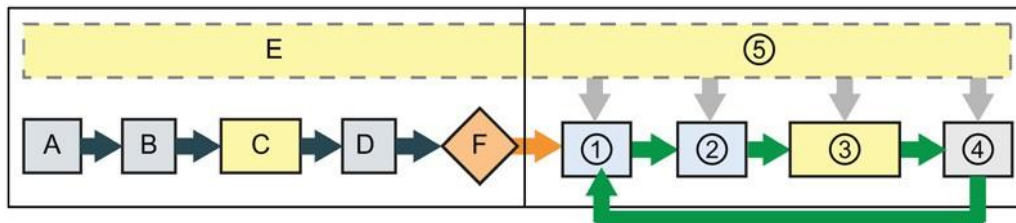
Поддающиеся исправлению ошибки могут вызвать переход ЦПУ в режим STOP.

ЦПУ может перейти в режим STOP из-за поддающихся исправлению ошибок, таких как неисправность съемного сигнального модуля или временные ошибки, такие как помехи питающей сети или колебания питающего напряжения. Подобные условия могли бы привести к материальному ущербу..

Если Вы выбрали в конфигурации "Warm restart - mode prior to POWER OFF", ЦПУ переходит в рабочий режим, в котором он находился до выключения питания или ошибки. Если ЦПУ был в режиме STOP во время выключения питания или ошибки, то он переходит в режим STOP после подачи питания и остается в нем, пока не получит команду на переход в режим RUN. Если ЦПУ в момент выключения питания или ошибки находился в режиме RUN, то при следующей подаче питания он переходит в режим RUN, если не обнаруживает ошибок, которые запретили бы переход в режим RUN.

Вы можете изменить текущий рабочий режим, используя команды "STOP" или "RUN" (стр. 1075) из он-лайн инструментов программного пакета. Вы можете также включать инструкцию STP (стр. 289) в Вашу программу, чтобы переводить ЦПУ в режим STOP. Это позволяет Вам останавливать выполнение своей программы, на основании логики.

- ЦПУ в режиме STOP обрабатывает любые коммуникационные запросы (как соответствующие) и выполняет самодиагностику. ЦПУ не обрабатывает пользовательскую программу, и не выполняются автоматические обновления образа процесса.
- В режимах STARTUP и RUN, ЦПУ выполняет задачи, показанные на следующем рисунке.



STARTUP

- A Очищает I (образ) область памяти
- B Инициализирует Q (образ) область памяти нулевыми, последними или подстановочными значениями, как это было сконфигурировано, и обнуляет PB, PN и AS-i выходы.
- C Инициализирует несохраняемую M память и блоки данных на их начальные значения и разрешает сконфигурированные циклические прерывания и события по времени суток.
Выполняет стартовые ОБ.
- D Копирует состояние физических входов в I память
- E Ставит любые события прерываний в очередь, которая будет обработана после входа в режим RUN
- F Разрешает запись Q памяти в физические выходы

RUN

- ① Записывает Q память в физические выходы
- ② Копирует состояние физических входов в I память
- ③ Выполняет ОБ программного цикла
- ④ Выполняет самотестирование
- ⑤ Обрабатывает прерывания и коммуникации во время любого этапа цикла сканирования

Процедура запуска (STARTUP)

Каждый раз, когда рабочий режим изменяется из STOP в RUN, ЦПУ очищает входы образа процесса, инициализирует выходы образа процесса и обрабатывает стартовые ОВ. Любые доступы на чтение входов образа процесса посредством инструкций в стартовых ОВ прочитывают ноль, а не текущее значение физического входа. Поэтому, чтобы прочитать текущее состояние физического входа во время режима запуска, Вы должны выполнить непосредственное обращение. Пусковые ОВ и любые связанные с ними FC и FB обрабатываются по очереди. Если существует более одного стартового ОВ, каждый исполняется в очередности согласно номеру ОВ, начиная с наименьшего номера.

Каждый стартовый ОВ включает в себя начальную информацию, которая помогает Вам определить действительность сохраняемых данных и часов времени суток. Вы можете программировать инструкции в стартовых ОВ, чтобы оценить эти начальные значения и принять соответствующие меры. Следующие начальные ячейки поддерживаются стартовыми ОВ:

Таблица 5- 1 Стартовые ячейки, поддерживаемые ОВ запуска

Вход	Тип данных	Описание
LostRetentive	Bool	Этот бит принимает значение «истина», если области размещения сохраняемых данных были потеряны
LostRTC	Bool	Этот бит принимает значение «истина», если часы времени суток (часы реального времени) были потеряны

ЦПУ также выполняет следующие задачи во время процедуры запуска

- Прерывания помещаются в очередь, но не обрабатываются во время фазы запуска
- Во время фазы запуска не выполняется контроль времени цикла
- Изменения конфигурации модулей HSC (высокоскоростной счетчик), PWM (шиотно-импульсная модуляция) и PtP (коммуникация точка-к-точке) могут быть выполнены при запуске
- Фактическая работа модулей HSC, PWM и коммуникаций точка-к-точке имеет место только в RUN

После того как исполнение ОВ запуска заканчивается, ЦПУ переходит в режим RUN и обрабатывает задачи управления в непрерывном цикле сканирования.

5.1.2. Процедура цикла сканирования в режиме RUN

Для каждого цикла сканирования ЦПУ записывает выходы, читает входы, выполняет пользовательскую программу, обновляет коммуникационные модули и отвечает на пользовательские прерывания по событию и коммуникационные запросы. Коммуникационные запросы периодически обрабатываются в процессе сканирования.

Эти действия (за исключением пользовательских прерывания по событию) обслуживаются регулярно и в последовательном порядке. Пользовательские прерывания по событию, которые разрешены, обслуживаются согласно приоритету в порядке, в котором они происходят. Для прерываний по событию ЦПУ читает входы, выполняет ОВ и затем записывает выходы, используя связанный раздел образа процесса (PIR), если имеет место.

Система гарантирует, что цикл сканирования будет закончен в течение периода, называемом максимальным временем цикла; иначе генерируется событие ошибки времени.

- Каждый цикл сканирования начинается, с извлечения текущих значений цифровых и аналоговых выходов из образа процесса и последующей записи их в физические выходы ЦПУ, SB и SM модулей, сконфигурированных для автоматического обновления ввода/вывода (конфигурация по умолчанию). Когда к физическим выходам обращается инструкция, как образ процесса по выходам так и сами физические выходы обновляются.
- Цикл сканирования продолжается чтением текущего состояния цифровых и аналоговых входов из ЦПУ, SB, и модулей SM сконфигурированных для автоматического обновления ввода/вывода (конфигурация по умолчанию) и последующей записью этих значений в образ процесса. Когда к физическому входу обращается инструкция, значение физического входа доступно инструкции, но образ процесса по входам не обновляется.
- После чтения входов пользовательская программа выполняется от первой до последней инструкции. Это включает в себя все ОВ программного цикла плюс все связанные с ними FC и FB. ОВ программного цикла выполняются в порядке согласно номеру ОВ, начиная с наименьшего.

Обработка коммуникаций происходит периодически в течение сканирования, возможно прерывая выполнение пользовательской программы.

Самодиагностика включает в себя периодические проверки состояния системы и модулей ввода/вывода.

Прерывания могут произойти во время любой части цикла сканирования и управляются событиями. Когда событие имеет место, ЦПУ прерывает цикл сканирования и вызывает ОВ, которая сконфигурирована для обработки этого события. После того, как ОВ заканчивает обрабатывать событие, ЦПУ продолжает выполнение пользовательской программы от точки прерывания.

5.1.3. Организационные блоки (ОВ)

ОВ управляют выполнением пользовательской программы. Определенные события в ЦПУ вызывают выполнение организационного блока. ОВ не могут вызывать друг друга или быть вызваны из FC или FB. Только событие, такое как диагностическое прерывание или временной интервал, может запустить выполнение ОВ. ЦПУ обращается с ОВ согласно их соответствующим классам приоритета, ОВ с более высоким приоритетом выполняются перед ОВ с более низким приоритетом. Наименьший класс приоритета равняется 1 (для основного программного цикла), а самый высокий класс приоритета равняется 26.

5.1.3.1. ОВ обработки программного цикла

ОВ программного цикла выполняются циклически, пока ЦПУ находится в режиме RUN. Главный блок программы - это ОВ программного цикла. Это блок, куда Вы помещаете инструкции, которые управляют Вашей программой, и где Вы вызываете дополнительные пользовательские блоки. У Вас может быть несколько ОВ программного цикла, которые ЦПУ обрабатывает в порядке номеров. Главным по умолчанию является ОВ1.

События программного цикла

Событие программного цикла происходит однажды во время каждого программного цикла (или сканирования). Во время программного цикла ЦПУ пишет выходы, читает входы и выполняет ОВ программного цикла. Событие программного цикла требуется и всегда разрешено. У Вас могло бы не быть ОВ программного цикла, или Вам можно было бы выбрать несколько ОВ для события программного цикла. После того, как событие программного цикла происходит, ЦПУ выполняет ОВ программного цикла с наименьшим номером (обычно "Main" ОВ 1). ЦПУ выполняет другие ОВ программного цикла последовательно (в порядке номеров) в рамках программного цикла. Обработка программы циклична, так что событие программного цикла имеет место в следующих случаях:

- Когда завершается обработка последнего ОВ запуска
- Когда завершается обработка последнего ОВ программного цикла

Таблица 5- 2 Стартовая информация для ОВ программного цикла

Вход	Тип данных	Описание
Initial_Call	Bool	Истина для первоначального вызова ОВ
Remanence	Bool	Истина, если доступны сохраняемые данные

5.1.3.2. ОВ обработки запуска

ОВ запуска обрабатываются однократно, когда рабочий режим ЦПУ изменяется из STOP в RUN, включая подачу питания в режиме RUN и переходы STOP-в-RUN посредством команд. После завершения обрабатывается главный "Программный цикл".

События запуска

Событие запуска происходит один раз при переходе из STOP в RUN и заставляет ЦПУ выполнять ОВ запуска. Вы можете сконфигурировать несколько ОВ для события запуска. ОВ запуска обрабатываются в порядке номеров.

Таблица 5- 3 Стартовая информация для ОВ запуска

Вход	Тип данных	Описание
LostRetentive	Bool	Истина, если сохраняемые данные потеряны
LostRTC	Bool	Истина, если дата и время потеряны

5.1.3.3. ОВ обработки прерывания с задержкой по времени

ОВ прерывания с задержкой по времени обрабатываются по истечению задержки, которую Вы конфигурируете.

События прерывания с задержкой по времени

Вы конфигурируете события прерывания с задержкой по времени, которые наступят после того, как истекло заданное время задержки. Вы назначаете время задержки с помощью инструкции SRT_DINT. События с задержкой по времени прерывают программный цикл, чтобы обработать соответствующий ОВ прерывания с задержкой по времени. Вы можете привязать только один ОВ к событию с задержкой по времени. ЦПУ поддерживает четыре события с задержкой по времени.

Таблица 5- 4 Стартовая информация для ОВ прерывания с задержкой по времени

Вход	Тип данных	Описание
Sign	Word	Идентификатор , передаваемый в триггерный вызов SRT_DINT

5.1.3.4. ОВ обработки циклического прерывания

ОВ циклического прерывания обрабатываются с заданным интервалом. Вы можете сконфигурировать в общей сложности до четырех событий циклических прерываний, с одним ОВ, соответствующим каждому из событий.

События циклических прерываний

События циклических прерываний позволяют Вам сконфигурировать выполнение ОВ прерывания с заданным временем цикла. Вы задаете начальное время цикла, когда создаете ОВ циклического прерывания. Циклическое событие прерывает программный цикл и выполняет соответствующий ОВ циклического прерывания. Обратите внимание на то, что событие циклического прерывания находится в классе с более высоким приоритетом, чем событие программного цикла.

Вы можете прикрепить только один ОВ обработки прерывания к циклическому событию.

Вы можете назначить фазовый сдвиг для каждого циклического прерывания так, чтобы обработка циклических прерываний могла быть смещена от друг относительно друга на величину фазового сдвига. Например, если у Вас есть циклические события с интервалами 1 мс и 2 мс, то каждые две миллисекунды, оба события имеют место одновременно. Если Вы выполните фазовый сдвиг события с интервалом 1 мс на 500 мкс, а с интервалом 2 мс на 0 мкс, то эти два события никогда не наступят одновременно.

Фазовый сдвиг по умолчанию равен 0. Чтобы изменить начальный фазовый сдвиг или изменить начальное время цикла для циклического события, щелкните правой кнопкой мыши по ОВ циклического прерывания в дереве проекта, выберите "Properties" из контекстного меню, затем щелкните по "Cyclic interrupt" и введите новые начальные значения. Вы можете также опросить и изменить время сканирования и фазовый сдвиг из Вашей программы, используя инструкции Опрос циклического прерывания (QRY_CINT) и Установка циклического прерывания (SET_CINT). Значения времени сканирования и фазового сдвига, заданные инструкцией SET_CINT, не сохраняются после цикла подачи питания или перехода в режим STOP; значения времени сканирования и фазового сдвига возвращаются к начальным значениям после цикла подачи питания или перехода в режим STOP. ЦПУ поддерживает в общей сложности четыре события циклических прерываний.

5.1.3.5. ОВ обработки аппаратных прерываний

ОВ аппаратных прерываний обрабатываются, когда имеет место соответствующее аппаратное событие. ОВ аппаратных событий выполняет прерывание нормального циклического выполнения программы в качестве реакции на сигнал от аппаратного события.

События аппаратных прерываний

Изменения в аппаратных средствах, такие как нарастающий или спадающий фронт входного канала или событие в HSC (Высокоскоростной счетчик) вызывают события аппаратного прерывания. S7-1200 поддерживает один ОВ перерыв для каждого события аппаратного прерывания. Вы разрешаете аппаратные события в конфигурации устройства и назначаете ОВ для события в конфигурации устройства или с помощью инструкции АТТАЧН в пользовательской программе. ЦПУ поддерживает несколько событий аппаратных прерываний. Модель CPU и число каналов ввода точно определяют доступные события.

Есть следующие пределы для событий аппаратных прерываний:

Фронты:

- События нарастающего фронта: максимум 16
- События спадающего фронта: максимум 16

События HSC:

- CV=PВ: максимум 6
- Изменено направление: максимум 6
- Внешний сброс: максимум 6

5.1.3.6. ОВ обработки прерываний ошибки времени

Если сконфигурирован, то ОВ обработки ошибки времени (ОВ 80) выполняется, когда либо цикл сканирования превышает максимальное время цикла, либо наступает событие ошибки времени. Если он вызван, то обрабатывается, прерывая нормальное циклическое выполнение программы или любой другой ОВ обработки события.

Возникновение любого из этих событий генерирует запись в диагностический буфер, описывающую событие. Запись в диагностический буфер генерируется независимо от существования ОВ обработки прерывания вследствие ошибки времени.

События прерывания вследствие ошибки времени

Возникновение любого из нескольких различных состояний ошибки времени приводит к событию ошибки времени:

- Цикл сканирования превышает максимальное время цикла
Состояние "максимальное время цикла превышено" наступает, если цикл программы не завершается в течение заданного максимального времени цикла сканирования. Смотри раздел "Контроль времени цикла" в Руководстве по системному S7-1200 (стр. 101) для получения более подробной информации относительно состояния максимального времени цикла; как конфигурировать максимальное время цикла сканирования в свойствах ЦПУ и как сбросить таймер цикла.
- ЦПУ не может запустить запрашиваемый ОВ, поскольку второе прерывание по времени (циклическое или с временной задержкой) запускается прежде, чем ЦПУ завершает обработку ОВ для первого прерывания
- Произошло переполнение очереди
Состояние "произошло переполнение очереди" наступает, если прерывания происходят быстрее, чем ЦПУ может обработать их. ЦПУ ограничивает количество ожидающих (в очереди) событий при помощи различных очередей для каждого типа событий. Если событие происходит, когда соответствующая очередь полна, ЦПУ генерирует событие ошибки времени.

Все события ошибки времени вызывают исполнение ОВ обработки прерывания ошибки времени, если он существует. Если ОВ обработки прерывания ошибки времени не существует, то конфигурация ЦПУ определит его реакцию на ошибку времени:

- Конфигурацией ЦПУ по умолчанию для ошибок времени, таких как старт второго циклического прерывания, прежде чем ЦПУ закончил обработку первого, является оставаться в RUN.
- Конфигурацией ЦПУ по умолчанию для превышения максимального времени является переход в STOP.

Пользовательская программа может продлить время выполнения цикла программы до десяти раз от сконфигурированного максимального времени цикла, посредством выполнения инструкции RE_TRIGR (стр. 288), чтобы перезапустить монитор времени

цикла. Однако, если два события "максимальное время цикла превышено" происходят в пределах того же самого цикла программы, не сбрасывая таймер цикла, то ЦПУ переходит в STOP, независимо от того, существует ли ОВ обработки прерывания ошибки времени. См. раздел "Контроль времени цикла" в Руководстве по системному S7-1200 (стр. 101).

ОВ обработки прерывания ошибки времени включает в себя стартовую информацию, которая помогает Вам определить, какое событие и ОВ произвели ошибку времени. Вы можете программировать инструкции в ОВ, чтобы оценить эту стартовую информацию и принять соответствующие меры.

Таблица 5- 5 Стартовая информация для ОВ ошибки времени (ОВ 80)

Вход	Тип данных	Описание
fault_id	BYTE	16#01 – максимальное время цикла превышено 16#02 - запрошенный ОВ не может быть запущен 16#07 и 16#09 – произошло переполнение очереди
csg_OBnr	OB_ANY	Количество ОВ, которые выполнялись, когда произошла ошибка
csg_prio	UINT	Приоритет ОВ, вызвавшего ошибку

Чтобы включать ОВ обработки ошибки времени в Ваш проект, Вы должны добавить прерывание по ошибке времени посредством двойного щелчка по "Add new block" под "Program blocks" в дереве, последующего выбора "Organization block", а затем - "Time error interrupt".

Приоритет для нового ЦПУ V4.0 равен 22. Если Вы заменяете ЦПУ V3.0 на ЦПУ V4.0 (стр. 1287), то приоритет равняется 26; приоритету, который был действительным для V3.0. В любом случае, поле приоритета редактируемое, и Вы можете установить любое значение приоритета в диапазоне от 22 до 26.

5.1.3.7. ОВ обработки диагностического прерывания

ОВ обработки прерывания ошибки диагностики выполняется, когда ЦПУ обнаруживает диагностическую ошибку, или, если способный к диагностике модуль распознает ошибку, и Вы разрешили для модуля прерывание ошибки диагностики. ОВ обработки ошибки диагностики прерывает нормальное циклическое выполнение программы. Вы можете включать инструкцию STP в ОВ обработки ошибки диагностики, чтобы перевести, при необходимости, ЦПУ в режим STOP, при получении этого типа ошибки.

Если Вы не включаете ОВ обработки прерывания ошибки диагностики в Вашу программу, то ЦПУ игнорирует ошибку и остается в режиме RUN.

События ошибки диагностики

Аналоговые (локальные), PROFINET, PROFIBUS и некоторые цифровые (локальные) устройства способны обнаруживать и сообщать о диагностических ошибках. Возникновение или исчезновение любого из нескольких различных состояний ошибки диагностики приводят к событию ошибки диагностики. Поддерживаются следующие диагностические ошибки:

- Отсутствие питания пользователя
- Выход за верхний предел
- Выход за нижний предел
- Обрыв провода

- Короткое замыкание

События ошибки диагностики вызывают выполнение ОВ обработки прерывания ошибки диагностики (ОВ 82), если он существует. Если ОВ не существует, ЦПУ игнорирует ошибку.

Чтобы включать ОВ обработки ошибки диагностики в Ваш проект, Вы должны добавить прерывание по ошибке диагностики посредством двойного щелчка по "Add new block" под "Program blocks" в дереве, последующего выбора "Organization block", а затем - "Diagnostic error interrupt".

Примечание

Диагностические ошибки для многоканальных локальных аналоговых устройств (ввод / вывод, терморезисторы и термопары)

ОВ обработки прерывания ошибки диагностики может обработать диагностическую ошибку только одного канала за один раз.

Если в двух каналах многоканального устройства присутствует ошибка, то вторая ошибка вызывает ОВ обработки прерывания ошибки диагностики только при следующих условиях: первая ошибка канала очищается, ОВ обработки прерывания ошибки, вызванной первой, завершен, и вторая ошибка все еще существует.

ОВ обработки прерывания ошибки диагностики включает в себя стартовую информацию, которая помогает Вам определить, вызвано ли событие возникновением или исчезновением ошибки, а также устройство и канал, которые сообщили об ошибке. Вы можете запрограммировать инструкции в ОВ обработки прерывания ошибки диагностики, чтобы оценить эти стартовые значения и принять соответствующие меры.

Примечание

Стартовая информация ОВ ошибки диагностики ссылается на субмодуль в целом, если никакое диагностическое событие не находится на рассмотрении

В V3.0 стартовая информация для уходящего события ошибки диагностики всегда указывала на источник события. В V4.0, если уходящее событие оставляет субмодуль без ожидающей оценки диагностики, стартовая информация ссылается на субмодуль в целом (16#8000), даже если источник события был определенным каналом.

Например, если обрыв провода вызывает событие ошибки диагностики в канале 2, то ошибка затем корректируется, и событие ошибки диагностики очищается, стартовая информация не будет ссылаться на канал 2, но на субмодуль (16#8000).

Таблица 5- 6 Стартовая информация для ОВ прерывания ошибки диагностики

Вход	Тип данных	Описание
IOWstate	WORD	Состояние ввода / вывода устройства: • Bit 0 = 1, если конфигурация корректна, и = 0, если конфигурация уже некорректна. • Bit 4 = 1, if an error is preseесли присутствует ошибка (к примеру, обрыв провода). (Bit 4 = 0, если ошибок нет.) • Bit 5 = 1, если конфигурация некорректна, и = 0, если конфигурация вновь корректна. • Bit 6 = 1, если имела место ошибка доступа к вводу/выводу. Обратитесь к LADDR для получения аппаратного идентификатора ввода/вывода с ошибкой доступа. (Bit 6 = 0, если ошибок нет.)
LADDR	HW_ANY	Аппаратный идентификатор устройства или функционального модуля, который сообщил об ошибке ¹
Channel	UINT	Номер канала
MultiError	BOOL	ИСТИНА, если присутствует более одной ошибки

¹ Входная информация LADDR содержит аппаратный идентификатор устройства или функционального модуля, который возвратил ошибку. Аппаратный идентификатор назначается автоматически, когда компоненты вставлены в устройство, или представление сети, и появляются на вкладке Constants в тегх ПЛК. Имя аппаратному идентификатору также назначается автоматически. Эти записи во вкладке Constants в тегх ПЛК не могут быть изменены.

5.1.3.8. ОВ обработки удаления или вставки модулей

ОВ "удаления или вставки модулей" выполняется, когда сконфигурированный и неотключенный модуль распределенного ввода/вывода или submodule (PROFIBUS, PROFINET, AS-i) генерирует событие, связанное со вставкой или удалением модуля

Событие удаления или вставки модулей

Следующие состояния генерируют событие удаления или вставки модулей:

- Кто-то удаляет или вставляет сконфигурированный модуль
- Сконфигурированный модуль физически отсутствует в стойке расширения
- В стойке расширения находится несовместимый модуль, который не соответствует сконфигурированному модулю
- В стойке расширения находится совместимый со сконфигурированным модуль, но конфигурация не позволяет замены
- У модуля или submodule есть ошибки параметризации

Если Вы не запрограммировали этот ОВ, то ЦПУ остается в режиме RUN, когда наступает любое из этих состояний для сконфигурированного и неотключенного модуля распределенного ввода/вывода.

Независимо от того, запрограммировали ли Вы этот ОВ, ЦПУ перейдет в режим STOP, когда любое из этих состояний наступает для модуля в центральной стойке.

Таблица 5- 7 Стартовая информация ОВ обработки удаления или вставки модуля

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Event_Class	Byte	16#38: модуль вставлен 16#29: модуль удален
Fault_ID	Byte	Идентификатор ошибки

5.1.3.9. ОВ обработки отказа стойки или станции

ОВ "отказа стойки или станции" выполняется, когда ЦПУ обнаруживает отказ или потерю обмена для распределенной стойки или станции.

Событие отказа стойки или станции

ЦПУ генерирует событие ошибки стойки или станции, когда он обнаруживает одно из следующих состояний:

- Отказ DP мастер-системы или PROFINET IO системы (в случае наступающего или уходящего события).
- Отказ ведомого устройства DP или IO устройства (в случае наступающего или уходящего события)
- Отказ некоторых субмодулей I-устройства PROFINET

Если Вы не запрограммировали этот ОВ, ЦПУ остается в режиме RUN, когда наступает любое из этих состояний.

Таблица 5- 8 Стартовая информация ОВ обработки отказа стойки или станции

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Event_Class	Byte	16#32: Активация ведомого устройства DP или IO устройства 16#33: Деактивация ведомого устройства DP или IO устройства 16#38: уходящее событие 16#39: наступающее событие
Fault_ID	Byte	Идентификатор ошибки

5.1.3.10. ОВ обработки прерывания по времени суток

ОВ обработки прерывания по времени суток обрабатываются на основании сконфигурированных условий времени суток. ЦПУ поддерживает два ОВ прерываний по времени суток.

События времени суток

Вы можете сконфигурировать событие прерывания по времени суток, которое происходит однократно в указанную дату или время или циклически с одним из следующих периодов:

- Каждую минуту: Каждую минуту происходит прерывание.
- Ежечасно: Каждый час происходит прерывание.
- Ежедневно: прерывание происходит каждый день в требуемое время (час и минута).
- Еженедельно: прерывание происходит каждую неделю в требуемое время в указанный день недели (например, каждый вторник в 4:30 днем).
- Ежемесячно: прерывание происходит каждый месяц в требуемое время в указанный день месяца. Число должно быть между 1 и 28, включительно.
- Каждый конец месяца: прерывание происходит в последний день каждого месяца в требуемое время.
- Ежегодно: прерывание происходит каждый год в указанную дату (месяц и день). Вы не можете определить дату 29 февраля.

Таблица 5- 9 Стартовая информация ОВ обработки прерывания по времени суток

Вход	Тип данных	Описание
CaughtUp	Bool	Вызов ОВ подхвачен, из-за перевода часов вперед
SecondTimes	Bool	Вызов ОВ стартовал повторно, из-за перевода часов назад

5.1.3.11. ОВ обработки изменения состояния

ОВ обработки изменения состояния выполняются, если ведомое устройство DPV1 или PNIO вызывают прерывание состояния. Это могло бы иметь место, если компонент (модуль или стойка) ведомого устройства DPV1 или PNIO изменяет свой рабочий режим, например из RUN в STOP.

События состояния

Для получения подробной информации о событиях, которые вызывают прерывание при изменении состояния, обратитесь к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5- 10 Стартовая информация ОВ обработки изменения состояния

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Признак тревоги

5.1.3.12. ОВ обработки обновления

ОВ обработки обновления выполняются, если ведомое устройство DPV1 или PNIO генерирует прерывание обновления.

События обновления

Для получения подробной информации о событиях, которые вызывают прерывание обновления, обратитесь к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5- 11 Стартовая информация ОВ обработки изменения состояния

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Признак тревоги

5.1.3.13. ОВ обработки специфического для профиля прерывания

ОВ обработки специфического для профиля прерывания выполняются, если DPV1 или PNIO генерирует специфическое для профиля прерывание.

События, связанные с профилем

Для получения подробной информации о событиях, которые вызывают специфическое для профиля прерывание, обратитесь к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5- 12 Стартовая информация ОВ обработки специфического для профиля прерывания

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Признак тревоги

5.1.3.14. Приоритеты обработки событий и организации очереди

Обработка в ЦПУ управляется событиями. Событие вызывает ОВ обработки прерывания, которая будет выполнен. Вы можете определить ОВ обработки прерывания для события во время создания блока, во время конфигурации устройства, или с помощью инструкций ATTACH или DETACH. Некоторые события происходят на регулярной основе, такие как программный цикл или циклические события. Другие события происходят лишь однократно, например событие запуска и события с временной задержкой. Некоторые события происходят, когда аппаратные средства генерируют событие, такие как событие фронта во входном канале или события в высокоскоростном счетчике. Такие события, как ошибка диагностики и ошибка времени происходят только при возникновении ошибки. Приоритеты и очереди событий используются, чтобы определить порядок обработки для ОВ прерываний по событию.

ЦПУ обрабатывает события в порядке приоритета, где 1 - это самый низкий приоритет, а 26 - самый высокий приоритет. До V4.0 ЦПУ S7-1200 каждый тип ОВ принадлежал к определенному классу приоритета (1 - 26). Начиная с версии V4.0, Вы можете назначить класс приоритета для каждого конфигурируемого ОВ. Вы задаете номер приоритета в атрибутах свойств ОВ.

Прерываемые и непрерывные режимы обработки

ОВ (стр. 88) выполняются в порядке приоритета событий, которые их вызывают. Начиная с версии V4.0, Вы можете конфигурировать обработку ОВ, которая может быть прерываемой или непрерывной. Обратите внимание на то, что ОВ программного цикла всегда прерываемые, но Вы можете сконфигурировать все другие ОВ, как обрабатываемые непрерывно так и с возможностью прерывания

Если Вы устанавливаете прерываемый режим, то, если ОВ обрабатывается и наступает событие с более высоким приоритетом, прежде чем завершается обработка ОВ, то работающий ОВ прерывается, чтобы позволить обработке ОВ с более высоким приоритетом. ОВ обработки события с более высоким приоритетом выполняется, и по его завершении обработка ОВ, который был прерван, продолжается. Когда наступают несколько событий, в то время как выполняется прерываемый ОВ, ЦПУ обрабатывает эти события в порядке приоритета.

Если Вы не выбираете режим прерывания обработки, то ОВ работает до завершения, когда он вызван независимо от любых других событий, которые вызываются в течение времени его обработки.

Рассмотрите следующие два случая, когда события прерывания вызывают циклический ОВ и ОВ с временной задержкой. В обоих случаях ОВ с временной задержкой (ОВ 201) не имеет никакой привязки к разделу образа процесса (стр. 79) и выполняется с приоритетом 4. Циклический ОВ (ОВ 200) имеет привязку к разделу образа процесса PIP1 и выполняется с приоритетом 2. Следующие иллюстрации показывают различие в обработке между непрерывным и прерываемыми способами выполнения:

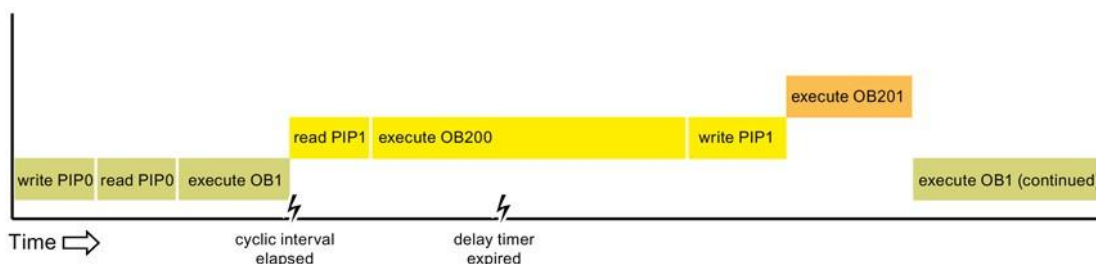


Рисунок 5-1 Случай 1: Непрерывная обработка ОВ

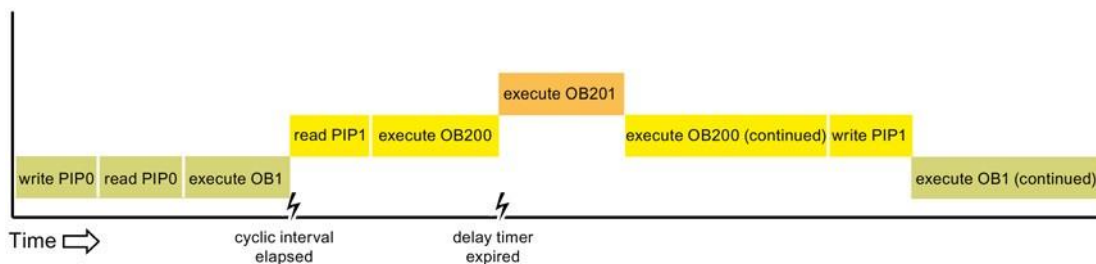


Рисунок 5-2 Случай 2: Прерываемая обработка ОВ

Примечание

Если Вы конфигурируете непрерывный способ выполнения ОВ, то ОВ ошибки времени не может прервать обработку ОВ, кроме ОВ программного цикла. До V4.0 ЦПУ S7-1200 ОВ ошибки времени мог прервать выполнение любого ОВ. Начиная с версии V4.0, Вы должны сконфигурировать прерываемое выполнение ОВ, если Вы хотите, чтобы ОВ ошибки времени (или ОВ с более высоким приоритетом) был в состоянии прервать выполнение ОВ, которые не являются ОВ программного цикла.

Понимание приоритетов обработки событий и организации очереди

ЦПУ ограничивает количество ожидающих (в очереди) событий из единственного источника, используя различные очереди для каждого типа событий. После достижения предела ожидающих событий данного типа следующее событие теряется. Вы можете использовать ОВ обработки прерывания ошибки времени (стр. 91), чтобы отреагировать на переполнение очереди.

Каждое событие ЦПУ имеет назначенный приоритет. Как правило, сервисные события ЦПУ обладают наивысшим приоритетом. Сервисные события ЦПУ с тем же самым приоритетом обслуживаются по принципу "первый пришел, первый обслужен".

Таблица 5- 13 ОВ обработки событий

Событие	Разрешенное количество	Приоритет ОВ по умолчанию
Программный цикл	1 событие программного цикла Допустимо несколько ОВ	1 ⁴
Запуск	1 событие запуска ¹ Допустимо несколько ОВ	1 ⁴
Задержка по времени	До 4 событий 1 ОВ на событие	3
Циклическое прерывание	До 4 событий 1 ОВ на событие	8
Аппаратное прерывание	До 50 событий аппаратных прерываний ² 1 ОВ на событие, но Вы можете использовать тот же самый ОВ для нескольких событий	18
		18
Ошибка времени	1 событие (только если сконфигурировано) ³	22 или 26 ⁴
Ошибка диагностики	1 событие (только если сконфигурировано)	5
Удаление или вставка модулей	1 событие	6
Отказ стойки или станции	1 событие	6
Время суток	До 2 событий	2
Состояние	1 событие	4
Обновление	1 событие	4
Профиль	1 событие	4

¹ Событие запуска и событие программного цикла никогда не происходят в одно и то же время, потому что событие запуска выполняется до завершения перед началом обработки события программного цикла.

² У Вас может быть больше чем 50 ОВ обработки событий аппаратных прерываний, если Вы используете инструкции DETACH и ATTACH.

³ Вы можете сконфигурировать ЦПУ так, чтобы он остался в режиме RUN, если цикл сканирования превышает максимальное время цикла, или Вы можете использовать инструкцию RE_TRIGR для перезапуска времени контроля цикла. Однако ЦПУ переходит в режим STOP, когда во второй раз один цикл сканирования превышает максимальное время цикла сканирования.

⁴ Приоритет для нового ЦПУ V4.0 или V4.1 равен 22. Если Вы замените ЦПУ V3.0 на V4.0 или V4.1, приоритет равняется 26: приоритет, который был действительным для V3.0. В любом случае поле приоритета является редактируемым, и Вы можете установить любое значение приоритета в диапазоне от 22 до 26.

Обратитесь к разделу "Замена ЦПУ V3.0 на ЦПУ V4.1 (стр. 1287)" для получения более подробных сведений.

Кроме того, ЦПУ распознает другие события, у которых нет связанного ОВ. Следующая таблица описывает эти события и соответствующие действия ЦПУ:

Таблица 5- 14 Дополнительные события

Событие	Описание	Действие ЦПУ
Ошибка доступа к вводу/выводу	Ошибка прямого чтения / записи ввода / вывода	ЦПУ регистрирует первое проявление в диагностическом буфере и остается в режиме RUN.
Ошибка максимального времени цикла	ЦПУ превышает заданное время цикла дважды	ЦПУ регистрирует ошибку в диагностическом буфере и переходит в режим STOP.
Ошибка доступа к периферии	Ошибка ввода/вывода при обновлении образа процесса	ЦПУ регистрирует первое проявление в диагностическом буфере и остается в режиме RUN.
Программная ошибка	Ошибка обработки программы	Если блок с ошибкой обеспечивает обработку ошибок, он обновляет структуру ошибок; в противном случае ЦПУ регистрирует ошибку в диагностическом буфере и остается в режиме RUN.

Задержка прерывания

Задержка прерываний по событию (время от уведомления ЦПУ что событие наступило до момента начала обработки ЦПУ первой инструкции в ОВ, обслуживающем событие) составляет приблизительно 175 мкс, при условии, что программный цикл ОВ является единственной сервисной процедурой, активной в момент события прерывания.

5.1.4. Контроль и конфигурирование времени цикла

Время цикла - это время, которое требуется операционной системе ЦПУ, чтобы выполнить циклическую фазу режима RUN. ЦПУ обеспечивает два метода контроля времени цикла:

- Максимальное время цикла сканирования
- Минимальное время цикла сканирования

Контроль цикла сканирования начинается после того, как событие запуска завершено. Настройка этой функции выполняется в "конфигурации устройства" для ЦПУ в пункте "Cycle time".

ЦПУ всегда контролирует цикл сканирования и реагирует, если максимальное время цикла сканирования превышено. Если заданное максимальное время цикла сканирования превышено, ошибка генерируется и обрабатывается одним из двух способов:

- Если пользовательская программа не содержит ОВ обработки прерывания ошибки времени, то событие ошибки таймера генерирует запись в диагностический буфер, но ЦПУ остается в режиме RUN. (Вы можете изменить конфигурацию ЦПУ, чтобы он переходил в режим STOP, когда обнаруживает ошибку времени, но настройкой по умолчанию является сохранение режима RUN.)
- Если пользовательская программа содержит ОВ обработки прерывания ошибки времени, то ЦПУ выполняет его.

Инструкция RE_TRIGR (стр. 288) (перезапуск контроля времени цикла) позволяет Вам сбрасывать таймер, который измеряет время цикла. Если истекшее время для текущей обработки цикла программы в десять раз меньше, чем сконфигурированное максимальное время цикла сканирования, инструкция RE_TRIGR перезапускает контроль времени цикла, и возвращается с ENO = ИСТИНА. В противном случае инструкция RE_TRIGR не перезапускает контроль времени цикла. Она возвращает ENO = ЛОЖЬ.

Как правило, цикл сканирования выполняется настолько быстро, насколько это возможно, и следующий цикл сканирования начинается, как только заканчивается текущий. В зависимости от пользовательской программы и коммуникационных задач, период времени для цикла сканирования может измениться от скана к скану. Чтобы устранить это изменение, ЦПУ поддерживает дополнительное минимальное время цикла сканирования. Если Вы активируете эту дополнительную опцию и предусматриваете минимальное время цикла сканирования в мс, то ЦПУ приостанавливается после выполнения ОВ программного цикла пока не истечет минимальное время цикла сканирования, прежде чем повторить программный цикл.

Если ЦПУ завершает нормальный цикл сканирования быстрее, чем указанное минимальное время цикла, то ЦПУ тратит дополнительное время цикла сканирования, выполняя оперативную диагностику и/или обрабатывая коммуникационные запросы.

Если ЦПУ не завершает цикл сканирования за указанное минимальное время цикла, то ЦПУ завершает скан как обычно (включая обработку коммуникаций) и не активирует реакцию системы в результате превышения минимального времени сканирования. Следующая таблица определяет диапазоны и настройки по умолчанию для функций контроля времени цикла.

Таблица 5- 15 Диапазон времени цикла

Время цикла	Диапазон (мс)	По умолчанию
Минимальное время цикла сканирования ¹	от 1 до 6000	150 мс
Минимальное время цикла сканирования ²	от 1 до максимального времени цикла сканирования	Неактивно

¹ Максимальное время цикла сканирования всегда активировано. Задайте время цикла от 1 до 6000 мс. По умолчанию равно 150 мс.

² Минимальное время цикла сканирование дополнительное, и деактивировано по умолчанию. При необходимости задайте время цикла от 1 мс до максимального времени цикла сканирования.

Конфигурирование времени цикла и коммуникационной нагрузки

Вы используете свойства ЦПУ в конфигурации устройств, чтобы задать следующие параметры:

- **Cycle:** Вы можете ввести максимальное время контроля цикла сканирования. Вы можете также активировать и ввести минимальное время цикла сканирования.

- **Communication load:** Вы можете задать процент времени, которое будет выделено для коммуникационных задач.

Для получения дополнительной информации о цикле сканирования, смотри "Контроль времени цикла". (стр. 101).

5.1.5. Память ЦПУ

Управление памятью

ЦПУ обеспечивает следующие области памяти, чтобы сохранить пользовательскую программу, данные и конфигурацию:

- Загрузочная память - это энергонезависимое ЗУ для пользовательской программы, данных и конфигурации. Когда Вы загружаете проект в ЦПУ, он вначале сохраняет программу в загрузочной области памяти. Эта область расположена или на карте памяти (если есть) или в ЦПУ. ЦПУ поддерживает эту энергонезависимую область памяти при потере питания. Карта памяти обеспечивает большее место для хранения, чем встроенная память ЦПУ.
- Рабочая память -это энергозависимое ЗУ для некоторых элементов пользовательского проекта при обработке пользовательской программы. ЦПУ копирует некоторые элементы проекта из загрузочной памяти в рабочую память. Содержимое этой энергозависимой области теряется, когда выключается питание, и восстанавливается ЦПУ, когда питание включается вновь.
- Сохраняемая память - это энергонезависимое ЗУ для ограниченного количества значений из рабочей памяти. ЦПУ использует сохраняемую область памяти, чтобы сохранить значения выбранных пользователем ячеек памяти на время отключения питания. Когда происходит выключение или перебой питания, ЦПУ восстанавливает эти сохраняемые значения при включении питания.

Чтобы показать использование памяти для скомпилированного программного блока, щелкните правой кнопкой мыши по блоку в папке "Program blocks" в дереве STEP 7 проекта и выберите "Resources" в контекстном меню. В разделе Compilation отображаются размер в загрузочной и рабочей памяти для скомпилированного блока.

Чтобы показать использование памяти для ЦПУ он-лайн, дважды щелкните по "Online and diagnostics" в STEP 7, раскройте "Diagnostics" и выберите "Memory".

Сохраняемая память

Вы можете избежать потери данных после перебоя в питании, отметив определенные данные, как сохраняемые. ЦПУ позволяет Вам сконфигурировать следующие данные, как сохраняемые:

- Битовая память (M): Вы можете определить размер сохраняемой памяти для битовой памяти в таблице тегов ПЛК или в списке назначения. Сохраняемая битовая память всегда начинается в MB0 и занимает последовательно конкретное количество байтов. Определите эту значение в таблице тегов ПЛК или в списке назначения, щелкнув по символу "Retain" в панели инструментов. Введите количество M байтов для сохранения, начиная с MB0.

Примечание: Для любого блока Вы можете отобразить список назначения, выбрав блок в папке Program Blocks, а затем выбирая команду меню **Tools > Assignment list**.

- Теги функционального блока (FB): Если FB был создан с настройкой "Optimized", то редактор интерфейса для этого FB включает колонку "Retain". В этой колонке Вы можете выбрать или "Retentive", "Non-retentive" или "Set in IDB" индивидуально для каждого тега. Экземплярный DB, который был создан, когда этот FB помещен в программный редактор, также показывает эту колонку сохранения. Вы можете

изменить состояние сохранения тега из редактора интерфейса экземплярного DB только в том случае, если Вы выбрали "Set in IDB" (Установка в экземплярном блоке данных) в поле Retain для тега в оптимизированном FB.

Если FB был создан с настройкой "Standard - compatible with S7-300/400", то редактор интерфейса для этого FB не содержит колонку "Retain". Экземплярный DB, созданная, когда этот FB вставляется в программном редакторе, отображает колонку "Retain", которая доступна для редактирования. В этом случае выбор опции "Retain" для любого тега приводит к выбору всех тегов. Точно так же снятие опции для любого тега приводит к деактивации сохраняемости для всех тегов. Для FB, который был сконфигурирован, как "Standard - compatible with S7-300/400", Вы можете изменить состояние сохраняемости в редакторе экземплярного DB, но все теги принимают вместе ту же самую настройку.

После того, как Вы создадите FB, Вы не можете изменить опцию "Standard - compatible with S7-300/400". Вы можете только выбрать эту опцию только при создании FB. Чтобы определить был ли сконфигурирован существующий FB с опцией "Optimized" или "Standard - compatible with S7-300/400", щелкните правой кнопкой мыши по FB в дереве проекта, выберите "Properties", а затем - "Attributes". Если в поле "Optimized block access" установлен флажок, то блок оптимизирован. Иначе, он является стандартным и совместимым с S7-300/400 ЦПУ.

- Теги глобального блока данных: поведение глобального DB относительно назначения состояния сохраняемости подобно FB. В зависимости от настройки доступа к блоку, Вы можете определить состояния сохраняемости или для отдельных тегов или для всех тегов глобального блока данных.
 - Если Вы выбираете "Optimized" при создании DB, то Вы можете выбрать состояние сохраняемости для каждого отдельного тега.
 - Если Вы выбираете "Standard - compatible with S7-300/400" при создании DB, то выбор состояния сохраняемости применяется ко всем тегам DB; либо все теги сохраняемые, либо не один из них не сораняется.

ЦПУ поддерживает в общей сложности 10240 байтов сохраняемых данных. Чтобы просмотреть, сколько из них доступно из таблицы тегов ПЛК или списка назначения, щелкните по символу "Retain" в панели инструментов. Хотя это область определена для сохранения M памяти, вторая колонка указывает на общую сохраняемую память, доступную для M и DB вместе. Обратите внимание на то, чтобы получить точное значение, Вы должны скомпилировать все блоки данных с сохраняемыми тегами.

Примечание

Загрузка программы не очищает и не вносит какие-либо изменения в существующие значения в сохраняемой памяти. Если Вы хотите очистить сохраняемую память перед загрузкой, то выполните сброс Вашего ЦПУ на заводские настройки до загрузки программы.

5.1.5.1. Системная и тактовая память

Вы используете свойства ЦПУ, чтобы активировать байты для "системной памяти" и "тактовой памяти". Ваша программная логика может сослаться на отдельные биты этих функций по их именам тега.

- Вы можете назначить один байт в М области для системной памяти. Байт системной памяти обеспечивает следующие четыре бита, на которые может сослаться Ваша пользовательская программа по следующим именам тега:
 - Первый цикл: (имя тега "FirstScan") бит установлен в 1 на время первого скана после того, как завершится ОВ запуска. (После выполнения первого скана бит "первый скан" устанавливается в 0.)
 - Статус диагностики изменился: (имя тега: "DiagStatusUpdate"), установлен в 1 на один скан после того, как ЦПУ регистрирует диагностическое событие. Поскольку ЦПУ не устанавливает бит "DiagStatusUpdate" до конца первого скана всех ОВ программного цикла, Ваша пользовательская программа не может обнаружить, было ли изменение состояния диагностики во время выполнения ОВ запуска или во время первого скана ОВ программного цикла.
 - Всегда 1 (высокий уровень): (имя тега "AlwaysTRUE") тег всегда установлен в 1.
 - Всегда 0 (низкий уровень): (имя тега "AlwaysFALSE") тег всегда установлен в 0.
- Вы можете назначить один байт в М области для тактовой памяти. Каждый бит в байте, сконфигурированного как тактовая память, генерирует прямоугольный импульс. Байт тактовой памяти обеспечивает 8 различных частот от 0.5 Гц (медленно) до 10 Гц (быстро). Вы можете использовать эти биты в качестве битов управления, особенно, при комбинировании с инструкциями выделения фронта, чтобы активировать действия в пользовательской программе на циклической основе.

ЦПУ инициализирует эти байты при переходе из режима STOP в режим STARTUP. Биты тактовой памяти изменяются синхронно по отношению к часам ЦПУ при нахождении в режимах STARTUP и RUN.



ПРЕДУПРЕЖДЕНИЕ

Риски связанные с перезаписью битов системной памяти или тактовой памяти

Перезапись битов системной или тактовой памяти может испортить данные в этих функциях и заставить Вашу пользовательскую программу работать неправильно, что может привести к материальному ущербу и травмам персонала.

Поскольку и тактовая и системная память не зарезервированы в М области памяти, инструкции или коммуникации могут выполнить запись в эти ячейки и испортить данные.

Избегайте записывать данные в эти ячейки, чтобы гарантировать правильную работу этих функций, и всегда используйте схему аварийного останова для Вашего процесса или машины.

Системная память представляет собой байт с битами, которые становятся активными (значение = 1) при определенном событии.

System memory bits

☒ Enable the use of system memory byte

Address of system memory byte (MBx):

First cycle:

Diagnostics status changed:

Always 1 (high):

Always 0 (low):

Таблица 5- 16 Системная память

7	6	5	4	3	2	1	0
Зарезервировано Значение 0				Всегда выкл. Значение 0	Всегда вкл. Значение 1	Индикатор состояния диагностики • 1: Изменилось • 0: Без изменений	Индикатор первого скана • 1: Первый скан после запуска • 0: Не первый скан

Тактовая память представляет собой байт, который циклически активирует и деактивирует отдельные биты с фиксированными интервалами. Каждый тактовый бит генерирует прямоугольный импульс в соответствующем бите М памяти. Эти биты могут использоваться в качестве битов управления, особенно, когда используются вместе с инструкциями выделения фронта, чтобы вызвать действия в пользовательском коде на циклической основе.

Clock memory bits

☒ Enable the use of clock memory byte

Address of clock memory byte (MBx):

10 Hz clock:

5 Hz clock:

2.5 Hz clock:

2 Hz clock:

1.25 Hz clock:

1 Hz clock:

0.625 Hz clock:

0.5 Hz clock:

Таблица 5- 17 Тактовая память

Номер бита	7	6	5	4	3	2	1	0
Имя тега								
Период (с)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Частота (Гц)	0.5	0.625	1	1.25	2	2.5	5	10

Поскольку тактовая память работает асинхронно к циклу ЦПУ, статус тактовой памяти может изменяться несколько раз в течение долгого цикла.

5.1.6. Диагностический буфер

ЦПУ поддерживает диагностический буфер, который содержит запись для каждого диагностического события. Каждая запись включает в себя дату и время наступления события, категорию и описание события. Записи отображаются в хронологическом порядке с более новыми событиями наверху. До 50 последних событий доступны в этом журнале. Когда журнал полон, последнее событие заменяет самое старое событие в журнале. При выключении питания события сохраняются.

Следующие типы событий регистрируются в диагностическом буфере:

- Каждое системное диагностическое событие; например, ошибки ЦПУ и модулей
- Каждое изменение состояния ЦПУ (каждое включение питания, каждый переход в STOP, каждый переход в RUN)

Чтобы обратиться к диагностическому буферу (стр. 1076), Вы должны быть он-лайн. Из представления "Online & diag-nostics", найдите диагностический буфер под "Diagnostics > Diagnostics buffer".

5.1.7. Часы времени суток

ЦПУ поддерживает часы времени суток. Суперконденсатор поставляет энергию, требуемую для поддержки часов, работающих в течение времени, когда питание ЦПУ выключено. Суперконденсатор заряжается пока у ЦПУ есть питание. После того как ЦПУ был запитан по крайней мере на 24 часа, суперконденсатор обладает достаточным зарядом, чтобы поддерживать работоспособность часов в течение, как правило, 20 дней.

STEP 7 устанавливает часы времени суток на системное время, что является значением по умолчанию при поставке или после сброса на заводские установки. Чтобы использовать часы времени суток, Вы должны установить их. Метки времени, такие как метки для записей в диагностический буфер, файлов системного журнала данных и записей в журнале данных основаны на системном времени. Вы устанавливаете время суток из функции "Set time of day" (стр. 1072) в представлении "Online & diagnostics" для ЦПУ, находящегося в он-лайн. Затем STEP 7 вычисляет системное время из введенного Вами времени плюс или минус смещение ОС Windows из UTC (Международное скоординированное время). Настройка времени суток на текущее местное время переводит системное время на UTC, если Ваши параметры настройки ОС Windows для часового пояса и летнего времени соответствуют Вашей местности.

STEP 7 содержит инструкции (стр. 318) для чтения и записи системного времени (RD_SYS_T и WR_SYS_T), чтения местного времени (RD_LOC_T) и установки часового пояса (SET_TIMEZONE). Инструкция RD_LOC_T вычисляет местное время, используя часовой пояс и сдвиг летнего времени, которые Вы устанавливаете в конфигурации "Time of day" в общих свойствах ЦПУ (стр. 162). Эти настройки позволяют Вам установить свой часовой пояс на местное время, дополнительно разрешить переход на летнее время и определить даты и время начала и окончания периода летнего времени. Вы можете также использовать инструкции SET_TIMEZONE для изменения этих настроек.

5.1.8. Конфигурирование выходов для перехода из RUN в STOP

Вы можете сконфигурировать поведение цифровых и аналоговых выходов, когда ЦПУ находится в режиме STOP. Для любого выхода ЦПУ, SB или SM Вы можете выбрать замораживание значения или использование значения замещения:

- Подстановка замещающего выходного значения (по умолчанию): Вы вводите значение замещения для каждого выхода (канала) ЦПУ, SB или SM устройства. Значением замещения по умолчанию для цифровых выходных каналов является OFF, а значением замещения по умолчанию для аналоговых выходных каналов является 0.
- Замораживание выходов, чтобы остаться в последнем состоянии: выходы сохраняют свое текущее значение во время перехода из RUN в STOP. После подачи питания, выходы устанавливаются в значение замещения по умолчанию

Вы конфигурируете поведение выходов в конфигурации устройства. Выберите отдельные устройства и используйте вкладку "Properties", чтобы сконфигурировать выходы для каждого устройства.

Примечание

Некоторые модули распределенного ввода/вывода предлагают дополнительные настройки для реакции на остановку ЦПУ. Выберите их из списка в конфигурации устройства для этих модулей.

При переходе из RUN в STOP ЦПУ сохраняет образ процесса и записывает соответствующие значения как для цифровых, так и для аналоговых выходов, на основании настроек конфигурации.

5.2 Хранение данных, области памяти, ввод/вывод и адресация

5.2.1. Доступ к данным в S7-1200

STEP 7 облегчает символьное программирование. Вы создаете символьные имена или "теги" для адресов данных, либо как теги ПЛК, связанные с адресами в памяти и каналами ввода/вывода, или как локальные переменные, используемые в пределах кодового блока. Чтобы использовать эти теги в Вашей пользовательской программе, просто присвойте имя тега параметру инструкции.

Для лучшего понимания того, как ЦПУ структурирует и адресует области памяти, следующие параграфы объясняют "абсолютное" обращение, на которое ссылаются теги ПЛК. ЦПУ предоставляет несколько возможностей для того, чтобы хранить данные во время выполнения пользовательской программы:

- Глобальная память: ЦПУ предлагает множество специализированных областей памяти, включая входы (I), выходы (Q) и битовую память (M). Эта память доступна всем кодовым блокам без ограничения.
- Таблица тегов ПЛК: Вы можете ввести символьные имена в STEP 7 таблице тегов ПЛК для определенных ячеек памяти. Эти теги глобальны по отношению к STEP 7 программе и позволяют программировать, используя имена, которые являются значащими для Вашего приложения.
- Блок данных (DB): Вы можете включать DB в свою пользовательскую программу, чтобы хранить данные для кодовых блоков. Записанные данные сохраняются, когда выполнение связанного кодового блока завершается. "Глобальный" DB хранит данные, которые могут использоваться всеми кодовыми блоками, в то время как экземплярный DB хранит данные для определенного FB и структурирован параметрами для FB.
- Временная память: Каждый раз, когда вызывается кодовый блок, операционная система ЦПУ выделяет временную или локальную память (L) для использования во время выполнения блока. Когда выполнение кодового блока завершается, ЦПУ перераспределяет локальную память для использования другими кодовыми блоками.

У каждой отдельной ячейки памяти есть уникальный адрес. Ваша пользовательская программа использует эти адреса, чтобы получить доступ к информации в ячейке памяти. Ссылки на области памяти ввода (I) или вывода (Q), такие как I0.3 или Q1.7, получают доступ к образу процесса. Чтобы получить доступ сразу к физическому входу или выходу, добавьте к ссылке ":P" (например, I0.3:P, Q1.7:P или "Stop:P").

Таблица 5- 18 Области памяти

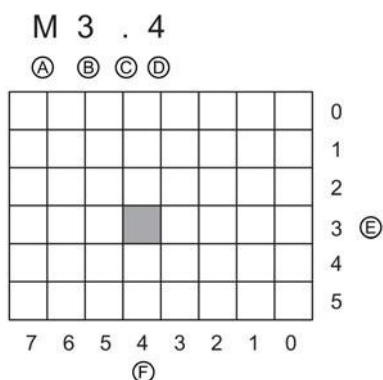
Область памяти	Описание	Форсирование	Сохраняемость
I Вход образа процесса I_:P ¹ (Физический вход)	Скопирован из физических входов в начале цикла сканирования	Нет	Нет
	Непосредственное чтение каналов физического ввода в ЦПУ, SB и SM	Да	Нет
Q Выход образа процесса Q_:P ¹ (Физический выход)	Копируется в физические выходы в начале цикла сканирования	Нет	Нет
	Непосредственная запись в каналы физического вывода в ЦПУ, SB и SM	Да	Нет
M Битовая память	Память данных и сигналов управления	Нет	Да (по желанию)
L Временная память	Временные данные для блока, локальные для этого блока	Нет	Нет
DB Блок данных	Память данных, а также память параметров для FB	Нет	Да (по желанию)

¹ Для непосредственного доступа к (на чтение или запись) физическим входам и физическим выходам, добавьте ":P" к адресу или тегу (например, I0.3:P, Q1.7:P или "Stop:P").

У каждой отдельной ячейки памяти есть уникальный адрес. Ваша пользовательская программа использует эти адреса, чтобы получить доступ к информации в ячейке памяти. Абсолютный адрес состоит из следующих элементов:

- Идентификатор области памяти (такой как, I, Q или M)
- Размер данных, к которым выполняется обращение ("B" для Byte, "W" для Word или "D" для DWord)
- Начальный адрес данных (например, байт 3 или слово 3)

При обращении к биту в адресе для Булева значения, Вы не вводите мнемонику для размера. Вы вводите только область памяти, номер байта и номер бита для данных (например, I0.0, Q0.1 или M3.4).



- | | |
|--|----------------------------|
| A Идентификатор области памяти | E Байты области памяти |
| B Адрес байта: байт 3 | F Биты выбранного байта |
| C Разделитель ("байт.бит") | |
| D Положение бита в байте (бит 4 из 8) | |

В примере за областью памяти и адресом байта (M = битовая область памяти, и 3 = Байт 3) следует точка (". "), чтобы отделить битовый адрес (бит 4).

Обращение к данным в областях памяти ЦПУ

STEP 7 упрощает символьное программирование. Как правило, теги создаются в ПЛК тегах, блоке данных или в интерфейсе в верхней части OB, FC или FB. Эти теги включают в себя имя, тип данных, смещение и комментарий. Кроме того, в блоке данных может быть определено стартовое значение. Вы можете использовать эти теги при программировании, вводя имя тега в параметре инструкции. Дополнительно Вы можете ввести абсолютный операнд (область памяти, размер и смещение) в параметре инструкции. Примеры в следующих разделах показывают, как ввести абсолютные операнды. Символ % вставляется редактором программы автоматически перед абсолютным операндом. В редакторе программы Вы можете выбрать одно из представлений: символьное, символьное и абсолютное или абсолютное.

I (вход образа процесса): ЦПУ формирует образ периферийных (физических) каналов ввода перед выполнением OB в каждом цикле сканирования и записывает эти значения в образ процесса по входам. Вы можете обратиться к образу процесса по входам, используя бит, байт, слово или двойное слово. Разрешен доступ как на чтение, так и на запись, но обычно, входы образа процесса только читаются.

Таблица 5- 19 Абсолютная адресация для I памяти

Бит	I[адрес байта].[адрес бита]	I0.1
Байт, слово или двойное слово	I[размер][начальный адрес байта]	IB4, IW5 или ID12

Добавляя ":P" к адресу, Вы можете сразу читать цифровые и аналоговые входы ЦПУ, SB, SM или распределенного модуля. Различие между доступом, с использованием I_:P вместо I в том, что данные поступают непосредственно из адресуемых каналов, а не из образа процесса по входам. Этот доступ I_:P упоминается как "непосредственное чтение", потому что данные получены напрямую из источника вместо получения из копии, которая была сделана последний раз при обновлении образа процесса по входам.

Поскольку физические входные каналы получают свои значения непосредственно от полевых устройств, подключенных к этим каналам, запись в эти каналы запрещена. Т.е. обращения I_:P разрешены только на чтение, напротив I обращения могут быть на чтение или запись.

Обращения I_:P также ограничены количеством входов, поддерживаемых самим ЦПУ, SB или SM, с округлением до ближайшего байта. Например, если входы SB 2 DI / 2 DQ сконфигурированы с начальным адресом I4.0, то к входным каналам можно получить доступ как I4.0:P и I4.1:P или как IB4:P. Обращения от I4.2:P до I4.7:P не отклоняются, но не имеют никакого смысла, так как эти каналы не используются. Доступы к IW4:P и ID4:P запрещены, так как они превышают байтовое смещение, связанное с SB.

Доступы с использованием I_:P не влияют на соответствующее значение, сохраненное в образе процесса по входам.

Таблица 5- 20 Абсолютная адресация для I памяти (прямое обращение)

Бит	I[адрес байта].[адрес бита] :P	I0.1:P
Байт, слово или двойное слово	I[размер][начальный адрес байта] :P	IB4:P, IW5:P или ID12:P

Q (образ процесса по выходу): ЦПУ копирует значения, сохраненные в образе процесса по выходам в физические выходные каналы. Вы можете получить доступ к образу процесса по выходам, используя бит, байт, слово или двойное слово. К выходам образа процесса разрешен доступ как на чтение, так и на запись.

Таблица 5- 21 Абсолютная адресация для Q памяти

Бит	Q [адрес байта].[адрес бита]	Q1.1
Байт, слово или двойное слово	Q[размер][начальный адрес байта]	QB5, QW10 или QD40

Добавляя ":P" к адресу, Вы можете записать сразу в физические цифровые и аналоговые выходы ЦПУ, SB, SM или распределенного модуля. Различие между доступом с использованием Q_:P вместо Q в том, что данные поступают непосредственно в адресуемые каналы, а также в образ процесса по выходам (запись в оба местоположения). Этот доступ Q_:P иногда упоминается как "непосредственная запись", потому что данные сразу отправляются в целевой канал; целевой канал может не ожидать следующего обновления образа процесса по выходам.

Поскольку физические выходные каналы непосредственно управляют полевыми устройствами, которые подключены к этим каналам, чтение из этих каналов запрещено. Т.е. обращения Q_:P возможны только на запись, напротив обращения Q могут быть как на чтение, так и на запись.

Обращения Q_:P также ограничены количеством выходов, поддерживаемых самим ЦПУ, SB или SM, с округлением до ближайшего байта. Например, если выходы SB 2 DI / 2 DQ сконфигурирован с начальным адресом Q4.0, то к выходным каналам можно получить доступ как Q4.0:P и Q4.1:P или как QB4:P. Обращения от Q4.2:P до Q4.7:P не отклоняются, но не имеют никакого смысла, так как эти каналы не используются. Доступы к QW4:P и QD4:P запрещены, так как они превышают байтовое смещение, связанное с SB.

Обращения с использованием Q_:P влияют как на физические выходы, так и на соответствующее значение, сохраненное в образе процесса по выходам.

Таблица 5- 22 Абсолютная адресация для Q памяти (прямое обращение)

Бит	Q[адрес байта].[адрес бита] :P	Q1.1:P
Байт, слово или двойное слово	Q[размер][начальный адрес байта] :P	QB5:P, QW5:P или QD40:P

М (область битовой памяти): Используйте битовую область памяти (М память) как для сигналов управления, так и для данных, чтобы сохранять промежуточное рабочее состояние или другую управляющую информацию. Вы можете обращаться к битовой области памяти, используя бит, байт, слово или двойное слово. Для М области памяти разрешены обращения на запись и на чтение..

Таблица 5- 23 Абсолютная адресация для М памяти

Бит	M [адрес байта].[адрес бита]	M26.7
Байт, слово или двойное слово	M[размер][начальный адрес байта]	MB20, MW30 или MD50

Темп (временная память): ЦПУ выделяет временную память по мере необходимости. ЦПУ выделяет временную память для кодового блока и инициализирует ячейки памяти в 0 в то время, когда он запускает блок кода (для ОВ) или вызывает блок кода (для FC или FB).

Временная память подобна М памяти за одним главным исключением: М память является "глобальной", а временная память - "локальной":

- М память: Любой ОВ, FC или FB могут обратиться к данным в М памяти, то есть данные доступны глобально для всех элементов пользовательской программы.
- Временная память: ЦПУ ограничивает доступ к данным во временной памяти теми ОВ, FC или FB, которые создали или объявили временную ячейку памяти. Временные ячейки памяти остаются локальными, и различные блоки кода не могут совместно использовать временную память, даже когда блок кода вызывает другой блок кода. Например: Когда ОВ вызывает FC, FC не может получить доступ к временной памяти ОВ, из которого он вызван.

ЦПУ обеспечивает временную (локальную) память для каждого уровня приоритета ОВ:

- 16 кбайт для запуска и программного цикла, включая вызываемые FB и FC
- 6 кбайт для каждого дополнительного потока события прерывания, включая вызываемые FB и FC. Вы обращаетесь к временной памяти только символично.

DB (блок данных): Используйте память DB для хранения различных типов данных, включая промежуточные рабочие состояния или другую управляющую информацию, параметры для FB и структурированные данные, требуемые для многих инструкций, таких как таймеры и счетчики. Вы можете обратиться к памяти блоков данных, используя бит, байт, слово или двойное слово. Для блоков данных чтения-записи разрешен доступ на чтение и на запись. К блокам данных только для чтения возможно обращение только на чтение.

Таблица 5- 24 Абсолютная адресация для DB памяти

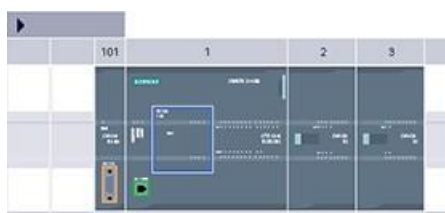
Бит	DB[номер блока данных].DBX[адрес байта].[адрес бита]	DB1.DBX2.3
Байт, слово или двойное слово	DB[номер блока данных].DB [размер] [начальный адрес байта]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Примечание

Когда Вы определяете абсолютный адрес в LAD или FBD, STEP 7 добавляет перед этим адресом символ "%", чтобы указать, что это - абсолютный адрес. При программировании Вы можете ввести абсолютный адрес с символом "%" или без него (например, %I0.0 или I0.0). Если он пропущен, STEP 7 вставляет символ "%".

В SCL Вы должны ввести "%" перед адресом, чтобы указать, что это - абсолютный адрес. Без "%" STEP 7 генерирует ошибку неопределенного тега во время компиляции.

Конфигурирование ввода-вывода в ЦПУ и модулях ввода-вывода



Device overview						
Module	Slot	I address	Q address	Type	Order	
	103					
	102					
RS485_1	101			CM 1241 (RS485)	6ES7	
PLC_1	1			CPU 1214C DC DCI	6ES7	
DI14/DO10	1.1	0...1	0...1	DI14/DO10		
AI2	1.2	64...67		AI2		
AO1 x 12bit	1.3		80...81	AO1 signal board	6ES7	
HSC_1	1.16	1000...		High speed counts		
HSC_2	1.17			High speed counts		
HSC_3	1.18			High speed counts		
HSC_4	1.19			High speed counts		
HSC_5	1.20			High speed counts		
HSC_6	1.21			High speed counts		
Pulse_1	1.32			Pulse generator (P)		
Pulse_2	1.33			Pulse generator (P)		
PROFINET L X1				PROFINET interface		
DI8 x 24VDC	2	8		SM 1221 DI8 x 24	6ES7	

Когда Вы добавляете ЦПУ и модули ввода-вывода в Вашу конфигурацию устройства, STEP 7 автоматически назначает адреса I и Q. Вы можете изменить значение адреса по умолчанию, выбрав поле адреса в конфигурации устройства и введя новые значения.

- STEP 7 назначает цифровые входы и выходы группами из 8 каналов (1 байт), независимо от того использует ли модуль все каналы или нет.
- STEP 7 распределяет аналоговые входы и выходы группами по 2, где каждый аналоговый канал занимает 2 байта (16 битов).

На рисунке показан пример ЦПУ 1214C с двумя SM и одной SB. В этом примере Вы могли бы изменить адрес модуля DI8 на 2 вместо 8. Инструмент помогает Вам, меняя диапазоны адресов, которые являются неподходящими по размеру или конфликтуют с другими адресами.

5.3 Обработка аналоговых значений

Модули аналогового сигнала обеспечивают входные сигналы или ожидают выходные значения, которые представляют диапазон значений напряжения или тока. Используются следующие диапазоны: ± 10 В, ± 5 В, ± 2.5 В или 0 - 20 мА. Значения, возвращенные модулями, являются целыми числами от 0 до 27648, представляющими номинальный диапазон для тока, и от -27648 до 27648 - для напряжения. Значения, находящиеся вне диапазона представляют собой переполнение или потерю значимости. Смотри таблицы представления аналогового входа (стр. 1194) и аналогового выхода (стр. 1195) для получения дополнительной информации о типах значений, находящихся вне диапазона.

В Вашей управляющей программе Вам, возможно, потребуется использовать эти значения в инженерных единицах, например чтобы представить объем, температуру, вес или другое количественное значение. Чтобы сделать это для аналогового входа, Вы должны вначале нормализовать аналоговое значение в вещественное значение (с плавающей точкой) от 0.0 до 1.0. Затем Вы должны смасштабировать его в значение, находящееся между минимальным и максимальным значениям инженерных единиц, которые оно представляет. Для значений, представленных в инженерных единицах, которые Вы должны преобразовать в значение аналогового выхода, Вы сначала нормализуете значение в значение от 0.0 до 1.0, а затем масштабируете его в диапазон от 0 до 27648 или от -27648 до 27648, в зависимости от диапазона аналогового модуля. STEP 7 предоставляет для этих целей инструкции NORM_X и SCALE_X (стр. 275). Вы можете также использовать инструкцию CALCULATE (стр. 237), чтобы масштабировать аналоговые значения (стр. 39).

Пример: Обработка аналогового значения

Рассмотрим, например, аналоговый вход, имеющий токовый диапазон 0 - 20 мА. Модуль аналогового входа возвращает значения в диапазоне от 0 до 24768 для измеренных значений. Для данного примера предположите, что Вы используете это значение аналогового входа, чтобы измерить диапазон температуры от 50 °C до 100 °C. У нескольких пробных измерений были бы следующие значения:

Аналоговое входное значение	Инженерные единицы
0	50 °C
6192	62.5 °C
12384	75 °C
18576	87.5 °C
24768	100 °C

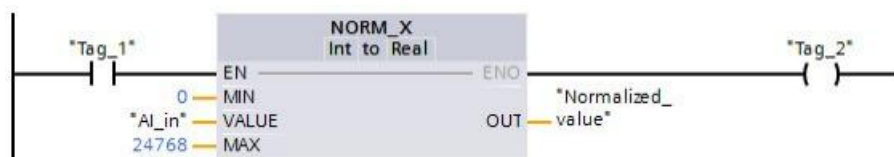
Расчет для определения инженерных единиц из значения аналогового входа в этом примере следующий:

Значение в инженерных единицах = $50 + (\text{Аналоговое входное значение}) * (100 - 50) / (24768 - 0)$. Для общего случая уравнение можно представить в виде:

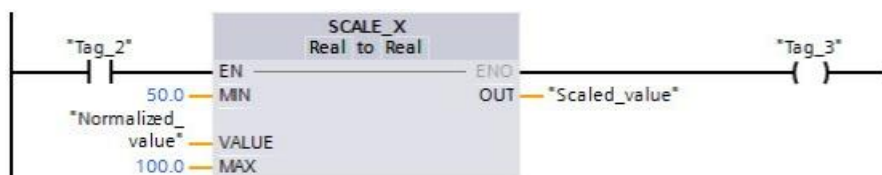
Значение в инженерных единицах = (Нижний предел инженерных единиц) +
 $(\text{Аналоговое входное значение}) * \frac{(\text{Верхний предел инженерных единиц} - \text{Нижний предел инженерных единиц})}{(\text{Максимум диапазона аналогового значения} - \text{Минимум диапазона аналогового значения})}$

В приложениях ПЛК типовым методом является нормализация значения аналогового входа к значению с плавающей точкой в диапазоне от 0.0 до 1.0. Затем Вы могли бы смасштабировать полученное значение к значению с плавающей точкой в диапазоне Ваших инженерных единиц. Для облегчения решения следующие LAD инструкции используют постоянные значения для диапазонов; Вы могли бы их использовать вместе с тегами.

Сегмент 1



Сегмент 2



5.4 Типы данных

Типы данных используются, чтобы определить как размер элемента данных, так и способ интерпретации данных. Каждый параметр инструкции поддерживает по крайней мере один тип данных, а некоторые параметры поддерживают несколько типов данных. Задержите курсор над полем параметра инструкции, чтобы видеть, какие типы данных поддерживаются для данного параметра.

Формальный параметр - это идентификатор в инструкции, который отмечает расположение данных, которые будут использоваться этой инструкцией (например: вход IN1 инструкции ADD). Фактический параметр - это ячейка памяти (которой предшествует символ "%") или константа, содержащие данные, которые будут использоваться инструкцией (пример %MD400 "Number_of_Widgets"). Тип данных фактического параметра, определенного Вами, должен соответствовать одному из поддерживаемых типов данных формального параметра, определенного инструкцией.

При определении фактического параметра Вы должны определить либо тег (символ), либо абсолютный (прямой) адрес в памяти. Теги связывают символьное имя (имя тега) с типом данных, ячейкой памяти, смещением памяти и комментарием, и могут быть созданы в редакторе ПЛК тегов или в редакторе интерфейса для блока (OB, FC, FB и DB). Если Вы вводите абсолютный адрес, у которого нет связанного тега, Вы должны использовать надлежащий размер, который соответствует поддерживаемому типу данных, и после ввода будет создан тег по умолчанию.

Все типы данных за исключением String, Struct, Array и DTL доступны в редакторе ПЛК тегов и редакторах интерфейса блока. String, Struct, Array и DTL доступны только в редакторах интерфейса блока. Вы можете также ввести постоянное значение для многих входных параметров.

- Бит и комбинация битов (стр. 118): Bool (Булево или битовое значение), Byte (8-битовое значение байта), Word (16-битовое значение), DWord (32-битовое значение двойного слова)
- Целое (стр. 119)
 - USInt (беззнаковое 8-битовое целое), SInt (8-битовое целое со знаком),
 - UInt (беззнаковое 16-битовое целое), Int (16-битовое целое со знаком)
 - UDInt (беззнаковое 32-битовое целое), DInt (32-битовое целое со знаком)
- Действительное с плавающей точкой (стр. 119): Real (действительное или с плавающей точкой значение 32-бита), LReal (действительное или с плавающей точкой значение 64-бита)
- Время и дата (стр. 120): Time (32-битовое значение МЭК времени), Date (16-битовое значение даты), TOD (32-битовое значение времени суток), DTL (12-байтовая структура даты и времени)
- Символ и строка (стр. 122): Char (8-битовый одиночный символ), String (строка переменной длины из максимум 254 символов)
- Массив (стр. 124)
- Структура данных (стр. 125): Struct
- Тип данных ПЛК (стр. 125)
- Указатели (стр. 126): Pointer, Any, Variant

Хотя они недоступны как типы данных, следующий BCD числовой формат поддерживается инструкциями преобразования.

Таблица 5- 25 Размер и диапазон BCD формата

Формат	Размер (в битах)	Числовой диапазон	Пример ввода константы
BCD16	16	от -999 до 999	123, -123
BCD32	32	от -9999999 до 9999999	1234567, -1234567

5.4.1. Типы данных Bool, Byte, Word и DWord

Таблица 5- 26 Битовые типы данных

Тип данных	Размер в битах	Тип числа	Диапазон числа	Примеры констант	Примеры адресов
Bool	1	Булевый	ЛОЖЬ или ИСТИНА	TRUE, 1,	I1.0 Q0.1 M50.7 DB1.DBX2.3 Имя_тега
		Двоичный	0 или 1	0, 2#0	
		Восьмеричный	8#0 или 8#1	8#1	
		Шестнадцатеричный	16#0 или 16#1	16#1	
Byte	8	Двоичный	от 2#0 до 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 Имя_тега
		Беззнаковое целое	от 0 до 255	15	
		Восьмеричный	от 8#0 до 8#377	8#17	
		Шестнадцатеричный	от B#16#0 до B#16#FF	B#16#F, 16#F	
Word	16	Двоичный	от 2#0 до 2#1111111111111111	2#1111000011110000	MW10 DB1.DBW2 Имя_тега
		Беззнаковое целое	от 0 до 65535	61680	
		Восьмеричный	от 8#0 до 8#177777	8#170360	
		Шестнадцатеричный	от W#16#0 до W#16#FFFF, от 16#0 до 16#FFFF	W#16#F0F0, 16#F0F0	
DWord	32	Двоичный	от 2#0 до 2#11111111111111111111111111111111	2#11110000111111110001111	MD10 DB1.DBD8 Имя_тега
		Беззнаковое целое	от 0 до 4294967295	15793935	
		Восьмеричный	от 8#0 до 8#377777777777	8#74177417	
		Шестнадцатеричный	от DW#16#0000_0000 до DW#16#FFFF_FFFF, от 16#0000_0000 до 16#FFFF_FFFF	DW#16#F0FF0F, 16#F0FF0F	

5.4.2. Целочисленные типы данных

Table 5- 27 Целочисленные типы данных (U = беззнаковое, S = короткое, D= двойное)

Тип данных	Размер в битах	Диапазон числа	Примеры констант	Примеры адресов
USInt	8	от 0 до 255	78, 2#010011110	MB0, DB1.DB4, Имя_тега
SInt	8	от -128 до 127	+50, 16#50	
UInt	16	от 0 до 65,535	65295, 0	MW2, DB1.DBW2, Имя_тега
Int	16	от -32,768 до 32,767	30000, +30000	
UDInt	32	от 0 до 4,294,967,295	4042322160	MD6, DB1.DBD8, Имя_тега
DInt	32	от -2,147,483,648 до 2,147,483,647	-2131754992	

5.4.3. Типы данных для действительных чисел с плавающей точкой

Вещественные (или с плавающей точкой) числа представлены как 32-разрядные числа одинарной точности (Real), или 64-разрядные числа двойной точности (LReal), как описано в ANSI/IEEE 754-1985 стандарте. Числа одинарной точности с плавающей точкой точны, до 6 значащих цифр, а числа с плавающей точкой двойной точности точны до 15 значащих цифр. Вы можете определить максимум 6 значащих (Real) или 15 (LReal) цифр при вводе константы с плавающей точкой, чтобы поддержать точность.

Таблица 5- 28 Типы данных для действительных чисел с плавающей точкой (L=длинное)

Тип данных	Размер в битах	Диапазон числа	Примеры констант	Примеры адресов
Real	32	от -3.402823e+38 до -1.175 495e-38, ±0, от +1.175 495e-38 до +3.402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DBD8, Имя_тега
LReal	64	от -1.7976931348623158e+308 до -2.2250738585072014e-308, ±0, от +2.2250738585072014e-308 до +1.7976931348623158e+308	12345.123456789e40, 1.2E+40	Имя_ДВ.имя_пер. Правила: • Не поддерживает прямую адресацию • Может быть назначен в таблице интерфейса блока OB, FB или FC

Вычисления, которые содержат длинную последовательность значений, включая очень большие и очень малые числа, могут привести к неточным результатам. Это может произойти, если числа отличаются в 10 в степени x раз, где (Реальный) x > 6 (Real) или 15 (LReal). Например (Real): 100 000 000 + 1 = 100 000 000.

5.4.4. Данные даты и времени

Таблица 5- 29 Типы данных даты и времени

Тип данных	Размер в битах	Диапазон числа	Примеры ввода константы
Time	32 бита	от T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms Сохраняется как: от -2,147,483,648 до +2,147,483,647 мс	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
Date	16 битов	от D#1990-1-1 до D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 бита	от TOD#0:0:0.0 до TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Дата и время длинные)	12 байтов	Мин.: DTL#1970-01-01-00:00:00.0 Макс.: DTL#2262-04-11:23:47:16.854 775 807	DTL#2008-12-16-20:30:20.250

Time

Данные TIME хранятся как двойное целое число со знаком, интерпретируемое как миллисекунды. Формат редактора может использовать информацию о дне (d), часах (h), минутах (m), секундах (s) и миллисекундах (ms).

Нет необходимости задавать все единицы времени. Например, T#5h10s и 500h допустимы.

Итоговое значение, образованное из указанных значений всех единиц не может превышать верхний или нижний пределы в миллисекундах для типа данных Time (от -2 147 483 648 мс до +2 147 483 647 мс).

Date

Данные DATE хранятся как целое значение без знака, которое интерпретируется как число дней, добавленных к исходной дате 01.01.1990 , чтобы получить заданную дату. Формат редактора должен определить год, месяц и день.

TOD

TOD (TIME_OF_DAY) данные хранятся, как двойное целое число без знака, которое интерпретируется как число миллисекунд с полуночи до заданного времени суток (Полночь = 0 мс). Должны быть определены час (24часа/день), минута и секунда. Указание долей секунд по желанию.

DTL

DTL (Дата и время длинные) тип данных использует 12 байтовую структуру, которая сохраняет информацию о дате и времени. Вы можете определить DTL данные либо во временной памяти блока, либо в DB. Значение для всех компонентов должно быть введено в столбце "Start value" редактора DB.

Таблица 5- 30 Размер и диапазон для DTL

Длина (байты)	Формат	Диапазон значений	Пример ввода значения
12	Часы и календарь Год-Месяц- День:Час:Минута: Секунда.Наносекунды	Мин.: DTL#1970-01-01-00:00:00.0 Макс.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

Каждый компонент DTL имеет отличный тип данных и диапазон значений. Тип данных заданного значения должен соответствовать типу данных соответствующих компонентов.

Таблица 5- 31 Элементы DTL структуры

Байт	Компонент	Тип данных	Диапазон значений
0	Год	UINT	от 1970 до 2554
1			
2	Месяц	USINT	от 1 до 12
3	День	USINT	от 1 до 31
4	День недели ¹	USINT	от 1(воскресенье) до 7(суббота) ¹
5	Час	USINT	от 0 до 23
6	Минута	USINT	от 0 до 59
7	Секунда	USINT	от 0 до 59
8	Наносекунды	UDINT	от 0 до 999 999 999
9			
10			
11			

¹ Формат Год-Месяц-День:Час:Минута: Секунда.Наносекунда не включает в себя день недели.

5.4.5. Символьные и строковые типы данных

Таблица 5- 32 Символьные и строковые типы данных

Тип данных	Размер	Диапазон	Примеры ввода констант
Char	8 битов	от 16#00 до 16#FF	'A', 't', '@', 'ä', 'Σ'
WChar	16 битов	от 16#0000 до 16#FFFF	'A', 't', '@', 'ä', 'Σ', азиатские символы, кириллические символы и другие
String	n+ 2 бфйтов	n = (от 0 до 254 байтов)	"ABC"
WString	n+ 2 слов	n = (от 0 до 65534 слов)	"ä123@XYZ.COM"

Char и WChar

Char занимает один байт в памяти и хранит единственный символ, закодированный в формате ASCII, включая расширенные коды ASCII символов. WChar занимает одно слово в памяти и может содержать любое представление символа в виде двух байтов

Синтаксис редактора использует символ одинарной кавычки до и после символа. Вы можете использовать видимые символы и управляющие символы.

String и WString

ЦПУ поддерживает тип данных String для хранения последовательности однобайтовых символов. Тип данных String содержит общее количество символов (число символов в строке) и текущее количество символов. Тип String обеспечивает до 256 байтов для хранения максимального общего количества символов (1 байт), текущего количества символов (1 байт) и до 254 байтов в строке. Каждый байт в типе данных String может быть любым значением от 16#00 до 16#FF.

Тип данных WString предусматривает более длинные строки из значений по одному слову (двойной байт) . Первое слово содержит максимальное общее количество символов; следующее слово содержит общее количество символов и следующая строка может содержать до 65534 слов. Каждое слово в типе данных WString может быть любым значением от 16#0000 до 16#FFFF.

Вы можете использовать буквенные строки (константы) для параметров инструкции типа IN с помощью одинарных кавычек. Например, 'ABC' - строка из трех символов, которая могла бы использоваться в качестве ввода для параметра IN инструкции S_CONV. Вы можете также создать строковые переменные, выбрав тип данных "String", или "WString" в редакторах интерфейса блоков для OB, FC, FB и DB. Вы не можете создать строку в редакторе тегов ПЛК.

Вы можете определить максимальный размер строки в байтах (String) или словах (WString), введя квадратные скобки после ключевого слова "String" или "WString" после того, как Вы выберете один из этих типов данных в выпадающем списке. Например, запись "MyString String [10]" определила бы максимальный размер в 10 байтов для MyString. Если Вы не добавляете квадратные скобки с максимальным размером, то 254 принимается для String, а 65534 для WString. Запись "MyWString WString[1000]" определила бы WString длиной в 1000 слов.

Следующий пример определяет String с помощью максимального количества символов 10 и текущего количества символов 3. Это означает, что String в настоящее время содержит 3 однобайтовых символа, но мог бы быть расширен до 10 однобайтовых символов.

Таблица 5- 33 Пример типа данных String

Общее количество символов	Текущее количество символов	Символ 1	Символ 2	Символ 3	...	Символ 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	...	Байт 11

Следующий пример определяет WString с максимальным количеством символов 500 и текущим количеством символов 300. Это означает, что строка в настоящее время содержит 300 символов по одному слову, но могла бы быть расширена до 500 символов по одному слову.

Таблица 5- 34 Пример типа данных WString

Общее количество символов	Текущее количество символов	Символ 1	Символы со 2 по 299	Символ 300	...	Символ 500
500	300	'ä' (16#0084)	Слова ASCII символов	'M' (16#004D)	...	-
Слово 0	Слово 1	Слово 2	Слова с 3 по 300	Слово 301	...	Слово 501

Управляющие ASCII символы могут использоваться в типах данных Char, Wchar, String и WString. Следующая таблица показывает примеры синтаксиса управляющих символов.

Таблица 5- 35 Действительные управляющие ASCII символы

Управляющие символы	ASCII шестн. знач. (Char)	ASCII шестн. знач. (WChar)	Управляющая функция	Примеры
\$L или \$l	16#0A	16#000A	Перевод строки	'\$LText', '\$0AText'
\$N или \$n	16#0A и 16#0D	16#000A и 16#000D	Разрыв строки Новая строка показывает два символа в последовательности.	'\$NText', '\$0A\$0DText'
\$P или \$p	16#0C	16#000C	Подача страницы	'\$PText', '\$0CText'
\$R или \$r	16#0D	16#000D	Возврат каретки (CR)	'\$RText', '\$0DText'
\$T или \$t	16#09	16#0009	Табулятор	'\$TText', '\$09Text'
\$\$	16#24	16#0024	Знак доллара	'100\$\$', '100\$24'
\$'	16#27	16#0027	Одинарная кавычка	'\$'Text\$', '\$27Text\$ 27'

5.4.6. Тип данных Array

Массивы

Вы можете создать массив, который содержит несколько элементов одного и того же типа данных. Массивы могут быть созданы в редакторе интерфейса блоков для OB, FC, FB и DB. Вы не можете создать массив в редакторе тегов ПЛК.

Чтобы создать массив в редакторе интерфейса блока, дают массиву имя и выбирают тип данных "Array [lo .. hi] of type", затем редактируют "lo", "hi" и "type" следующим образом:

- lo - стартовый (наименьший) индекс для Вашего массива
- hi - конечный (наивысший) индекс для Вашего массива
- type - один из типов данных, таких как BOOL, SINT, UDINT

Таблица 5- 36 Правила для типа данных ARRAY

Тип данных	Синтаксис массива		
ARRAY	Имя [индекс1_мин.. индекс1_макс, индекс2_мин.. индекс2_макс] of <тип данных>		
	• Все элементы массива должны иметь один и тот же тип данных. • Индекс может быть отрицательным, но нижний предел должен быть меньше или равен верхнему пределу. • Массивы могут иметь от одного до шести измерений. • Объявления индексов многомерного массива мин..макс разделяются символами запятой. • Вложенные массивы или массивы массивов недопустимы. • Объем памяти, занимаемый массивов = (размер одного элемента * общее количество элементов в массиве)		
	Индекс массива	Допустимые типы данных индекса	Правила для индекса массива
	Константа или переменная	USInt, SInt, UInt, Int, UDInt, DInt	• Пределы значений: от -32768 до +32767 • Допустимо: Комбинированные константы и переменные • Допустимо: Константные выражения • Недопустимо: Переменные выражения

**Пример:
объявления
массива**

ARRAY[1..20] of REAL
ARRAY[-5..5] of INT
ARRAY[1..2, 3..4] of CHAR

Одно измерение, 20 элементов
Одно измерение, 11 элементов
Два измерения, 4 элемента

**Пример:
адресация
элементов
массива**

ARRAY1[0]
ARRAY2[1,2]
ARRAY3[i,j]

ARRAY1 элемент 0
ARRAY2 элемент [1,2]
если i =3 и j=4, то адресуется элемент
ARRAY3[3, 4]

5.4.7. Структурированные типы данных

Вы можете использовать тип данных "Struct", чтобы определить структуру данных, состоящих из других типов данных. Структурированный тип данных может использоваться, чтобы обработать группу связанных данных процесса как отдельный блок данных. Типу данных Struct присваивают имя и определяют внутреннюю структуру данных в редакторе блока данных или редакторе интерфейса блока.

Массивы и структуры могут также быть собраны в большую структуру. Структура может быть вложенной до восьми уровней. Например, Вы можете создать структуру структур, которые содержат массивы.

5.4.8. ПЛК тип данных

Редактор ПЛК типа данных позволяет Вам определять структуры данных, которые Вы можете многократно использовать в Вашей программе. Вы создаете ПЛК тип данных при открытии ветви дерева проекта "PLC data types" и двойном щелчке по элементу "Add new data type". На вновь создаваемом элементе типа данных PLC используйте два одиночных щелчка, чтобы изменить его имя по умолчанию и дважды щелкнуть, чтобы открыть редактора ПЛК типа данных.

Вы создаете пользовательскую структуру ПЛК типа данных, используя те же методы редактирования, которые используются в редакторе блока данных. Добавьте новые строки для любых типов данных, которые необходимы, чтобы создать нужную структуру данных.

Если создается новый ПЛК тип данных, то новое имя ПЛК типа появится в выпадающих списках селектора типа данных в редакторе DB и редакторе интерфейса кодового блока.

Возможные варианты использования ПЛК типов данных:

- ПЛК типы данных могут использоваться непосредственно в качестве типа данных в интерфейсе кодового блока или в блоках данных.
- ПЛК типы данных могут использоваться в качестве шаблона для создания множества глобальных блоков данных, которые используют ту же самую структуру данных.

Например, ПЛК тип данных мог бы быть рецептом для смешивания цветов. Вы можете в этом случае присвоить этот ПЛК тип данных множеству блоков данных. Каждый блок данных может тогда иметь скорректированные переменные для создания определенного цвета.

5.4.9. Типы данных указателей

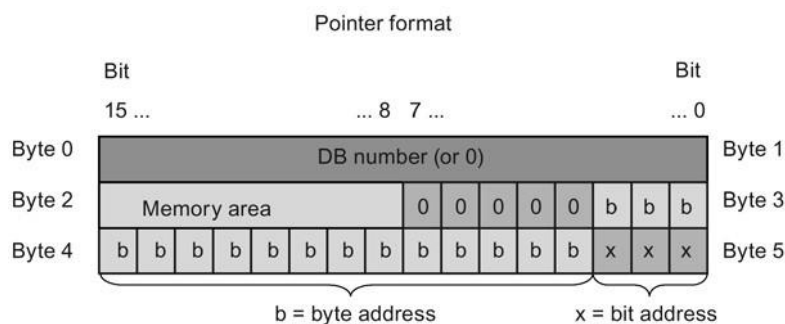
Типы данных указателей (Pointer, Any и Variant) могут использоваться в таблицах интерфейса для FB и FC кодовых блоков. Вы можете выбрать тип данных указателя из выпадающих списков типа данных интерфейса блока.

Тип данных Variant также используется для параметров инструкций.

5.4.9.1. Тип данных указателя "Pointer"

Тип данных Pointer указывает на определенную переменную. Он занимает в памяти 6 байтов (48 битов) и может содержать следующую информацию:

- Номер DB или 0, если данные хранятся не в DB
- Область памяти CPU
- Адрес переменной



В зависимости от инструкции Вы можете объявить следующие три типа указателей:

- Внутризонный указатель: содержит данные адреса переменной
- Межзонный указатель: содержит данные области памяти и адреса переменной
- Указатель DB: содержит номер блока данных и адрес переменной

Таблица 5- 37 Типы указателей:

Тип	Формат	Пример ввода
Внутризонный указатель	P#Байт.Бит	P#20.0
Межзонный указатель	P#Область_памяти_Байт.Бит	P#M20.0
DB-указатель	P#Блок_данных.Элемент_данных	P#DB10.DBX20.0

Вы можете ввести параметр с типом Pointer без префикса (P #). Ваша запись будет автоматически преобразована в формат указателя.

Таблица 5- 38 Коды областей памяти в указателе Pointer:

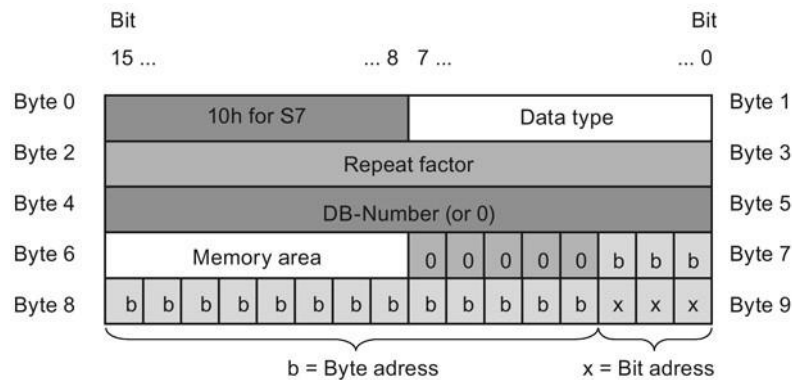
Шестнадцатеричный код	Тип данных	Описание
b#16#81	I	Образ процесса по входам
b#16#82	Q	Образ процесса по выходам
b#16#83	M	Битовая память
b#16#84	DBX	Блок данных
b#16#85	DIX	Экземплярный блок данных
b#16#86	L	Локальные данные
b#16#87	V	Локальные данные вызывающего блока

5.4.9.2. Тип указателя " ANY"

Тип данных указателя ANY ("любой") указывает на начало области данных и определяет ее длину. Указатель ANY использует 10 байтов памяти и может содержать следующую информацию:

- Тип данных: Тип данных элементов области данных
- Коэффициент повторения: Количество элементов данных
- Номер DB: Блок данных, в котором сохранены элементы данных
- Область хранения: область памяти ЦПУ, в которой сохранены элементы данных
- Начальный адрес: "Байт.Бит" начала области данных

Следующий рисунок показывает структуру указателя ANY:



Указатель не может обнаружить структуры. Он может адресовать только элементарные переменные.

Таблица 5- 39 Формат и примеры указателя ANY:

Формат	Пример ввода	Описание
P#Блок_данных.Область_памяти Адрес_данных Тип Количество	P#DB 11.DBX 20.0 INT 10	10 слов в глобальном DB 11, начиная с DBB 20.0
P# Область_памяти Ад- рес_данных Тип Количество	P#M 20.0 BYTE 10	10 байтов, начиная с MB 20.0
	P#I 1.0 BOOL 1	Вход I1.0

Таблица 5- 40 Коды типов данных в указателе ANY

Шестнадцатеричный код	Тип данных	Описание
b#16#00	Null	Нулевой указатель
b#16#01	Bool	Биты
b#16#02	Byte	Баты, 8 битов
b#16#03	Char	8-битовый символ
b#16#04	Word	16-битов-слово
b#16#05	Int	16-битов-целое
b#16#37	SInt	8-битов-целое
b#16#35	UInt	16-битов беззнаковое целое
b#16#34	USInt	8-битовое беззнаковое целое
b#16#06	DWord	32-битовое двойное слово
b#16#07	DInt	32-битовое двойное целое
b#16#36	UDInt	32-битовое-беззнаковое двойное целое
b#16#08	Real	32-битовое с плавающей точкой
b#16#0B	Time	Время
b#16#13	String	Символьная строка

Таблица 5- 41 Коды областей памяти в указателе ANY:

Шестнадцатеричный код	Тип данных	Описание
b#16#81	I	Образ процесса по входам
b#16#82	Q	Образ процесса по выходам
b#16#83	M	Битовая память
b#16#84	DBX	Блок данных
b#16#85	DIX	Экземплярный блок данных
b#16#86	L	Локальные данные
b#16#87	V	Локальные данные вызывающего блока

5.4.9.3. Тип указателя "Variant"

Тип данных Variant может указать на переменные различных типов данных или параметры. Указатель Variant может указать на структуры и отдельные компоненты структур. Указатель Variant не занимает места в памяти.

Таблица 5- 42 Свойства указателя Variant

Длина (байты)	Представление	Формат	Пример ввода
0	Символьное	Операнд	MyTag
		Имя_DB.Имя_структуры.Имя_элемента	MyDB.Struct1.pressure1
	Абсолютное	Операнд	%MW10
		Имя_DB.Операнд Тип Длина	P#DB10.DBX10.0 INT 12

5.4.10. Обращение к "срезу" тегированного типа данных

К тегам ПЛК и тегам блока данных можно обращаться на уровне бита, байта или слова в зависимости от их размера. Синтаксис для доступа к такому срезу данных является следующим:

- "<Имя ПЛК тега>".xp (обращение к биту)
- "<Имя ПЛК тега>".bp (обращение к байту)
- "<Имя ПЛК тега>".wp (обращение к слову)
- "<Имя блока данных>".<имя тега>.xp (обращение к биту)
- "<Имя блока данных>".<имя тега>.bp (обращение к байту)
- "<Имя блока данных>".<имя тега>.wp (обращение к слову)

К тегу размера двойного слова можно обратиться, используя биты 0 - 31, байты 0 - 3, или слова 0 - 1. К тегу размера слова можно обратиться, используя биты 0 - 15, байты 0 - 1 или слово 0. К тегу размера байта можно обратиться, используя биты 0 - 7, или байт 0. Битовый, байтовый и пословный срезы могут использоваться везде, где биты, байты или слова - ожидаемые операнды.

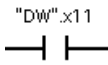
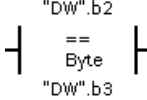
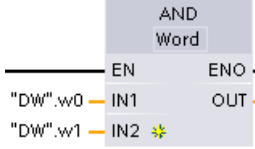
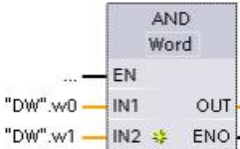
																								BYTE							
																WORD															
DWORD																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

Примечание

Допустимыми типами данных, к которым можно обратиться посредством среза являются Byte, Char, Conn_Any, Date, DInt, DWord, Event_Any, Event_Att, Hw_Any, Hw_Device, HW_Interface, Hw_Io, Hw_Pwm, Hw_SubModule, Int, OB_Any, OB_Att, OB_Cyclic, OB_Delay, OB_WHINT, OB_PCYCLE, OB_STARTUP, OB_TIMEERROR, OB_Tod, Port, Rtm, SInt, Time, Time_Of_Day, UDInt, UInt, USInt и Word. К ПЛК тегам с типом Real может обратиться, используя срез, но к тегам блока данных с типом Real нельзя.

Примеры

В таблице ПЛК тегов, "DW" объявлен, как тег с типом DWORD. Примеры демонстрируют обращение к биту, байту и слову, используя срез:

	LAD	FBD	SCL
Обращение к биту			<pre>IF "DW".x11 THEN ... END_IF;</pre>
Обращение к байту			<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
Обращение к слову			<pre>out := "DW".w0 AND "DW".w1;</pre>

5.4.11. Обращение к тегу с использование AT наложения

AT наложение для тега позволяет Вам обратиться к уже объявленному тегу блока со стандартным доступом, используя накладываемое объявление другого типа данных. Вы можете, например, адресовать отдельные биты тега с типами Byte, Word или DWord с помощью массива из Bool.

Объявление

Чтобы выполнить наложение на параметр, объявите дополнительный параметр непосредственно после параметра, на который должно быть выполнено наложение и выберите тип данных "AT". Редактор создает наложение, и Вы можете тогда выбрать тип данных, структуру или массив, который Вы хотите использовать для наложения.

Пример

Настоящий пример показывает входные параметры FB со стандартным доступом. На байтовый тег B1 выполняется наложение массива булевских элементов:

	B1	Byte	0.0
	▼ OV	AT*B1	Array[0..7] of Bool
	OV[0]	Bool	0.0
	OV[1]	Bool	0.1
	OV[2]	Bool	0.2
	OV[3]	Bool	0.3
	OV[4]	Bool	0.4
	OV[5]	Bool	0.5
	OV[6]	Bool	0.6
	OV[7]	Bool	0.7

Другой пример является тег DWord на который, накладывается структура, включающая в себя Word, Byte и два элемента Bool:

DW1	DWord	2.0
DW1_Struct AT*DW1*	Struct	2.0
W1	Word	0.0
B1	Byte	2.0
BO1	Bool	3.0
BO2	Bool	3.1

Столбец Offset интерфейса блока показывает расположение наложенных типов данных относительно исходного тега.

Вы можете адресовать наложенные типы непосредственно в логике программы:

LAD	FBD	SCL
		IF #OV[1] THEN ... END_IF;
		IF #DW1_Struct.W1 = W#16#000C THEN ... END_IF;
		out1 := #DW1_Struct.B1;
		IF #OV[4] AND #DW1_Struct.BO2 THEN ... END_IF;

Правила

- Наложение тегов возможно только в FB и FC блоках со стандартным (не оптимизированным) доступом.
- Вы можете наложить параметры для всех типов блока и всех разделов объявления.
- Вы можете использовать наложенный параметр как любой другой параметр блока.
- Вы не можете наложить параметры типа VARIANT.
- Размер накладываемого параметра должен быть меньше или равен размеру параметра, на который выполняется наложение.
- Вы должны объявить накладываемую переменную сразу после переменной, на которую она накладывается, и выбрать ключевое слово "AT" в качестве исходного типа данных.

5.5 Использование карты памяти

Примечание

ЦПУ поддерживает только предварительно отформатированные карты памяти SIMATIC (стр. 1266).

Прежде чем Вы скопируете любую программу на отформатированную карту памяти, удалите любую ранее сохраненную программу из карты памяти.

Используйте карту памяти либо в качестве карты передачи, либо в качестве карты программирования. Любая программа, которую Вы копируете на карту памяти, содержит все кодовые блоки и блоки данных, любые технологические объекты и конфигурацию устройства. Скопированная программа не содержит форсированные значения.

- Используйте карту передачи (стр. 135), чтобы скопировать программу во внутреннюю загрузочную память ЦПУ, не используя STEP 7. После того, как Вы вставите карту передачи, ЦПУ вначале стирает пользовательскую программу и любые форсированные значения из внутренней загрузочной памяти, а затем копирует программу из карты передачи во внутреннюю загрузочную память. Когда процесс передачи завершен, Вы должны удалить карту передачи.

Вы можете использовать пустую карту передачи, чтобы получить доступ к защищенному паролем ЦПУ, когда пароль был потерян или забыт (стр. 144). Вставка пустой карты передачи удаляет защищенную паролем программу во внутренней загрузочной памяти ЦПУ. Вы можете тогда загрузить новую программу в ЦПУ.

- Используйте карту программирования (стр. 138) в качестве внешней загрузочной памяти для ЦПУ. Вставка карты программирования в ЦПУ стирает всю внутреннюю загрузочную память ЦПУ (пользовательская программа и любые форсированные значения). ЦПУ в этом случае выполняет программу во внешней загрузочной памяти (карта программирования). Загрузка в ЦПУ, у которого есть карта программирования, обновляет только внешнюю загрузочную память (карта программирования). Поскольку внутренняя загрузочная память ЦПУ была стерта, когда Вы вставили карту программирования, карта программирования должна остаться в ЦПУ. Если Вы удаляете карту программирования, ЦПУ переходит в режим STOP. (Светодиод ошибки загорается, чтобы указать на то, что карта программирования была удалена.)

Скопированная на карту памяти программа включает в себя кодовые блоки, блоки данных, технологические объекты и конфигурацию устройства. Карта памяти не содержит форсированные значения. Форсированные значения не являются частью программы, но сохраняются в загрузочной памяти, либо внутренней, либо внешней (карта программирования). Если карта программирования вставлена в ЦПУ, STEP 7 применяет форсированные значения только к внешней загрузочной памяти на карте программирования.

Вы также используете карту памяти при загрузке обновлений встроенного ПО (стр. 141).

5.5.1. Вставка карты памяти в ЦПУ

ЗАМЕТКА

Защитите карту памяти и гнездо от электростатического разряда

Электростатический разряд может повредить карту памяти или гнездо в ЦПУ.

Прикоснитесь к заземленной контактной площадке и/или наденьте заземленный браслет, когда Вы работаете с картой памяти. Храните карту памяти в проводящем контейнере.



Проверьте, что карта памяти не защищена от записи. Сдвиньте переключатель защиты в сторону от позиции "Lock".

⚠ ПРЕДУПРЕЖДЕНИЕ

Убедитесь, что ЦПУ не управляет процессом прежде, чем вставить карту памяти.

Если Вы вставляете карту памяти (сконфигурированную как карта программирования, карту передачи или карту обновления встроенного ПО) в работающий ЦПУ, то ЦПУ сразу переходит в режим STOP, что могло бы вызвать нарушение процесса, послужившее причиной смерти или тяжелых телесных повреждений.

Прежде, чем вставить или удалить карту памяти, всегда убедитесь, что ЦПУ не находится в активной фазе управления машиной или процессом. Всегда используйте цепь аварийного останова для своего приложения или процесса.

Примечание

Не вставляйте карты передачи программы V3.0 в ЦПУ S7-1200 V4.0.

Карты передачи программы версии 3.0 не совместимы с ЦПУ S7-1200 версии V4.0. Вставка карты памяти, которая содержит программу V3.0, вызывает ошибку ЦПУ.

Если Вы действительно вставляете карту передачи программы недопустимой версии (стр. 135), Вы должны удалить карту и выполнить переход из STOP в RUN, сброс памяти (MRES) или цикл питания. После того, как Вы выведете ЦПУ из состояния ошибки, Вы можете загрузить допустимую программу для ЦПУ V4.0.

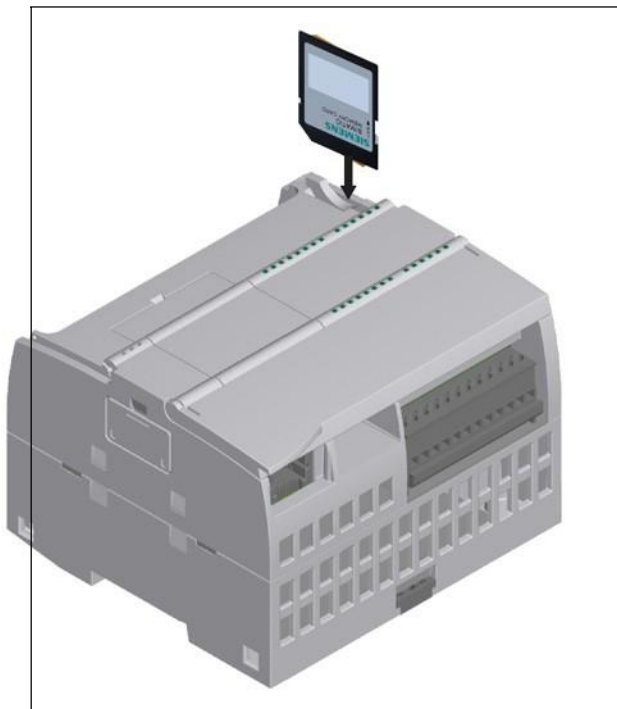
Для перевода программы V3.0 в программу V4.0, Вы должны использовать TIA Portal, чтобы сменить устройство в аппаратной конфигурации.

Примечание

Не вставляйте карты передачи программы V3.0 в ЦПУ S7-1200 V4.0.

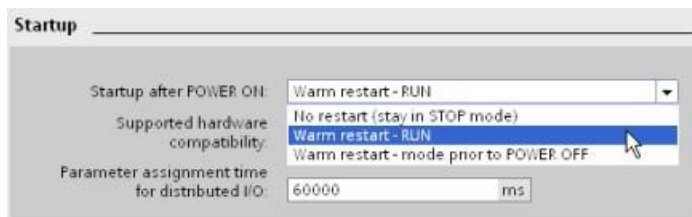
Если Вы вставляете карту памяти в ЦПУ, находящемся в режиме STOP, диагностический буфер выводит на экран сообщение о том, что инициирована оценка карты памяти. ЦПУ оценит карту памяти при следующем переходе ЦПУ в режим RUN, сбросе памяти ЦПУ с помощью MRES или цикл включения и выключения питания ЦПУ.

Таблица 5- 43 Вставка карты памяти

	Чтобы вставить карту памяти, откройте верхнюю дверцу ЦПУ и вставьте карту памяти в слот. Нажимной соединитель допускает простую вставку и удаление. Карта памяти снабжена ключом для надлежащей установки.
---	---

5.5.2. Конфигурирование параметра запуска ЦПУ прежде, чем копировать проект на карту памяти

Когда Вы копируете программу на карту передачи или карту программирования, программа содержит в себе параметр запуска для ЦПУ. Прежде, чем скопировать программу в карту памяти, убедитесь, что Вы сконфигурировали рабочий режим для ЦПУ после цикла включения и выключения питания. Выберите, должен ли запускаться ЦПУ в режиме STOP, режиме RUN или в предыдущем режиме (до цикла включения и выключения питания).



5.5.3. Карта передачи

ЗАМЕТКА

Защитите карту памяти и гнездо от электростатического разряда

Электростатический разряд может повредить карту памяти или гнездо в ЦПУ.

Прикоснитесь к заземленной контактной площадке и/или наденьте заземленный браслет, когда Вы работаете с картой памяти. Храните карту памяти в проводящем контейнере.

Создание карты передачи

Никогда не забывайте конфигурировать параметр запуска ЦПУ (стр. 135) прежде, чем скопировать программу на карту передачи. Чтобы создать карту передачи, выполните следующие шаги:

1. Вставьте пустую карту памяти SIMATIC, которая не защищена от записи в считыватель SD-карт, подключенный к Вашему компьютеру. (Если карта защищена от записи, сдвиньте переключатель защиты в сторону от позиции "Lock".)

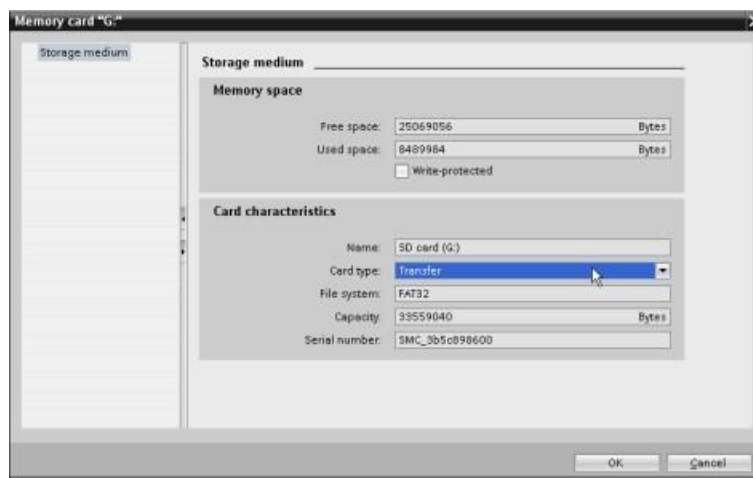
Если Вы повторно используете карту памяти SIMATIC, которая содержит пользовательскую программу или обновление встроенного ПО, Вы **должны** удалить программные файлы прежде, чем повторно использовать карту. Используйте Windows Explorer, чтобы вывести на экран содержимое карты памяти и удалить файл "S7_JOB.S7S", а также удалить любые существующие папки "Data Logs" и папку каталога (такие как "SIMATIC.S7S" или "FWUPDATE.S7S").

ЗАМЕТКА

Не удаляйте скрытые файлы " LOG " и "crdinfo.bin" с карты памяти.

Файлы " LOG " и "crdinfo.bin" требуются для карты памяти. Если Вы удаляете эти файлы, Вы не можете использовать карту памяти с ЦПУ.

2. В дереве проекта (представление Project view), разверните папку "SIMATIC Card Reader" и выберите свой считыватель карт.
3. Выведите на экран диалоговое окно "Memory card", щелкнув правой кнопкой по букве диска, соответствующей карте памяти в считывателе и выбрав "Properties" из контекстного меню.
4. В диалоговом окне "Memory card" выберите "Transfer" из выпадающего меню "Card type".
В этот момент STEP 7 создает пустую карту передачи. Если Вы создаете пустую карту передачи, например, чтобы восстановиться после потерянного пароля ЦПУ (стр. 144), удалите карту передачи из считывателя.



5. Добавьте программу, выбрав ЦПУ (например PLC_1 [CPU 1214C DC/DC/DC]) в дереве проекта и перетащив ЦПУ на карту памяти. (Другой метод состоит в копировании ЦПУ и вставке его на карте памяти.) Копирование ЦПУ на карту памяти открывает диалоговое окно "Load preview".

6. В диалоговом окне "Load preview" нажмите кнопку "Load", чтобы скопировать ЦПУ на карту памяти.
7. Когда диалоговое окно выводит на экран сообщение, что ЦПУ (программа) было загружено без ошибок, нажмите кнопку "Finish".

Использование карты передачи

ПРЕДУПРЕЖДЕНИЕ

Убедитесь, что ЦПУ не управляет процессом прежде, чем вставить карту памяти.

Вставка карты памяти заставит ЦПУ перейти в режим STOP, что могло бы повлиять на работающие процесс или машину. Непредсказуемый ход процесса или работа машины могли бы привести к смерти или травмам персонала и/или материальному ущербу.

Прежде, чем вставить карту передачи, каждый раз убеждайтесь, что ЦПУ находится в режиме STOP, и Ваш процесс находится в безопасном состоянии.

Примечание

Не вставляйте карты передачи программы V3.0 в ЦПУ S7-1200 V4.0.

Карты передачи программы версии 3.0 не совместимы с ЦПУ S7-1200 версии V4.0.

Вставка карты памяти, которая содержит программу V3.0, вызывает ошибку ЦПУ.

Если Вы действительно вставляете карту передачи программы недопустимой версии (стр. 135), Вы должны удалить карту и выполнить переход из STOP в RUN, сброс памяти (MRES) или цикл питания. После того, как Вы выведете ЦПУ из состояния ошибки, Вы можете загрузить допустимую программу для ЦПУ V4.0.

Чтобы перенести программу в ЦПУ, выполните следующие шаги:

1. Вставьте карту передачи в ЦПУ (стр. 133). Если ЦПУ будет в режиме RUN, то ЦПУ перейдет в режим STOP. Светодиод обслуживания (MAINT) мигает, указывая, что карта памяти должна быть оценена.
2. Выполните цикл включения и выключения питания ЦПУ, чтобы оценить карту памяти. Альтернативными методами для перезагрузки ЦПУ являются выполнение перехода STOP-в-RUN или сброс памяти (MRES) из STEP 7.
3. После перезагрузки и оценки карты памяти, ЦПУ копирует программу во внутреннюю загрузочную память. Светодиод RUN/STOP поочередно мигает зеленым и желтым, чтобы указать, на процесс копирования программы. Когда светодиод RUN/STOP загорается (чисто желтым) и светодиод MAINT мигает, процесс копирования окончен. Вы можете удалить карту памяти.
4. Перезагрузите ЦПУ (выполняя цикл питания или альтернативными методами), чтобы оценить новую программу, которая была передана во внутреннюю загрузочную память.

ЦПУ затем переходит в режим запуска (RUN или STOP), который Вы сконфигурировали для проекта.

Примечание

Вы должны удалить карту передачи прежде, чем перевести ЦПУ в режим RUN.

5.5.4. Карта программирования

ЗАМЕТКА

Электростатический разряд может повредить карту памяти или гнездо в ЦПУ.

Прикоснитесь к заземленной контактной площадке и/или наденьте заземленный браслет, когда Вы работаете с картой памяти. Храните карту памяти в проводящем контейнере.



Проверьте, что карта памяти не защищена от записи. Сдвиньте переключатель защиты в сторону от позиции "Lock".

Прежде чем Вы скопируете любые элементы программы на карту программирования, удалите любые ранее сохраненные программы из карты памяти.

Создание карты программирования

Карта памяти, если она используется в качестве карты программирования, является внешней загрузочной памятью ЦПУ. Если Вы удаляете карту программирования, то внутренняя загрузочная память ЦПУ пуста.

Примечание

Если Вы вставляете пустую карту памяти в ЦПУ и выполняете оценку карты памяти, выполняя цикл питания ЦПУ, переход из STOP в RUN сброс памяти (MRES), программа и форсированные значения, находящиеся во внутренней загрузочной памяти ЦПУ копируются на карту памяти. (Карта памяти после этого становится картой программирования.) После того, как копирование было завершено, программа во внутренней загрузочной памяти ЦПУ стерта. ЦПУ затем переходит в сконфигурированный режим запуска (RUN или STOP).

Никогда не забывайте конфигурировать параметр запуска ЦПУ (стр. 135) прежде, чем скопировать программу на карту передачи. Чтобы создать карту программирования, выполните следующие шаги:

1. Вставьте пустую карту памяти SIMATIC, которая не защищена от записи в считыватель SD-карт, подключенный к Вашему компьютеру. (Если карта защищена от записи, сдвиньте переключатель защиты в сторону от позиции "Lock".)

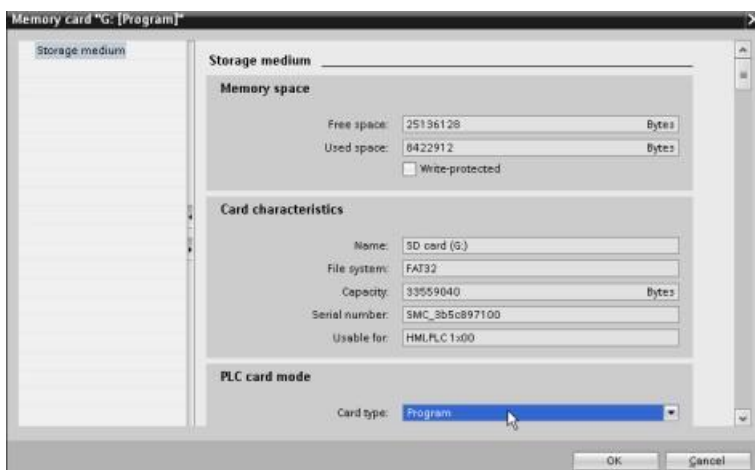
Если Вы повторно используете карту памяти SIMATIC, которая содержит пользовательскую программу или обновление встроенного ПО, Вы **должны** удалить программные файлы прежде, чем повторно использовать карту. Используйте Windows Explorer, чтобы вывести на экран содержимое карты памяти и удалить файл "S7_JOB.S7S", а также удалить любые существующие папки "Data Logs" и папку каталога (такие как "SIMATIC.S7S" или "FWUPDATE.S7S").

ЗАМЕТКА

Не удаляйте скрытые файлы " LOG " и "crdinfo.bin" с карты памяти.

Файлы " LOG " и "crdinfo.bin" требуются для карты памяти. Если Вы удаляете эти файлы, Вы не можете использовать карту памяти с ЦПУ.

2. В дереве проекта (представление Project view), разверните папку "SIMATIC Card Reader" и выберите свой считыватель карт.
3. Выведите на экран диалоговое окно "Memory card", щелкнув правой кнопкой по букве диска, соответствующей карте памяти в считывателе и выбрав "Properties" из контекстного меню.
4. В диалоговом окне "Memory card" выберите "Program" из выпадающего меню.



5. Добавьте программу, выбрав ЦПУ (например PLC_1 [CPU 1214C DC/DC/DC]) в дереве проекта и перетащив ЦПУ на карту памяти. (Другой метод состоит в копировании ЦПУ и вставке его на карту памяти.) Копирование ЦПУ на карту памяти открывает диалоговое окно "Load preview".
6. В диалоговом окне "Load preview" нажмите кнопку "Load", чтобы скопировать ЦПУ на карту памяти.
7. Когда диалоговое окно выводит на экран сообщение, что ЦПУ (программа) было загружено без ошибок, нажмите кнопку "Finish".

Использование карты программирования в качестве загрузочной памяти Вашего CPU



ПРЕДУПРЕЖДЕНИЕ

Риски, связанные со вставкой карты программирования

Убедитесь, что ЦПУ не находится в активной фазе управления процессом прежде, чем вставить карту памяти.

Вставка карты памяти заставит ЦПУ перейти в режим STOP, что могло бы повлиять на работу работающих процесса или машины. Непредсказуемый ход процесса или работа машины могли бы привести к смерти или травмам персонала и/или материальному ущербу.

Прежде, чем вставить карту передачи, каждый раз убеждайтесь, что ЦПУ находится в автономном режиме и безопасном состоянии.

Чтобы использовать карту программирования в Вашем ЦПУ, выполните следующие шаги:

1. Вставьте карту программирования в ЦПУ (стр. 133). Если ЦПУ будет в режиме RUN, то ЦПУ перейдет в режим STOP. Светодиод обслуживания (MAINT) мигает, указывая, что карта памяти должна быть оценена.
2. Выполните цикл включения и выключения питания ЦПУ, чтобы оценить карту памяти. Альтернативными методами для перезагрузки ЦПУ являются выполнение перехода STOP-в-RUN или сброс памяти (MRES) из STEP 7.
3. После перезагрузки и оценки карты памяти, ЦПУ очищает внутреннюю загрузочную память.

ЦПУ затем переходит в режим запуска (RUN или STOP), который Вы сконфигурировали для него.

Карта программирования должна оставаться в ЦПУ. Удаление карты программирования оставляет ЦПУ без программы во внутренней загрузочной памяти.



ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с удалением карты программирования

Если Вы удаляете карту программирования, ЦПУ теряет свою внешнюю загрузочную память и генерирует ошибку. ЦПУ переходит в режим STOP и зажигает светодиод ошибки.

Управляющие устройства могут перестать работать в небезопасном состоянии, приводящем к непредвиденной работе управляемого оборудования. Такие непредвиденные условия могли бы привести к смерти или серьезным травмам персонала и/или повреждению оборудования.

Не удаляйте карту программирования, не осознав, что Вы удаляете программу из ЦПУ.

Обновление встроенного ПО

Вы можете использовать карту памяти для выполнения обновления встроенного ПО. Альтернативными методами являются использование страницы информации о модуле (стр. 804) на веб-сервере или использование он-лайн и диагностических функций STEP 7, чтобы выполнить обновление встроенного ПО (стр. 1074). Настоящая глава объясняет метод, который использует карту памяти.

ЗАМЕТКА

Защитите карту памяти и гнездо от электростатического разряда

Электростатический разряд может повредить карту памяти или гнездо в ЦПУ.

Прикоснитесь к заземленной контактной площадке и/или наденьте заземленный браслет, когда Вы работаете с картой памяти. Храните карту памяти в проводящем контейнере.

Вы используете карту памяти, когда загружаете обновления встроенного ПО с сайта поддержки пользователей (<http://www.siemens.com/tiaportal>). На этом сайте перейдите к **Automation Technology > Automation Systems > SIMATIC Industrial Automation Systems > PLC > Modular controllers SIMATIC S7 > SIMATIC S7-1200**. Отсюда продолжайте перемещаться к определенному типу модуля, который Вы должны обновить. Под "Support" щелкните по ссылке для "Software Downloads", чтобы перейти к следующему этапу.

В качестве альтернативы Вы можете обратиться напрямую к веб-странице загрузок для S7-1200 (<http://support.automation.siemens.com/WW/view/en/34612486/133100>).

Примечание

Вы не можете обновить S7-1200 ЦПУ V3.0 или более ранней версии до S7-1200 V4.0 или V4.1 посредством обновления встроенного ПО.

ЗАМЕТКА

Не используйте утилиту Windows или любую другую утилиту, чтобы выполнить форматирование карты памяти

Если карта памяти Siemens будет переформатирована, используя утилиту Microsoft Windows, то карта памяти больше не будет годной к использованию в ЦПУ S7-1200.

Чтобы загрузить обновление встроенного ПО на Вашу карту памяти, выполните следующие шаги:

1. Вставьте пустую карту памяти SIMATIC, которая не защищена от записи в считыватель SD-карт, подключенный к Вашему компьютеру. (Если карта защищена от записи, сдвиньте переключатель защиты в сторону от позиции "Lock".)
Вы можете повторно использовать карту памяти SIMATIC, которая содержит пользовательскую программу или другое обновление встроенного ПО, но Вы должны удалить некоторые файлы на карте памяти.

Если Вы повторно используете карту памяти SIMATIC, которая содержит пользовательскую программу или обновление встроенного ПО, Вы **должны** удалить программные файлы прежде, чем повторно использовать карту. Используйте Windows Explorer, чтобы вывести на экран содержимое карты памяти и удалить файл "S7_JOB.S7S", а также удалить любые существующие папки "Data Logs" и папку каталога (такие как "SIMATIC.S7S" или "FWUPDATE.S7S").

ЗАМЕТКА

Не удаляйте скрытые файлы " LOG " и "crdinfo.bin" с карты памяти.

Файлы " LOG " и "crdinfo.bin" требуются для карты памяти. Если Вы удаляете эти файлы, Вы не можете использовать карту памяти с ЦПУ.

2. Выберите самораспаковывающийся файл (.exe) для обновления встроенного ПО, которое соответствует Вашему модулю, и загрузите его на свой компьютер. Дважды щелкните по файлу обновления, установите целевой путь файла на корневой каталог карты памяти SIMATIC и запустите процесс распаковки. После того, как распаковка завершена, корневой каталог (папка) карты памяти будет содержать каталог "FWUPDATE.S7S" и файл "S7_JOB.S7S".

3. Аккуратно извлеките карту памяти из считывателя.

Чтобы установить обновление встроенного ПО, выполните следующие шаги:



ПРЕДУПРЕЖДЕНИЕ

Убедитесь, что ЦПУ не находится в активной фазе управления процессом прежде, чем обновлять встроенное ПО.

Установка обновления встроенного ПО заставит ЦПУ перейти в режим STOP, что могло бы повлиять на работу работающих процесса или машины. Непредсказуемый ход процесса или работа машины могли бы привести к смерти или травмам персонала и/или материальному ущербу.

Прежде, чем вставить карту памяти, каждый раз убеждайтесь, что ЦПУ находится в автономном режиме и безопасном состоянии.

1. Вставьте карту памяти в ЦПУ. Если ЦПУ будет в режиме RUN, то ЦПУ перейдет в режим STOP. Светодиод обслуживания (MAINT) мигает, указывая, что карта памяти должна быть оценена.
2. Выполните цикл питания ЦПУ, чтобы запустить обновление встроенного ПО. Альтернативными методами для перезагрузки ЦПУ являются выполнение перехода STOP-в-RUN или сброс памяти (MRES) из STEP 7.

Примечание

Чтобы завершить обновление встроенного ПО, Вы должны обеспечить наличие внешнего питания 24 В пост.т. для модуля.

После перезагрузки ЦПУ запускается обновление встроенного ПО. Светодиод RUN/STOP мигает поочередно зеленым и желтым цветом, указывая, что обновление копируется. Когда светодиод RUN/STOP будет гореть непрерывно (чисто желтым) и будет мигать светодиод MAINT, процесс копирования завершен. Вы должны затем удалить карту памяти.

3. После удаления карты памяти вновь перезагрузите ЦПУ (или выполняя цикл питания или альтернативными методами перезагрузки), чтобы загрузить новое встроенное ПО.

Обновление встроенного ПО не затрагивает пользовательскую программу и аппаратную конфигурацию. Когда на ЦПУ подается питание, он переходит в сконфигурированное состояние запуска. (Если режим запуска для Вашего ЦПУ был сконфигурирован на "Warm restart - mode before POWER OFF", то ЦПУ будет в режиме STOP, потому что последним состоянием ЦПУ было STOP.)

Примечание**Обновление нескольких модулей, подключенных к ЦПУ**

Если Ваша аппаратная конфигурация содержит несколько модулей, которые соответствуют одному файлу обновления встроенного ПО на карте памяти, то ЦПУ применяет обновления ко всем подходящим модулям (CM, SM и SB) в порядке конфигурации, т.е. в порядке увеличения позиции модуля в конфигурации устройства в STEP 7.

Если Вы загрузили несколько обновлений встроенного ПО на карту памяти для нескольких модулей, то ЦПУ применяет обновления в порядке, в котором Вы загружали их на карту памяти.

5.6 Восстановление из состояния потерянного пароля

Если Вы потеряли пароль для защищенного паролем ЦПУ, используйте пустую карту передачи, чтобы удалить защищенную паролем программу. Пустая карта передачи стирает внутреннюю загрузочную память ЦПУ. Вы можете тогда загрузить новую пользовательскую программу в ЦПУ из STEP 7.

Для получения информации о создании и использовании пустой карты передачи, смотрите раздел для карт передачи (стр. 135).



ПРЕДУПРЕЖДЕНИЕ

Убедитесь, что ЦПУ не управляет процессом прежде, чем вставить карту памяти.

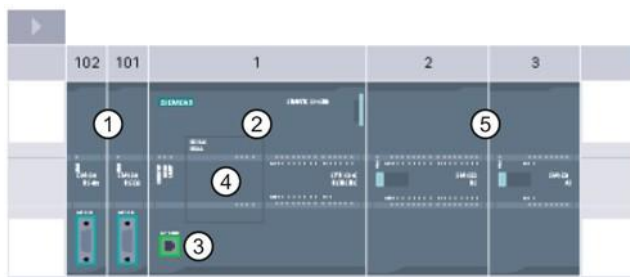
Если Вы вставляете карту передачи в работающий ЦПУ, то он перейдет в режим STOP. Управляющие устройства могут прекратить работу в небезопасном состоянии, приводящем к неожиданной работе управляемого оборудования. Такие непредвиденная работа могла бы привести к смерти или серьезным травмам персонала и/или повреждению оборудования.

Прежде, чем вставить карту передачи, всегда убеждайтесь, что ЦПУ находится в режиме STOP, и Ваш процесс находится в безопасном состоянии.

Вы должны удалить карту передачи прежде, чем перевести ЦПУ в режим RUN.

Конфигурирование устройств

Вы создаете конфигурацию устройств для своего ПЛК, добавляя ЦПУ и дополнительные модули в Ваш проект.



- ① Коммуникационный модуль (CM) или коммуникационный процессор (CP): максимум 3, вставляемых в слоты 101, 102 и 103
- ② ЦПУ: слот 1
- ③ PROFINET порт ЦПУ
- ④ Сигнальная плата (SB), коммуникационная плата (CB) или батарейная плата (BB): максимум 1, вставляемая в ЦПУ
- ⑤ Сигнальный модуль (SM) для цифрового или аналогового ввода/вывода: максимум 8, вставляемые в слоты со 2 по 9
(CPU 1214C, CPU 1215C и CPU 1217C позволяют 8, CPU 1212C позволяет 2, CPU 1211C не позволяет ни одного)

Управление конфигурацией

Конфигурация устройства для S7-1200 также поддерживает "управление конфигурацией (стр. 151)", где Вы можете сконфигурировать максимальную конфигурацию для проекта включая модули, которые Вы могли бы фактически не использовать. Эта функция, иногда также называемая "оперирование опциями", позволяет Вам сконфигурировать максимальную конфигурацию, которую Вы могли бы использовать с изменениями в части установленных модулей в нескольких приложениях.

6.1 Вставка ЦПУ

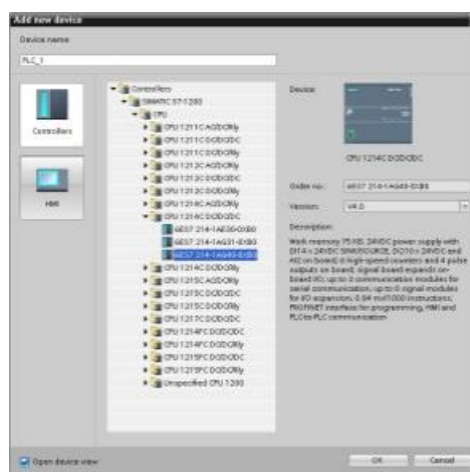
Вы создаете свою конфигурацию устройства, вставляя ЦПУ в Ваш проект.

- В представлении Portal view выберите "Devices & Networks" и щелкните по "Add new device".
- В представлении Project view, под именем проекта, дважды щелкните по "Add new device".

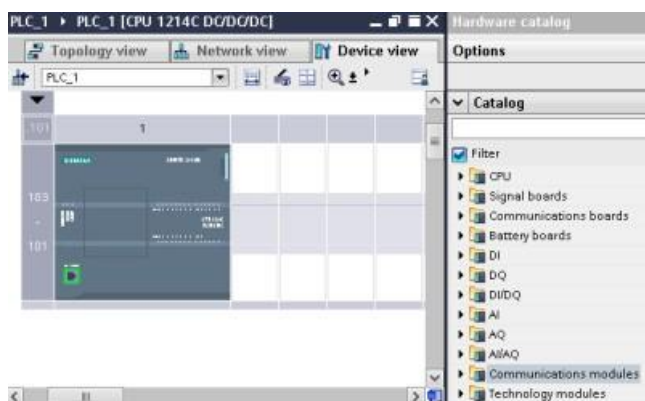


Убедитесь, что Вы вставляете корректную модель и версию встроенного ПО из списка. При выборе ЦПУ в диалоговом окне "Add new device" создается стойка и ЦПУ.

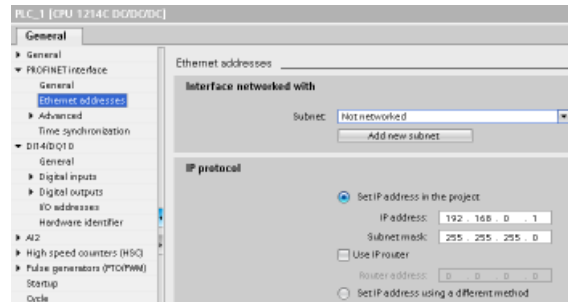
Диалог "Add new device"



Представление Device view аппаратной конфигурации



Выбор ЦПУ в представлении Device view выводит свойства ЦПУ в окне инспектора.



Примечание

ЦПУ не имеет предварительно сконфигурированного IP-адреса. Вы должны вручную присвоить IP-адрес ЦПУ во время конфигурации устройства. Если Ваш ЦПУ соединен с маршрутизатором в сети, Вы также вводите IP-адрес для маршрутизатора.

6.2 Выгрузка конфигурации подключенного ЦПУ

STEP 7 предлагает два метода для выгрузки аппаратной конфигурации подключенного ЦПУ:

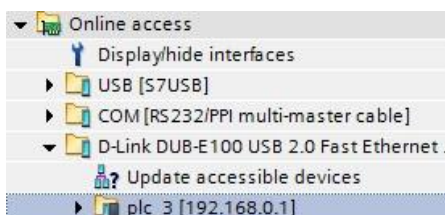
- Выгрузка подключенного устройства как новую станцию
- Конфигурирование неустановленного ЦПУ и обнаружение аппаратной конфигурации подключенного ЦПУ

Имейте ввиду, однако, что первый метод выгружает как аппаратную конфигурацию, так и программное обеспечение подключенного ЦПУ.

Выгрузка устройства как новой станции

Чтобы выгрузить подключенное устройство как новую станцию, выполните следующие шаги:

1. Раскройте свой коммуникационный интерфейс из узла дерева проекта "Online access".
2. Дважды щелкните по "Update accessible devices".
3. Выберите ПЛК из обнаруженных устройств.



4. Из он-лайн меню STEP 7 выберите команду меню "Upload device as new station (аппаратная и программная часть)".

STEP 7 выгружает аппаратную конфигурацию и программные блоки.

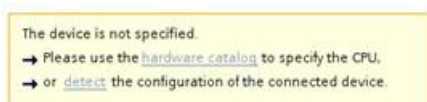
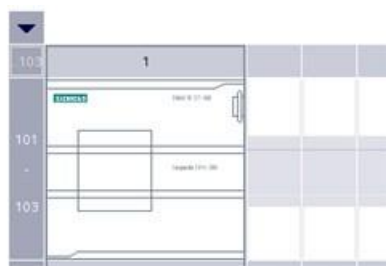
Обнаружение аппаратной конфигурации неустановленного ЦПУ



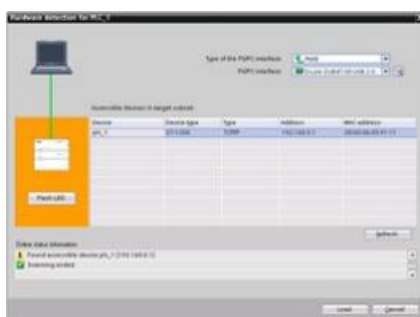
Если Вы подключаетесь к ЦПУ, Вы можете выгрузить конфигурацию этого ЦПУ, включая любые модули в Ваш проект. Просто создайте новый проект и выберите "unspecified CPU" вместо того, чтобы выбрать определенный ЦПУ. (Вы можете также полностью пропустить конфигурацию устройства, выбирая "Create a PLC program" в процедуре "First steps". STEP 7 в этом случае автоматически создает неустановленный ЦПУ.)

В программном редакторе Вы выбираете команду "Hardware detection" из меню "Online".

Из редактора конфигурации устройства Вы выбираете опцию для обнаружения конфигурации подключенного устройства.



После того, как Вы выберете CPU в он-лайн диалоговом окне и нажмете кнопку Load, STEP 7 выгружает аппаратную конфигурацию из ЦПУ, включая любые модули (SM, SB или CM). Вы можете затем сконфигурировать параметры для ЦПУ и модулей (стр. 162).



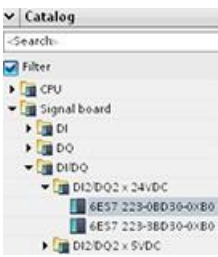
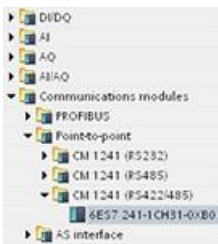
6.3 Добавление модулей в конфигурацию

Используйте аппаратный каталог, чтобы добавить модули к ЦПУ:

- Сигнальный модуль (SM) обеспечивает дополнительные цифровые или аналоговые каналы ввода-вывода. Эти модули подключаются с правой стороны ЦПУ.
- Сигнальная плата (SB) обеспечивает лишь несколько дополнительных каналов ввода-вывода для ЦПУ. SB устанавливается на передней стороне ЦПУ.
- Батарейная плата 1297 (BB) обеспечивает долгосрочное резервное копирование часов реального времени. BB устанавливается на передней стороне ЦПУ.
- Коммуникационная плата (CB) обеспечивает дополнительный коммуникационный порт (такой как RS485). CB устанавливается на передней стороне ЦПУ.
- Коммуникационный модуль (CM) и коммуникационный процессор (CP) обеспечивают дополнительный коммуникационный порт, такой как PROFIBUS или GPRS. Эти модули подключаются с левой стороны ЦПУ.

Чтобы вставить модуль в конфигурацию устройства, выберите модуль в аппаратном каталоге и либо дважды щелкните, либо перетащите модуль в выделенный слот. Вы должны добавить модули в конфигурацию устройства и загрузите аппаратную конфигурацию в ЦПУ, чтобы модули стали бы функциональными.

Таблица 6- 1 Добавление модуля в конфигурацию устройства

Модуль	Выберите модуль	Вставьте модуль	Результат
SM			
SB, BB или CB			
CM или CP			

С помощью функции "управления конфигурацией" (стр. 151) Вы можете добавить сигнальные модули и сигнальные платы в Вашу конфигурацию устройства, которая могла бы не соответствовать фактическим аппаратным средствам для определенного приложения, но она будет использоваться в связанных приложениях, которые совместно используют пользовательскую программу, модель ЦПУ и, возможно, некоторые сконфигурированные модули.

6.4 Управление конфигурацией

6.4.1. Преимущества и практическое использование управления конфигурацией

Управление конфигурацией может быть полезным, когда Вы создаете решение для автоматизации (машина), которую Вы намереваетесь использовать с изменениями в нескольких установках.

Вы можете загрузить STEP 7 конфигурацию устройства и пользовательскую программу в различные установленные ПЛК конфигурации. Вы должны выполнить лишь несколько простых доработок, чтобы сделать проект STEP 7 соответствующим фактической установке.

6.4.2. Конфигурирование центральной установки и дополнительных модулей

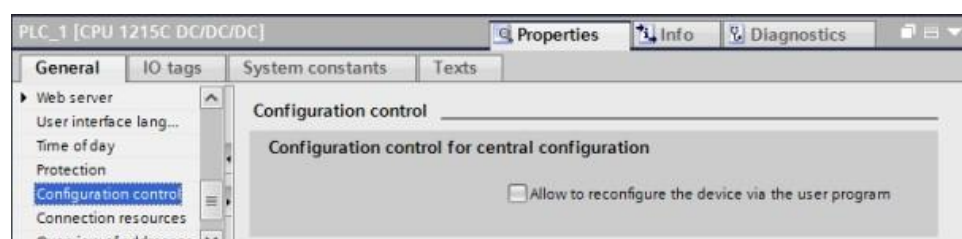
Управление конфигурацией с помощью STEP 7 и S7-1200 позволяет Вам сконфигурировать максимальную конфигурацию для стандартной машины и управлять версиями (опциями), которые используют разновидности этой конфигурации. Руководство PROFINET со STEP 7 (<http://support.automation.siemens.com/WW/view/en/49948856>) ссылается на эти типы проектов как на "проект стандартной машины".

Запись управляющих данных, которую Вы программируете в блоке запуска, уведомляет ЦПУ, относительно тех модулей, которые отсутствуют в реальной установке по сравнению с конфигурацией или какие модули расположены в других слотах по сравнению с конфигурацией. Управление конфигурацией не оказывает влияние на назначение параметров модулей.

Управление конфигурацией дает Вам гибкость, чтобы вносить поправки в установку, пока Вы можете получить реальную конфигурацию из максимальной конфигурации устройства в STEP 7.

Чтобы активировать управление конфигурацией и структурировать требуемую запись управляющих данных, выполните следующие шаги:

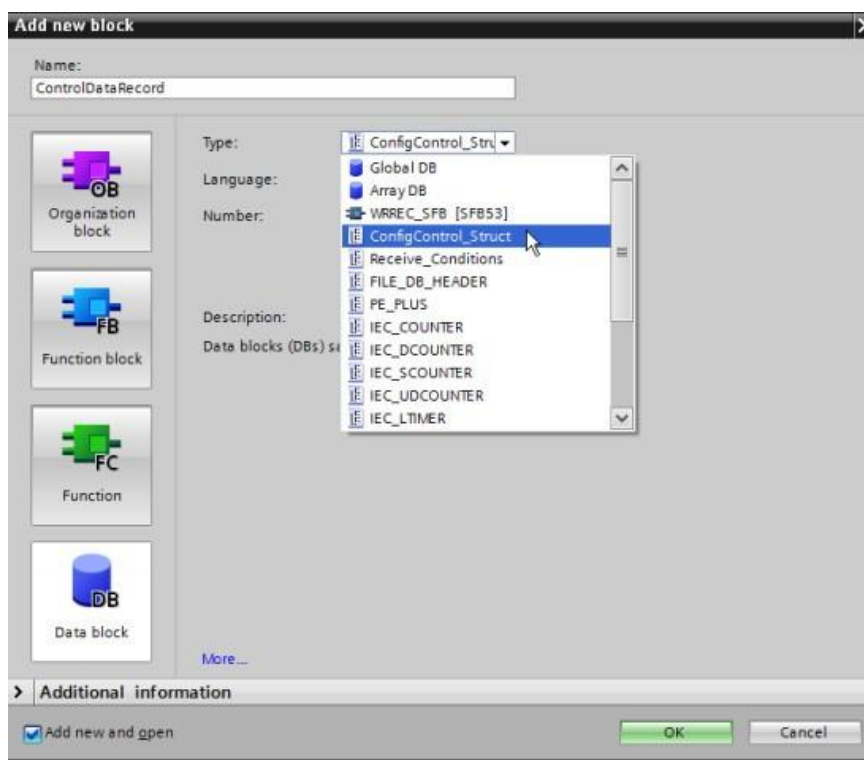
1. Дополнительно, выполните сброс ЦПУ на заводские настройки, чтобы гарантировать, что несовместимая запись управляющей информации не присутствует в ЦПУ.
2. Выберите ЦПУ в конфигурации устройства в STEP 7.
3. Из узла Configuration control в свойствах ЦПУ установите флажок "Enable reconfiguration of device with user program".



4. Создайте ПЛК тип данных, чтобы хранить запись управляющих данных. Сконфигурируйте его как структуру, которая включает в себя четыре элемента USint для информации об управлении конфигурацией и дополнительные USint, соответствующие слотам максимальной конфигурации устройства S7-1200, следующим образом:

ConfigControl_Struct				
	Name	Data type	Default value	Comment
1	▼ ConfigControl	Struct		
2	Block_length	USInt	16	Length of control data record, including header
3	Block_ID	USInt	196	Data record number
4	Version	USInt	5	
5	Subversion	USInt	0	
6	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
7	Slot_2	USInt	255	Configured slot 2/ Assigned "real" slot
8	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
9	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
10	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
11	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
12	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
13	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
14	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
15	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
16	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
17	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

5. Создайте блок данных на основе ПЛК типа данных, который Вы создали.



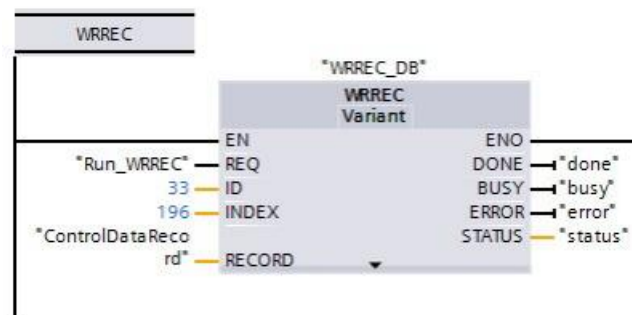
6. В этом блоке данных сконфигурируйте Block_length, Block_ID, Version и Subversion, как показано ниже. Сконфигурируйте значения для слотов на основании их наличия или отсутствия и позиции в Вашей текущей установке:
- 0: сконфигурированный модуль отсутствует в реальной конфигурации. (слот пустой.);
 - от 1 до 9, от 101 до 103: реальная позиция для сконфигурированного слота;
 - 255: STEP 7 конфигурация устройства не содержит модуля в этом слоте.

ControlDataRecord				
	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex c..
8	Slot_2	USInt	255	Configured slot 2/ Assigned "real" slot
9	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

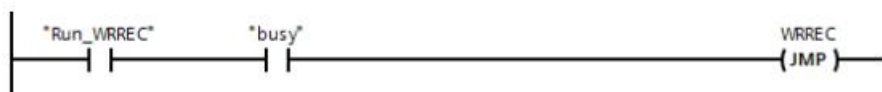
Смотри пример управления конфигурацией (стр. 157) для разъяснений относительно того, как назначить значения для слотов.

7. В ОВ запуска вызовите расширенную инструкцию WRREC (Write data record), чтобы перенести записи управляющих данных, которую Вы создали в индекс 196 аппаратного устройства с ID 33. Используйте инструкции метки и JMP (прыжок), чтобы дождаться завершения инструкции WRREC.

Network 1:



Network 2:



Примечание

Управление конфигурацией не имеет силы пока инструкция WRREC в ОВ запуска не переносит запись управляющих данных. Если Вы активировали управление конфигурацией, а в ЦПУ отсутствует запись управляющих данных, то ЦПУ перейдет в режим STOP после выхода из режима STARTUP. Убедитесь, что Вы запрограммировали ОВ запуска для передачи записи управляющих данных.

Упорядочение модулей

В следующей таблице представлены назначения номеров слотов:

Слот	Модули
1	Сигнальная плата или коммуникационная плата (дополнительная карта ЦПУ)
от 2 до 9	Сигнальные модули
от 101 до 103	Коммуникационные модули

Запись управляющих данных

Запись управляющих данных 196 содержит назначения слотов и представляет актуальную конфигурацию, как показано ниже:

Байт	Элемент	Значение	Пояснение
0	Длина блока	16	Заголовок
1	ID блока	196	
2	Версия	5	
3	Подверсия	0	
4	Назначение для дополнительной платы ЦПУ	Реальная дополнительная плата, 0 или 255*	Управляющий элемент В каждом элементе описывает, какой из реальных слотов устройства назначен сконфигурированному слоту.
5	Назначение сконфигурированного слота 2	Реальный слот, 0 или 255*	
...	...	...	
12	Назначение сконфигурированного слота 9	Реальный слот, 0 или 255*	
13	Назначение сконфигурированного слота 101	Реальный слот или 255*	В отличие от сигнальных модулей реальный слот для физически существующих коммуникационных модулей должен быть тем же, что и сконфигурированный слот.
14	Назначение сконфигурированного слота 102	Реальный слот или 255*	
15	Назначение сконфигурированного слота 103	Реальный слот или 255*	

*Значения для слотов:

- 0: Сконфигурированный модуль отсутствует в реальной конфигурации. (слот пустой.)
- от 1 до 9, от 101 до 103: реальная позиция для сконфигурированного слота
- 255: STEP 7 конфигурация устройства не содержит модуля в этом слоте.

Примечание

Альтернативный способ создания ПЛК типа данных

В качестве альтернативы по созданию пользовательского ПЛК типа данных Вы можете создать блок данных непосредственно со всеми элементами структуры для записи управляющих данных. Вы могли бы сконфигурировать несколько структур в этом блоке данных, чтобы использовать его как хранилище нескольких конфигураций записи управляющих данных. Любой из вариантов является эффективным способом передачи записи управляющих данных в процессе запуска.

Правила

Соблюдайте следующие правила:

- Управление конфигурацией не поддерживает изменения позиции для коммуникационных модулей. Позиции слотов записи управляющих данных для слотов 101 - 103 должны соответствовать фактической установке. Если Вы не сконфигурировали модуль для слота, введите 255 для этой позиции слота.
- Вы не можете встроить пустые (неиспользованные) слоты между заполненными (используемыми) слотами. Для примера, если в фактической конфигурации есть модуль в слоте 4, то у фактической конфигурации должны также быть модули в слотах 2 и 3. Соответственно, если в фактической конфигурации есть коммуникационный модуль в слоте 102, то в фактической конфигурации должен также быть модуль в слоте 101.
- Если Вы разрешили управление конфигурацией, ЦПУ не готов к работе без записи управляющих данных. ЦПУ возвращается из запуска в режим STOP, если ОВ запуска не переносит действительную запись управляющих данных. ЦПУ не инициализирует центральный ввод-вывод в этом случае и вводит причину перехода в режим STOP в диагностический буфер.
- ЦПУ сохраняет успешно переданную запись управляющих данных в сохраняемой памяти, это означает, что нет необходимости снова переносить запись управляющих данных 196 при перезапуске, если Вы не изменяли конфигурацию.
- Каждый реальный слот должен присутствовать только один раз в записи управляющих данных.
- Вы можете назначить реальный слот только одному сконфигурированному слоту.

Примечание

Изменение конфигурации

Передача записи управляющих данных с измененной конфигурацией вызывает следующую автоматическую реакцию ЦПУ: сброс памяти и последующий запуск с измененной конфигурацией.

В результате этой реакции ЦПУ удаляет исходную запись управляющих данных и переносит новую запись управляющих данных с сохранением.

Поведение во время работы

Для он-лайн индикации и для индикации в буфере диагностики (модуль исправен или модуль дефектный), STEP 7 использует конфигурацию устройства а не отличающуюся реальную конфигурацию.

Пример: выходные данные диагностики модуля. Этот модуль сконфигурирован в слоте 4, но фактически вставлен в слот 3. Он-лайн представление показывает, что сконфигурированный слот 4 дефектный. В реальной конфигурации модуль в слоте 3 указывает на ошибку при помощи светодиодного индикатора.

Если Вы сконфигурировали модуль как отсутствующий в записи управляющих данных (0 запись), система автоматизации ведет себя следующим образом:

- Модули, определяемые как не существующие в записи управляющих данных, не предоставляют диагностические данные, и их состояние всегда в порядке. Значение статуса - в порядке.
- Прямой образцу процесса по выходам, которые отсутствуют не имеет эффекта; ЦПУ не сообщает о наличии какой-либо ошибки.
- Прямой доступ на чтение к входам или доступ на чтение к образцу процесса по входам, которые отсутствуют приводит к передаче значения "0" для каждого входа; ЦПУ не сообщает о наличии какой-либо ошибки.
- Передача записи данных в модуль, который отсутствует не имеет эффекта; ЦПУ не сообщает о наличии какой-либо ошибки.
- Попытка прочитать запись данных из модуля, который отсутствует приводит к ошибке, потому что ЦПУ не может вернуть допустимую запись данных.

Сообщения об ошибках

ЦПУ возвращает следующие сообщения об ошибках, если ошибка происходит во время передачи записи управляющих данных:

Код ошибки	Значение
16#80B1	Недопустимая длина; информация о длине в записи управляющих данных некорректна.
16#80B5	Параметры управления конфигурацией не назначены
16#80E2	Запись данных была передана в неправильном ОВ. Запись данных должна быть передана в ОВ запуска.
16#80B0	Тип блока (байт 2) в записи управляющих данных не равен 196.
16#80B8	Ошибка параметра; модуль сигнализирует о недопустимых параметрах, например: • Запись управляющих данных пытается изменить конфигурацию коммуникационного модуля или дополнительной коммуникационной карты. Реальная конфигурация для коммуникационных модулей и дополнительной коммуникационной карты должна совпадать с конфигурацией STEP 7. • Назначенное значение для несконфигурированного слота в STEP 7 проекте не равно 255. • Назначенное значение для сконфигурированного слота вне диапазона. • В назначенной конфигурации есть "внутренний" пустой слот, например, слот n назначен, а слот n-1 не назначен.

Пример управления конфигурацией

Настоящий пример описывает конфигурацию, состоящую из ЦПУ и трех модулей ввода-вывода. Модуль в слоте 3 отсутствует в первой реальной установке, таким образом Вы используете управление конфигурацией, чтобы "скрыть" его.

Во второй установке приложение включает в себя модуль, который был первоначально скрыт, но теперь установлен в последний слот. Измененная запись управляющих данных обеспечивает информацию относительно назначений слотов для модулей.

Пример: Реальная установка со сконфигурированным, но неиспользуемым модулем

Конфигурация устройства содержит все модули, которые могут присутствовать в реальной установке (максимальная конфигурация). В данном случае модуль, находящийся в слоте 3 в конфигурации устройства, отсутствует в реальной установке.

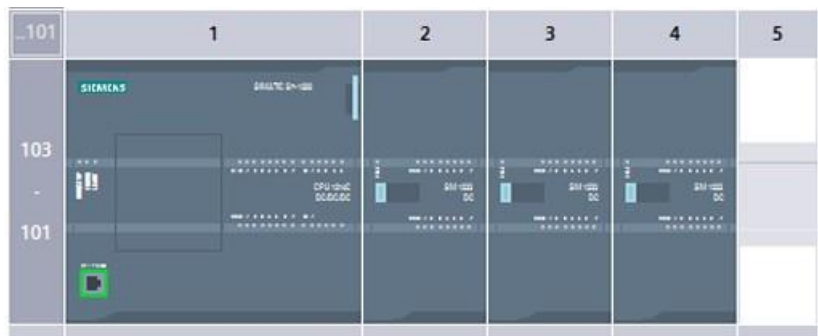


Рисунок 6-1 Максимальная конфигурация устройства



Рисунок 6-2 Реальная установка с отсутствующим модулем, сконфигурированным для слота 3, и модулем, сконфигурированным для слота 4, находящемся в слоте 3

Чтобы указать на отсутствие модуля Вы должны задать 0 в записи управляющих данных для модуля 3.

ControlDataRecord				
	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ca..
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	0	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

Пример: Реальная установка с модулем, впоследствии добавляемым в другой слот

Во втором примере модуль в слоте 3 конфигурации устройства присутствует в реальной установке, но находится в слоте 4.

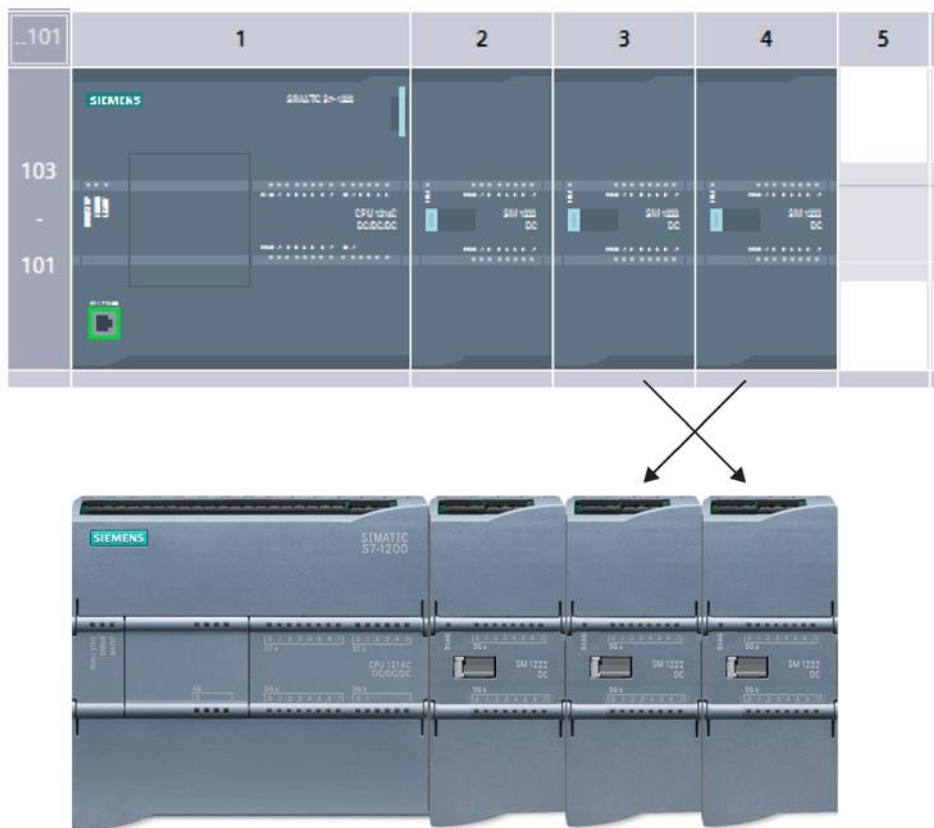


Рисунок 6-3 Конфигурация устройства в сравнении с реальной установкой, где модули в слотах 3 и 4 поменяны местами

Чтобы связать конфигурацию устройства с реальной установкой отредактируйте запись управляющих данных, чтобы привязать модули к правильным позициям слотов.

ControlDataRecord				
	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	4	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

6.5 Замена устройства

Вы можете изменить тип сконфигурированного ЦПУ или модуля. Находясь в конфигурации устройства, щелкните правой клавишей мыши по устройству и выберите "Change device" из контекстного меню. В диалоговом окне найдите и выберите ЦПУ или модуль, который Вы хотите заменить. Диалоговое окно Change device отображает информацию относительно совместимости двух устройств.

Примечание

Замена устройства: замена ЦПУ V3.0 на ЦПУ V4.1

Вы можете открыть проект STEP 7 V12 в STEP 7 V13 и заменить ЦПУ V3.0 на ЦПУ V4.1. Вы не можете заменить ЦПУ с версиями до V3.0. Когда Вы заменяете ЦПУ V3.0 на ЦПУ V4.1, примите во внимание различия (стр. 1287) в свойствах и поведении между двумя версиями и меры, которые Вам следует предпринять.

Если у Вас есть проект для ЦПУ с версией более ранней, чем V3.0, Вы должны вначале выполнить обновление до ЦПУ V3.0, а затем обновить до V4.1.

6.6 Конфигурирование работы ЦПУ

6.6.1. Обзор

Для того чтобы сконфигурировать рабочие параметры ЦПУ, выберите ЦПУ в представлении Device view (голубой контур вокруг ЦПУ), и используйте вкладку "Properties" окна инспектора.

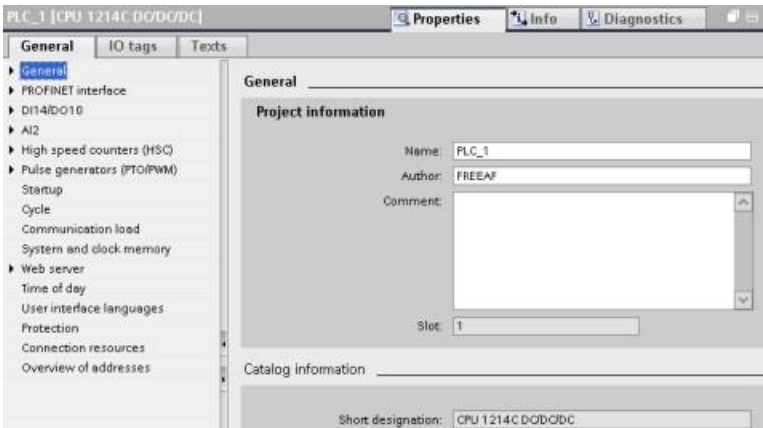


Таблица 6- 2 Свойства ЦПУ

Свойство	Описание
PROFINET interface	Устанавливает IP-адрес для ЦПУ и синхронизацию времени.
DI, DO и AI	Конфигурирует поведение локального (встроенного) цифрового и аналогового ввода-вывода (например, времена фильтра для цифрового ввода и реакция на остановку ЦПУ для цифрового вывода).
High-speed counters (стр. 457) и pulse generators (стр. 407)	Разрешает и конфигурирует высокоскоростные счетчики (HSC) и генераторы импульсов, используемые для операций с пачкой импульсов (PTO) и широтно-импульсной модуляции (PWM). Когда Вы конфигурируете выходы ЦПУ или сигнальной платы как генераторы импульсов (для использования с PWM или инструкциями управления движением), соответствующие выходные адреса удаляются из Q памяти и не могут использоваться для других целей в Вашей пользовательской программе. Если Ваша пользовательская программа пишет значение на выход, используемый в качестве генератора импульсов, ЦПУ не пишет это значение в физический выход.
Startup (стр. 83)	Startup after POWER ON: Выбирает поведение ЦПУ после цикла выключения-включения питания, например, запуск в режиме STOP или переход в режим выполнения RUN после "теплого" перезапуска. Supported hardware compatibility: Конфигурирует стратегию замены для всех компонентов системы (SM, SB, CM, CP и CPU): • Разрешает приемлемую замену • Разрешает любую замену (по умолчанию) Каждый модуль внутренне содержит требования по совместимости замены на основе количеству каналов ввода-вывода, электрической совместимости и других соответствующих факторов сравнения. Например, SM с 16 каналами мог бы стать приемлемой заменой для SM с 8 каналами, но 8-канальный SM не мог бы быть приемлемой заменой для 16-канального SM. Если Вы выбираете "Allow acceptable substitute", STEP 7 ужесточает правила замены; иначе, STEP 7 позволяет любую замену.

Свойство	Описание
	Parameter assignment time for distributed I/O: Конфигурирует максимальное количество времени (по умолчанию: 60000 мс) для распределенного ввода-вывода, запускаемого в работу. (СМ и СР получают питание и коммуникационные параметры от ЦПУ во время запуска. Это время назначения учитывает время для запуска в работу ввода-вывода, подключенного к СМ или СР.) ЦПУ переходит в RUN, как только распределенный ввод-вывод запущен в работу, независимо от времени назначения. Если распределенный ввод-вывод не был переведен в режим он-лайн в течение этого времени, ЦПУ тем не менее переходит в RUN без распределенного ввода-вывода. Примечание: Если Ваша конфигурация использует СМ 1243-5 (ведущее устройство PROFIBUS), не устанавливайте этот параметр меньше 15 секунд (15000 мс), чтобы гарантировать, что модуль может быть переведен в режим онлайн.
Cycle (стр. 101)	Определяет максимальное время цикла или фиксированное минимальное время цикла
Communication load	Выделяет процент процессорного времени, которое будет выделено коммуникационным задачам
System and clock memory (стр. 105)	Разрешает байт для функций "системной памяти" и байт для функций "тактовой памяти" (где каждый бит изменяет состояние из 0 в 1 с фиксированной частотой).
Web server (стр. 785)	Разрешает и конфигурирует функцию веб-сервера.
Time of day	Выбирает часовой пояс и конфигурирует летнее время
User interface languages	Выбирает язык для веб-сервера и дисплея ЦПУ, соответственно языку проекта. Максимум двум языкам проекта Вы можете присвоить соответствующий язык пользовательского интерфейса для дисплея ЦПУ и веб-сервера.
Protection (стр. 197)	Устанавливает защиту чтения-записи и пароли для доступа к ЦПУ.
Connection resources (стр. 613)	Обеспечивает сводку коммуникационных ресурсов, которые доступны для ЦПУ и числа ресурсов, которые были сконфигурированы.
Overview of addresses	Обеспечивает сводку адресов ввода-вывода, которые были сконфигурированы для ЦПУ.

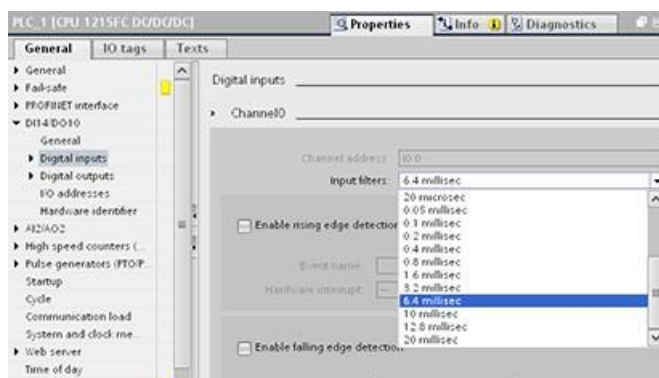
6.6.2. Конфигурирование времен фильтра для цифрового входа

Фильтры цифровых входов защищают Вашу программу от отклика на нежелательные быстрые изменения входных сигналов, что может быть вызванодребезгом контактов переключателя или электрических помех. Время фильтра по умолчанию в 6.4 мс блокирует нежелательные переходы от типовых механических контактов. Различные каналы в Вашем приложении могут потребовать более короткого времени фильтра для обнаружения и отклика на сигналы от быстрых датчиков или более длительного времени фильтра, чтобы блокировать медленный дребезг контакта или более длительный импульсный шум.

Время фильтра для входа в 6.4 мс означает, что одиночное изменение сигнала из '0' в '1' или из '1' в '0', должно продолжаться приблизительно 6.4 мс, чтобы быть обнаруженным, и одиночный импульс высокого или низкого уровня короче, чем приблизительно 6.4 мс, не обнаруживается. Если входной сигнал переключается между '0' и '1' более быстро, чем время фильтра, значение входного канала может измениться в пользовательской программе, когда накопленная продолжительность импульсов нового значения по импульсам прежнего значения превышает время фильтра.

Фильтр цифрового входа работает следующим образом:

- Когда на входе "1", он считает на увеличение, останавливаясь через время фильтра. Канал регистра образа изменяется из "0" в "1", когда количество сравнивается с временем фильтра.
- Когда на входе "0", он считает на уменьшение, останавливаясь в "0". Канал регистра образа изменяется из "1" в "0", когда количество сравнивается с "0".
- Если сигнал на входе изменяется скачкообразно, то счетчик будет считать понемногу вверх и в обратно. Регистр образа изменится, когда чистое накопление импульсов сравняется или с временем фильтра или с "0".
- Быстро изменяющийся сигнал с большим количеством "0", чем "1" в конечном счете перейдет в "0", а если будет больше "1", чем "0", то регистр образа в конечном счете изменится в "1".



Каждый входной канал имеет единственную конфигурацию фильтра, которая применяется ко всем случаям использования: технологические входы, прерывания, захват импульсов и входы высокоскоростных счетчиков. Чтобы сконфигурировать времена фильтра для входа выберите "Digital Inputs".

Время фильтра по умолчанию для цифровых входов составляет 6.4 мс. Вы можете выбрать время фильтра из выпадающего списка фильтров входов. Допустимый диапазон времен фильтра составляет от 0.1 мкс до 20.0 мс.

 ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с изменениями времени фильтра для канала цифрового ввода

Если время фильтра для канала цифрового ввода изменено по сравнению с предыдущей настройкой, новое входное значение уровня "0", возможно, должно быть удержано на максимальное время, соответствующее накопленным 20.0 мс, прежде чем фильтр полностью станет быстро реагировать на новые входные значения. В течение этого времени короткие "0" импульсы продолжительностью менее 20.0 мс не могут быть обнаружены или подсчитаны.

Это изменение времени фильтра может привести к непредвиденной работе машины или процесса, что может вызвать смерть или серьезную травму персонала и/или повреждение оборудования.

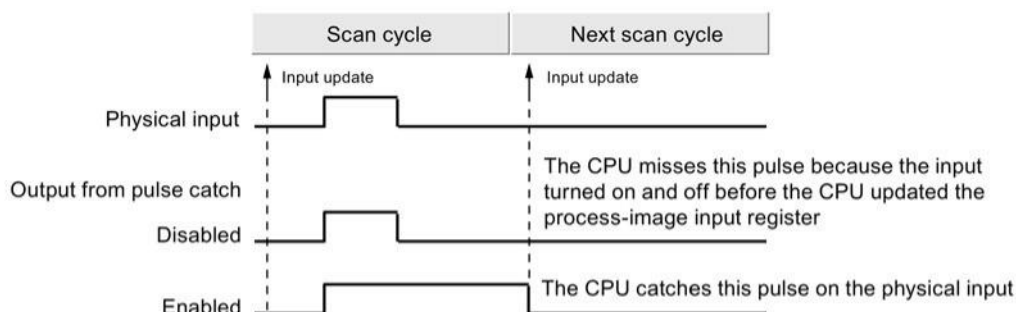
Чтобы гарантировать, что новое время фильтра сразу вступает в силу, необходимо выполнить цикл включения и выключения питания ЦПУ.

6.6.3. Захват импульсов

Центральный процессор S7-1200 обеспечивает функцию захвата импульсов для цифровых входных каналов. Функция захвата импульсов позволяет Вам захватить импульсы высокого или низкого уровней, который имеет такую короткую длительность, что они не всегда были бы замечены, когда ЦПУ читает цифровые входы в начале цикла сканирования.

Когда для входа разрешен захват импульсов, изменение состояния входа фиксируется и удерживается до следующего цикла обновления входа. Это гарантирует, что импульс, который длится в течение короткого периода времени, будет захвачен и удержан, пока ЦПУ не прочтает входы.

Рисунок ниже показывает основные операции ЦПУ S7-1200 с разрешенным захватом импульсов и без него:



Поскольку функция захвата импульсов воздействует на вход после того, как он проходит через входной фильтр, Вы должны подобрать входное время фильтра так, чтобы импульс не был отсеян фильтром. Рисунок ниже показывает блок-схему цифрового входа:

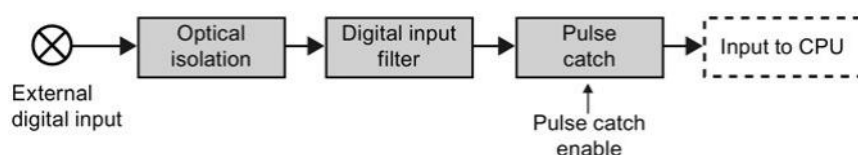
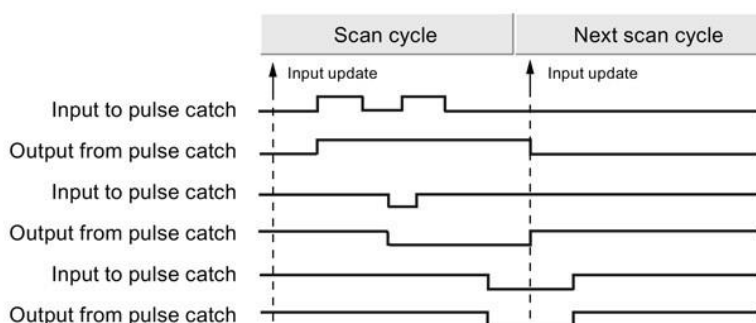


Рисунок ниже показывают отклик разрешенной функции захвата импульсов при различных условиях на входе. Если у Вас присутствует больше чем один импульс в данном скане, только первый импульс будет считан. Если у Вас присутствует несколько импульсов в данном скане, Вам следует использовать прерывания по нарастающему/спадающему фронту:



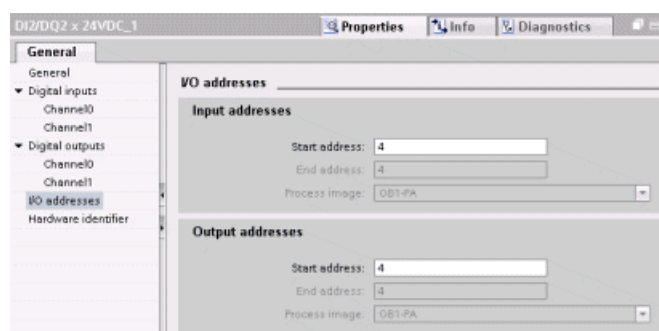
6.7 Конфигурирование параметров модулей

Чтобы сконфигурировать эксплуатационные параметры для модулей, выберите модуль в представлении Device view и используйте вкладку "Properties" окна инспектора для задания параметров модуля.

Конфигурирование сигнального модуля (SM) или сигнальной платы (SB)

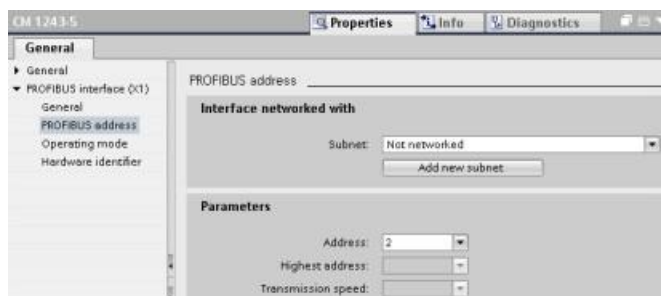
Конфигурация устройства для сигнальных модулей и сигнальных плат обеспечивает возможность по конфигурированию следующего:

- Цифровой ввод/вывод: Вы можете сконфигурировать входы для обнаружения нарастающего или спадающего фронта (каждое связано с событием и аппаратным прерыванием) или для "захвата импульса" (чтобы задержаться после мгновенного импульса) вплоть до следующего обновления образа процесса по входам. Выходы могут использовать замороженное или подстановочное значение.
- Аналоговый ввод/вывод: Для отдельных входов сконфигурируйте параметры, такие как тип измерения (напряжение или ток), диапазон и сглаживание, и разрешите диагностику переполнения или потери значимости. Аналоговые выходы обеспечивают параметры, такие как тип выхода (напряжение или ток), а также связанные с диагностикой, такие как короткое замыкание (для выходов по напряжению) или диагностика верхнего/нижнего предела. Вы не конфигурируете диапазоны аналоговых входов и выходов в инженерных единицах в диалоге Properties. Вы должны выполнить соответствующую обработку в своей программной логике, как описано в теме "Обработка аналоговых значений (стр. 115)".
- Адреса ввода/вывода: Вы конфигурируете начальный адрес для набора входов и выходов модуля. Вы можете также назначить входы и выходы разделам образа процесса (PIR0, PIR1, PIR2, PIR3, PIR4) или выбрать для них автоматическое обновление или не использовать разделение образа процесса. См. "Выполнение пользовательской программы" (стр. 79) для получения сведений об образе процесса и разделах образа процесса.



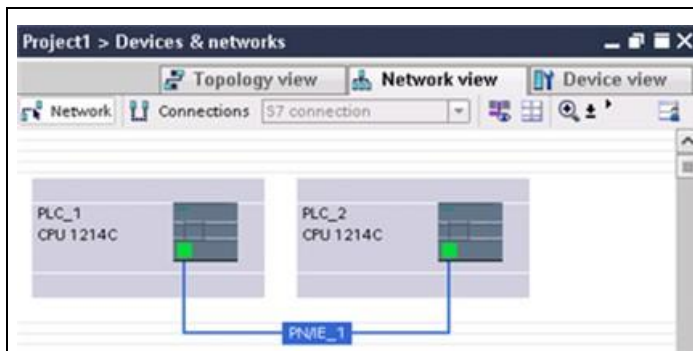
Конфигурирование коммуникационного интерфейса (СМ, СР или СВ)

В зависимости от типа коммуникационного интерфейса Вы конфигурируете параметры для сети.



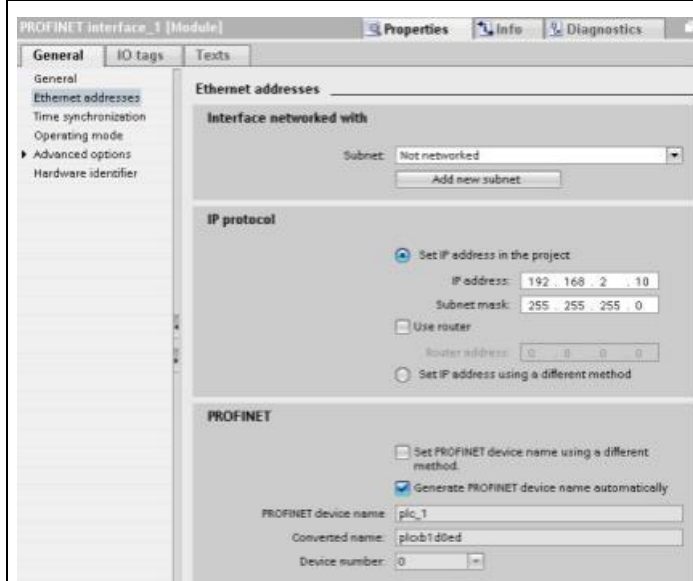
6.8 Конфигурирование ЦПУ для коммуникаций

S7-1200 разработан, чтобы удовлетворить Ваши коммуникационные и сетевые потребности, поддерживая не только самую простую из сетей, но также и более сложные сети. S7-1200 также обеспечивает инструменты, которые позволяют Вам осуществлять обмен с другими устройствами, такими как принтеры и весы, которые используют свои собственные коммуникационные протоколы.



Используйте представление "Network view" конфигурации устройства, чтобы создать сетевые соединения между устройствами в Вашем проекте. После создания сетевого соединения используйте вкладку "Properties" окна инспектора, чтобы сконфигурировать параметры сети.

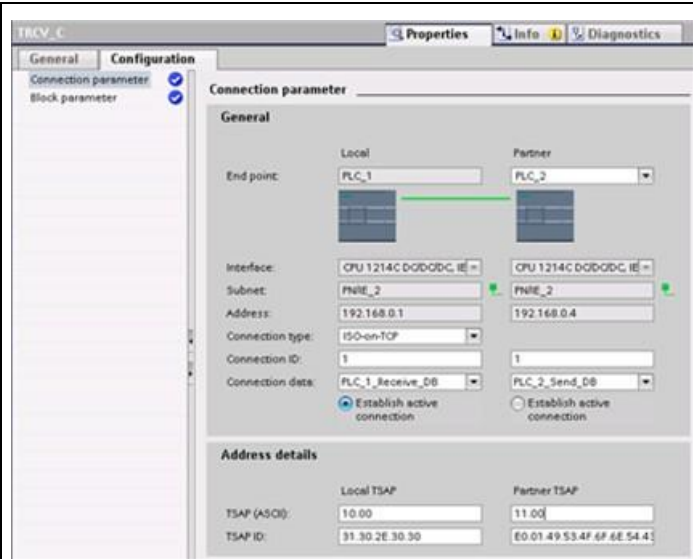
Обратитесь к разделу "Создание сетевого соединения" (стр. 618) для получения дополнительной информации.



В окне Properties выберите конфигурационную запись "Ethernet addresses". STEP 7 отображает диалог конфигурирования Ethernet адреса, который связывает проект программного обеспечения с IP-адресом ЦПУ, который принимает этот проект.

Примечание: У ЦПУ S7-1200 нет предварительно сконфигурированного IP-адреса. Вы должны вручную назначить IP-адрес для ЦПУ.

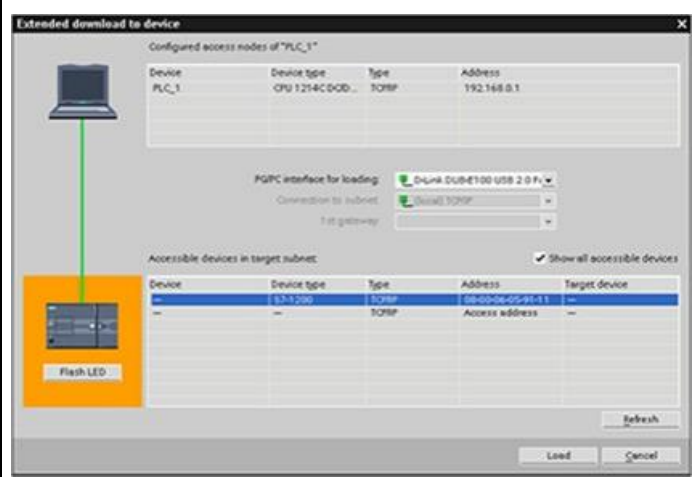
Обратитесь к разделу "Назначению адресов Интернет Протокола (IP)" (стр. 622) для получения дополнительной информации.



Для TCP, ISO-on-TCP и UDP Ethernet-протоколов, используют "свойства" инструкции (TSEND_C, TRCV_C или TCON), чтобы сконфигурировать "локального/партнерского" соединения.

Рисунок показывает "свойства соединения" вкладки "Configuration" для ISO-on-TCP соединения.

Обратитесь к разделу "Конфигурирование пути локального/партнерского соединения" (стр. 619) для получения дополнительной информации.



После завершения конфигурации загрузите проект в ЦПУ. Все IP-адреса сконфигурированы, когда Вы загружаете проект.

Обратитесь к разделу "Тестированию сети PROFINET" (стр. 629) для получения дополнительной информации.

Примечание

Чтобы установить соединение с Вашим ЦПУ, Ваша карта сетевого интерфейса (NIC) и ЦПУ должны быть в том же самом классе сети и в той же самой подсети. Вы можете либо настроить свою карту сетевого интерфейса, чтобы соответствовать IP-адресу по умолчанию ЦПУ, либо Вы можете изменить IP-адрес ЦПУ, чтобы соответствовать классу сети и подсети Вашей карты сетевого интерфейса.

Основы программирования

7.1 Рекомендации по разработке ПЛК системы

При разработке ПЛК системы Вы можете выбирать из множества методов и критериев. Следующие общие руководящие принципы могут применяться ко многим дизайн-проектам. Конечно, Вы должны следовать директивам процедур своей собственной компании и сложившейся практике на основе Вашего собственного обучения и территориального расположения.

Таблица 7- 1 7.1 Рекомендации по разработке ПЛК системы

Рекомендуемые этапы	Задачи
Разделите свой процесс или машину	Разделите свой процесс или машину на разделы, у которых есть уровень независимости друг от друга. Эти разделы определяют границы между контроллерами и влияют на спецификации функционального описания и назначение ресурсов.
Создайте функциональные спецификации	Выполните описания работы для каждого раздела процесса или машины, такие как каналы ввода-вывода, функциональное описание работы, состояния, которые должны быть достигнуты прежде, чем разрешить работу каждого исполнительного механизма (такого как соленоид, двигатель или привод), описание интерфейса оператора и любых интерфейсов
Разработайте аварийные схемы безопасности	Идентифицируйте любое оборудование, которое могло бы потребовать аппаратной логики для безопасности. Помните, что управляющие устройства могут перестать работать в небезопасном состоянии, что может произвести неожиданный запуск или изменение в работе оборудования. Там где непредсказуемая или неправильная работа оборудования могла бы привести к телесным повреждениям персонала или значительному материальному ущербу, рассмотреть реализацию электромеханических блокировок (которые работают независимо от ПЛК), чтобы предотвратить небезопасную работу. Следующие задачи следует включить в проект разработки цепей аварийной защиты: - Идентифицируйте любую ненадлежащую или неожиданную работу исполнительных механизмов, которые могли бы быть опасными. - Идентифицируйте условия, которые гарантировали бы, что работа не опасна, и определите, как обнаружить эти условия независимо от ПЛК. - Идентифицируйте, как ПЛК влияет на процесс, когда питание подается и снимается, а также идентифицируйте, как и когда обнаруживаются ошибки. Используйте эту информацию только для разработки нормальной и ожидаемой аварийной работы. Вы не должны полагаться на этот "наилучший" сценарий по соображениям безопасности. - Спроектируйте ручные или электромеханические защитные блокировки, которые блокируют опасную работу независимую от ПЛК. - Обеспечьте передачу надлежащей информации о состоянии независимых схем в ПЛК так, чтобы программа и любые интерфейсы оператора обладали необходимой информацией. - Идентифицируйте любые другие связанные с безопасностью требования для безопасной работы процесса.
Спланируйте безопасность системы	Определите, какой Вам необходим уровень защиты (стр. 197) для доступа к Вашему процессу. Вы можете при помощи пароля защитить от несанкционированного доступа ЦПУ и программные блоки.

Рекомендуемые этапы	Задачи
Определите операторские станции	На основе требований функциональных спецификаций создайте следующие чертежи операторских станций: • Обзорный чертеж, который показывает расположение каждой операторской станции относительно процесса или машины. • Схема расположения оборудования для операторских станций, таких как дисплей, коммутаторы и световые сигналы. • Электрические схемы с соответствующим вводом-выводом ПЛК и сигнальными модулями.
Создайте чертежи конфигурации	На основании требований функциональной спецификации создайте схемы расположения контрольно-измерительных приборов: • Обзорный чертеж, который показывает расположение каждого ПЛК относительно процесса или машины. • Механический чертеж каждого ПЛК и любых модулей ввода-вывода, включая любые шкафы и другое оборудование. • Электрические схемы для каждого ПЛК и любых модулей ввода-вывода, включая номера моделей устройств, коммуникационные адреса и адреса ввода-вывода.
Создайте список символьных имен	Создайте список символьных имен для абсолютных адресов. Включите в него не только физические сигналы ввода-вывода, но также и другие элементы (такие как имена тегов), чтобы использовать их в программе.

7.2 Структурирование Вашей пользовательской программы

При создании пользовательской программы для задач автоматизации, Вы вставляете инструкции в кодовые блоки:

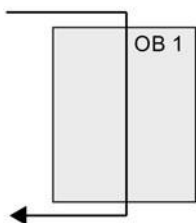
- Организационный блок (ОВ) реагирует на определенное событие в ЦПУ и может прервать выполнение пользовательской программы. По умолчанию циклическая обработка пользовательской программы (ОВ1) обеспечивает основу для Вашей пользовательской программы. Если Вы включаете в свою программу другие ОВ, то они прерывают обработку ОВ 1. Другие ОВ выполняют специфические функции, такие как обработка задач запуска, обработка прерываний и ошибок или обработка определенного программного кода с установленным временным интервалом.
- Функциональный блок (ФВ) - это подпрограмма, которая выполняется, когда вызывается из другого кодового блока (ОВ, ФВ или FC). Вызывающий блок передает параметры в ФВ, а также идентифицируют определенный блок данных (DB), который хранит данные для определенного вызова или, по-другому, экземпляра этого ФВ. Изменение DB экземпляра позволяет универсальному ФВ управлять работой ряда устройств. Например, один ФВ может управлять несколькими насосами или клапанами с различным DB экземплярами, содержащими специфические рабочие параметры для каждого насоса или клапана.
- Функция (FC) является подпрограммой, которая выполняется, когда вызывается из другого кодового блока (ОВ, ФВ или FC). У FC нет связанного DB экземпляра. Вызывающий блок передает параметры в FC. Выходные значения FC должны быть записаны в ячейку памяти или в глобальный DB.

Выбор структуры Вашей пользовательской программы

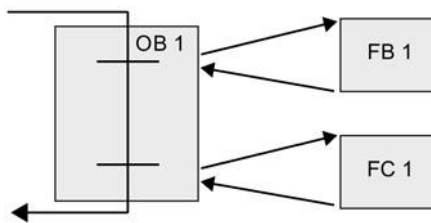
На основании требований Вашего приложения Вы можете выбрать линейную или модульную структуру для создания Вашей пользовательской программы:

- Линейная программа выполняет все инструкции для Ваших задач автоматизации последовательно, одну за другой. Как правило, линейная программа содержит все инструкции в ОВ циклической обработки (ОВ 1).
- Модульная программа вызывает определенные кодовые блоки, которые выполняют определенные задачи. Чтобы создать модульную структуру, Вы делите сложную задачу автоматизации на меньшие подчиненные задачи, которые соответствуют технологическим функциям процесса. Каждый кодовый блок обеспечивает сегмент программы для каждой подчиненной задачи. Вы структурируете свою программу, вызывая кодовый блок из другого кодового блока.

Линейная структура:



Модульная структура:



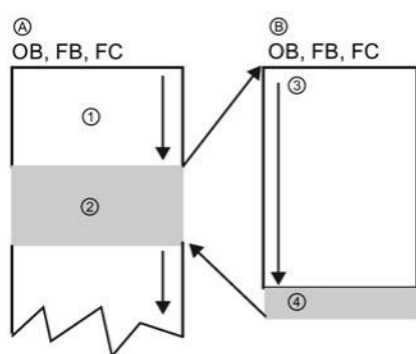
Создавая универсальные кодовые блоки, которые могут быть повторно использованы в пользовательской программе, Вы можете упростить разработку и реализацию пользовательской программы. Применение универсальных кодовых блоков обладает многими преимуществами:

- Вы можете создать допускающие повторное использование кодовые блоки для стандартных задач, например, для управления насосом или двигателем. Вы можете также сохранить эти универсальные кодовые блоки в библиотеке, которой могут пользоваться различные приложения или решения.
- Когда Вы разбиваете пользовательскую программу на модульные компоненты, которые относятся к функциональным задачам, проект Вашей программы может быть проще в понимании и управлении. Модульные компоненты не только помогают стандартизировать проектирование программы, но и могут также упростить и ускорить обновление или изменение программного кода.
- Создание модульных компонентов упрощает отладку Вашей программы. Структурируя всю программу как ряд модульных программных сегментов, Вы можете протестировать функциональность каждого кодового блока, как это запланировано.
- Создание модульных компонентов, которые относятся к определенным технологическим функциям поможет упростить и уменьшить время, связанное с вводом в эксплуатации полного приложения.

7.3 Использование блоков для структурирования Вашей программы

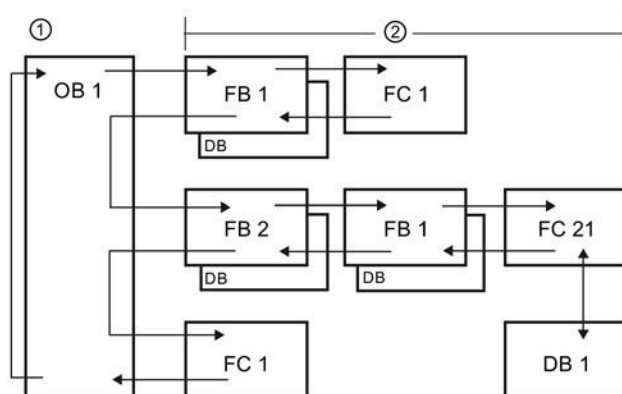
Разрабатывая FB и FC для решения универсальных задач, Вы создаете модульные кодовые блоки. Затем Вы структурируете свою программу посредством вызова другими кодовыми блоками этих допускающих повторное использование модулей. Вызывающий блок передает специфичные для устройства параметры в вызванный блок.

Когда кодовый блок вызывает другой кодовый блок, ЦПУ обрабатывает программный код в вызванном блоке. После того, как выполнение вызванного блока завершено, ЦПУ возобновляет выполнение вызывающего блока. Обработка продолжается выполнением инструкции, которая следует за вызовом блока.



- A Вызывающий блок
- B Вызываемый (или прерывающий) блок
- ① Обработка программы
- ② Инструкция или событие, которое инициирует обработку другого блока
- ③ Обработка программы
- ④ Окончание блока (возврат в вызывающий блок)

Вы можете вложить вызовы блоков для получения более модульной структуры. В следующем примере глубина вложения равняется 3: программный цикл OB плюс 3 уровня вызовов кодовых блоков.



- ① Начало цикла
- ② Глубина вложения

7.3.1. Организационный блок (OB)

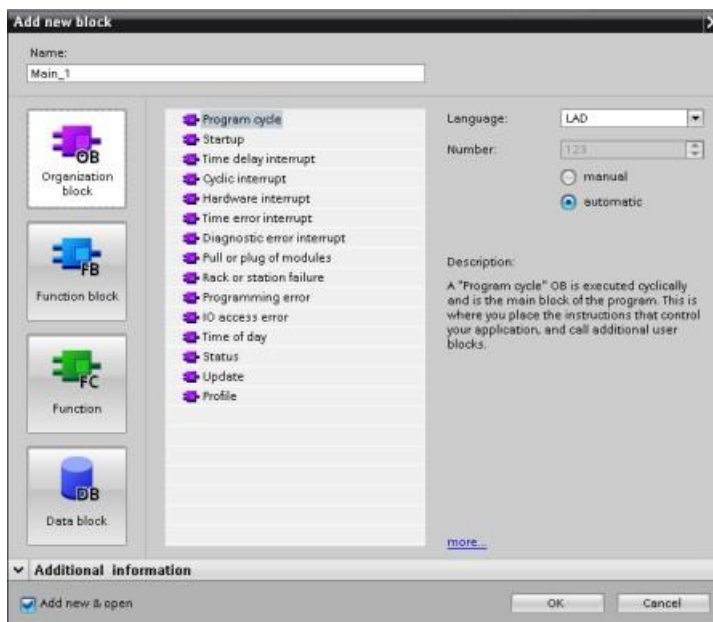
Организационные блоки обеспечивают структуру Вашей программы. Они служат интерфейсом между операционной системой и пользовательской программой. OB управляются событиями. Событие, такое как диагностическое прерывание или временной интервал, заставляет ЦПУ обрабатывать OB. Некоторые OB обладают заранее определенными стартовыми событиями и поведением.

OB программного цикла содержит Вашу основную программу. Вы можете включить более одного OB программного цикла в Вашу пользовательскую программу. В режиме RUN организационные блоки программного цикла выполняются с самым низким уровнем приоритета и могут быть прерваны всеми другими типами событий. OB запуска не прерывает OB программного цикла, потому что ЦПУ обрабатывает OB запуска до перехода в режим RUN.

После окончания обработки OB программного цикла ЦПУ сразу приступает к их повторной обработке. Эта циклическая обработка является "обычным" типом обработки, используемым в контроллерах с программируемой логикой. Для многих приложений вся пользовательская программа размещена в единственном OB программного цикла.

Вы можете создать другой OB для выполнения определенных функций, таких как обработка прерываний и ошибок, или для обработки определенного программного кода с установленным временным интервалом. Эти OB прерывают выполнение OB программного цикла.

Используйте диалоговое окно "Add new block", чтобы создать новый OB в Вашей пользовательской программе.



Обработка прерываний всегда управляется событиями. Когда такое событие имеет место, ЦПУ прерывает выполнение пользовательской программы и вызывает OB, который был сконфигурирован, чтобы обработать это событие. После окончания OB обработки прерывания ЦПУ возобновляет выполнение пользовательской программы с точки на которой она была прервана.

ЦПУ определяет порядок обработки событий прерываний по приоритету. Вы можете назначить несколько событий прерывания тому же классу приоритета. Для получения дополнительной информации обратитесь к темам по организационным блокам (стр. 88) и обработке пользовательской программы (стр. 79).

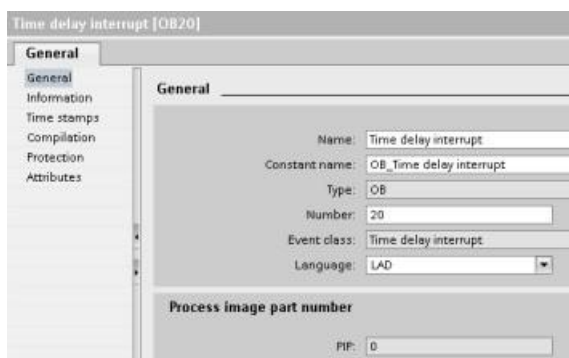
Создание дополнительных ОВ

Вы можете создать несколько ОВ для своей пользовательской программы, даже для событий программного цикла и запуска. Используйте диалоговое окно "Add new block", чтобы создать ОВ и ввести имя для него.

Если Вы создаете несколько ОВ программного цикла для Вашей пользовательской программы, то ЦПУ выполняет каждый ОВ программного цикла в числовой последовательности, начиная с ОВ программного цикла с наименьшим номером (таким как ОВ 1). Например: после завершения первого ОВ программного цикла (такого как ОВ 1) ЦПУ выполняет ОВ программного цикла со следующим по большинству номером.

Конфигурирование свойств ОВ

Вы можете изменить свойства ОВ. Например, Вы можете сконфигурировать номер ОВ или язык программирования.



Примечание

Обратите внимание на то, что Вы можете назначить номер раздела образа процесса ОВ, что соответствует PIP0, PIP1, PIP2, PIP3 или PIP4. Если Вы вводите номер для раздела образа процесса, то ЦПУ создает этот раздел образа процесса. См. тему "Выполнение пользовательской программы (стр. 79)" для получения более подробных сведений о разделах образа процесса.

7.3.2. Функция (FC)

Функция (FC) является кодовым блоком, который обычно выполняет специфическую обработку ряда входных значений. FC сохраняет результаты этой работы в ячейках памяти. Например, используйте FC, чтобы выполнить стандартные и допускающие повторное использование операции (такие, как математические вычисления) или технологические функции (такие, как индивидуальное управление с использованием битовых логических операций). FC можно также несколько раз вызывать в различных точках программы. Это повторное использование упрощает программирование часто повторяющихся задач.

У FC нет связанного блока данных экземпляра (DB). FC использует локальный стек данных для временных данных, используемых в процессе вычислений. Временные данные не сохраняются. Чтобы сохранить данные постоянно, присвойте выходное значение ячейке из глобальной памяти, такой как M память или к глобальный DB.

7.3.3. Функциональный блок (FB)

Функциональный блок (FB) - это кодовый блок, который использует экземплярный блок данных для своих параметров и статических данных. У FB есть память переменных, которая расположена в блоке данных (DB) или DB "экземпляре". DB экземпляр обеспечивает блок памяти, который связан с этим экземпляром (или вызовом) FB и хранит данные после того, как FB завершается. Вы можете связать различные DB экземпляры с различными вызовами FB. DB экземпляры позволяют Вам использовать один универсальный FB для управления многими устройствами. Вы структурируете свою программу при наличии одного кодового блока, вызывающего FB и DB экземпляр. ЦПУ в этом случае обрабатывает программный код в этом FB и сохраняет параметры и статические локальные данные блока в DB экземпляре. Когда обработка FB завершается, ЦПУ возвращается к кодовому блоку, который вызвал FB. DB экземпляр сохраняет значения для этого экземпляра FB. Эти значения доступны при последующих вызовах функционального блока или в том же или других циклах сканирования.

Повторно используемый кодовые блоки со связанной памятью

Вы обычно используете FB, чтобы управлять работой задач или устройств, которые не заканчивают работу в одном цикле сканирования. Чтобы сохранить рабочие параметры так, чтобы к ним можно было быстро получить доступ от одного до другого сканирования, у каждого FB в Вашей пользовательской программе есть один или несколько DB экземпляров. Когда Вы вызываете FB, Вы также определяете DB экземпляр, который содержит параметры и статические локальные данные блока для этого вызова или "экземпляра" FB. DB экземпляр сохраняет эти значения после того, как завершается выполнение FB.

Создавая FB для универсальных задач управления, Вы можете повторно использовать FB для многих устройств, выбирая различные DB экземпляры для различных вызовов FB.

FB сохраняет Input, Output и InOut параметры, а также Static переменные в DB экземпляре.

Вы можете также изменить и загрузить интерфейс функционального блока в режиме RUN (стр. 1092).

Назначение стартовых значений в DB экземпляре

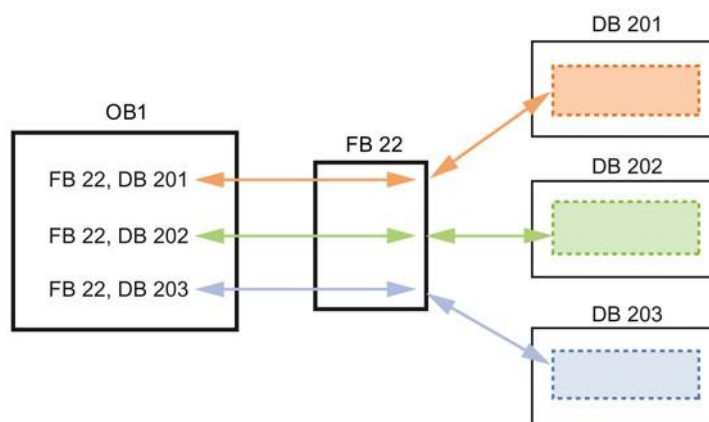
DB экземпляр хранит и значение по умолчанию и стартовое значение для каждого параметра. Стартовое значение обеспечивает значение, которое будет использоваться, когда FB выполняется. Стартовое значение может затем быть изменено во время обработки Вашей пользовательской программы.

Интерфейс FB также обеспечивает столбец "Default value", который позволяет Вам присваивать новое стартовое значение для параметра, когда Вы пишете программный код. Это значение по умолчанию в FB затем передается в стартовое значение в связанном DB экземпляре. Если Вы не присваиваете новое стартовое значение для параметра в интерфейсе FB, значение по умолчанию из DB экземпляра копируется в стартовое значение.

Использование единственного FB с несколькими DB

Следующий рисунок показывает ОВ, который вызывает один FB три раза, используя различные блоки данных для каждого вызова. Эта структура позволяет одному универсальному FB управлять несколькими подобными устройствами, такими как двигатели, назначая различные блоки данных экземпляры каждому вызову для различных устройств.

Каждый DB экземпляр сохраняет данные (такие как скорость, время разгона и общее время работы) для отдельного устройства.



В этом примере FB 22 управляет тремя отдельными устройствами, с DB 201, хранящим рабочие данные для первого устройства, DB 202, хранящим рабочие данные для второго устройства, и DB 203, хранящим рабочие данные для третьего устройства.

7.3.4. Блок данных (DB)

Вы создаете блоки данных (DB) в своей пользовательской программе, чтобы хранить данные для кодовых блоков. Все программные блоки в пользовательской программе могут получить доступ к данным в глобальном DB, но DB экземпляр хранит данные для определенного функционального блока (FB).

Данные, сохраненные в DB, не удаляются, когда выполнение связанного кодового блока завершается. Существует два типа DB:

- Глобальный DB хранит данные для кодовых блоков в Вашей программе. Любые OB, FB или FC могут получить доступ к данным в глобальном DB.
- DB экземпляр хранит данные для определенного FB. Структура данных в DB экземпляре отражает параметры (Input, Output и InOut) и статические данные для FB. (Временная память для FB не сохраняется в DB экземпляре.)

Примечание

Несмотря на то, что DB экземпляр отражает данные для определенного FB, любой кодовый блок может получить доступ к данным в DB экземпляре.

Вы можете также изменить и загрузить блоки данных в режиме RUN (стр. 1092).

Блоки данных только для чтения

Вы можете сконфигурировать DB, используемый только для чтения:

1. Щелкните правой кнопкой мыши по DB в навигаторе проекта и выберите "Properties" из контекстного меню.
2. В диалоговом окне "Properties" выберите "Attributes".
3. Выберите опцию "Data block write-protected in the device" и нажмите "OK".

Оптимизированные и стандартные блоки данных

Вы можете также сконфигурировать блок данных в качестве стандартного или оптимизированного блока данных. Стандартный DB совместим с программными инструментами STEP 7 Classic и классическими ЦПУ S7-300 и S7-400. У блоков данных с оптимизированным доступом нет фиксированной определенной структуры. Элементы данных содержат в объявлении только символьное имя и для них отсутствует фиксированный адрес в блоке. ЦПУ сохраняет элементы автоматически в доступной области памяти блока так, чтобы не было никаких разрывов в памяти. Это делается для оптимального использования объема памяти.

Чтобы установить оптимизированный доступ для блока данных, выполните следующие шаги:

1. Раскройте папку программных блоков в дереве проекта STEP 7.
2. Щелкните правой кнопкой по блоку данных и выберите "Properties" из контекстного меню.
3. В атрибутах выберите "Optimized block access".

Обратите внимание на то, что оптимизированный доступ к блоку выбран по умолчанию для новых блоков данных. Если Вы отменяете выбор "Optimized block access", то блок использует стандартный доступ.

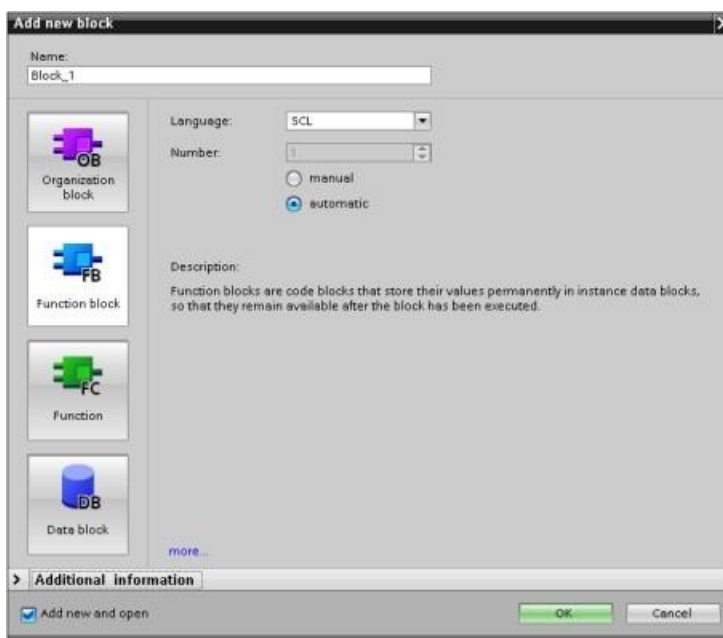
Примечание

Тип доступа к блоку для FB и его DB экземпляра

Убедитесь, что, если для Вашего FB выбрана настройка "Optimized block access", тогда настройкой для DB экземпляра этого FB также будет "Optimized block access". Точно также, если Вы не выбрали "Optimized block access" для FB, то есть доступ к FB осуществляется стандартным образом, будьте уверены, что DB экземпляр имеет также стандартный, по-другому, неоптимизированный доступ.

Если у Вас нет совместимых типов доступа к блоку, то изменения в значениях InOut параметра FB от HMI во время обработки FB могли бы быть потеряны.

7.3.5. Создание повторно используемого кодовых блоков



Используйте диалоговое окно под "Add new block" под "Program blocks" в навигаторе проекта, чтобы создать OB, FB, FC и глобальный DB.

Когда Вы создаете кодовый блок, Вы выбираете язык программирования для блока. Вы не выбираете язык для DB, потому что он только хранит данные.

Отмеченный флажок "Add new and open" (по умолчанию) открывает кодовый блок в представлении Project view.

Вы можете хранить объекты, которые Вы хотите повторно использовать, в библиотеках. Для каждого проекта есть библиотека, которая подключена к проекту. В дополнение к библиотеке проекта Вы можете создать любое количество глобальных библиотек, которыми можно пользоваться в нескольких проектах. Так как библиотеки совместимые друг с другом, элементы библиотеки могут быть скопированы и перемещены из одной библиотеки в другую.

Библиотеками пользуются, например, чтобы создать шаблоны для блоков, которые Вы сначала вставляете в библиотеку проекта, а затем продолжаете разработку. Наконец, Вы копируете блоки из библиотеки проекта в глобальную библиотеку. Вы делаете глобальную библиотеку доступной для других коллег, работающих над Вашим проектом. Они используют блоки и далее адаптируют их к своим индивидуальным требованиям, в случае необходимости.

Для получения дополнительной информации о работе с библиотеками, обратитесь к темам по библиотекам в онлайн-справке STEP 7.

7.3.6. Передача параметров в блоки

У функциональных блоков (FB) и функций (FC) есть три различных типа интерфейса:

- IN
- IN/OUT
- OUT

FB и FC получают параметры через интерфейсы типа IN и IN/OUT. Блоки обрабатывают параметры и возвращают значения вызывающему блоку через интерфейсы типа IN/OUT и OUT.

Пользовательская программа передает параметры, используя один из двух методов.

Передача по значению

Когда пользовательская программа передает параметр в функцию как "вызов по значению", пользовательская программа копирует фактическое значение параметра во входной параметр блока для интерфейса типа IN. Эта операция требует дополнительной памяти для скопированного значения.



Когда пользовательская программа вызывает блок, она копирует значения.

Передача по ссылке

Когда пользовательская программа передает параметр функции как "вызов по ссылке", пользовательская программа ссылается на адрес фактического параметра для интерфейса типа IN/OUT и не копирует значение. Эта операция не требует дополнительной памяти.



Когда пользовательская программа вызывает блок, она ссылается на адрес фактических параметров.

Примечание

Как правило тип интерфейса IN/OUT используют для структурированных тегов (например, ARRAY, STRUCT и STRING) во избежание излишнего увеличения требуемой памяти данных.

Оптимизация блока и передача параметров

Пользовательская программа передает параметры FC как "вызов по значению" для простых типов данных (например, INT, DINT и REAL). Она передает сложные типы данных (например, STRUCT, ARRAY и STRING) как "вызов по ссылке".

Пользовательская программа обычно передает параметры FB в блок данных (DB) экземпляра, связанный с FB:

- Пользовательская программа передает простые типы данных (например, INT, DINT и REAL) как "вызов по значению", копируя параметры в/из DB экземпляра.
- Пользовательская программа копирует сложные типы данных (например, STRUCT, ARRAY и STRING) в и из DB экземпляра для параметров типа IN и OUT.
- Пользовательская программа передает сложные типы данных как "вызов по ссылке" для интерфейса типа IN/OUT.

DB могут быть созданы как "оптимизированные" или как "стандартные" (неоптимизированные). Оптимизированные блоки данных более компактны, чем неоптимизированные блоки данных. Кроме того, упорядочение элементов данных в DB отличается для оптимизированного по сравнению с неоптимизированным DB. Обратитесь к разделу "Оптимизированные блоки" Руководства по S7-программированию для S7-1200/1500, STEP 7 (TIA Portal), 03/2014 (<http://support.automation.siemens.com/WW/view/en/81318674>) для получения сведений об оптимизированных блоках.

Вы создаете FB и FC для обработки оптимизированных или неоптимизированных данных. Вы можете установить флажок "Optimized block access" как один из атрибутов для блока. Пользовательская программа оптимизирует программные блоки по умолчанию, и программные блоки ожидают данные, передаваемые в блок в оптимизированном формате.

Когда пользовательская программа передает сложный параметр (например, STRUCT) в функцию, система проверяет выбор оптимизации для блока данных, содержащего структуру и выбор оптимизации программного блока. Если Вы оптимизируете и блок данных и функцию, то пользовательская программа передает STRUCT как "вызов по ссылке". Также происходит, если Вы выбираете неоптимизированный доступ и для блока данных и для функции.

Однако, если Вы выбираете оптимизацию функционального блока и блока данных отличающейся (подразумевается, что Вы оптимизировали один блок а другой нет), STRUCT должен быть преобразован в формат, ожидаемый функцией. Например, если Вы выбираете неоптимизированный доступ для блока данных и оптимизированный для функции, затем STRUCT в блоке данных должен быть преобразован в оптимизированный формат, прежде чем функция сможет обработать STRUCT. Система выполняет это преобразование, делая "копию" STRUCT и преобразовывая его в оптимизированный формат, который ожидает функция.

Таким образом, когда пользовательская программа передает сложный тип данных (например, STRUCT) в функцию как IN/OUT параметр, функция ожидает, что пользовательская программа передаст STRUCT как "вызов по ссылке":

- Если Вы выбираете оптимизированный или неоптимизированный доступ и для блока данных, содержащего STRUCT, и для функции, пользовательская программа передает данные как "вызов по ссылке".

- Если Вы не конфигурируете блок данных и функцию с теми же настройками оптимизации (один оптимизирован, а другой нет), система должна сделать копию STRUCT прежде, чем передать его в функцию. Поскольку система должна сделать эту копию структуры, она фактически преобразовывает "вызов по ссылке" в "вызов по значению".

Эффект настройки оптимизации в пользовательских программах

Копирование параметра может вызвать проблему в пользовательской программе, если HMI или ОВ прерывания изменяет элементы структуры. Например, есть IN/OUT параметр функции (обычно передаваемый как "вызов по ссылке"), но параметры настройки оптимизации блока данных и функции отличаются:

1. Когда пользовательская программа готова вызвать функцию, система должна сделать "копию" структуры, чтобы изменить формат данных, чтобы соответствовать функции.
2. Пользовательская программа вызывает функцию со ссылкой на "копию" структуры.
3. ОВ прерывания вызывается, в то время как выполняется функция, и ОВ прерывания изменяет значение в исходной структуре.
4. Функция завершается и, так как структура находится в IN/OUT параметре, система копирует значения назад в исходную структуру в исходном формате.

Эффект создания копии структуры, чтобы изменить формат состоит в том, что данные, записанные ОВ прерывания, теряются. То же может произойти при записи значения от HMI. HMI может прервать пользовательскую программу и записать значение таким же образом как ОВ прерывания.

Есть несколько способов исправить эту проблему:

- Лучшее решение этой проблемы состоит в том, чтобы выбрать идентичными настройки оптимизации программного блока и блока данных при использовании сложных типов данных (например, STRUCT). Это гарантирует, что пользовательская программа всегда передает параметры как "вызов по ссылке".
- Другое решение состоит в том, что ОВ прерывания или HMI непосредственно не изменяют элемент в структуре. ОВ или HMI могут изменить другую переменную, а затем Вы можете скопировать эту переменную в структуру в определенной точке пользовательской программы.

7.4 Понимание согласованности данных

ЦПУ поддерживает согласованность для всех элементарных типов данных (таких как Word или DWord) и всех определенных системой структур (например, IEC_TIMER или DTL).

Чтение или запись значения не могут быть прерваны. (Например, ЦПУ защищает доступ к значению DWord, пока четыре байта DWord не были считаны или записаны.), Чтобы гарантировать, что ОБ программного цикла и ОБ прерывания не смогут выполнить запись в одну и ту же ячейку памяти одновременно, ЦПУ не обрабатывает ОБ прерывания пока операции чтения или записи в ОБ программного цикла не будет завершена.

Если Ваша пользовательская программа совместно делит несколько значений в памяти между ОБ программного цикла и ОБ прерывания, то Ваша пользовательская программа должна также гарантировать, что эти значения изменяются или читаются согласованно. Вы можете использовать DIS_AIRT (запретить аварийное прерывание) и EN_AIRT (разрешить аварийное прерывание) инструкции в Вашем ОБ программного цикла, чтобы защитить любой доступ к совместно используемым значениям.

- Вставьте инструкцию DIS_AIRT в кодовый блок, чтобы гарантировать, что ОБ прерывания не может быть обработан во время операции чтения или операции записи.
- Вставьте инструкции, которые читают или записывают значения, которые могли бы быть изменены ОБ прерывания.
- Вставьте инструкцию EN_AIRT в конце последовательности, чтобы отменить DIS_AIRT и разрешить обработку ОБ прерывания.

Коммуникационный запрос от устройства HMI или другого ЦПУ может также прервать выполнение ОБ программного цикла. Коммуникационные запросы могут также вызвать проблемы с согласованностью данных. ЦПУ гарантирует, что элементарные типы данных всегда читаются и записываются инструкциями пользовательской программы согласованно. Поскольку пользовательская программа периодически прерывается коммуникациями, невозможно гарантировать, что несколько значений в ЦПУ будут все обновлены от HMI одновременно. Например, значения, выводимые на данном экране HMI, могли бы быть из различных циклов сканирования ЦПУ.

Инструкции PtP ("точка-в-точку"), инструкции PROFINET (такие как TSEND_C и TRCV_C), инструкции распределенного ввода-вывода PROFINET (стр. 350) и инструкции распределенного ввода-вывода PROFIBUS (стр. 350) передают буферы данных, которые могли бы быть прерваны. Гарантируйте согласованность для буферов данных, избегая любой операции чтения или записи в буферы как в ОБ программного цикла, так и в ОБ прерывания. Если необходимо изменить значения в буфере для этих инструкций в ОБ прерывания, используйте инструкцию DIS_AIRT, чтобы задержать любое прерывание (ОБ прерывания или коммуникационное прерывание от HMI или другого ЦПУ) до выполнения инструкции EN_AIRT.

Примечание

Использование инструкции DIS_AIRT задерживает обработку ОБ прерывания, пока не будет выполнена инструкция EN_AIRT, влияя на задержку прерывания (время от момента наступления события до момента, когда выполняется ОБ прерывание) Вашей пользовательской программы.

7.5 Язык программирования

STEP 7 предлагает следующие стандартные языки программирования для S7-1200:

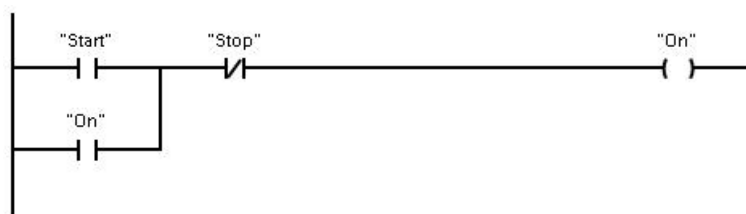
- LAD (лестничная логика) является графическим языком программирования. Представление основывается на принципиальных схемах (стр. 186).
- FBD (функционально-блочная схема) является языком программирования, который основывается на графических логических символах, используемых в Булевой алгебре (стр. 187).
- SCL (структурированный язык управления) является основанным на тексте, высокоуровневым языком программирования (стр. 188).

Когда Вы создаете кодовый блок, Вы выбираете язык программирования, который будет использоваться этим блоком.

Ваша пользовательская программа может использовать кодовые блоки, создаваемые на любом из языков программирования.

7.5.1. Лестничная логика (LAD)

Элементы принципиальной схемы, такие как нормально замкнутые, нормально разомкнутые контакты и катушки соединяются, чтобы сформировать сегменты.



Для создания логики сложных операций, Вы можете вставить ответвления, чтобы описать логику параллельных цепей. Параллельные ответвления открыты вниз или соединены непосредственно с шиной питания. Вы завершаете ответвления вверх.

LAD обеспечивает инструкцию "прямоугольник" для множества функций, таких как математика, таймер, счетчик и пересылка.

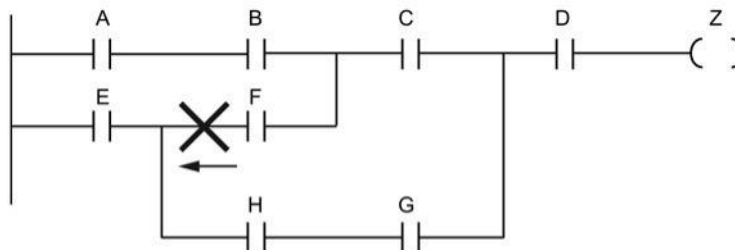
STEP 7 не ограничивает число инструкций (строки и столбцы) в сегменте LAD.

Примечание

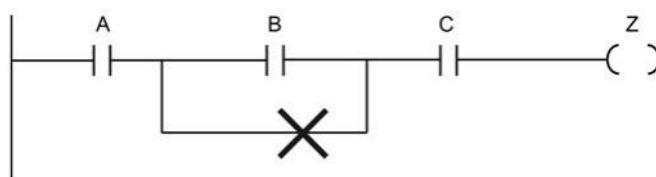
Каждый сегмент LAD должна завершиться катушкой или инструкцией прямоугольника.

Учитывайте следующие правила при создании сегмента LAD:

- Вы не можете создать ответвление, которое могло бы привести к потоку энергии в обратном направлении.

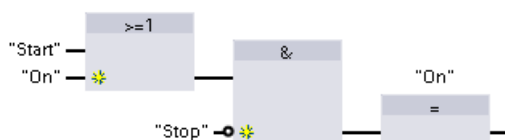


- Вы не можете создать ответвление, которое вызвало бы короткое замыкание.



7.5.2. Функционально-блочная диаграмма (FBD)

FBD также как и LAD является графическим языком программирования. Представление логики основывается на графических логических символах, используемых в Булевой алгебре.



Чтобы создать логику для сложных операций, вставьте параллельные ответвления между прямоугольниками.

Математические функции и другие сложные функции могут быть представлены непосредственно в соединении с логическими прямоугольниками.

STEP 7 не ограничивает количество инструкций (строки и столбцы) в сегменте FBD.

7.5.3. SCL

Structured Control Language (SCL) - это язык программирования высокого уровня для ЦПУ SIMATIC S7, основанный на языке PASCAL. SCL поддерживает блочную структуру STEP 7 (стр. 175). Ваш проект может включать программные блоки, созданные на любом из этих трех языков программирования: SCL, LAD и FBD.

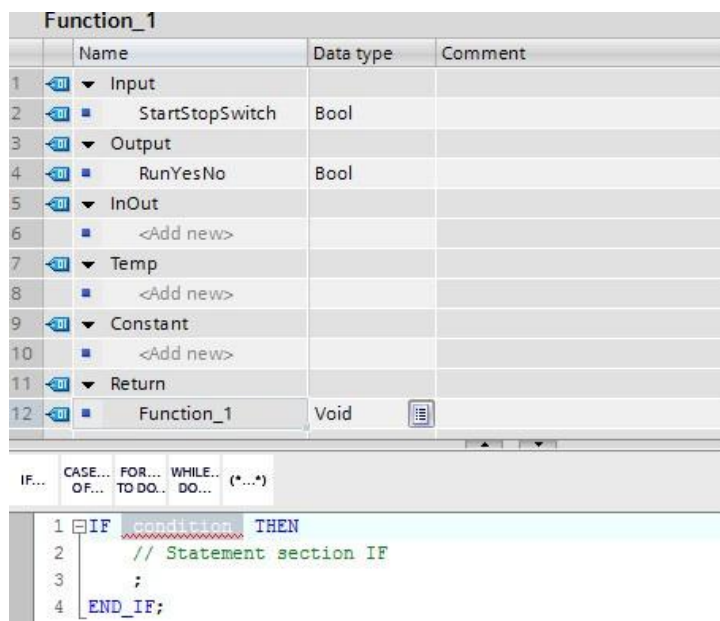
Инструкции SCL используют типовые операторы программирования, такие как присвоение (: =), математические функции (+ для сложения, - для вычитания, * для умножения, и / для деления). SCL также использует стандартные операции программного управления языка PASCAL, такие как IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO и RETURN. Вы можете использовать любую ссылку PASCAL для синтаксических элементов языка программирования SCL. Многие из других инструкций для SCL, такие как таймеры и счетчики, соответствуют LAD и FBD инструкциям. Для получения более подробной информации о специфических инструкциях обратитесь к описаниям этих инструкций в главах Базовые инструкции (стр. 209) и Расширенные инструкции (стр. 315).

7.5.3.1. Программный редактор SCL

Вы можете определять любой тип блока (OB, FB или FC) для использования с языком программирования SCL во время создания блока. STEP 7 обеспечивает программный редактор SCL, который включает в себя следующие элементы:

- Интерфейсный раздел для определения параметров кодового блока
- Раздел для программного кода
- Дерево инструкций, которое содержит инструкции SCL, поддерживаемые ЦПУ

Вы вводите SCL код своей инструкции непосредственно в разделе кода. Редактор содержит кнопки для общих конструкций кода и комментариев. Для более сложных инструкций просто перетащите инструкции SCL из дерева инструкций в Вашу программу. Вы можете также использовать любой текстовый редактор, чтобы создать программу SCL, а затем импортировать этот файл в STEP 7.



В разделе интерфейса SCL кодового блока Вы можете объявить следующие типы параметров:

- Input, Output, InOut и Ret_Val: Эти параметры определяют входные теги, выходные теги и возвращаемое значение для кодового блока. Имя тега, которое Вы вводите здесь, используется локально во время исполнения кодового блока. Вам обычно не следовало бы использовать глобальные имена тегов в таблице тегов.
- Static (только для FB; рисунок выше представляет FC): Кодовый блок использует статические теги для хранения статических промежуточных результатов в блоке данных экземпляре. Блок сохраняет статические данные, пока они не будут перезаписаны, что может произойти через несколько циклов. Имена блоков, которые этот блок вызывает как мультитекмпляры, также сохраняются в статических локальных данных.
- Temp: Эти параметры являются временными тегами, которые используются во время исполнения кодового блока.
- Constant: Они являются именованными постоянными значениями для Вашего кодового блока.

Если Вы вызываете SCL кодовый блок из другого кодового блока, то параметры SCL кодового блока появляются как входы или выходы.



В этом примере теги для "Start" и "On" (из таблицы тегов проекта) соответствуют "StartStopSwitch" и "RunYesNo" в таблице объявления SCL программы.

7.5.3.2. SCL выражения и операции

Создание SCL выражения

SCL выражение - это формула для вычисления значения. Выражение состоит из операндов и операторов (таких как *,/, + или-). Операнды могут быть тегами, константами или выражениями.

Оценка выражения происходит в определенном порядке, который определен следующими факторами:

- У каждого оператора есть заранее определенный приоритет, при этом вначале выполняется операция с самым высоким приоритетом.
- В случае равного приоритета операторы обрабатываются последовательно слева направо.
- Вы используете круглые скобки, чтобы определять последовательность операторов, которые будут вычислены вместе.

Результат выражения может использоваться или для присвоения значения тегу, используемому Вашей программой, как условие, которое будет использоваться оператором управления, как параметры для другой SCL инструкции или для вызова кодового блока.

Таблица 7- 2 Операторы в SCL

Тип	Операция	Оператор	Приоритет
Скобки	(Выражение)	(,)	1
Математические	Степень	**	2
	Знак (унарный плюс)	+	3
	Знак (унарный минус)	-	3
	Умножение	*	4
	Деление	/	4
	Модуль	MOD	4
	Сложение	+	5
	Вычитание	-	5
Сравнение	Меньше чем	<	6
	Меньше или равно чем	<=	6
	Больше чем	>	6
	Больше или равно чем	>=	6
	Равно	=	7
	Не равно	<>	7
Битовая логика	Отрицание (унарное)	NOT	3
	Логическая операция И	AND или &	8
	Логическая операция исключающее ИЛИ	XOR	9
	Логическая операция ИЛИ	OR	10
Присвоение	Присвоение	:=	11

Как высокоуровневый язык программирования, SCL использует стандартные операторы для основных задач:

- Оператор присвоения: :=
- Математические функции: +, -, * и /
- Адресация глобальных переменных (тегов): "<имя тега>" (имя тега или имя блока данных, заключенное в двойные кавычки)
- Адресация локальных переменных: #<имя тега> (имя переменной, которому предшествует символ "#"). Следующие примеры, показывают выражения для различных случаев использования.

"C" := #A+#B;	Присваивает сумму двух локальных переменных тегу
"Data_block_1".Tag := #A;	Присвоение тегу блока данных
IF #A > #B THEN "C" := #A;	Условие для IF-THEN оператора
"C" := SQRT (SQR (#A) + SQR (#B));	Параметры для инструкции SQRT

Арифметические операторы могут обработать различные типы числовых данных. Тип данных результата определяется типом наиболее значащих операндов. Например, операция умножения, которая использует операнды INT и REAL, приводит к результату с типом REAL.

Управляющие операторы

Управляющий оператор - это специализированный тип SCL выражения, которое выполняет следующие задачи:

- Ветвление программы
- Повторение разделов SCL программного кода
- Прыжок в другие части SCL программы
- Выполнение по условию

Управляющие операторы SCL включают инструкции IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO и RETURN.

Одиночный оператор обычно занимает одну строку кода. Вы можете ввести несколько операторов в одну строку, или Вы можете разбить оператор на несколько строк кода, чтобы сделать код проще для чтения. Разделители (такие как табуляция, разрывы строк и дополнительные пробелы) игнорируются при проверке синтаксиса. Оператор END завершает управляющий оператор.

Следующие примеры показывают, управляющий оператор FOR-TO-DO. (Обе формы кодирования синтаксически допустимы.)

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;  
FOR x := 0 TO max DO  
    sum := sum + value(x);  
END_FOR;
```

Управляющий оператор может также снабжаться меткой. Метка выделяется двоеточием в начале оператора:

```
Label: <Statement>;
```

Онлайн-справка STEP 7 обеспечивает полную ссылку по языку программирования SCL.

Условия

Условие - это выражение сравнения или логическое выражение, результат которого имеет тип BOOL (со значением ИСТИНА или ЛОЖЬ). Следующий пример показывает условия различных типов.

#Temperature > 50	Выражение сравнения
#Counter <= 100	
#CHAR1 < 'S'	
#Alpha <> 12) AND NOT #Beta	Сравнение и логическое выражение
5 + #Alpha	Арифметическое выражение

Условие может использовать арифметические выражения:

- Условие выражения принимает значение ИСТИНА, если результатом является какое-либо значение кроме нуля.
- Условие выражения принимает значение ЛОЖЬ, если результат равняется нулю.

Вызов других кодовых блоков из Вашей SCL программы

Чтобы вызвать другой кодовый блок в Вашей пользовательской программе, просто введите имя (или абсолютный адрес) FB или FC с параметрами. Для FB Вы должны обеспечить DB экземпляр, который вызывается с FB.

<DB name> (Parameter list)	Вызов одиночного экземпляра
<#Instance name> (Parameter list)	Вызов мульти-экземпляра
"MyDB"(MyInput:=10,MyInOut:="Tag1");	
<FC name> (Parameter list)	Стандартный вызов
<Operand>:=<FC name> (Parameter list)	Вызов из выражения
"MyFC"(MyInput:=10,MyInOut:="Tag1");	

Вы можете также перетащить блоки из навигационного дерева в редактор SCL программы и завершить присвоением параметров.

Добавление комментариев блока в SCL код

Вы можете добавить комментарий блока в свой SCL код включением текста комментария между (* и *). У Вас может быть любое число строк комментария между (* и *). Ваш программный SCL блок может содержать много комментариев. Для удобства программирования редактор SCL содержит кнопку комментария блока наряду с общими управляющими операторами:



Адресация

Также как LAD и FBD, SCL позволяет Вам использовать любые теги (символьная адресация) или абсолютные адреса в Вашей пользовательской программе. SCL также позволяет Вам использовать переменную в качестве индекса массива.

Абсолютная адресация

%I0.0	Добавляйте спереди абсолютного адреса символ "%".
%MB100	Без "%", STEP 7 генерирует ошибку неопределенного тега во время компиляции.

Символьная адресация

"PLC_Tag_1"	Тег в таблице ПЛК тегов
"Data_block_1".Tag_1	Тег в блоке данных
"Data_block_1".MyArray[#i]	Элемент массива в блоке данных

7.5.3.3. Индексная адресация с помощью PEEK и POKE инструкций

SCL предлагает PEEK и POKE инструкции, которые позволяют Вам выполнять чтение/запись из/в блоки данных, ввод-вывод или память. Вы обеспечиваете параметры для определенных байтовых или битовых смещений, необходимые для работы.

Примечание

Чтобы использовать PEEK и POKE инструкции с блоками данных, Вы должны использовать стандартные (не оптимизированные) блоки данных. Также обратите внимание на то, что PEEK и POKE инструкции, просто передают данные. Они не знают о типах данных по указанным адресам.

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

Читает байт, на который указывает byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример ссылки на блок данных:

```
%MB100 := PEEK(area:=16#84,
               dbNumber:=1, byteOffset:=#i);
```

Пример ссылки на байт входов IB3:

```
%MB100 := PEEK(area:=16#81,
               dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Читает слово, на которое указывает byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MW200 := PEEK_WORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

Читает двойное слово, на которое указывает byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MD300 := PEEK_DWORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

Читает булево значение, на которое указывают bitOffset и byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                     dbNumber:=1, byteOffset:=#ii,
                     bitOffset:=#j);
```

```
POKE(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

Записывает значение (Byte, Word или DWord) в указанное byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример ссылки на блок данных:

```
POKE(area:=16#84, dbNumber:=2,
      byteOffset:=3, value:="Tag_1");
```

Пример ссылки на байт выходов QB3:

```
POKE(area:=16#82, dbNumber:=0,
      byteOffset:=3, value:="Tag_1");
```

Записывает булево значение в указанные bitOffset и byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
            byteOffset:=3, bitOffset:=5, value:=0);
```

Записывает заданное в "count" число байтов, начиная с указанного байтового смещения, указанного исходного блока данных, ввода-вывода или области памяти в указанный byteOffset целевого блока данных, ввода-вывода или области памяти

Пример:

```
POKE_BLK(area_src:=16#84,
          dbNumber_src:=#src_db, byteOffset_src:=#src_byte,
          area_dest:=16#84,
          dbNumber_dest:=#src_db, byteOffset_dest:=#src_byte,
          count:=10);
```

Для инструкций PEEK и POKE применимы следующие значения для параметров "area", "area_src" и "area_dest". Для областей отличных от блоков данных параметр dbNumber должен быть всегда равен 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

7.5.3.4. EN и ENO для LAD, FBD и SCL

Определение "потока энергии" (EN и ENO) для инструкции

Определенные инструкции (такие как математические инструкции и инструкции пере-сылки) обеспечивают параметры для EN и ENO. Эти параметры касаются потока энергии в LAD или FBD и определяют, будет ли выполняться инструкция во время этого ска-на. SCL также позволяет Вам устанавливать параметр ENO для кодового блока.

- EN (Разрешить вход) является булевым входом. Поток энергии (EN = 1) должен присутствовать на этом входе прямоугольника инструкции, которая будет выпол-няться. Если вход EN прямоугольника LAD будет соединен непосредственно с ле-вой шиной питания, то инструкция будет всегда выполняться.
- ENO (Разрешить выход) является булевым выходом. Если у прямоугольника есть по-ток энергии на входе EN, и прямоугольник выполняет свою функцию без ошибок, то выхода ENO передает поток энергии (ENO = 1) следующему элементу. Если при вы-полнении прямоугольника инструкции обнаружена ошибка, то поток энергии преры-вается (ENO = 0) на прямоугольнике инструкции, который сгенерировал ошибку.

Таблица 7- 3 Операнды для EN и ENO

Программный редактор	Входы/выходы	Операнды	Тип данных
LAD	EN, ENO	Поток энергии	Bool
FBD	EN	I, I:P, Q, M, DB, Temp, Поток энергии	Bool
	ENO	Поток энергии	Bool
SCL	EN ¹	TRUE, FALSE	Bool
	ENO ²	TRUE, FALSE	Bool

¹ Использование EN допустимо только для FB.

² Использование ENO с SCL кодовым блоком является необязательным. Вы должны сконфи-гурировать компилятор SCL, чтобы устанавливать ENO, когда кодовый блок завершается.

Конфигурирование SCL для установки ENO

Для того чтобы сконфигурировать SCL компилятор для установки ENO, выполните следующие шаги:

1. Выберите команду "Settings" из меню "Options".
2. Разверните свойства "PLC programming" и выберите "SCL (Structured Control Language)".
3. Выберите опцию "Set ENO automatically".

Использование ENO в программном коде

Вы можете также использовать ENO в своем программном коде, например, присваивая ENO ПЛК тегу, или оценивая ENO в локальном блоке.

Примеры:

```
"MyFunction"
( IN1 :=
  ... , IN2
  := ... ,
  OUT1 => #myOut,
  ENO => #statusFlag ); // ПЛК тег statusFlag содержит значение
  ENO

"MyFunction"
( IN1 := ...
  IN2 := ... ,
  OUT1 => #myOut,
  ENO => ENO ); // флаг состояния блока "MyFunction"
               // сохраняется в локальном блоке

IF ENO = TRUE THEN
  // выполнять код только, если MyFunction возвращает ENO истинным
```

Влияние параметров Ret_Val или Status на ENO

Некоторые инструкции, такие как коммуникационные инструкции или инструкции преобразования строк, обеспечивают выходной параметр, который содержит информацию о работе инструкции. Например, некоторые инструкции обеспечивают Ret_Val (возвращаемое значение) параметр, который обычно имеет тип данных Int и содержит информацию о статусе в диапазоне от -32768 до +32767. Другие инструкции обеспечивают параметр Status, который обычно имеет тип данных Word и хранит информацию о статусе в диапазоне шестнадцатеричных значений от 16#0000 до 16#FFFF. Численное значение, сохраненное в параметре Ret_Val или Status, определяет состояние ENO для этой инструкции.

- Ret_Val: значение от 0 до 32767 обычно устанавливает ENO = 1 (или ИСТИНА). Значение от -32768 до -1 обычно устанавливает ENO = 0 (или ЛОЖЬ). Чтобы оценить Ret_Val, измените представление на шестнадцатеричное.
- Status: значение от 16#0000 до 16#7FFF обычно устанавливает ENO = 1 (или ИСТИНА). Значение от 16#8000 до 16#FFFF обычно устанавливает ENO = 0 (или ЛОЖЬ).

Инструкции, которые требуют для их выполнения больше чем один скан, часто обеспечивают параметр Busy (Bool), чтобы сигнализировать, что инструкция активна, но не завершила выполнение. Эти инструкции часто также обеспечивают параметры Done (Bool) и Error (Bool). Done сигнализирует, что инструкция была завершена без ошибки, а Error, что инструкция была завершена в состоянии ошибки.

- Когда Busy = 1 (или TRUE), ENO = 1 (или TRUE).
- Когда Done = 1 (или TRUE), ENO = 1 (или TRUE).
- Когда Error = 1 (или TRUE), ENO = 0 (или FALSE).

Смотри также

Инструкции OK (Проверить правильность) и NOT_OK (Проверить неправильность) (стр. 234)

7.6 Защита

7.6.1. Защита доступа для ЦПУ

ЦПУ обеспечивает четыре уровня безопасности для ограничения доступа к определенным функциям. Когда Вы конфигурируете уровень безопасности и пароль для ЦПУ, Вы ограничиваете функции и области памяти, к которым можно получить доступ, не вводя пароль.

Каждый уровень позволяет определенным функциям быть доступными без пароля. Состояние по умолчанию для ЦПУ состоит в отсутствии каких-либо ограничений и защиты паролем. Чтобы ограничить доступ к ЦПУ, Вы конфигурируете свойства ЦПУ и вводите пароль.

Ввод пароля по сети не ставит под угрозу защиту паролем для ЦПУ. Защита паролем не применяется к выполнению инструкций программы пользователя, включая коммуникационные функции. Ввод корректного пароля обеспечивает доступ ко всем функциям на этом уровне.

Коммуникации ПЛК-ПЛК (использующие коммуникационные инструкции в кодовых блоках) не ограничены уровнем безопасности в ЦПУ.

Таблица 7- 4 Уровни безопасности для ЦПУ

Уровень безопасности	Ограничения доступа
Полный доступ (без защиты)	Предоставляет полный доступ без защиты паролем.
Доступ на чтение	Предоставляет доступ HMI и всем видам ПЛК-ПЛК коммуникаций без защиты паролем. Пароль требуется для изменения (записи в) ЦПУ и для смены режима ЦПУ (RUN/STOP).
Доступ от HMI	Предоставляет доступ HMI и всем видам ПЛК-ПЛК коммуникаций без защиты паролем. Пароль требуется для чтения данных в ЦПУ, для изменения (записи в) ЦПУ и для смены режима ЦПУ (RUN/STOP).
Без доступа (полная защита)	Не позволяет доступ без защиты паролем. Пароль требуется для доступа HMI, чтения данных в ЦПУ и для изменения (записи в) ЦПУ.

Пароли чувствительны к регистру. Чтобы сконфигурировать уровень защиты и пароли, выполните следующие действия:

1. В "Device configuration", выберите ЦПУ.
2. В окне инспектора выберите вкладку "Properties".
3. Выберите свойство "Protection", чтобы установить уровень защиты и ввести пароли.

Protection

Select the access level for the PLC.

Access level	Access			Access permission	
	HMI	Read	Write	Password	Confirmation
☐ Full access (no protection)	✓	✓	✓	*****	*****
☐ Read access	✓	✓		*****	*****
☒ HMI access	✓				
☐ No access (complete protection)					

Когда Вы загружаете эту конфигурацию в ЦПУ, пользователь обладает доступом от HMI и может получить доступ к функциям HMI без пароля. Чтобы считать данные, пользователь должен ввести сконфигурированный пароль для "Read access" или пароль для "Full access (no protection)". Чтобы записать данные, пользователь должен ввести сконфигурированный пароль для "Full access (no protection)".

ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к защищенному ЦПУ

У пользователей с полномочиями полного доступа к ЦПУ есть право читать и записывать переменные ПЛК. Независимо от уровня доступа для ЦПУ у пользователей веб-сервера могут быть полномочия читать и записывать переменные ПЛК. Несанкционированный доступ к ЦПУ или изменение переменных ПЛК на недопустимые значения могло бы нарушить работу процесса и привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Авторизованные пользователи могут выполнять изменение рабочего режима, запись в данные ПЛК и обновления встроенного ПО. Сименс рекомендует, чтобы Вы соблюдали следующие меры безопасности:

- Защитить сильными паролями уровни доступа к ЦПУ и идентификаторы пользователей веб-сервера (стр. 789). Сильные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.
- Разрешите доступ к веб-серверу только по протоколу HTTPS.
- Не расширяйте минимальные по умолчанию полномочия пользователя веб-сервера "Everybody".
- Выполните проверку ошибок и проверку диапазонов Ваших переменных в Вашей программной логике, так как пользователи Веб-страницы могут изменить переменные ПЛК на недопустимые значения.

Механизм подключения

Чтобы получить доступ к удаленным партнерам по коммуникациям с помощью инструкции PUT/GET, у пользователя должно также быть разрешение.

По умолчанию опция "Permit access with PUT/GET communication" не разрешена. В этом случае доступ на чтение и доступ на запись к данным ЦПУ возможен только для коммуникационных соединений, которые требуют конфигурации или программирования как для локального ЦПУ, так и для коммуникационного партнера. Доступ, например, с помощью инструкции BSEND/BRCV возможен.

Соединения, для которых локальный ЦПУ является только сервером (подразумевается, что никакого конфигурирования/программирования обмена с коммуникационным партнером не существует в локальном ЦПУ), таким образом невозможны во время работы ЦПУ, например:

- PUT/GET, FETCH/WRITE или FTP доступ через коммуникационные модули
- PUT/GET доступ от других S7 ЦПУ
- HMI доступ через PUT/GET коммуникации

Если Вы хотите предоставить доступ к данным ЦПУ со стороны клиента, т.е. Вы не хотите ограничивать коммуникационные службы ЦПУ, выполнять следующие действия:

1. Сконфигурируйте любой, кроме "No access (complete protection)" защитный уровень доступа.
2. Установите флажок "Permit access with PUT/GET communication".



Когда Вы загружаете эту конфигурацию, ЦПУ разрешает PUT/GET коммуникации от удаленных партнеров

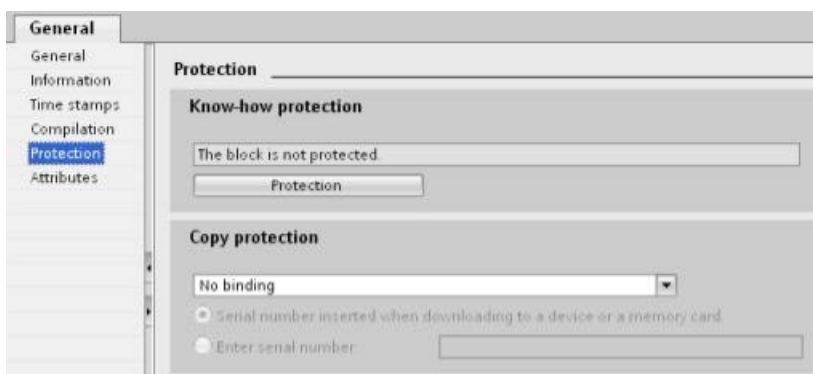
7.6.2. Защита "ноу-хау"

Защита "ноу-хау" позволяет Вам предотвратить несанкционированный доступ к одному или нескольким кодовым блокам (OB, FB, FC или DB) в Вашей программе. Вы создаете пароль, чтобы ограничить доступ к кодовому блоку. Защита паролем предотвращает несанкционированное чтение или изменение кодового блока. Без пароля Вы можете только читать следующую информация о кодовом блоке:

- Заголовок блока, комментарий блока и свойства блока
- Параметры передачи (IN, OUT, IN_OUT, Return)
- Структура вызова программы
- Глобальные теги в перекрестных ссылках (без информации о точке использования), но локальные теги скрыты

Когда Вы конфигурируете блок для защиты "ноу-хау", к коду внутри блока нельзя получить доступ кроме как после ввода пароля.

Используйте карту задач "Properties" кодового блока, чтобы сконфигурировать защиту ноу-хау для этого блока. После открытия кодового блока выберите "Protection" в свойствах.



1. В свойствах для кодового блока нажмите кнопку "Protection", чтобы вывести на экран диалоговое окно "Know-how protection".
2. Нажмите кнопку "Define", чтобы ввести пароль.



После ввода и подтверждения пароля, нажмите "OK".



7.6.3. Защита от копирования

Дополнительное средство защиты позволяет Вам связывать использование программных блоков с определенной картой памяти или ЦПУ. Эта функция особенно полезна для защиты Вашей интеллектуальной собственности. Когда Вы связываете программный блок с определенным устройством, то Вы ограничиваете использование программы или кодового блока только определенной картой памяти или ЦПУ. Эта функция позволяет Вам распространять программу или кодовый блок электронно (по Интернету или по электронной почте) или отправляя картридж памяти. Защита от копирования доступна для ОВ (стр. 176), FB (стр. 178) и FC (стр. 178). ЦПУ S7-1200 поддерживает три типа защиты блока:

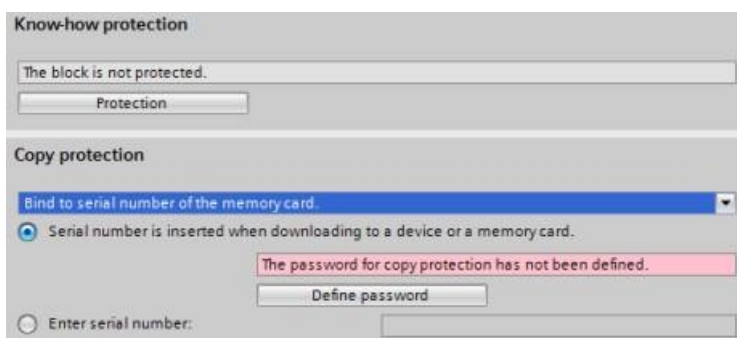
- Привязка к серийному номеру ЦПУ
- Привязка к серийному номеру карты памяти
- Динамическая привязка с обязательным пароле

Используйте карту задач "Properties" кодового блока, чтобы связать блок с определенным ЦПУ или картой памяти.

1. После открытия кодового блока выберите "Protection".



2. Из выпадающего списка под задачей "Copy protection" выберите тип защиты от копирования, который Вы хотите использовать.



3. Для привязки к серийному номеру ЦПУ или карты памяти, выберите либо вставить порядковый номер при загрузке либо ввести порядковый номер для карты памяти или ЦПУ.

Примечание

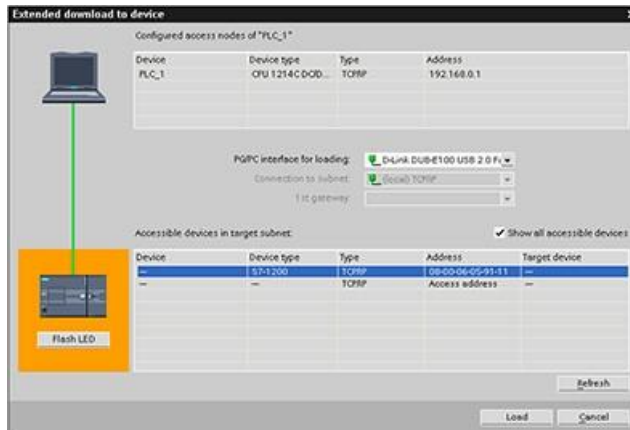
Серийный номер чувствителен к регистру.

Для динамической привязки с обязательным паролем определите пароль, который Вы должны использовать для загрузки или копирования блока.

Когда Вы впоследствии загружаете (стр. 203) блок с динамической привязкой, Вы должны ввести пароль, чтобы быть в состоянии загрузить блок. Обратите внимание на то, что пароль защиты от копирования и пароль защиты ноу-хау (стр. 200) являются двумя отдельными паролями.

7.7 Загрузка элементов Вашей программы

Вы можете загрузить элементы своего проекта из устройства программирования в ЦПУ. Когда Вы загружаете проект, ЦПУ сохраняет пользовательскую программу (OB, FC, FB и DB) в долговременной памяти.



Вы можете загрузить свой проект из устройства программирования в Ваш ЦПУ в любом из следующих расположений:

- "Дерево проекта": Щелкните правой кнопкой по элементу программы, а затем щелкните по контекстно-зависимому элементу "Download".
- Меню "Online": Щелкните левой кнопкой по элементу "Download to device".
- Панель инструментов: Щелкните левой кнопкой по символу "Download to device".

Обратите внимание на то, что, если Вы использовали динамическую привязку с обязательным паролем (стр. 201) к какому-либо из программных блоков, то Вы должны ввести пароль для защищенных блоков, чтобы загрузить их. Если Вы сконфигурировали этот тип защиты от копирования для нескольких блоков, Вы должны ввести пароль для каждого из защищенных блоков, чтобы загрузить их.

Примечание

Загрузка программы не очищает или вносит какие-либо изменения в существующие значения в сохраняемой памяти. Если Вы хотите очистить сохраняемую память перед загрузкой, то выполните сброс Вашего ЦПУ к заводским настройкам до загрузки программы.

Вы можете также загрузить проект панели Basic HMI (стр. 30) из TIA Portal на карту памяти в ЦПУ S7-1200.

7.8 Выгрузка из он-лайн ЦПУ

Вы можете также скопировать программные блоки из он-лайн ЦПУ или карты памяти, связанных с Вашим устройством программирования.

Подготовьте офф-лайн проект для копируемых программных блоков:

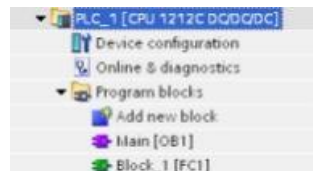
1. Добавьте устройство ЦПУ, которое соответствует он-лайн ЦПУ.
2. Раскройте узел ЦПУ так, чтобы была видна папка "Program blocks".



Чтобы выгрузить программные блоки из он-лайн ЦПУ в  офф-лайн проект, выполните следующие действия:

1. Щелкните левой клавишей по папке "Program blocks" в офф-лайн проекте. 
2. Щелкните левой клавишей по кнопке "Go online".
3. Щелкните левой клавишей по кнопке "Upload".
4. Подтвердите свое решение в диалоговом окне выгрузки (стр. 1069).

Когда выгрузка завершена, STEP 7 отображает все выгруженные программные блоки в проекте.



7.8.1. Сравнение он-лайн ЦПУ и офф-лайн ЦПУ

Вы можете использовать редактор "Compare" (стр. 1077) в STEP 7, чтобы найти различия между он-лайн и офф-лайн проектами. Это могло бы оказаться полезным для Вас до выгрузки из ЦПУ.

7.9 Отладка и тестирование программы

7.9.1. Наблюдение и изменение данных в ЦПУ

Как показано в следующей таблице, Вы можете наблюдать и изменять значения в онлайн ЦПУ.

Таблица 7- 5 Наблюдение и изменение данных с помощью STEP 7

Редактор	Наблюдение	Изменение	Форсирование
Таблица наблюдения	Да	Да	Нет
Таблица форсирования	Да	Нет	Да
Программный редактор	Да	Да	Нет
Таблица тегов	Да	Нет	Нет
Редактор DB	Да	Нет	Нет



	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I0.0	Bool	☐ FALSE	
2	"Off"	%I0.1	Bool	☐ FALSE	
3	"Run"	%Q0.0	Bool	☐ FALSE	

Наблюдение с помощью таблицы



Наблюдение с помощью редактора LAD

Обратитесь к главе "Online and diagnostics" для получения дополнительной информации о наблюдении и изменении данных в ЦПУ (стр. 1078).

7.9.2. Таблицы наблюдения и таблицы форсирования

Вы используете "таблицы наблюдения" для наблюдения и изменения значений в пользовательской программе, обрабатываемой он-лайн ЦПУ. Вы можете создать и сохранить различные таблицы наблюдения в своем проекте, чтобы поддерживать разнообразие тестовых сред. Это позволяет Вам воспроизводить тесты во время ввода в эксплуатацию или с целью ремонта и обслуживания.

С таблицей наблюдения Вы можете контролировать и взаимодействовать с ЦПУ, поскольку он выполняет пользовательскую программу. Вы можете отобразить или изменить значения не только для тегов кодовых блоков и блоков данных, но также и для областей памяти ЦПУ, включая входы и выходы (I и Q), периферийные входы (I:P), битовую память (M), и блоки данных (DB).

С таблицей наблюдения Вы можете разрешить физические выходы (Q:P) ЦПУ в режиме STOP. Например, Вы можете присвоить определенные значения выходам при тестировании проводного монтажа ЦПУ.

STEP 7 также обеспечивает таблицу форсирования для "принудительной" установки тега в определенное значение. Для получения большей информации о форсировании смотрите раздел по форсированию значений в ЦПУ (стр. 1085) в главе "Он-лайн и диагностика".

Примечание

Форсированные значения сохраняются в ЦПУ, а не в таблице наблюдения.

Вы не можете форсировать вход (или "I" адрес). Однако Вы можете форсировать периферийный вход. Чтобы форсировать периферийный вход, добавьте ":P" к адресу (например: "Op:P").

STEP 7 также обеспечивает возможность трассировки и записи переменных программы на основании условий запуска (стр. 1097).

7.9.3. Перекрестная ссылка, чтобы показать использование

Окно инспектора отображает информацию по перекрестным ссылкам, т.е. как выбранный объект используется во всем проекте: пользовательской программе, ЦПУ и любых устройствах HMI. Вкладка "Cross-reference" отображает экземпляры, где используется выбранный объект и другие объекты используют его. Окно инспектора также включает блоки, которые доступны только он-лайн в перекрестных ссылках. Чтобы отобразить перекрестные ссылки, выберите команду "Show cross-references". (В представлении Project view найдите перекрестные ссылки в меню "Tools".)

Примечание

Вы не должны закрывать редактор, чтобы увидеть информацию по перекрестным ссылкам.

Вы можете сортировать записи в перекрестных ссылках. Список перекрестных ссылок обеспечивает обзор использования адресов памяти и тегов в пользовательской программе.

- При создании и изменении программы, Вы сохраняете обзор операндов, тегов и вызовов блоков, которые Вы использовали.
- От перекрестных ссылок Вы можете перейти непосредственно к месту использования операндов и тегов.
- Во время теста программы или при поиске и устранении неисправностей, Вы получаете уведомление о том, какая ячейка памяти обрабатывается, какой командой, в каком блоке, какой используется тег, на каком экране и какой из блоков вызывается каким-либо другим блоком.

Таблица 7- 6 Элементы поперечных ссылок

Колонка	Описание
Object	Имя объекта, который использует низкоуровневые объекты или используется низкоуровневыми объектами
Number	Количество использований
Point of use	Каждое расположение использования, например, сеть
Property	Специальные свойства объектов, на которые ссылаются, например, имена тегов в объявлениях мульти-экземпляров
as	Показывает дополнительную информацию об объекте, такую как, используется ли DB экземпляр в качестве шаблона или в качестве мульти-экземпляра
Access	Тип доступа, является ли доступ к операнду доступом на чтение (R) и/или доступом на запись (W)
Address	Адрес операнда
Type	Информация о типе и языке, использованном для создания объекта
Path	Путь к объекту в дереве проекта

В зависимости от установленных продуктов таблица перекрестных ссылок выводит на экран дополнительные или отличные столбцы.

7.9.4. Структура вызовов, чтобы исследовать иерархию вызовов

Структура вызовов описывает иерархию вызовов блоков в Вашей пользовательской программе. Она обеспечивает обзор использованных блоков, вызовы других блоков, отношения между блоками, требования к данным для каждого блока и состояние блоков. Вы можете открыть программный редактор и отредактировать блоки из структуры вызова.

Отображение структуры вызова предоставляет Вам список блоков, используемых в пользовательской программе. STEP 7 выделяет первый уровень структуры вызова и отображает любые блоки, которые не вызывает никакой другой блок в программе. Первый уровень структуры вызова отображает ОВ и любые FC, FB и DB, которые не вызывает ОВ. Если кодовый блок вызывает другой блок, то вызванный блок отображается как отступ под вызывающим блоком. Структура вызовов отображает только те блоки, вызываемые кодовым блоком.

Вы можете выборочно показать только блоки, вызывающие конфликты в структуре вызовов. Следующие условия вызывают конфликты:

- Блоки, которые выполняют любые вызовы с более старыми или более новыми метками времени кода
- Блоки, которые вызывают блок с измененным интерфейсом
- Блоки, которые используют тег с измененным типом адреса и/или типом данных
- Блоки, которые не вызываются ОВ ни непосредственно, ни косвенно
- Блоки, которые вызывают несуществующий или недостающий блок

Вы можете сгруппировать несколько вызовов блоков и блоков данных как группу. Вы используете выпадающий список, чтобы видеть ссылки на различные расположения вызова.

Вы можете также выполнить проверку согласованности, чтобы показать конфликты по метки времени. При изменении метки времени блока во время или после того, как сгенерирована программа, может привести к конфликтам метки времени, которые поочередно вызывают несогласованности среди блоков, которые вызывают и вызываются.

- Большая часть конфликтов метки времени и интерфейса могут быть исправлены, путем перекомпиляции кодовых блоков.
- Если компиляции не удастся разрешить несоответствия, используйте ссылку в столбце "Details", чтобы перейти к источнику проблемы в программном редакторе. Вы тогда можете вручную устранить любые несогласованности.
- Должны быть перекомпилированы любые блоки, отмеченные красным цветом.

Базовые инструкции

8.1 Операции бинарной логики

8.1.1. Инструкции бинарной логики

Языки LAD и FBD очень удобны при работе с двоичной логикой. При этом SCL, который в свою очередь имеет ряд преимуществ при работе со сложными математическими выражениями и структурами управления, может быть также использован для работы с двоичной логикой.

Язык контактного плана LAD

Таблица 8- 1 Нормально открытые и нормально закрытые контакты

LAD	SCL	Описание
"IN" 	<pre>IF in THEN Statement; ELSE Statement; END_IF;</pre>	Нормально открытые и нормально закрытые контакты: Вы можете соединить одни контакты с другими, при этом построив свою логику программы. Если входной бит использует идентификатор памяти I (вход) или Q (выход), то значение бита может быть прочитано из регистра области отображения. Физический контакт в Вашем процессе подключен к I клеммам ПЛК. ЦПУ сканирует подключенные входные сигналы и непрерывно обновляет соответствующие состояния в области отображения.
"IN" 	<pre>IF NOT (in) THEN Statement; ELSE Statement; END_IF;</pre>	Вы можете прочитать физический вход при помощи ":P" после соответствующего смещения (например: "%I3.4:P"). При непосредственном чтении битовые значения считываются непосредственно из физического входа, а не из образа процесса. Непосредственное чтение не обновляет образ процесса.

Таблица 8- 2 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Bool	Назначенный бит

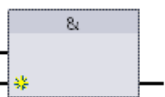
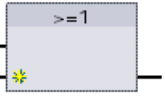
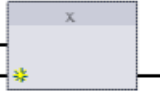
- Нормально открытый контакт пропускает сигнал (ON), когда значение бита равно 1.
- Нормально закрытый контакт пропускает сигнал (ON), когда значение бита равно 0.
- Контакты, соединенные последовательно образуют логическое И.
- Контакты, соединенные параллельно образуют логическое ИЛИ.

FBD AND, OR и XOR инструкции

При программировании на языке FBD, сегменты с LAD контактами преобразуются в соответствующие AND (&), OR (≥ 1), и EXCLUSIVE OR (x) сегменты с блоками, в которых Вы можете назначить значения для входов и выходов блока. Вы также можете соединить их с другими логическими блоками, создав при этом собственную логику работы программы. После того, как блок был добавлен в сегмент, Вы можете перенести инструмент "Insert input" (Вставить вход) из панели "Favorites" (Избранное) или дерева инструкции, после чего отпустить его прямо на вход соответствующего блока. Вы также можете добавить вход, нажав правой кнопкой мыши на входе блока и, выбрав "Insert input".

Входы и выходы блоков могут быть соединены с другими логическими блоками, или Вы можете указать адрес бита или символьное имя для неподключенного входа. При выполнении блока, текущее состояние входов обрабатывается согласно двоичной логике блока, и если значение истинно, то выход будет также истинным.

Таблица 8- 3 Инструкции AND, OR, и XOR

FBD	SCL ¹	Описание
	<code>out := in1 AND in2;</code>	Все входы инструкции AND должны иметь значение TRUE, чтобы выход принял значение TRUE.
	<code>out := in1 OR in2;</code>	Какой-либо из входов инструкции OR должен иметь значение TRUE, чтобы выход принял значение TRUE.
	<code>out := in1 XOR in2;</code>	Нечетное количество входов инструкции XOR должно иметь значение TRUE, чтобы выход принял значение TRUE.

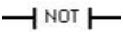
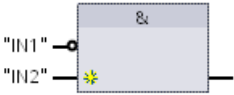
¹ В SCL: Вы должны присвоить результат операции переменной, которая будет использоваться в другом выражении.

Таблица 8- 4 Типы данных параметров

Параметр	Тип данных	Описание
IN1, IN2	Bool	Входной бит

NOT – логический инвертер

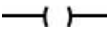
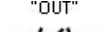
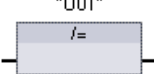
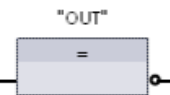
Таблица 8- 5 Инвертировать RLO (результат логической операции)

LAD	FBD	SCL	Описание
	 	NOT	При работе с языком программирования FBD, Вы можете захватить инструкцию "Invert RLO" из панели "Favorites" (Избранное) или дерева инструкций, после чего отпустить на вход или выход для создания логического инвертера на данном соединителе блока. При работе с языком LAD, контакт NOT инвертирует логическое состояния прохождения сигнала. - Если отсутствует прохождение сигнала к NOT контакту, то на выходе есть сигнал. - Если есть прохождение сигнала к NOT контакту, то на выходе нет сигнала.

Выходная катушка и инструкция назначения

Инструкция выходной катушки записывает значение в выходной бит. Если выходной бит, назначенный Вами, использует идентификатор Q, то ЦПУ переводит выходной бит в регистре образа процесса в активное или неактивное состояния, назначив указанному биту значение прохождения сигнала. Выходные сигналы для Ваших приводов подключены к Q клемма ЦПУ. В режиме RUN ЦПУ непрерывно сканирует Ваши входные сигналы, обрабатывает их состояния в соответствии с логикой Вашей программы, и после этого устанавливает новые значения выходов в регистре образа процесса. ЦПУ передает новые значения состояния выходов, которые хранятся в регистре образа процесса на выходные клеммы.

Таблица 8- 6 Назначение и инвертированное назначение

LAD	FBD	SCL	Описание
		out := <Boolean expression>;	При работе с языком программирования FBD, LAD катушки преобразуются в блоки назначения (= и /=), в которых Вы задаете битовый адрес для выхода блока. Входы и выходы блока могут быть соединены с другими логическими блоками или Вы можете указать битовый адрес. Вы можете произвести непосредственную запись в физический выход при помощи ":P" после соответствующего смещения Q (например: "%Q3.4:P"). При непосредственной записи битовые значения записываются в образ процесса и непосредственно в физический выход.
		out := NOT <Boolean expression>;	
			

8.1 Операции бинарной логики

Таблица 8- 7 Типы данных параметров

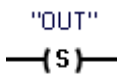
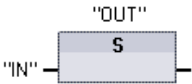
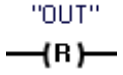
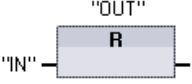
Параметр	Тип данных	Описание
OUT	Bool	Назначенный бит

- Если есть прохождение сигнала через выходную катушку или FBD-блок "=" активирован, то значение выхода устанавливается в 1.
- Если нет прохождения сигнала через выходную катушку или FBD-блок "=" не активирован, то значение выхода устанавливается в 0.
- Если есть прохождение сигнала через инвертированную выходную катушку или FBD-блок "/=" активирован, то значение выхода устанавливается в 0.
- Если нет прохождения сигнала через инвертированную выходную катушку или FBD-блок "/=" не активирован, то значение выхода устанавливается в 1.

8.1.2. Инструкции установки и сброса

Установка и сброс одного бита

Таблица 8- 8 Инструкции S и R

LAD	FBD	SCL	Описание
		Недоступно	Установить выход: При активации S (Установить), значение адреса на выходе OUT устанавливается в 1. Если S не активируется, то значение на выходе OUT не изменится.
		Недоступно	Сбросить выход: При активации R (Сбросить), значение адреса на выходе OUT устанавливается в 0. Если R не активируется, то значение на выходе OUT не изменится.

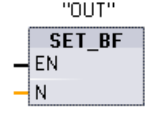
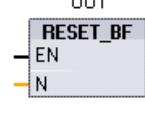
- ¹ В LAD и FBD: Данные инструкции могут быть вставлены в любом месте сегмента.
- ² В SCL: Вы должны написать код для работы данной функции в Вашем приложении.

Таблица 8- 9 Типы данных параметров

Параметр	Тип данных	Описание
IN (или соединение с контактной или блочной логикой)	Bool	Битовый тег контролируемой ячейки
OUT	Bool	Битовый тег устанавливаемой или сбрасываемой ячейки

Установка и сброс битового поля

Таблица 8- 10 Инструкции SET_BF и RESET_BF

LAD ¹	FBD	SCL	Описание
		Недоступно	Установить битовое поле: При активации SET_BF, значение 1 присваивается "n" битам, начиная с адреса тега OUT. Если SET_BF не активируется, то OUT не меняется.
		Недоступно	Сбросить битовое поле: RESET_BF записывает значение 0 в "n" биты, начиная с адреса тега OUT. Если инструкция RESET_BF не активируется, то OUT не меняется.

¹ В LAD и FBD: Данные инструкции должны быть крайними справа в сегменте.

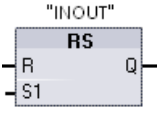
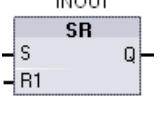
² В SCL: Вы должны написать код для работы данной функции в Вашем приложении.

Таблица 8- 11 Типы данных параметров

Параметр	Тип данных	Описание
OUT	Bool	Начальный элемент битового поля, которое необходимо установить или сбросить (Например: #MyArray[3])
n	Константа (UInt)	Количество битов для записи

Триггер с приоритетом установки и сброса

Таблица 8- 12 Инструкции RS и SR

LAD / FBD	SCL	Описание
	Недоступно	Триггер сброса/установки: RS представляет собой триггер-защелку с приоритетом, в котором приоритет имеет установка. Если оба сигнала установки (S1) и сброса (R) истинны, то ячейка по адресу INOUT будет иметь состояние 1.
	Недоступно	Триггер установки/сброса: SR представляет собой триггер-защелку с приоритетом, в котором приоритет имеет сброс. Если оба сигнала установки (S) и сброса (R1) истинны, то ячейка по адресу INOUT будет иметь состояние 0.

¹ В LAD и FBD: Данные инструкции должны быть крайней справа инструкцией в сегменте.

² В SCL: Вы должны написать код для работы данной функции в Вашем приложении.

Таблица 8- 13 Типы данных параметров

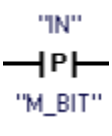
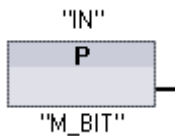
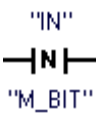
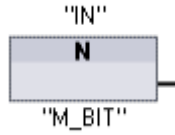
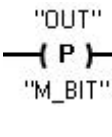
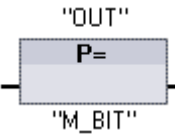
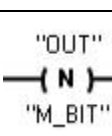
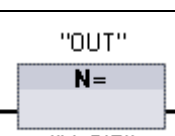
Параметр	Тип данных	Описание
S, S1	Bool	Установка входа; 1 означает приоритет
R, R1	Bool	Сброс входа; 1 означает приоритет
INOUT	Bool	Назначенный тег "INOUT"
Q	Bool	Состояние тега "INOUT"

Тег "INOUT" назначает битовый адрес для установки или сброса. Дополнительный выход Q показывает состояния сигнала с адресом "INOUT".

Инструкция	S1	R	Бит "INOUT"
RS	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	1
SR	S1	R1	
	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	0

8.1.3. Инструкции выделения положительного и отрицательного фронта

Таблица 8- 14 Положительное и отрицательное изменение фронта

LAD	FBD	SCL	Описание
		Недоступно ¹	Сканировать операнд на положительный фронт. LAD: Контакт имеет состояние TRUE, когда происходит переход из 0 в 1 (OFF в ON) в назначенном бите "IN". Затем логическое состояние контакта комбинируется с состоянием прохождения сигнала на входе, чтобы установить состояние сигнала на выходе. Р-контакт может быть расположен в любом месте сегмента, но не в конце ветви. FBD: Состояние выхода логики равно TRUE, когда происходит переход из 0 в 1 (OFF в ON) в назначенном бите "IN". Р-блок может быть расположен только в начале ветви.
		Недоступно ¹	Сканировать операнд на отрицательный фронт. LAD: Контакт имеет состояние TRUE, когда происходит переход из 1 в 0 (ON в OFF) в назначенном входном бите. Затем логическое состояние контакта комбинируется с состоянием прохождения сигнала на входе, чтобы установить состояние сигнала на выходе. N-контакт может быть расположен в любом месте сегмента, но не в конце ветви. FBD: Состояние выхода логики равно TRUE, когда происходит переход из 1 в 0 (ON в OFF) в назначенном входном бите. N-блок может быть расположен только в начале сегмента.
		Недоступно ¹	Установить операнд по положительному фронту. LAD: Назначенный бит "OUT" будет иметь состояние TRUE, при обнаружении перехода из 0 в 1 (OFF в ON) состояния прохождения сигнала через катушку. Состояние прохождения сигнала на входе всегда передается через катушку в качестве состояния сигнала на выходе. Р-катушка может находиться в любом месте сегмента. FBD: Назначенный бит "OUT" будет иметь состояние TRUE при обнаружении перехода из 0 в 1 (OFF в ON) на входе блока или в назначенном на входе бите, если блок находится в начале ветви. Состояние входа всегда передается через блок в качестве состояния выхода. Блок P= может находиться в любом месте ветви.
		Недоступно ¹	Установить операнд по отрицательному фронту. LAD: Назначенный бит "OUT" будет иметь состояние TRUE, при обнаружении перехода из 1 в 0 (ON в OFF) состояния прохождения сигнала через катушку. Состояние прохождения сигнала на входе всегда передается через катушку в качестве состояния сигнала на выходе. N-катушка может находиться в любом месте сегмента. FBD: Назначенный бит "OUT" будет иметь состояние TRUE при обнаружении перехода из 1 в 0 (ON в OFF) на входе блока или в назначенном на входе бите, если блок находится в начале ветви. Состояние входа всегда передается через инструкцию в качестве состояния выхода. Блок N= может находиться в любом месте ветви.

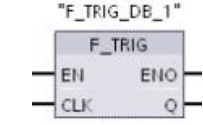
¹ В SCL: Вы должны написать код для работы данной функции в Вашем приложении.

Таблица 8- 15 P_TRIG и N_TRIG

LAD / FBD	SCL	Описание
	Недоступно ¹	Сканирует RLO (результат логической операции) на положительный фронт. Прохождение сигнала или логическое состояние на выходе Q равно TRUE, когда обнаруживается положительный фронт (OFF-ON) состояния входа CLK (FBD) или прохождения сигнала на входе CLK (LAD). В LAD инструкция P_TRIG не может быть расположена в начале или конце сегмента. В FBD инструкция P_TRIG может быть расположена в любом месте, кроме окончания ветви.
	Недоступно ¹	Сканирует RLO (результат логической операции) на отрицательный фронт. Прохождение сигнала или логическое состояние на выходе Q равно TRUE, когда обнаруживается отрицательный фронт (ON - OFF) состояния входа CLK (FBD) или прохождения сигнала на входе CLK (LAD). В LAD инструкция N_TRIG не может быть расположена в начале или конце сегмента. В FBD инструкция N_TRIG может быть расположена в любом месте, кроме окончания ветви.

¹ В SCL: Вы должны написать код для работы данной функции в Вашем приложении.

Таблица 8- 16 Инструкции R_TRIG и F_TRIG

LAD / FBD	SCL	Описание
	<pre>"R_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);</pre>	Установить тег по положительному фронту. Назначенный экземпляр блока данных (DB) используется для сохранения предыдущего состояния входа CLK. Выход Q будет иметь состояние TRUE при обнаружении положительного перехода (OFF в ON) состояния входа CLK (FBD) или прохождения сигнала на входе CLK (LAD). В LAD, инструкция R_TRIG не может находиться в начале или конце сегмента. В FBD, инструкция R_TRIG может находиться в любом месте сегмента, кроме окончания ветви.
	<pre>"F_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);</pre>	Установить тег по отрицательному фронту. Назначенный экземпляр блока данных (DB) используется для сохранения предыдущего состояния входа CLK. Выход Q будет иметь состояние TRUE при обнаружении отрицательного перехода (ON в OFF) состояния входа CLK (FBD) или прохождения сигнала на входе CLK (LAD). В LAD, инструкция F_TRIG не может находиться в начале или конце сегмента. В FBD, инструкция F_TRIG может находиться в любом месте сегмента, кроме окончания ветви.

При работе с R_TRIG и F_TRIG, когда Вы вставляете инструкцию в программу, автоматически открывается диалоговое окно "Call options" (Варианты вызова). В нем, Вы можете определить

будет ли сохраняться бит изменения фронта в собственном блоке данных (экземпляре) или в качестве локального тега (мультиэкземпляр) в интерфейсе блока. Если, Вы создадите отдельный блок данных, Вы сможете найти его в дереве проекта в каталоге "Program resources" (Ресурсы программы)

в разделе "Program blocks > System blocks" (Программные блоки > Системные блоки).

Таблица 8- 17 Типы данных параметров (P и N контакты/катушки, P=, N=, P_TRIG и N_TRIG)

Параметр	Тип данных	Описание
M_BIT	Bool	Битовая ячейка, в которой сохраняется предыдущее состояние входа
IN	Bool	Входной бит, фронт перехода которого обнаруживается
OUT	Bool	Выходной бит, который указывает на обнаружение фронта перехода
CLK	Bool	Прохождение сигнала или входной бит, фронт перехода которого обнаруживается
Q	Bool	Выход, который указывает на обнаружение фронта

Все инструкции по работе с фронтом используют битовую ячейку памяти (M_BIT: P/N контакты/катушки, P_TRIG/N_TRIG) или бит в экземплярном DB (R_TRIG, F_TRIG) для сохранения предыдущего состояния входного сигнала. Изменение фронта определяется сравнением состояния входа с предыдущим состоянием. Если состояние входа отличается от своего предыдущего состояния, то изменение фронта записывается на выход как TRUE. В противном случае, состояние выхода будет иметь значение FALSE.

Примечание

Инструкции фронта оценивают значения входа и битовой ячейки памяти каждый раз, когда они выполняются, включая первое выполнение. Вы должны принять в расчет начальные состояния входа и ячейки памяти при разработке Вашей программы или не обнаруживать фронт при первом сканировании.

Поскольку бит памяти должен сохраняться от одного скана до другого, Вы должны использовать уникальный бит для каждой инструкции фронта, и Вы не должны использовать этот бит в каком-либо другом месте в Вашей программе. Вы должны также избегать использования временной памяти и памяти, на которую может воздействовать другая системная функция, такая как обновление ввода-вывода. Используйте только М-память, глобальный DB или статическую память (в экземплярном DB) для назначения ячейки M_BIT.

8.2 Таймеры

Вы можете использовать соответствующие инструкции для создания задержек по времени. Максимальное количество таймеров, которые Вы можете создать, зависит от объема памяти ЦПУ. Каждая инструкция использует 16 байтовую структуру с типом IEC_Timer в DB для хранения данных таймера, которая задается в верхней части инструкции. При добавлении инструкции STEP 7 автоматически создает DB.

Таблица 8- 18 Таймерные инструкции

LAD / FBD инструкции	LAD катушки	SCL	Описание
IEC_Timer_0 TP Time IN Q PT ET	TP_DB (TP) "PRESET_Tag"	<pre>"IEC_Timer_0_DB".TP(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Таймер TP генерирует импульс с заданным временем.
IEC_Timer_1 TON Time IN Q PT ET	TON_DB (TON) "PRESET_Tag"	<pre>"IEC_Timer_0_DB".TON (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Таймер TON устанавливает выход Q в активное состояние, спустя заданное время.
IEC_Timer_2 TOF Time IN Q PT ET	TOF_DB (TOF) "PRESET_Tag"	<pre>"IEC_Timer_0_DB".TOF (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Таймер TOF сбрасывает выход Q в неактивное состояние, спустя заданное время.
IEC_Timer_3 TONR Time IN Q R ET PT	TONR_DB (TONR) "PRESET_Tag"	<pre>"IEC_Timer_0_DB".TONR (IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Таймер TONR устанавливает выход Q в активное состояние, спустя заданное время. Истекшее время накапливается за несколько периодов тактирования до тех пор, пока вход R не сбросит истекшее время.
Только FBD: PT PT	TON_DB (PT) "PRESET_Tag"	<pre>PRESET_TIMER(PT:=_time_in_, TIMER:=_iec_timer_in_);</pre>	Катушка PT (предустановка таймера) загружает новое значение времени в IEC_Timer.
Только FBD: RT	TON_DB (RT)	<pre>RESET_TIMER(_iec_timer_in_);</pre>	Катушка RT (Сброс таймера) сбрасывает указанный IEC_Timer.

¹ STEP 7 автоматически создает DB при добавлении инструкции.

² В примерах на языке SCL, "IEC_Timer_0_DB" является именем экземплярного DB.

Таблица 8- 19 Типы данных параметров

Параметр	Тип данных	Описание
Блок: IN Катушка: Прохождение сигнала	Bool	ТР, TON, и TONR: Блок: 0=Отключить таймер, 1=Активировать таймер Катушка: Нет прохождения сигнала=Отключить таймер, Есть прохождение сигнала=Активировать таймер TOF: Блок: 0= Активировать таймер, 1= Отключить таймер Катушка: Нет прохождения сигнала = Активировать таймер, Есть прохождение сигнала = Отключить таймер
R	Bool	Только для инструкции TONR: 0=Без сброса 1=Сброс истекшего времени и установка бита Q в 0
Блок: PT Катушка: "PRESET_Tag"	Time	Блок таймера или катушка: Вход установки времени
Блок: Q Катушка: DBdata.Q	Bool	Блок таймера: Выход блока Q или бит Q в DB таймера Катушка таймера: Вы можете обратиться только к биту Q в DB таймера
Блок: ET Катушка: DBdata.ET	Time	Блок таймера: Выход блока ET (истекшее время) или значение ET в DB таймера Катушка таймера: Вы можете обратиться только к значению времени ET в DB таймера.

Таблица 8- 20 Реакция на изменение значений в параметрах PT и IN

Таймер	Изменения в параметрах блока PT и IN и соответствующих параметрах катушки
TP	- Изменение PT не влияет на работу, пока таймер запущен. - Изменение IN не влияет на работу, пока таймер запущен.
TON	- Изменение PT не влияет на работу, пока таймер запущен. - Изменение параметра IN на FALSE, пока таймер запущен, сбрасывает и останавливает таймер.
TOF	- Изменение PT не влияет на работу, пока таймер запущен. - Изменение параметра IN на TRUE, пока таймер запущен, сбрасывает и останавливает таймер.
TONR	- Изменение PT не влияет на работу, пока таймер запущен, но влияет, когда таймер продолжает работать. - Изменение параметра IN на FALSE, пока таймер запущен, останавливает его, но не сбрасывает. Изменение IN обратно на TRUE запустит таймер с накопленного значения.

Значения PT (заданное время) и ET (истекшее время) хранятся в назначенном блоке данных IEC_TIMER, как двойные целые числа со знаком, представленные в миллисекундах. Тип TIME использует идентификатор T# и может быть задан в простом (T#200ms или 200) или комбинированном формате T#2s_200ms.

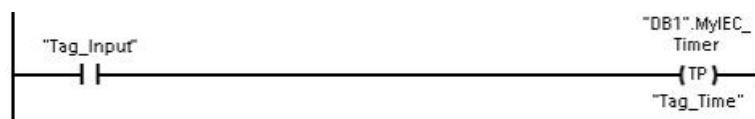
Таблица 8- 21 Размер и диапазон значений типа данных TIME

Тип данных	Размер	Допустимые диапазоны значений ¹
TIME	32 бита, хранится как DInt	от T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms сохраняется как диапазон (-2,147,483,648 мс -- +2,147,483,647 мс)

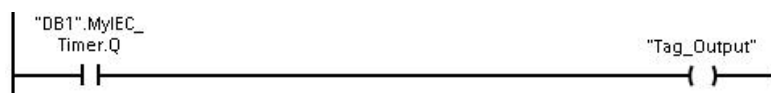
¹ Отрицательные значения типа данных TIME не могут быть использованы в таймерных инструкциях. Отрицательные значения PT (заданное время) сбрасываются в ноль при выполнении инструкции таймера. ET (истекшее время) всегда является положительным числом.

Пример таймерных катушек

Таймерные катушки -(TP)-, -(TON)-, -(TOF)- и -(TONR)- должны быть последними инструкциями в сегменте LAD. Как показано в примере с таймером, инструкция контакта в последующем сегменте оценивает бит Q в структуре IEC_Timer в DB. Аналогично, Вы должны адресовать элемент ELAPSED из структуры IEC_Timer в DB, если Вы хотите использовать значение истекшего времени в своей программе.



Импульсный таймер начинает свою работу при переходе из 0 в 1 тега Tag_Input. Таймер запускается на время, определенное значением Tag_Time.



Пока таймер работает, состояние DB1.MyIEC_Timer.Q=1, и состояние тега Tag_Output равно = 1. Когда значение Tag_Time истекло, параметры DB1.MyIEC_Timer.Q и Tag_Output сбрасываются в 0.

Катушки сброса таймера -(RT)- предустановки таймера -(PT)-

Эти инструкции катушек могут использоваться с таймерами в виде блока или катушки и могут быть помещены в промежуточную позицию. Состояние сигнала на выходе катушки всегда соответствует состоянию сигнала на входе. Когда катушка - (RT) - активирована, элемент ELAPSED указанного DB данных IEC_Timer сброшен в 0. Когда катушка - (PT) - активирована, в элемент PRESET указанного DB данных IEC_Timer загружено назначенное значение выдержки времени.

Примечание

Когда Вы помещаете инструкции таймера в FB, Вы можете выбрать опцию "Multi-instance data block". Имена структуры таймера могут отличаться от имен отдельных структур данных, но данные таймера содержатся в едином блоке данных и не требуют отдельного блока данных для каждого таймера. Это сокращает время обработки и объем используемой памяти, необходимые для обработки таймеров. Не существует какой-либо взаимосвязи между структурами данных таймера в совместно используемом мультитекстном DB.

Принцип работы таймеров

Таблица 8- 22 Типы IEC таймеров

Таймер	Диаграмма
TP: Генерировать импульс Таймер TP генерирует импульс с заданным временем.	
TON: Генерировать задержку включения Таймер TON устанавливает выход Q в активное состояние, спустя заданное время.	

Таймер	Диаграмма
TOF: Генерировать задержку выключения Таймер TOF устанавливает выход Q в неактивное состояние, спустя заданное время.	
TONR: Накопитель времени Таймер TONR устанавливает выход Q в активное состояние по истечении заданного времени. Истекшее время накапливается за несколько периодов тактирования, пока вход не будет использован вход R, чтобы сбросить истекшее время.	

Примечание

В ЦПУ нет выделенных ресурсов для инструкций типа таймер. Наоборот, каждый таймер использует свою собственную структуру в памяти DB и непрерывно работающей внутренней таймер ЦПУ.

Когда таймер запущен по изменению фронта на входе инструкции TP, TON, TOF или TONR, значение непрерывно работающего внутреннего таймера ЦПУ копируется в элемент START структуры DB, выделенной для этой инструкции таймера. Это стартовое значение остается неизменным, в то время как таймер продолжает работать и используется позже каждый раз, когда таймер обновляется. Каждый раз, когда таймер запущен, новое стартовое значение загружается в структуру таймера из внутреннего таймера ЦПУ.

Когда таймер обновляется, стартовое значение, описанное выше, вычитается из текущего значения внутреннего таймера ЦПУ, чтобы определить истекшее время. Истекшее время затем сравнивается с предварительной установкой, чтобы определить состояние Q бита таймера. Элементы ELAPSED и Q обновляются в структуре DB, выделенной для этого таймера. Обратите внимание на то, что истекшее время фиксируется на уровне предварительно установленного значения (таймер не продолжает накапливать истекшее время после того, как предварительная установка достигнута)

Обновление таймера выполняется только когда:

- Была выполнена таймерная инструкция (TP, TON, TOF, и TONR)
- Обращение к параметру "ELAPSED", структуры таймера блока данных DB, выполняется непосредственно из самой инструкции
- Обращение к параметру "Q", структуры таймера блока данных DB, выполняется непосредственно из самой инструкции

Программирование таймеров

При работе с таймерами, необходимо соблюсти следующие требования в Вашей программе:

- Вы можете обновить таймер несколько раз за один скан. Таймер обновляется каждый раз, когда выполняется инструкция (TP, TON, TOF, TONR) и каждый раз, когда элементы структуры таймера ELAPSED или Q используются в качестве параметра другой обрабатываемой инструкции, что является преимуществом, если Вам необходима наиболее последняя информация о таймере (а именно, быстрый доступ на чтение таймера). Тем не менее, если Вы решили работать с консистентными значениями при сканировании программы, то необходимо поместить Вашу таймерную инструкцию выше остальных инструкций, которым также требуются эти значения, и использовать теги с выходов таймера Q и ET вместо элементов структуры блока данных ELAPSED и Q.
- У Вас также могут быть сканы, в течение которых таймер не будет обновляться. Для этого, необходимо вызвать таймер из функции, после чего прекратить вызов данной функции на один или нескольких сканов. Если нет других инструкций, которые обращаются к элементам структуры таймера ELAPSED или Q, то обновление таймера не будет произведено. Новый процесс обновления не будет начат до тех пор, пока инструкция не будет выполнена повторно или не будет выполнена другая команда, использующая параметры структуры таймера ELAPSED или Q.
- Несмотря на то, что это не типично, Вы можете назначить ту же DB структуру таймера нескольким инструкциям таймера. Как правило, чтобы избежать неожиданного взаимного влияния, Вы должны использовать только одну инструкцию таймера (TP, TON, TOF, TONR) на одну DB структуру таймера.
- Таймеры с самосбросом полезны, чтобы инициировать действия, которые должны происходить периодически. Как правило, таймеры с самосбросом создаются, помещая нормально замкнутый контакт, который ссылается на бит таймера перед инструкцией таймера. Этот сегмент таймера обычно располагается выше одного или более зависимых сегментов, которые используют бит таймера, чтобы инициировать действия. Когда таймер истекает (истекшее время достигает предварительно установленного значения), бит таймера становится активным на один скан, позволяя выполнение зависимых логически сегментов, которыми управляет таймер. При следующем выполнении сегмента таймера нормально замкнутый контакт деактивирован, таким образом сбрасывая таймер и выходной бит таймера. При следующем скане нормально замкнутый контакт активирован, таким образом перезапуская таймер. При создании самосброса таймеров, таких как этот, не используйте элемент "Q" DB структуры таймера в качестве параметра для нормально замкнутого контакта перед инструкцией таймера. Вместо этого используйте тег, соединенный с выходом "Q" инструкции таймера. Причина подобного ограничения состоит в том, что это вызывает обновление таймера, и, если таймер будет обновлен из-за нормально замкнутого контакта, то контакт сразу сбросит инструкцию таймера. Выход Q инструкции таймера не будет активирован на один скан, и зависимые сегменты не будут выполнены.

Сохранение данных таймера после перехода RUN-STOP-RUN или цикла питания ЦПУ

Если режим RUN был завершен переходом в STOP или циклом выключения питания и осуществляется повторный переход ЦПУ в режим RUN, то данные таймера, сохраненные в предыдущей сессии режима RUN ЦПУ будут потеряны, если только структура данных не определена, как сохраняемая (TP, TON, TOF и TONR таймеры).

Если Вы принимаете настройки по умолчанию в окне вызова после добавления инструкции таймера в программном редакторе, Вы автоматически назначаете экземпляр DB, который **не может быть сохраняемым**. Чтобы Ваши данные таймера были сохраняемыми, Вы должны либо использовать глобальный блок данных DB или мультиэкземплярный DB.

Назначение глобального DB для возможности сохранения данных таймера

Данный способ работает независимо от того, где используется таймер (OB, FC или FB).

1. Создайте глобальный блок данных DB:
 - Двойной щелчок мыши "Add new block" (Добавить новый блок) в дереве проекта
 - Щелкните по пиктограмме блока данных (DB)
 - В качестве типа укажите глобальный DB
 - Если Вы хотите выбрать отдельные элементы данных в этом DB как сохраняемые, убедитесь, что опция "Optimized" отмечена флажком. Другой вариант DB - "Standard - compatible with S7-300/400" позволяет сделать либо все элементы блока данных сохраняемыми либо все несохраняемыми.
 - Щелкните по ОК
2. Добавьте структуру(-ы) DB для таймеров:
 - В новом глобальном DB, добавьте новый статический тег с типом IEC_Timer.
 - В столбце "Retain" (Сохраняемость), убедитесь, что установлен флаг для сохраняемости данной структуры.
 - Повторите данную процедуру для создания структур для всех таймеров, которые Вы хотите сохранять в блоке данных DB. Либо Вы можете поместить каждую структуру в уникальный глобальный DB, или Вы можете поместить несколько структур таймера в один глобальный DB. Вы также можете поместить остальные статические теги, помимо таймеров в данном глобальном DB. Добавление нескольких структур для таймера в один глобальный DB снизит общее количество, используемых Вами блоков.
 - При необходимости, переименуйте структуры таймеров.
3. Откройте программный блок для редактирования, в котором Вы будете использовать сохраняемый таймер (OB, FC или FB).
4. Добавьте инструкцию таймера там, где это необходимо.
5. При появлении диалогового окна с выбором вариантов вызова, закройте его.
6. Над новой инструкцией таймера, укажите имя (не используйте средств помощи для просмотра) глобального DB и структуру таймера, которые Вы создали ранее (пример: "Data_block_3.Static_1").

Назначение мультиэкземплярного DB для возможности сохранения данных таймера

Данная возможность доступна только если Вы добавили таймер в FB.

Необходимо убедиться, что выбран тип доступа к FB "Optimized block access" (только символьный доступ). Чтобы определить, как сконфигурирован атрибут доступа к FB, щелкните правой кнопкой мыши по FB в дереве проекта, перейдите к свойствам, а затем выберите Attributes.

Если FB имеет тип доступа "Optimized block access" (только символьный доступ):

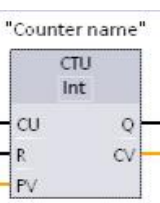
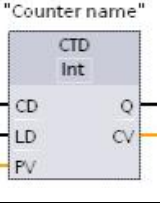
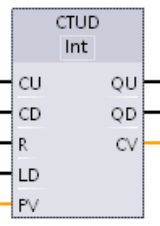
1. Откройте FB.
2. Добавьте таймерную инструкцию в нужный сегмент FB.
3. При появлении окна с вариантом вызова, нажмите на пиктограмму мультиэкземпляра. Данная опция доступна только если инструкция была добавлена в FB.
4. В появившемся окне вариантов вызова, переименуйте таймер, если это необходимо.
5. Нажмите ОК. Инструкция таймера появится в редакторе, а структура IEC_TIMER в интерфейсе FB в разделе Static.
6. При необходимости, откройте редактор интерфейса FB (может потребоваться нажатие на небольшую стрелку, для раскрытия представления).
7. В Static, найдите структуру таймера, которая была создана для Вас.
8. В столбце Сохраняемость (Retain) данной таймерной структуры, перейдите на "Retain". Всякий раз, когда данный FB будет вызван из другого программного блока, будет создан экземпляр DB с соответствующим интерфейсом, с сохраняемой таймерной структурой.

Если FB не обладает типом доступа "Optimized block access", это означает, что доступ к блоку является стандартным, что характерно для S7-300/400. При таком доступе возможен как символьный доступ, так и абсолютный. Для назначения мультиэкземпляра FB со стандартным типом доступа, проделайте следующие шаги:

1. Откройте FB.
2. Добавьте таймерную инструкцию в необходимый сегмент FB.
3. При появлении окна с вариантом вызова, нажмите на пиктограмму мультиэкземпляра. Данная опция доступна только если инструкция была добавлена в FB.
4. В появившемся окне вариантов вызова, переименуйте таймер, если это необходимо.
5. Нажмите ОК. Инструкция таймера появится в редакторе, а структура IEC_TIMER в интерфейсе FB в разделе Static.
6. Откройте блок, который будет использовать данный FB.
7. Добавьте данный FB. После этого будет создан экземплярный блок данных для этого FB.
8. Откройте созданный экземплярный блок данных.
9. В Static, найдите структуру таймера. В столбце Сохраняемость (Retain) данной структуры, установите флажок, чтобы сделать эту структуру сохраняемой.

8.3 Операции счета

Таблица 8- 23 Счетные инструкции

LAD / FBD	SCL	Описание
	<pre>"IEC_Counter_0_DB".CTU (CU:=_bool_in, R:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	Используйте данные инструкции для подсчета внутренних и внешних событий в программе. Каждый счетчик использует структуру, которая находится в блоке данных, чтобы хранить данные счетчика. При добавлении инструкции счетчика, Вам необходимо назначить блок данных. - CTU представляет собой инкрементный счетчик - CTD представляет собой декрементный счетчик - CTUD представляет собой реверсивный счетчик
	<pre>"IEC_Counter_0_DB".CTD (CD:=_bool_in, LD:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	
	<pre>"IEC_Counter_0_DB".CTUD D(CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LD:=_bool_in, PV:=_in, QU=>_bool_out, QD=>_bool_out, CV=>_out);</pre>	

1 В LAD и FBD: Выберите тип счетчика из выпадающего меню соответствующей инструкции.

2 STEP 7 автоматически создает DB при добавлении инструкции.

3 В примерах на языке SCL "IEC_Timer_0_DB" является именем экземплярного DB.

Таблица 8- 24 Типы данных параметров

Параметр	Тип данных ¹	Описание
CU, CD	Bool	Прямой и обратный счет на единицу
R (CTU, CTUD)	Bool	Сброс значения счетчика в ноль
LD (CTD, CTUD)	Bool	Вход управления загрузкой предустановки
PV	SInt, Int, DInt, USInt, UInt, UDim	Предустановка
Q, QU	Bool	True, если CV >= PV
QD	Bool	True, если CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDim	Текущее значение счетчика

¹ Диапазон допустимых значений зависит от выбранного типа данных. Если значение счетчика имеет тип целого без знака, то диапазон устанавливается от 0 до максимального значения. При выборе целого со знаком, счет ведется от нижней границы до верхней.

Количество счетчиков, которые Вы можете использовать в Вашей программе ограничено только объемом памяти ЦПУ. Счетчики используют следующие объемы памяти:

- Для типов данных SInt или USInt, счетчик использует 3 байта.
- Для типов данных Int или UInt, счетчик использует 6 байтов.
- Для типов данных Dint или UDInt, счетчик использует 12 байтов.

Данные инструкции используют программные счетчики, чья максимальная скорость счета ограничен частотой вызова OB в котором они находятся. OB в котором находятся инструкции должен выполняться достаточно часто, чтобы обнаружить все фронты входов CU или CD. Для более быстрого счета, см.инструкцию CTRL_HSC (стр. 457).

Примечание

При добавлении инструкций счетчики в FB, Вы можете выбрать мультиэкземплярный DB, имена структур для счетчиков могут отличаться от структур данных, однако данные счетчика хранятся в отдельном DB, при этом не требуется для каждого отдельного счетчика создавать отдельный DB. Это позволяет сэкономить время и память для хранения данных нескольких счетчиков. При этом связь между структурами в мультиэкземплярных DB отсутствует.

Принцип работы счетчиков

Таблица 8- 25 Работа счетчика CTU (счет на увеличение)

Счетчик	Принцип
Счетчик CTU каждый раз увеличивает значение на 1, когда значение параметра CU изменилось из 0 в 1. На временной диаграмме работы счетчика CTU, показана работа с беззнаковым целым типом данных (где PV = 3). • Если значение параметра CV (текущее значение счетчика) больше или равно значению параметра PV (заданное значение счетчика), тогда выходной параметр Q = 1. • Если значение параметра сброса R изменилось с 0 на 1, тогда текущее значение счетчика будет сброшено в 0.	The diagram shows four signals over time: CU (count input), R (reset input), CV (current value), and Q (output flag). CU has four rising edges. R has one rising edge. CV starts at 0 and increments by 1 on each rising edge of CU. When R goes high, CV resets to 0. Q becomes 1 when CV reaches the preset value PV (3) and resets to 0 when CV resets to 0 or reaches 0.

8.3 Операции счета

Таблица 8- 26 Работа счетчика CTD (счет на уменьшение)

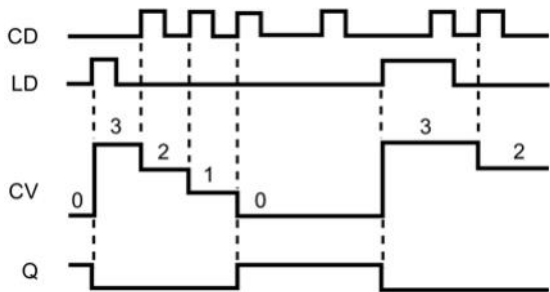
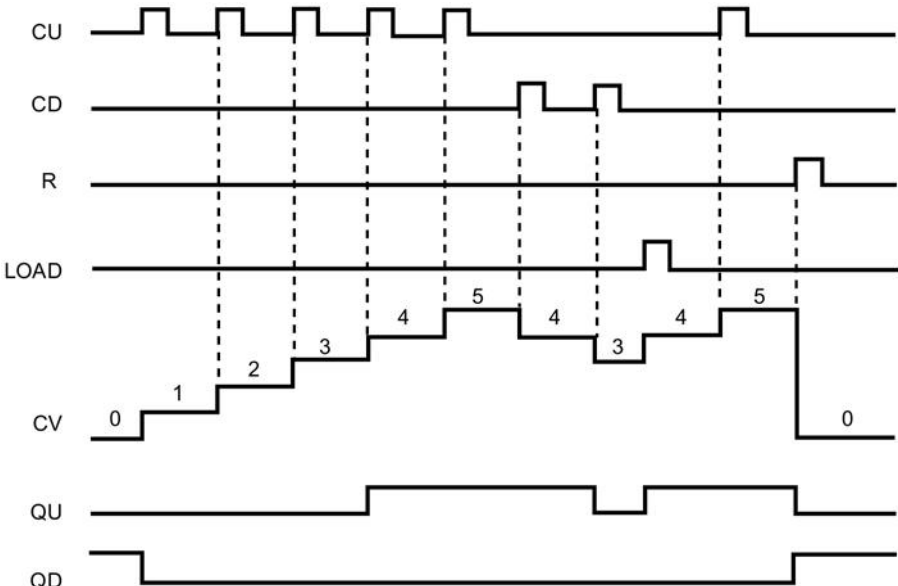
Счетчик	Принцип
Счетчик CTD каждый раз уменьшает значение на 1, когда значение параметра CD изменилось из 0 в 1. На временной диаграмме счетчика CTD, показана работа с беззнаковым целым типом данных (где PV = 3) • Если значение параметра CV (текущее значение счетчика) меньше или равно значению параметра PV (заданное значение счетчика), тогда выходной параметр Q = 1. • Если значение параметра LD изменилось из 0 в 1, тогда значение параметра PV (заданное значение) будет загружено в счетчик как новое CV (текущее значение счетчика)	

Таблица 8- 27 Работа счетчика CTUD (на понижение и повышение)

Счетчик	Принцип
Счетчик CTUD увеличивает свое значение на 1 при переходе из 0 в 1 на входе (CU) или уменьшает свое значение на 1 при переходе из 0 в 1 на входе (CD). На временной диаграмме счетчика CTUD, работа с беззнаковым целым типом данных (где PV = 4) • Если значение параметра CV больше или равно значению параметра PV, то значение параметра QU = 1. • Если значение параметра CV меньше или равно нулю, то значение параметра QD = 1. • Если значение параметра LOAD изменилось из 0 в 1, тогда значение параметра PV (заданное значение) будет загружено в счетчик как новое CV (текущее значение счетчика). • Если значение параметра сброса R изменилось из 0 в 1, тогда текущее значение счетчика будет сброшено в 0.	

Сохранение данных счетчика после перехода RUN-STOP-RUN или цикла питания ЦПУ

Если режим RUN был завершен переходом в STOP или циклом выключения питания и осуществляется повторный переход ЦПУ в режим RUN, то данные счетчика, сохраненные в предыдущей сессии режима RUN ЦПУ будут потеряны, если только структура данных не определена, как сохраняемая (CTU, CTD и CTUD счетчики).

Если Вы принимаете настройки по умолчанию в окне вызова после добавления инструкции счетчика в программном редакторе, Вы автоматически назначаете экземпляр DB, который **не может быть сохраняемым**. Чтобы Ваши данные счетчика были сохраняемыми, Вы должны либо использовать глобальный блок данных DB или мультиэкземплярный DB.

Назначение глобального DB для возможности сохранения данных счетчика

Данный способ работает независимо от того, где используется таймер (OB, FC или FB).

1. Создайте глобальный блок данных DB:
 - Двойной щелчок мыши "Add new block" (Добавить новый блок) в дереве проекта
 - Щелкните по пиктограмме блока данных (DB)
 - В качестве типа укажите глобальный DB
 - Если Вы хотите выбрать отдельные элементы данных в этом DB как сохраняемые, убедитесь, что используется блок данных с символьным типом доступа.
 - Щелкните по ОК
2. Добавьте структуру(-ы) DB для счетчиков:
 - В новом глобальном DB добавьте новый статический тег, используя один из типов данных счетчика. Постарайтесь использовать тип, который Вам необходим для Ваших значений Preset и Count.
 - В столбце "Retain" (Сохраняемость), убедитесь, что установлен флаг для сохраняемости данной структуры.
 - Повторите данную процедуру для создания структур для всех счетчиков, которые Вы хотите сохранять в блоке данных DB. Либо Вы можете поместить каждую структуру в уникальный глобальный DB, или Вы можете поместить несколько структур таймера в один глобальный DB. Вы также можете поместить остальные статические теги, помимо таймеров в данном глобальном DB. Добавление нескольких структур для таймера в один глобальный DB снизит общее количество, используемых Вами блоков.
 - При необходимости, переименуйте структуры счетчиков.
3. Откройте программный блок для редактирования, в котором Вы будете использовать сохраняемый счетчик (OB, FC или FB).
4. Добавьте счетчик в необходимом месте.
5. При появлении диалогового окна с выбором вариантов вызова, закройте его. Теперь Вы должны увидеть над именем счетчика и под ним - "???".
6. Над новой инструкцией счетчика укажите имя (без использования средств помощи при вводе) глобального DB и структуру данных, которые Вы создали до этого (например: "Data_block_3.Static_1"). После этого необходимо будет заполнить соответствующие поля предустановки и значения счетчика (например: UInt для структуры IEC_UCounter).

Тип данных счетчика	Соответствующий тип для предустановки и значения
IEC_Counter	INT
IEC_Scounter	SINT

IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

Назначение мультиэкземплярного DB для возможности сохранения данных счетчика

Данная возможность доступна только если Вы добавили счетчик в FB.

Необходимо убедиться, что выбран тип доступа к FB "Optimized block access" (только символьный доступ). Чтобы определить, как сконфигурирован атрибут доступа к FB, щелкните правой кнопкой мыши по FB в дереве проекта, перейдите к свойствам, а затем выберите Attributes.

Если FB имеет тип доступа "Optimized block access" (только символьный доступ):

1. Откройте FB.
2. Добавьте счетчик в нужный сегмент FB.
3. При появлении окна с вариантом вызова, нажмите на пиктограмму мультиэкземпляра. Данная опция доступна только если инструкция была добавлена в FB..
4. В появившемся окне вариантов вызова, переименуйте счетчик, если это необходимо.
5. Нажмите OK. Счетчик появится в редакторе, а структура IEC_COUNTER в интерфейсе FB в разделе Static.
6. При необходимости, измените тип счетчика с INT на другой. Структура счетчика изменится соответственно.
7. При необходимости, откройте редактор интерфейса FB (может потребоваться нажатие на небольшую стрелку, для раскрытия представления).
8. В Static, найдите структуру счетчика, которая была создана для Вас.
9. В столбце Сохраняемость (Retain) данной структуры счетчика, перейдите на "Retain". Всякий раз, когда данный FB будет вызван из другого программного блока, будет создан экземпляр DB с соответствующим интерфейсом, с сохраняемой структурой счетчика.

Если FB не обладает типом доступа "Optimized block access", это означает, что доступ к блоку является стандартным, что характерно для S7-300/400. При таком доступе возможен как символьный доступ, так и абсолютный. Для назначения мультиэкземпляра FB со стандартным типом доступа, проделайте следующие шаги:

1. Откройте FB.
2. Добавьте счетчик в нужный сегмент FB.
3. При появлении окна с вариантом вызова, нажмите на пиктограмму мультиэкземпляра. Данная опция доступна только если инструкция была добавлена в FB.
4. В появившемся окне вариантов вызова, переименуйте счетчик, если это необходимо.
5. Нажмите OK. Счетчик появится в редакторе с типом INT, а структура IEC_COUNTER в интерфейсе FB в разделе Static.
6. При необходимости, измените тип счетчика с INT на другой. Структура счетчика изменится соответственно.
7. Откройте блок, который будет использовать данный FB.

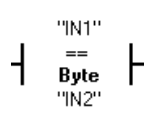
8. Добавьте FB. После этого будет создан экземплярный блок данных для данного FB.
9. Откройте созданный экземплярный блок данных.
10. В Static, найдите структуру счетчика. В столбце Сохраняемость (Retain) данной структуры, установите флажок, чтобы сделать эту структуру сохраняемой.

Тип отображаемый в инструкции счетчика (для значений предустановки и счета)	Соответствующий структурный тип, отображаемый в интерфейсе FB
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

8.4 Операции сравнения

8.4.1. Инструкции сравнения значений

Таблица 8- 28 Инструкции сравнения

LAD	FBD	SCL	Описание
		<pre> out := in1 = in2; or IF in1 = in2 THEN out := 1; ELSE out := 0; END_IF; </pre>	Сравнивают два значения с одним типом данных. Когда сравнение на LAD контакте равно TRUE, контакт активирован. Когда сравнение в FBD блоке равно TRUE, тогда выход блока равен TRUE.

¹ Для LAD и FBD: Щелкните на имени инструкции (таком как "=="), чтобы выбрать тип сравнения из выпадающего списка. Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 29 Типы данных параметров

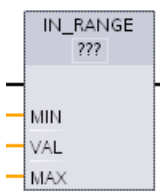
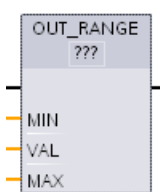
Параметр	Тип данных	Описание
IN1, IN2	Byte, Word, DWord, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, WString, Char, Time, Date, TOD, DTL, Constant	Сравниваемые значения

Таблица 8- 30 Описание функций сравнения

Тип	Результат сравнения ИСТИННЫЙ, когда ...
=	IN1 равен IN2
<>	IN1 не равен IN2
>=	IN1 больше или равен IN2
<=	IN1 меньше или равен IN2
>	IN1 больше чем IN2
<	IN1 меньше чем IN2

8.4.2. Инструкции IN_Range (Значение в диапазоне) и OUT_Range (Значение вне диапазона)

Таблица 8- 31 Инструкции проверки нахождения значения в пределах диапазона и вне диапазона

LAD / FBD	SCL	Описание
	<pre>out := IN_RANGE(min, val, max);</pre>	Проверяет, входит ли заданное значение в допустимый диапазон или нет. Если сравнение дает результат TRUE, то выход данной инструкции принимает значение TRUE.
	<pre>out := OUT_RANGE(min, val, max);</pre>	

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 32 Типы данных параметров

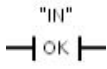
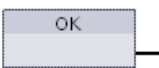
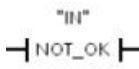
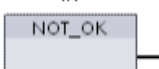
Параметр	Тип данных ¹	Описание
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Constant	Входы инструкции сравнения

¹ Входные параметры MIN, VAL, и MAX должны иметь один тип данных.

- Результат инструкции IN_RANGE равен TRUE, когда: MIN <= VAL <= MAX
- Результат инструкции OUT_RANGE равен TRUE, когда: VAL < MIN или VAL > MAX

8.4.3. Инструкции ОК (Проверить на достоверность) и NOT_OK (Проверить на недостоверность)

Таблица 8- 33 Инструкции ОК (проверить на достоверность) и Not OK (проверить на недостоверность)

LAD	FBD	SCL	Описание
		Недоступно	Проверяет вещественное значение на входе на соответствие IEEE спецификации 754.
		Недоступно	

¹ Для LAD и FBD: Когда контакт LAD имеет состояние TRUE, контакт активируется и реализует прохождение сигнала. Когда FBD блок имеет состояние TRUE, выход блока имеет состояние TRUE.

Таблица 8- 34 Типы данных параметров

Параметр	Тип данных	Описание
IN	Real, LReal	Входные данные

Таблица 8- 35 Выполнение

Инструкция	Результат проверки вещественного равен TRUE, если:
OK	Входное значение соответствует формату вещественного числа ¹
NOT_OK	Входное значение не соответствует формату вещественного числа ¹

¹ Значение с типом данных Real или LReal не соответствует формату, если оно равно +/- INF (бесконечность), NaN (Не является числовым значением) или является денормализованной величиной. Денормализованная величина – это значение близкое к нулевому. При этом, в дальнейших вычислениях ЦПУ заменит это значение нулем.

8.4.4. Инструкции сравнения типов данных Variant и Array

8.4.4.1 Инструкции проверки равенства и неравенства

ЦПУ S7-1200 предлагает инструкции для запросов на равенство или неравенство типа данных тега, на который указывает операнд Variant типу данных другого операнда.

Кроме того, ЦПУ S7-1200 предлагает инструкции для запросов на равенство или неравенство типа данных элемента массива типу данных другого операнда.

В этих инструкциях Вы сравниваете <Operand1> с <Operand2>. <Operand1> должен иметь тип данных Variant. <Operand2> может иметь элементарный тип данных ПЛК. В LAD и FBD, <Operand1> указывается над инструкцией. В LAD и FBD, <Operand2> указывается под инструкцией.

Для всех инструкций результат логической операции (RLO) равен 1 (истина), если проверка на равенство или неравенство истинна, и 0 (ложь) если нет.

Существуют следующие инструкции сравнения на равенство и неравенство:

Таблица 8- 36 Инструкции
EQ_Type (Сравнение типа данных на СООТВЕТСТВИЕ типу данных тега)
NE_Type (Сравнение типа данных на НЕСООТВЕТСТВИЕ типу данных тега)
EQ_ElemType (Сравнение типа данных элемента массива на СООТВЕТСТВИЕ типу данных тега)
NE_ElemType (Сравнение типа данных элемента массива на НЕСООТВЕТСТВИЕ типу данных тега)

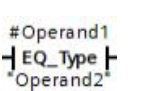
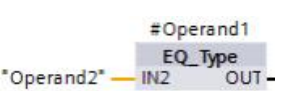
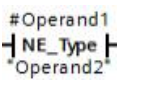
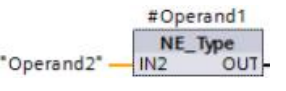
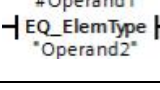
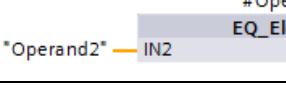
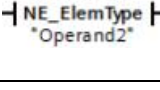
LAD	FBD	SCL	Описание
		Недоступно	Проверяет, имеет ли тег, на который указывает Variant в Operand1, тот же тип данных, что и тег в Operand2.
		Недоступно	Проверяет, имеет ли тег, на который указывает Variant в Operand1, тип данных отличный от типа данных тега в Operand2.
		Недоступно	Проверяет, имеет ли элемент массива, на который указывает Variant в Operand1, тот же тип данных, что и тег в Operand2.
		Недоступно	Проверяет, имеет ли элемент массива, на который указывает Variant в Operand1, тип данных отличный от типа данных тега в Operand2.

Таблица 8- 37 Типы данных параметров

Параметр	Тип данных	Описание
Operand1	Variant	Первый операнд
Operand2	Битовые строки, целые числа, числа с плавающей точкой, таймеры, дата и время, символьные строки, массивы, ПЛК типы данных.	Второй операнд

8.4.4.2 Функции сравнения с нулевым значением

Вы можете использовать инструкции IS_NULL и NOT_NULL, чтобы определить указывает ли значение на входе на объект или нет.

Для обеих инструкций, <Operand> должен иметь тип Variant.

Имеются следующие инструкции сравнения с нулем :

Таблица 8- 38 Инструкции
IS_NULL (Проверка указателя на РАВНО НУЛЮ)
NOT_NULL (Проверка указателя на НЕ РАВНО НУЛЮ)

LAD	FBD	SCL	Описание
#Operand └─ IS_NULL ─┘	#Operand IS_NULL OUT ─	Недоступно	Проверяет, является ли тег, на который указывает тип Variant в параметре Operand, нулем и поэтому не является объектом.
#Operand └─ NOT_NULL ─┘	#Operand NOT_NULL OUT ─	Недоступно	Проверяет, является ли тег, на который указывает тип Variant в параметре Operand, не нулем и поэтому является объектом.

Таблица 8- 39 Типы данных параметров

Параметр	Тип данных	Описание
Операнд	Variant	Операнд для оценки равенства или неравенства нулю

8.4.4.3 IS_ARRAY (Проверка на массив)

Вы можете использовать инструкцию "Check for ARRAY" , чтобы запросить, указывает ли Variant на тег с типом данных Array.

<Operand> должен иметь тип данных Variant.

Инструкция возвращает 1 (истина), если операнд является массивом.

Таблица 8- 40 IS_ARRAY (Проверка на тип данных ARRAY)

LAD	FBD	SCL	Описание
#Operand └─ IS_ARRAY ─┘	#Operand IS_ARRAY OUT ─	IS_ARRAY(_variant_in_)	Проверяет, является ли тег, на который указывает тип данных Variant массивом.

Таблица 8- 41 Типы данных параметров

Параметр	Тип данных	Описание
Операнд	Variant	Операнд для оценки соответствия типу данных массив.

8.5 Арифметические функции

8.5.1 Инструкция CALCULATE (вычисление)

Таблица 8- 42 Инструкция CALCULATE

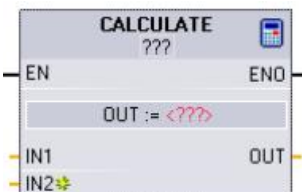
LAD / FBD	SCL	Описание
	Используйте стандартные SCL математические выражения для создания уравнения.	Инструкция CALCULATE позволяет Вам создать арифметическую функцию, работающую со входам (IN1, IN2, .. INn) и возвращающую результат в OUT, согласно уравнению, которое Вы определяете. - Сначала необходимо выбрать тип данных. Все входы и выход должны иметь одинаковый тип данных. - Для добавления нового входа, нажмите на пиктограмме последнего входа.

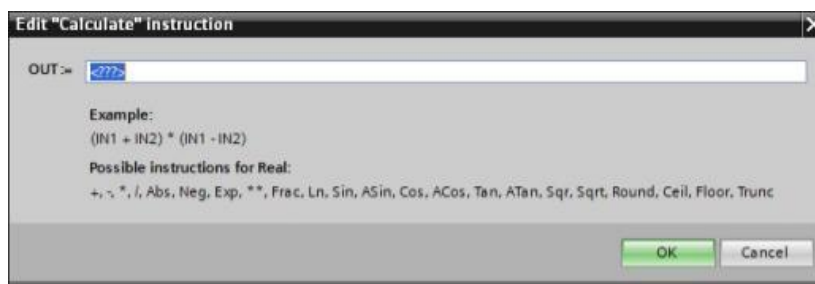
Таблица 8- 43 Типы данных параметров

Параметр	Тип данных ¹
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord

¹ Параметры IN и OUT должны иметь одинаковый тип данных (с неявными преобразованиями входных параметров). Например: Значение с типом SINT на входе будет преобразовано в INT или REAL если OUT имеет тип INT или REAL соответственно.

Щелкните по значку калькулятора, чтобы открыть диалоговое окно и определить Вашу математическую функцию. Вы вводите свое уравнение, сопрягая входы (такие как IN1 и IN2) и операции. Когда Вы нажимаете "OK", чтобы сохранить функцию, диалоговое окно автоматически создает входы для инструкции CALCULATE.

Диалоговое окно показывает пример и список возможных инструкций, которые Вы можете использовать на основе типа данных параметра OUT:



Примечание

Вы также должны создать вход для любых констант в Вашей функции. Значение константы будет указываться на соответствующем входе инструкции CALCULATE.

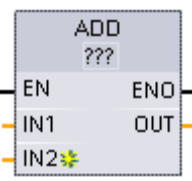
Назначив константы в качестве входов, Вы можете скопировать инструкцию CALCULATE в другие места Вашей программы без необходимости изменения функции. После этого, Вы можете изменить значения или теги на входах инструкции без модификации самой функции.

При выполнении CALCULATE и успешном завершении всех отдельных операций во время вычисления $ENO = 1$. В противном случае $ENO = 0$.

Пример использования инструкции CALCULATE, Вы можете найти в разделе "Создание сложного уравнения с помощью простой инструкции (стр. 39)".

8.5.2 Инструкции сложения, вычитания, умножения и деления

Таблица 8- 44 Инструкции сложения, вычитания, умножения и деления

LAD / FBD	SCL	Описание
	<pre> out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2; </pre>	• ADD: Сложение ($IN1 + IN2 = OUT$) • SUB: Вычитание ($IN1 - IN2 = OUT$) • MUL: Умножение ($IN1 * IN2 = OUT$) • DIV: Деление ($IN1 / IN2 = OUT$) Операция целочисленного деления отбрасывает дробную часть частного, чтобы сформировать целое число на выходе.

¹ В LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 45 Типы данных параметров (LAD и FBD)

Параметр	Тип данных ¹	Описание
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Constant	Входы арифметической инструкции
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Выход арифметической инструкции

¹ Параметры IN1, IN2, и OUT должны иметь одинаковый тип данных.



Для добавления нового входа к инструкции ADD или MUL, щелкните левой клавишей мыши на пиктограмме "Create" (Создать) или правой клавишей мыши на одном из входных параметров, после чего выберите "Insert input".

Для удаления входа, нажмите правой кнопкой мыши на одном из входных параметров (когда имеется более чем 2 собственных входа данной инструкции) и выберите "Delete" (Удалить).

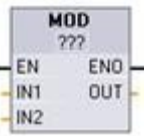
При активации (EN = 1) арифметическая инструкция начинает выполняться над входными значениями (IN1 и IN2) и сохраняет результат в ячейке памяти, заданной выходным параметром (OUT). После успешного выполнения инструкции, ENO= 1.

Таблица 8- 46 Состояние ENO

ENO	Описание
1	Без ошибок
0	Значение результата операции может быть вне действительного диапазона выбранного типа данных. Возвращается младшая значащая часть результата, которая помещается в целевой тип.
0	Деление на 0 (IN2 = 0): Результат не определен, и возвращен нуль.
0	Real/LReal: Если одно из входных значений представляет собой NaN (не число), тогда возвращается NaN.
0	ADD Real/LReal: Если оба значения IN представляют собой INT с различными знаками, то это недопустимая операция и возвращается NaN.
0	SUB Real/LReal: Если оба значения IN представляют собой INT с различными знаками, то это недопустимая операция и возвращается NaN.
0	MUL Real/LReal: Если одно значение IN равно нулю, а другое INT, то это недопустимая операция и возвращается NaN.
0	DIV Real/LReal: Если оба значение IN равны нулю или INT, то это недопустимая операция и возвращается NaN.

8.5.3 Инструкция MOD (возвращает остаток от деления)

Таблица 8- 47 Инструкция получения остатка (возвращает остаток от деления)

LAD / FBD	SCL	Описание
	<code>out := in1 MOD in2;</code>	Вы можете использовать инструкцию MOD, чтобы вернуть остаток операции целочисленного деления. Значение на входе IN1 делится на значение входа IN2, и остаток возвращается на выходе OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 48 Типы данных параметров

Параметр	Тип данных ¹	Описание
IN1 и IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Constant	Входы инструкции получения остатка.
OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Выход инструкции получения остатка.

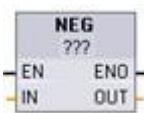
¹ Параметры IN1, IN2, и OUT должны иметь одинаковый тип данных.

Таблица 8- 49 Значения ENO

ENO	Описание
1	Без ошибок
0	IN2 = 0, на выход OUT будет записано нулевое значение

8.5.4 Инструкция NEG (Создать двоичное дополнение)

Таблица 8- 50 Инструкция NEG (Создать двоичное дополнение)

LAD / FBD	SCL	Описание
	-(in);	Инструкция NEG инвертирует арифметический знак переменной на входе IN и сохраняет результат в параметр OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 51 Типы данных параметров

Параметр	Тип данных ¹	Описание
IN	SInt, Int, DInt, Real, LReal, Constant	Вход арифметической инструкции
OUT	SInt, Int, DInt, Real, LReal	Выход арифметической инструкции

¹ Параметры IN и OUT должны иметь одинаковый тип данных.

Таблица 8- 52 Состояние ENO

ENO	Описание
1	Без ошибок
0	Результат выходит за диапазон допустимых значений выбранного типа данных. Пример для SInt: Результатом NEG (-128) является +128, что превышает максимум типа данных.

8.5.5 Инструкции INC (Инкремент) и DEC (Декремент)

Таблица 8- 53 Инструкции INC и DEC

LAD / FBD	SCL	Описание
	<code>in_out := in_out + 1;</code>	Увеличивает на единицу целое число со знаком или без знака: IN_OUT значение +1 = IN_OUT значение
	<code>in_out := in_out - 1;</code>	Уменьшает на единицу целое число со знаком или без знака: IN_OUT значение - 1 = IN_OUT значение

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 54 Типы данных параметров

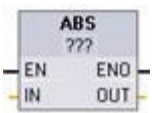
Параметр	Тип данных	Описание
IN/OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Вход и выход арифметической инструкции

Таблица 8- 55 Состояние ENO

ENO	Описание
1	Без ошибок
0	Результат выходит за диапазон допустимых значений используемого типа данных. Пример для SInt: Результатом INC (+127) является +128, что превышает максимум типа данных.

8.5.6 Инструкция ABS (Получить абсолютное значение)

Таблица 8- 56 Инструкция ABS (абсолютное значение)

LAD / FBD	SCL	Описание
	<code>out := ABS(in);</code>	Вычисляет абсолютное значение целого числа со знаком или вещественного числа на входе IN и сохраняет результат в параметре OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 54 Типы данных параметров

Параметр	Тип данных ¹	Описание
IN	SInt, Int, DInt, Real, LReal	Вход арифметической инструкции
OUT	SInt, Int, DInt, Real, LReal	Выход арифметической инструкции

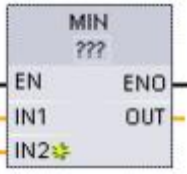
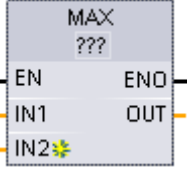
¹ Параметры IN и OUT должны иметь одинаковый тип данных.

Таблица 8- 58 Состояние ENO

ENO	Описание
1	Без ошибок
0	Результат выходит за диапазон допустимых значений используемого типа данных. Пример для SInt: Результатом ABS (-128) является +128, что превышает максимум типа данных.

8.5.7 Инструкции MIN (Вычислить минимум) и MAX (Вычислить максимум)

Таблица 8- 59 Инструкции MIN (вычислить минимум) и MAX (вычислить максимум)

LAD / FBD	SCL	Описание
	<pre>out:= MIN(in1:=_variant_in_, in2:=_variant_in_ [, ...in32]);</pre>	Инструкция MIN сравнивает значение двух параметров IN1 и IN2, после чего записывает минимальное значение в параметр OUT.
	<pre>out:= MAX(in1:=_variant_in_, in2:=_variant_in_ [, ...in32]);</pre>	Инструкция MAX сравнивает значение двух параметров IN1 и IN2, после чего записывает максимальное значение в параметр OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 60 Типы данных параметров

Параметр	Тип данных ¹	Описание
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD, Constant	Входы арифметической инструкции (до 32)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Выход арифметической инструкции

¹ Параметры IN1, IN2, и OUT должны иметь одинаковый типу данных.



Для добавления нового входа щелкните левой клавишей мыши на пиктограмме "Create" (Создать) или правой клавишей мыши на одном из входных параметров, после чего выберите "Insert input".

Для удаления входа, нажмите правой кнопкой мыши на одном из уже существующих входов, после чего нажмите "Delete" (Удалить).

Таблица 8- 61 Состояние ENO

ENO	Описание
1	Без ошибок
0	Только для типа данных REAL: - По крайней мере один из входов не является вещественным (REAL) числом (NaN). - Результат на выходе OUT +/- INF (бесконечность).

8.5.8 Инструкция LIMIT (Задать предел)

Таблица 8- 62 Инструкция LIMIT (задать предел)

LAD / FBD	SCL	Описание
	<pre>LIMIT(MN:=_variant_in_, IN:=_variant_in_, MX:=_variant_in_, OUT:=_variant_out_);</pre>	Инструкция LIMIT проверяет значение в параметре IN на принадлежность диапазону, заданному на входах MIN и MAX, и если значение выходит за диапазон, то она фиксирует значение равным входу MIN или MAX.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 63 Типы данных параметров

Параметр	Тип данных ¹	Описание
MN, IN, и MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD·Constant	Входы арифметической инструкции
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Выход арифметической инструкции

¹ Параметры MN, IN, MX, и OUT должны быть одного типа данных.

Если значение параметра IN принадлежит заданному диапазону, то оно записывается в параметр OUT. Если значение параметра IN не принадлежит заданному диапазону, то в параметр OUT записывается значение параметра MIN (если значение в IN меньше, чем значение MIN) или записывается значение параметра MAX (если значение в IN больше чем значение MAX).

Таблица 8- 64 Состояние ENO

ENO	Описание
1	Без ошибок
0	Real: Если одно или несколько значений MIN, IN и MAX представляют собой NaN (не число), тогда возвращается NaN.
0	Если MIN больше чем MAX, значение IN записывается в OUT.

Примеры на языке SCL:

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Результат: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); // Результат: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); // Результат: MyVal = 10

8.5.9 Экспоненциальные, логарифмические и тригонометрические инструкции

Вы можете использовать следующие инструкции при работе с вещественными числами с типом данных Real или LReal:

- SQR: Вычисление квадрата числа ($IN^2 = OUT$)
- SQRT: Квадратный корень из числа ($\sqrt{IN} = OUT$)
- LN: Натуральный логарифм ($LN(IN) = OUT$)
- EXP: Вычисление экспоненты ($e^{IN} = OUT$), по основанию $e = 2.71828182845904523536$
- EXPT: Возведение в степень ($IN1^{IN2} = OUT$)

Параметры IN1 и OUT инструкции EXPT должны всегда иметь одинаковый тип данных (Real или LReal). Вы можете выбрать тип данных для параметра показателя IN2 из многих типов данных.

- FRAC: Получение мантиссы (дробная часть числа с плавающей точкой на входе $IN = OUT$)
- SIN: Вычисление синуса ($\sin(IN \text{ в радианах}) = OUT$)
- ASIN: Вычисление арксинуса ($\arcsin(IN) = OUT$ в радианах), где $\sin(OUT \text{ в радианах}) = IN$
- COS: Вычисление косинуса ($\cos(IN \text{ в радианах}) = OUT$)
- ACOS: Вычисление арккосинуса ($\arccos(IN) = OUT$ в радианах), где $\cos(OUT \text{ в радианах}) = IN$
- TAN: Вычисление тангенса ($\tan(IN \text{ в радианах}) = OUT$)
- ATAN: Вычисление арктангенса ($\arctan(IN) = OUT$ в радианах), где $\tan(OUT \text{ в радианах}) = IN$

Таблица 8- 65 Примеры арифметических инструкций для чисел с плавающей точкой

LAD / FBD	SCL	Описание
	<pre>out := SQR(in);</pre> or <pre>out := in * in;</pre>	Вычисление квадрата числа: $IN^2 = OUT$ Например: Если $IN = 9$, то $OUT = 81$.
	<pre>out := in1 ** in2;</pre>	Обычное возведение в степень: $IN1^{IN2} = OUT$ Например: Если $IN1 = 3$ и $IN2 = 2$, то $OUT = 9$.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

² Для SCL: Вы можете использовать обычную арифметическую запись для создания арифметических выражений.

Таблица 8- 66 Типы данных параметров

Параметр	Тип данных	Описание
IN, IN1	Real, LReal, Constant	Входы
IN2	SInt, Int, DInt, USInt, UInt,UDInt, Real, LReal, Constant	Вход для инструкции EXPT
OUT	Real, LReal	Выходы

Таблица 8- 67 Состояние ENO

ENO	Инструкция	Условие	Результат (OUT)
1	Все	Без ошибок	Верный результат
0	SQR	Результат превышает допустимый диапазон Real/LReal	+INF
		IN = +/- NaN (не является числом)	+NaN
	SQRT	IN отрицательный	-NaN
		IN = +/- INF (бесконечность) или +/- NaN	+/- INF или +/- NaN
	LN	IN = 0.0, отрицательный, -INF, или -NaN	-NaN
		IN = +INF или +NaN	+INF или +NaN
	EXP	Результат превышает допустимый диапазон Real/LReal	+INF
		IN = +/- NaN	+/- NaN
	SIN, COS, TAN	IN = +/- INF или +/- NaN	+/- INF или +/- NaN
	ASIN, ACOS	IN вне диапазона (-1.0 : +1.0)	+NaN
		IN = +/- NaN	+/- NaN
	ATAN	IN = +/- NaN	+/- NaN
	FRAC	IN = +/- INF или +/- NaN	+NaN
	EXPT	IN1 = +INF и IN2 не -INF	+INF
		IN1 отрицательный или -INF	+NaN если IN2 - Real/LReal, -INF в противном случае
		IN1 или IN2 = +/- NaN	+NaN
		IN1 = 0.0 и IN2 с типом Real/LReal (только)	+NaN

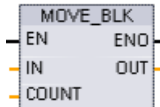
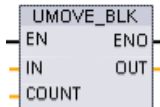
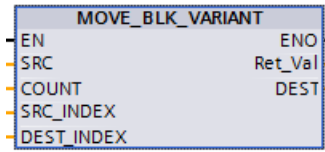
8.6 Операции пересылки

8.6.1. Инструкции MOVE (переслать значение), MOVE_BLK (переслать блок), UMOVE_BLK (переслать блок без прерываний), и MOVE_BLK_VARIANT (переслать блок)

Используйте инструкции пересылки для копирования элементов данных в новую область памяти и конвертации из одного типа данных в другой. При этом, исходные данные не меняются.

- Инструкция MOVE копирует один элемент данных из адреса источника, заданного параметром IN в адрес приемника, назначенный параметром OUT.
- У инструкций MOVE_BLK и UMOVE_BLK имеется еще один дополнительный параметр COUNT. COUNT определяет количество данных, которое необходимо скопировать. Количество скопированных байт на один элемент зависит от типа данных, назначенного тегам параметров IN и OUT в таблице тегов ПЛК.

Таблица 8- 68 Инструкции MOVE, MOVE_BLK, UMOVE_BLK, и MOVE_BLK_VARIANT

LAD / FBD	SCL	Описание
	<code>out1 := in;</code>	Копирует элемент данных, хранящийся в назначенном адресе, в новый адрес или множество адресов. ¹
	<code>MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</code>	Пересылка с возможностью прерывания, которая копирует блок с элементами данных в новый адрес.
	<code>UMOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</code>	Непрерывная пересылка, которая копирует блок с элементами данных в новый адрес.
	<code>MOVE_BLK(SRC:=_variant_in, COUNT:=_udint_in, SRC_INDEX:=_dint_in, DEST_INDEX:=_dint_in, DEST=>_variant_out);</code>	Пересылает содержимое области памяти источника в область памяти приемника. Вы можете скопировать весь массив или элементы массива в другой массив с тем же типом данных. Размер (количество элементов) источника и приемника при этом может различаться. Вы можете скопировать несколько или один элемент внутри массива. Вы можете использовать типы данных Variant для указания на массивы источника и приемника.

¹ Инструкция MOVE: Для добавления еще одного выхода, в LAD или FBD, нажмите на пиктограмму "Create" (Создать) у выходного параметра. В SCL, используйте выражение множественного присвоения. Вы могли бы также использовать одну из конструкций цикла.

Таблица 8- 69 Типы данных инструкции MOVE

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC типы данных, PLC типы данных	Адрес источника
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC типы данных, PLC типы данных	Адрес приемника



Чтобы добавить выходы инструкции MOVE, щелкните левой клавишей мыши на пиктограмме "Create" (Создать) или правой клавишей мыши на одном из существующих выходных контактов OUT и выберите команду "Insert output" (Вставить выход).

Для удаления выхода щелкните правой кнопкой мыши на одном из существующих выходных контактов OUT (если имеются уже созданные Вами) и воспользуйтесь командой "Delete" (Удалить).

Таблица 8- 70 Типы данных инструкций MOVE_BLK и UMOVE_BLK

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Начальный адрес источника
COUNT	UInt	Количество копируемых данных
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Адрес приемника

Таблица 8- 71 Типы данных инструкции MOVE_BLK_VARIANT

Параметр	Тип данных	Описание
SRC	Variant (тип который указывает на массив или его элементы)	Блок из которого будет производится копирование данных
COUNT	UDInt	Количество копируемых данных
SRC_INDEX	DInt	Индекс относительно нуля в массиве SRC
DEST_INDEX	DInt	Индекс относительно нуля в массиве DEST
RET_VAL	Int	Информация об ошибке
DEST	Variant (тип который указывает на массив или его элементы)	Область приемника, куда будет производится копирование содержимого блока

Примечание**Необходимо соблюдать следующие правила при копировании данных**

- Чтобы скопировать тип данных Bool, используйте SET_BF, RESET_BF, R, S, или выходную катушку (LAD) (стр. 212)
- Чтобы скопировать одиночные элементарные типы данных, используйте MOVE
- Чтобы скопировать массив, элементы которого имеют элементарный тип данных используйте MOVE_BLK или UMOVE_BLK
- Чтобы скопировать структуру, используйте MOVE
- Чтобы скопировать строку, используйте S_MOVE (стр. 326)
- Чтобы скопировать отдельный символ в строку, используйте MOVE
- Инструкции MOVE_BLK и UMOVE_BLK не могут быть использованы для копирования массивов или структур в I, Q, или M области памяти.

Инструкции MOVE_BLK и UMOVE_BLK отличаются методом обработки прерываний:

- В процессе выполнения инструкции MOVE_BLK события прерываний **помещаются в очередь и обрабатываются**. Используйте инструкцию MOVE_BLK, если данные в области приемника не используются внутри подпрограммы OB прерывания или, в случае использования, целевые данные не обязаны быть согласованными. Если операция MOVE_BLK прерывается, то последний элемент данных пересылается и является согласованным в адресе назначения. Операция MOVE_BLK возобновляется после завершения выполнения OB прерывания.
- События прерывания **помещаются в очередь, но не обрабатываются**, пока выполнение UMOVE_BLK не завершено. Используйте инструкцию UMOVE_BLK, когда операция пересылки должна быть завершена и целевые данные должны быть согласованными перед выполнением подпрограммы OB прерывания. Для получения дополнительной информации смотрите раздел по согласованности данных (стр. 185).

ENO всегда активирован при выполнении инструкции MOVE.

Таблица 8- 72 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Все элементы в количестве COUNT были успешно скопированы.
0	Либо диапазон источника (IN), либо приемника (OUT) превысил доступную область памяти.	Элементы, которые могут быть размещены копируются. Неполные элементы не копируются.

Таблица 8- 73 Коды ошибки инструкции MOVE_BLK_VARIANT

RET_VAL (W#16#...)	Описание
0000	Без ошибок
80B4	Несоответствие типов данных.
8151	Доступ к параметру SRC невозможен.
8152	Операнд в параметре SRC имеет неверный тип данных.
8153	Ошибка генерации кода в параметре SRC
8154	Операнд в параметре SRC имеет тип данных Bool.
8281	Неверное значение параметра COUNT.

RET_VAL (W#16#...)	Описание
8382	Значение в параметре SRC_INDEX не входит в диапазон допустимых значений типа данных Variant.
8383	Значение в параметре SRC_INDEX превышает максимальный размер массива.
8482	Значение в параметре DEST_INDEX не входит в диапазон допустимых значений типа данных Variant.
8483	Значение в параметре DEST_INDEX превышает максимальный размер массива.
8534	Параметр DEST защищен от записи.
8551	Доступ к параметру DEST невозможен.
8552	Неверный тип данных операнда в параметре DEST.
8553	Ошибка генерации кода в параметре DEST.
8554	Операнд в параметре DEST имеет тип данных Bool.
*Вы можете отобразить коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.	

8.6.2. Инструкция Deserialize

Вы можете воспользоваться инструкцией "Deserialize", чтобы выполнить обратное преобразование последовательного представления ПЛК типа данных (UDT) в ПЛК тип данных и заполнить его внутреннее содержимое. Если сравнение дает результат TRUE, то выход блока принимает значение TRUE.

Область памяти, в которой находится последовательное представление ПЛК типа данных, должна иметь тип данных Array of Byte и Вы должны объявить блок данных со стандартным доступом (неоптимизированный). Убедитесь, что имеется достаточное количество памяти до выполнения преобразования.

Инструкция позволяет Вам преобразовать несколько последовательные представления преобразованных ПЛК типов данных назад в их исходные типы данных.

Примечание

Если Вы хотите выполнить обратное преобразование только одного последовательного представления ПЛК типа данных (UDT), Вы можете также использовать инструкцию "TRCV: Получить данные через коммуникационное соединение".

Таблица 8- 74 Инструкция DESERIALIZE

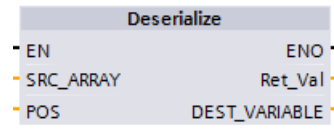
LAD / FBD	SCL	Описание
	<pre>ret_val := Deserialize(SRC_ARRAY:=_variant_in_, DEST_VARIABLE=>_variant_out); POS:=_dint_inout_;</pre>	Выполняет обратное преобразование последовательного представления ПЛК типа данных (UDT) в ПЛК тип данных и заполняет внутреннее содержимое

Таблица 8- 75 Параметры инструкции DESERIALIZE

Параметры	Тип	Тип данных	Описание
SRC_ARRAY	IN	Variant	Глобальный блок данных, в котором содержится поток данных
DEST_VARIABLE	INOUT	Variant	Тег в который сохраняется преобразованный ПЛК тип данных (UDT)
POS	INOUT	DInt	Количество байт, которые использует преобразованный ПЛК тип данных
RET_VAL	OUT	Int	Информация об ошибке

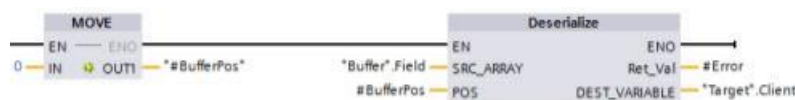
Таблица 8- 76 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Без ошибок
80B0	Перекрытие областей памяти для параметров SRC_ARRAY и DEST_VARIABLE.
8136	Блок данных в параметре DEST_VARIABLE не является блоком со стандартным типом доступа.
8150	Нет данных в фактическом параметре SRC_ARRAY.
8151	Ошибка генерации кода в параметре SRC_ARRAY.
8153	Недостаточно свободной памяти в параметре SRC_ARRAY.
8250	Нет данных в фактическом параметре DEST_VARIABLE.
8251	Ошибка генерации кода в параметре DEST_VARIABLE.
8254	Неверный тип данных в параметре DEST_VARIABLE.
8382	Значение в параметре POS превышает размер массива.
* Вы можете отобразить коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.	

Пример: Инструкция Deserialize

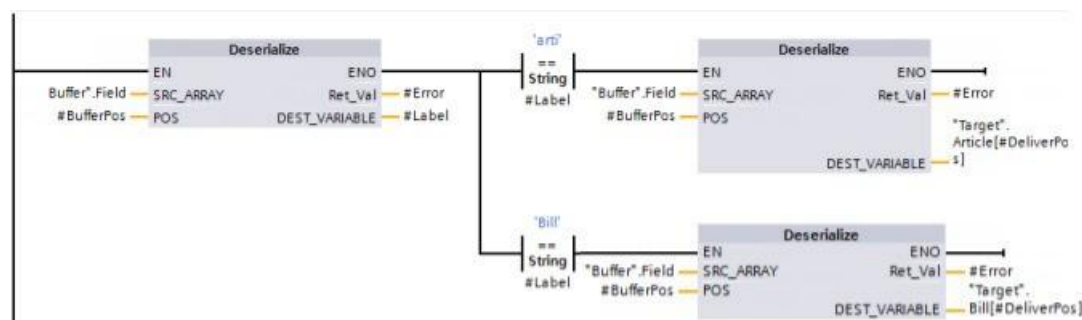
В следующем примере показан принцип работы данной инструкции:

Сегмент 1:



Инструкция "MOVE" перемещает значение "0" в тег блока данных "#BufferPos". Инструкция Deserialize затем выполняет преобразование последовательного представления пользовательских данных из блока данных "Buffer" и записывает его в блок данных "Target". Инструкция Deserialize вычисляет количество байтов, которое используется для преобразованных данных и сохраняет его в теге блока данных "#BufferPos".

Сегмент 2:



Инструкция "Deserialize" выполняет преобразование последовательного представления потока данных, на который указывает "Buffer" и записывает символы в операнд "#Label". Логика сравнивает символы, используя инструкции сравнения "arti" и "Bill". Если сравнение для "arti" = TRUE, данные являются статьей и должны быть преобразованы и записаны в структуру данных "Article" блока данных "Target". Если сравнение для "Bill" = TRUE, то это данные счета, которые должны быть преобразованы и записаны в структуру данных "Bill" блока данных "Target".

Интерфейс функционального блока (или функции):

Name	Data type
▼ Input	
DeliverPos	Int
► Output	
InOut	
► Static	
▼ Temp	
BufferPos	DInt
Error	Int
Label	String[4]

Пользовательские ПЛК типы данных:

Ниже показаны два ПЛК типа данных (UDT) для настоящего примера:

Article		
	Name	Data type
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Name	Data type
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Блоки данных:

Ниже отображены соответствующие блоки данных для настоящего примера:

Target		
	Name	Data type
1	Static	
2	Client	*Client*
3	Article	Array[0..10] of *Article*
4	Bill	Array[0..10] of Int

Buffer		
	Name	Data type
1	Static	
2	Field	Array[0..294] of Byte

8.6.3. Инструкция Serialize

Вы можете использовать инструкцию "Serialize" для преобразования нескольких ПЛК типов данных (UDT) в последовательное представление без потери структуры.

Вы также можете использовать данную инструкцию для временного сохранения нескольких структурированных элементов данных из Вашей программы в буфер, например в глобальный блок данных, и отправить их в другой ЦПУ. В области памяти, в которой сохраняются преобразованные ПЛК типы данных, должен иметь тип данных ARRAY of BYTE со стандартным типом доступа. Убедитесь, что имеется достаточное количество памяти для выполнения преобразования.

Параметр POS содержит информацию о количестве байтов, которое используют преобразованные ПЛК типы данных.

Примечание

Если Вы хотите передать только один ПЛК тип данных (UDT), Вы можете использовать инструкцию " TSEND: Передать данные через коммуникационное соединение".

Таблица 8- 77 Инструкция SERIALIZE

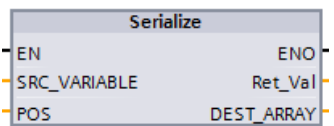
LAD / FBD	SCL	Описание
	<pre>ret_val := Serialize(SRC_VARIABLE=>_variant_in_, DEST_ARRAY:=_variant_out_, POS:=_dint_inout_);</pre>	Преобразует ПЛК тип данных (UDT) в последовательное представление.

Таблица 8- 78 Параметры инструкции SERIALIZE

Параметры	Тип	Тип данных	Описание
SRC_VARIABLE	IN	Variant	ПЛК тип данных (UDT), который будет преобразован в последовательное представление.
DEST_ARRAY	INOUT	Variant	Блок данных в который сохраняется сгенерированный поток данных.
POS	INOUT	DInt	Количество байт, которое используется преобразованными ПЛК типами данных. Параметр POS рассчитывается относительно нуля.
RET_VAL	OUT	Int	Информация об ошибке.

Таблица 8- 79 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Без ошибок
80B0	Перекрытие областей памяти для параметров SRC_VARIABLE и DEST_ARRAY.
8150	Нет данных в фактическом параметре SRC_VARIABLE.
8152	Ошибка генерации кода в параметре SRC_VARIABLE.
8236	Блок данных в параметре DEST_ARRAY не является блоком со стандартным типом доступа.
8250	Нет данных в фактическом параметре DEST_ARRAY.
8252	Ошибка генерации кода в параметре DEST_ARRAY.
8253	Недостаточно свободной памяти в параметре DEST_ARRAY.
8254	Неверный тип данных в параметре DEST_VARIABLE.
8382	Значение в параметре POS превышает размер массива.

* Вы можете просмотреть коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.

Пример: Инструкция Serialize

В следующем примере показан принцип работы данной инструкции:

Сегмент 1:



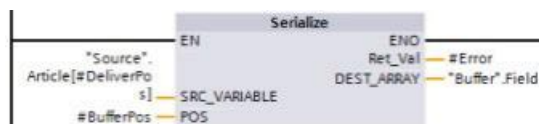
Инструкция "MOVE" пересылает значение "0" в параметр "#BufferPos". Инструкция "Serialize" преобразует пользовательские данные из блока данных "Source" и записывает их в последовательное представление в блок данных "Buffer". Инструкция сохраняет количество байтов, используемых последовательным представлением в параметре "#BufferPos".

Сегмент 2:



Далее логика вставляет некоторый текст разделителя, чтобы позднее упростить преобразование последовательного представления. Инструкция "S_MOVE" пересылает текстовую строку "arti" в параметр "#Label". Инструкция "Serialize" записывает эти символы после исходных клиентских данных в блок данных "Buffer". Инструкция добавляет количество байтов в текстовой строке "arti" к количеству, уже сохраненному в параметре "#BufferPos".

Сегмент 3:



Инструкция "Serialize" преобразует данные определенной статьи, которая вычисляется во времени выполнения, из блока данных "Source" и записывает его в последовательное представление в блок данных "Buffer" после символов "arti"

Интерфейс блока:

	Name	Data type
▼	Input	
■	DeliverPos	Int
▶	Output	
▶	InOut	
▶	Static	
▼	Temp	
■	BufferPos	DInt
■	Error	Int
■	Label	String[4]

Пользовательские ПЛК типы данных:

Ниже показаны две структуры ПЛК типов данных (UDT) для настоящего примера:

Article		
	Name	Data type
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Name	Data type
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Блоки данных:

Ниже отображены соответствующие блоки данных для настоящего примера:

Source		
	Name	Data type
1	▼ Static	
2	■ ▶ Client	"Client"
3	■ ▶ Article	Array[0..10] of "Article"

Buffer		
	Name	Data type
1	▼ Static	
2	■ ▶ Field	Array[0..294] of Byte

8.6.4. Инструкции FILL_BLK (Заполнить блок) и UFILL_BLK (Заполнить блок без прерываний)

Таблица 8- 80 Инструкции FILL_BLK и UFILL_BLK

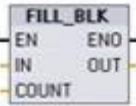
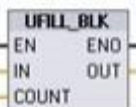
LAD / FBD	SCL	Описание
	FILL_BLK (in:=_variant_in, count:=int, out=>_variant_out);	Инструкция заполнения с возможностью прерывания: Заполняет диапазон адресов копиями назначенных элементов данных
	UFILL_BLK (in:=_variant_in, count:=int out=>_variant_out);	Инструкция заполнения без прерываний: Заполняет диапазон адресов копиями назначенных элементов данных

Таблица 8- 81 Типы данных параметров

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Адрес источника данных.
COUNT	UDInt, USInt, UInt	Количество копируемых элементов данных
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Адрес приемника данных.

Примечание

Правила для операций заполнения данных

- При работе с типом данных BOOL, используйте SET_BF, RESET_BF, R, S, или выходную катушку (LAD)
- Для заполнения одиночным элементарным типом данных, используйте MOVE
- Для заполнения массива элементарным типом данных, используйте FILL_BLK или UFILL_BLK
- Для заполнения одного символа в строке, используйте MOVE
- Инструкции FILL_BLK и UFILL_BLK не могут быть использованы для заполнения массивов в I, Q, или M областях памяти.

Инструкции FILL_BLK и UFILL_BLK копируют элемент данных источника IN в приемник, начиная с адреса, заданного в параметре OUT. Процесс копирования повторяется и блок с назначенными адресами заполняется до тех пор, пока не было достигнуто значение параметра COUNT.

Инструкции FILL_BLK и UFILL_BLK различаются по способу обработки прерываний:

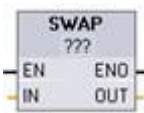
- События прерывания **помещаются в очередь и обрабатываются** во время выполнения FILL_BLK. Используйте инструкцию FILL_BLK, когда данные в адресе назначения пересылки не используются в подпрограмме прерывания OB или, если используется, целевые данные не обязаны быть согласованными.
- События прерывания **помещаются в очередь, но не обрабатываются**, пока выполнение UFILL_BLK не завершено. Используйте инструкцию UFILL_BLK, когда операция пересылки должна быть завершена, и целевые данные должны быть согласованными до выполнением подпрограммы OB прерывания.

Таблица 8- 82 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Элемент IN был успешно скопирован во все COUNT приемники.
0	Диапазон приемника (OUT) превышает доступную область памяти	Элементы, которые могут быть размещены копируются. Неполные элементы не копируются..

8.6.5. Инструкция SWAP (Обмен байтов)

Таблица 8- 83 Инструкция SWAP

LAD / FBD	SCL	Описание
	<code>out := SWAP(in);</code>	Меняет порядок байтов для двух или четырех байтовых элементов данных. При этом не происходят изменения в порядке битов самого байта. При выполнении данной инструкции ENO всегда имеет состояние TRUE.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 84 Типы данных параметров

Параметр	Тип данных	Описание
IN	Word, DWord	Байты данных в прямом порядке (IN)
OUT	Word, DWord	Байты данных в обратном порядке (OUT)

Пример 1	Параметр IN = MB0 (до выполнения)		Параметр IN = MB4 (после выполнения)	
Адрес	MW0	MB1	MW4	MB5
W#16#1234	12	34	34	12
WORD	MSB	LSB	MSB	LSB

Пример 2	Параметр IN = MB0 (до выполнения)				Параметр IN = MB4 (после выполнения)			
Адрес	MD0	MB1	MB2	MB3	MD4	MB5	MB6	MB7
DW#16#12345678	12	34	56	78	78	56	34	12
DWORD	MSB			LSB	MSB			LSB

8.6.6. Инструкции чтения / записи памяти

8.6.6.1 Инструкции PEEK и POKE (только SCL)

В SCL имеются инструкции PEEK и POKE, которые позволяют Вам прочитать или записать информацию из блоков данных, ввода-вывода или памяти. Для этого, Вам необходимо задать параметры байтового или битового смещения.

Примечание

Для использования инструкций PEEK и POKE с блоками данных, Вы должны применять только блоки со стандартным доступом (неоптимизированные). Также обратите внимание на то, что инструкции PEEK и POKE всего лишь передают данные. Они не анализируют типы данных в адресах.

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

Читает байт, на который ссылается byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример ссылки на блок данных:

```
%MB100 := PEEK(area:=16#84,
                dbNumber:=1, byteOffset:=#i);
```

Пример ссылки на IB3:

```
%MB100 := PEEK(area:=16#81,
                dbNumber:=0, byteOffset:=#i); // где
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Читает слово, на которое ссылается byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MW200 := PEEK_WORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

Читает двойное слово, на которое ссылается byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MD300 := PEEK_DWORD(area:=16#84,
                      dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

Читает булево значение, на которое ссылаются bitOffset и byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                       dbNumber:=1, byteOffset:=#ii,
                       bitOffset:=#j);
```


POKE(area:=_in_,
dbNumber:=_in_,
byteOffset:=_in_,
value:=_in_);

Записывает значение (Byte, Word или DWord) по ссылке в byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример ссылки на блок данных:

POKE(area:=16#84, dbNumber:=2,
byteOffset:=3, value:="Tag_1");

Пример ссылки на QB3:

POKE(area:=16#82, dbNumber:=0,
byteOffset:=3, value:="Tag_1");

POKE_BOOL(area:=_in_,
dbNumber:=_in_,
byteOffset:=_in_,
bitOffset:=_in_,
value:=_in_);

Записывает булево значение по ссылкам в bitOffset и byteOffset указанного блока данных, ввода-вывода или области памяти.

Пример:

POKE_BOOL(area:=16#84, dbNumber:=2,
byteOffset:=3, bitOffset:=5, value:=0);

POKE_BLK(area_src:=_in_,
dbNumber_src:=_in_,
byteOffset_src:=_in_,
area_dest:=_in_,
dbNumber_dest:=_in_,
byteOffset_dest:=_in_,
count:=_in_);

Записывает определенное значение "count" количество байтов, начиная со ссылки в byte Offset указанного блока данных, ввода-вывода или области памяти в указанную посредством ссылки в byteOffset область приемника в целевом блоке данных, вводе-выводе или области памяти.

Пример:

POKE_BLK(area_src:=16#84,
dbNumber_src:=#src_db, byteOffset_src:=#src_byte, area_dest:=16#84,
dbNumber_dest:=#src_db, byteOffset_dest:=#src_byte, count:=10);

Для инструкций PEEK и POKE, применимы следующие значения параметров "area", "area_src" и "area_dest". Для областей отличных от блоков данных, параметр dbNumber должен быть равен 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

8.6.6.2 Инструкции чтения и записи big endian и little endian (SCL)

В S7-1200 CPU имеется возможность использования SCL инструкций для чтения и записи данных в формате little endian и big endian. Формат little endian означает, что байт с самым младшим битом будет находиться по наименьшему адресу памяти. Формат big endian означает, что байт с самым старшим битом будет находиться по наименьшему адресу памяти.

Доступны четыре SCL инструкции для чтения и записи данных в формате little endian и big endian:

- READ_LITTLE (Чтение данных в формате little endian)
- WRITE_LITTLE (Запись данных в формате little endian)
- READ_BIG (Чтение данных в формате big endian)
- WRITE_BIG (Запись данных в формате big endian)

Таблица 8- 85 Инструкции чтения и записи big и little endian

LAD / FBD	SCL	Описание
Недоступно	<code>READ_LITTLE(src_array:=_variant_in_, dest_Variable =>_out_, pos:=_dint_inout)</code>	Считывает данные из области памяти и записывает их в одиночный тег в формате little endian.
Недоступно	<code>WRITE_LITTLE(src_variable:=_in_, dest_array =>_variant_inout_, pos:=_dint_inout)</code>	Записывает данные из одиночного тега в область памяти в формате little endian.
Недоступно	<code>READ_BIG(src_array:=_variant_in_, dest_Variable =>_out_, pos:=_dint_inout)</code>	Считывает данные из области памяти и записывает их в одиночный тег в формате big endian.
Недоступно	<code>WRITE_BIG(src_variable:=_in_, dest_array =>_variant_inout_, pos:=_dint_inout)</code>	Записывает данные из одиночного тега в область памяти в формате big endian.

Таблица 8- 86 Параметры инструкций READ_LITTLE и READ_BIG

Параметр	Тип данных	Описание
src_array	Array of Byte	Область памяти из которой производится чтение данных
dest_Variable	Битовые строки, целые числа, вещественные числа, таймеры, дата и время, символьные строки	Переменная в которую записываются данные
pos	DINT	Позиция относительно нуля с которой начинается чтение данных со входа src_array.

Таблица 8- 87 Параметры инструкций WRITE_LITTLE и WRITE_BIG

Параметр	Тип данных	Описание
src_variable	Битовые строки, целые числа, вещественные числа, LDT, TOD, LTOD, DATA, Char, WChar	Исходные данные из тега
dest_array	Array of Byte	Область памяти в которую будут записываться данные
pos	DINT	Позиция относительно нуля с которой начинается запись данных в выход dest_array.

Таблица 8- 88 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Без ошибок
80B4	SRC_ARRAY или DEST_ARRAY имеют тип отличный от Array of Byte
8382	Значение параметра POS превышает размер массива.
8383	Значение параметра POS в пределах допустимых значений массива, но размер области памяти превышает размер массива.
* Вы можете просмотреть коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.	

8.6.7. Инструкции для типа данных Variant

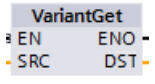
8.6.7.1 VariantGet (Чтение значения тега VARIANT)

Вы можете воспользоваться инструкцией "Read out Variant tag value" для чтения значения тега по указателю Variant в параметре на который ссылается SRC и после этого записать его в тег параметра DST.

Параметр SRC имеет тип данных Variant. Любой тип данных, за исключением Variant, может быть назначен параметру DST.

Тип данных тега в параметре DST должен соответствовать типу данных на который указывает Variant.

Таблица 8- 89 Инструкция VariantGet

LAD / FBD	SCL	Описание
	<pre>VariantGet(SRC:=_variant_in_, DST=>_variant_out_);</pre>	Считывает тег, упомянутый в параметре SRC и записывает его в тег параметра DST

Примечание

Для копирования структур и массивов, Вы можете использовать инструкцию "MOVE_BLK_VARIANT: Переслать блок".

Таблица 8- 90 Параметры инструкции VariantGet

Параметр	Тип данных	Описание
SRC	Variant	Указатель на источник данных
DST	Битовые строки, целые числа, числа с плавающей точкой, таймеры, дата и время, символьные строки, элементы массивов, ПЛК типы данных	Приемник для записи данных

Таблица 8- 91 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Инструкция скопировала тег данных, упомянутый посредством SRC в тег DST.
0	Состояние входа EN равно 0 или типы данных не соответствуют.	Инструкция не скопировала данные.

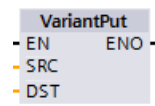
8.6.7.2 Инструкция VariantPut (Запись значения тега VARIANT)

Вы можете использовать инструкцию "Write VARIANT tag value", чтобы записать значение тега в параметре SRC в тегу в параметре DST, на который указывает VARIANT.

Параметр DST имеет тип данных VARIANT. Любой тип данных за исключением VARIANT может быть определен в параметре SRC.

Тип данных тега в параметре SRC должен соответствовать типу данных, на который указывает VARIANT.

Таблица 8- 92 Инструкция VariantPut

LAD / FBD	SCL	Описание
	<pre>VariantPut(SRC:=_variant_in_, DST=>_variant_in_);</pre>	Записывает тег, на который ссылается параметр SRC по ссылке, которая указана в параметре DST.

Примечание

Для копирования структур и массивов, Вы можете использовать инструкцию "MOVE_BLK_VARIANT: Переслать блок".

Таблица 8- 93 Параметры инструкции VariantPut

Параметр	Тип данных	Описание
SRC	Битовые строки, целые числа, числа с плавающей точкой, таймеры, дата и время, символьные строки, элементы массивов, ПЛК типы данных	Указатель на источник данных
DST	Variant	Приемник для записи данных

Таблица 8- 94 Состояние ENO

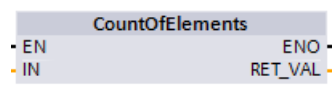
ENO	Условие	Результат
1	Без ошибок	Инструкция скопировала тег данных SRC в тег DST.
0	Состояние входа EN равно 0 или типы данных не соответствуют.	Инструкция не скопировала данные.

8.6.7.3 Инструкция CountOfElements (Количество элементов массива)

Вы можете воспользоваться инструкцией "Get number of ARRAY elements" для подсчета количества элементов массива в теге, на который указывает Variant.

Если массив одномерный, то инструкция вернет разницу между индексом последнего и первого элемента +1. При работе с многомерными массивами, инструкция вернет произведение всех измерений.

Таблица 8- 95 Инструкция CountOfElements

LAD / FBD	SCL	Описание
	<pre>Result := CountOfElements(_variant_in_);</pre>	Подсчитывает количество элементов в массиве, на который указывает параметр IN.

Примечание

Если Variant указывает на Array of Bool, инструкция подсчитывает элементы заполнения до ближайшей границы байта. Например, инструкция вернет 8, для массива Array [0..1] of Bool.

Таблица 8- 96 Параметры инструкции CountOfElements

Параметр	Тип данных	Описание
IN	Variant	Тег, указывающий на массив, элементы которого необходимо подсчитать.
RET_VAL	UDint	Результат инструкции

Таблица 8- 97 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Инструкция возвращает количество элементов массива.
0	Состояние сигнала на входе разрешения EN равно "0", или Variant не указывает на массив.	Инструкция возвращает 0.

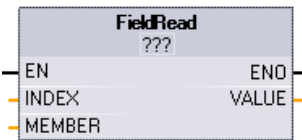
8.6.8. Унаследованные инструкции

8.6.8.1 Инструкции FieldRead (Чтение поля) и FieldWrite (Запись поля)

Примечание

STEP 7 V10.5 **не поддерживает** ссылку на переменную в качестве индекса массива или многомерные массивы. Инструкции FieldRead и FieldWrite использовались, чтобы обеспечить операции с переменным индексом для одномерного массива. STEP 7 V11 и выше **поддерживает** переменную в качестве индекса массива и многомерные массивы. FieldRead и FieldWrite включены в STEP 7 V11 и выше для совместимости с программами предыдущих версий, которые использовали эти инструкции

Таблица 8- 98 Инструкции FieldRead и FieldWrite

LAD / FBD	SCL	Описание
	<pre>value := member[index];</pre>	FieldRead считывает элемент массива с индексом INDEX из массива, первый элемент которого задан в параметре MEMBER. Значение элемента массива далее передается в область, заданную в параметре VALUE.
	<pre>member[index] := value;</pre>	WriteField передает значение из области, заданной в параметре VALUE в массив, у которого первый элемент задан в параметре MEMBER. Значение передается в элемент массива с индексом, заданным в параметре INDEX.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 99 Типы данных параметров

Параметр и тип		Тип данных	Описание
Index	Input	DInt	Индекс элемента массива, который необходимо прочитать или записать
Member ¹	Input	Двоичные числа, целые числа, вещественные числа, таймеры, DATE, TOD, CHAR и WCHAR как элементы тега ARRAY	Местоположение первого элемента в одномерном массиве, определенном в глобальном блоке данных или интерфейсе блока. Например: Если индекс массива назначен как [-2..4], то индекс первого элемента равен -2, а не 0.
Value ¹	Out	Двоичные числа, целые числа, вещественные числа, таймеры, DATE, TOD, CHAR, WCHAR	Местоположение куда будет скопирован заданный элемент массива (FieldRead) Местоположение значения, которое будет скопировано в назначенный элемент массива (FieldWrite)

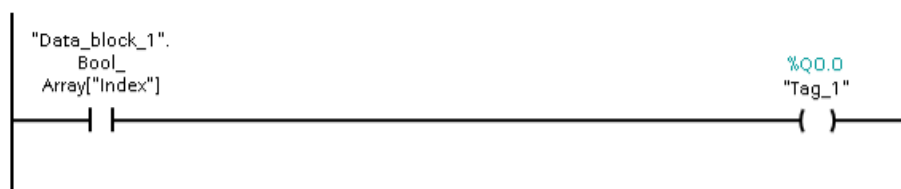
¹ Тип данных элемента массива, заданный в параметрах MEMBER и VALUE, должен быть одинаковым.

Выход ENO = 0, если удовлетворено одно из следующих условий:

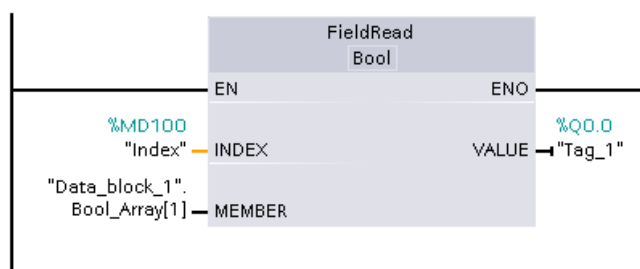
- Вход EN имеет состояние "0"
- Элемент массива, заданный в параметре INDEX, не определен в массиве, на который указывает параметр MEMBER
- В процессе работы возникли ошибки, например переполнение

Пример: Доступ к данным посредством индексации массива

Для получения доступа к элементам массива через переменную, достаточно использовать переменную в качестве индекса массива в Вашей программе. Например, в сегменте ниже выход устанавливается согласно булевому значению из массива в "Data_block_1", определяемому посредством ПЛК тега "Index".



Принцип схож с работой инструкции FieldRead:



Инструкции FieldWrite и FieldRead могут быть заменены логикой переменной индексации массива.

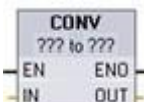
В SCL отсутствуют инструкции FieldRead или FieldWrite, но имеется косвенная адресация массива с помощью переменной:

```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```


8.7 Операции преобразования

8.7.1. Инструкция CONV (Преобразовать значение)

Таблица 8- 100 Инструкция Convert (CONV)

LAD / FBD	SCL	Описание
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	Преобразует элемент данных из одного типа данных в другой.

- Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.
- Для SCL: Вы можете преобразовать один тип в другой, определив один тип для входного параметра (in) и другой тип для выходного параметр (out). Например, DWORD_TO_REAL преобразует значение DWord в Real.

Таблица 8- 101 Типы данных параметров

Параметр	Тип данных	Описание
IN	Битовые строки ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Значение на входе
OUT	Битовые строки ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Преобразованное значение на выходе

- Инструкция не разрешит использование битовых строк (Byte, Word, DWord). Для выбора типа данных Byte, Word, или DWord, выберите в качестве типа данных целое число без знака с соответствующей длиной. Например, выберите USInt для Byte, UInt для Word, или UDInt для DWord.

После того, как Вы выберете тип данных из которого необходимо выполнить преобразование, в выпадающем списке Вы увидите варианты возможного выбора типа для преобразования. Преобразование из/в BCD16 ограничено типом данных Int. Преобразование из/в BCD32 ограничено типом данных DInt.

Таблица 8- 102 Состояние ENO

ENO	Описание	Результат
1	Без ошибок	Верный результат
0	Значение в IN = +/- INF или +/- NaN	+/- INF или +/- NaN
0	Результат превышает допустимый диапазон значений для типа данных OUT	В OUT записывается значение IN

8.7.2. Инструкции преобразования для SCL

Инструкции преобразования в SCL

Таблица 8- 103 Преобразование из Bool, Byte, Word или DWord

Тип данных	Инструкция	Результат
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	Значение передается в младший значащий бит целого типа данных
Byte	BYTE_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	BYTE_TO_WORD, BYTE_TO_DWORD	Значение передается в младший значащий байт целевого типа данных.
	BYTE_TO_SINT, BYTE_TO_USINT	Значение передается в целевой тип данных.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	Значение передается в младший значащий байт целевого типа данных.
Word	WORD_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	WORD_TO_BYTE	Младший значащий байт исходного значения передается в целевой тип данных
	WORD_TO_DWORD	Значение передается в младшее значащее слово целевого типа данных.
	WORD_TO_SINT, WORD_TO_USINT	Младший значащий байт исходного значения передается в целевой тип данных.
	WORD_TO_INT, WORD_TO_UINT	Значение передается в целевой тип данных.
	WORD_TO_DINT, WORD_TO_UDINT	Значение передается в младшее значащее слово целевого типа данных.
DWord	DWORD_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT	Младший значащий байт исходного значения передается в целевой тип данных.
	DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	Младшее значащее слово исходного значения передается в целевой тип данных.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	Значение передается в целевой тип данных.

Таблица 8- 104 Преобразование из короткого целого (SInt или USInt)

Тип данных	Инструкция	Результат
SInt	SINT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	SINT_TO_BYTE	Значение передается в целевой тип данных
	SINT_TO_WORD, SINT_TO_DWORD	Значение передается в младший значащий байт целого типа данных.
	SINT_TO_INT, SINT_TO_DINT, SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	Значение преобразуется.
USInt	USINT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	USINT_TO_BYTE	Значение передается в целевой тип данных
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	Значение передается в младший значащий байт целого типа данных.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	Значение преобразуется.

Таблица 8- 105 Преобразование из целого (Int или UInt)

Тип данных	Инструкция	Результат
Int	INT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	Значение преобразуется.
	INT_TO_WORD	Значение передается в целевой тип данных.
	INT_TO_DINT	Значение передается в младший значащий байт целого типа данных.
UInt	UINT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	Значение преобразуется.
	UINT_TO_WORD, UINT_TO_DATE	Значение передается в целевой тип данных.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	Значение передается в младший значащий байт целого типа данных.

Таблица 8- 106 Преобразование из двойного целого (Dint или UDim)

Тип данных	Инструкция	Результат
DInt	DINT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	Значение преобразуется.
	DINT_TO_DWORD, DINT_TO_TIME	Значение передается в целевой тип данных.
UDInt	UDINT_TO_BOOL	Младший значащий бит передается в целевой тип данных.
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	Значение преобразуется.
	UDINT_TO_DWORD, UDINT_TO_TOD	Значение передается в целевой тип данных.

Таблица 8- 107 Преобразование из вещественного числа (Real или LReal)

Тип данных	Инструкция	Результат
Real	REAL_TO_DWORD, REAL_TO_LREAL	Значение передается в целевой тип данных.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	Значение преобразуется.
LReal	LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING	Значение преобразуется.

Таблица 8- 108 Преобразование из Time, DTL, TOD или Date

Тип данных	Инструкция	Результат
Time	TIME_TO_DINT	Значение передается в целевой тип данных.
DTL	DTL_TO_DATE, DTL_TO_TOD	Значение преобразуется.
TOD	TOD_TO_UDINT	Значение преобразуется.
Date	DATE_TO_UINT	Значение преобразуется.

Таблица 8- 109 Преобразование из символьного типа (Char или String)

Тип данных	Инструкция	Результат
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	Значение преобразуется.
	CHAR_TO_STRING	Значение передается в первый символ строки.
String	STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL	Значение преобразуется.
	STRING_TO_CHAR	Первый символ строки копируется в Char.

8.7.3. Инструкции ROUND (Округлить численное значение) и TRUNC (Отбросить дробную часть численного значения)

Таблица 8- 110 Инструкции ROUND и TRUNC

LAD / FBD	SCL	Описание
	<code>out := ROUND (in);</code>	Преобразует вещественное число в целое. В LAD/FBD, нажмите на "???" в верхней части инструкции для выбора типа данных на выходе, например "DInt". В SCL, тип данных по умолчанию для данной инструкции - DINT. Для округления в другой тип выходных данных введите имя инструкции с ключевым словом типа данных, например, ROUND_REAL или ROUND_LREAL. Дробная часть вещественного числа округляется до ближайшего целого значения (IEEE – округление до ближайшего). Если дробная часть находится ровно посередине между двумя целыми (например, 10.5), то результат округляется до большего целого. Например: • ROUND (10.5) = 10 • ROUND (11.5) = 12
	<code>out := TRUNC(in);</code>	TRUNC преобразует вещественное число в целое. Дробная часть вещественного числа отбрасывается (IEEE – округляется до нуля).

¹ Для LAD и FBD: Нажмите на "???" (на имени инструкции) и выберите тип данных из выпадающего списка.

Таблица 8- 111 Типы данных параметров

Параметр	Тип данных	Описание
IN	Real, LReal	Вещественное число
OUT	SInt, Int, DInt, USInt, UInt, UDim, Real, LReal	Выходное значение после округления или отбрасывания дробной части.

Таблица 8- 112 Состояние ENO

ENO	Описание	Результат
1	Без ошибок	Действительный результат
0	Значение в IN = +/- INF или +/- NaN	+/- INF или +/- NaN

8.7.4. Инструкции CEIL и FLOOR (Округление до следующего большего или меньшего целого числа)

Таблица 8- 113 Инструкции CEIL и FLOOR

LAD / FBD	SCL	Описание
	<code>out := CEIL(in);</code>	Преобразует вещественное число (Real или LReal) в целое число большее или равное конвертируемому числу (IEEE "округление до +бесконечности").
	<code>out := FLOOR(in);</code>	Преобразует вещественное число (Real или LReal) в целое число меньшее или равное конвертируемому числу (IEEE "округление до -бесконечности").

¹ Для LAD и FBD: Нажмите на "???" (на имени инструкции) и выберите тип данных из выпадающего списка.

Таблица 8- 114 Типы данных параметров

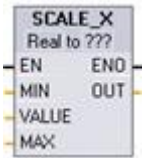
Параметр	Тип данных	Описание
IN	Real, LReal	Вещественное число
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Преобразованное число

Таблица 8- 115 Состояние ENO

ENO	Описание	Результат
1	Без ошибок	Действительный результат
0	Значение в IN = +/- INF или +/- NaN	+/- INF или +/- NaN

8.7.5. Инструкции SCALE_X (Масштабирование) и NORM_X (Нормализация)

Таблица 8- 116 Инструкции SCALE_X и NORM_X

LAD / FBD	SCL	Описание
	<pre>out :=SCALE_X(min:=_in_, value:=_in_, max:=_in_);</pre>	Масштабирует нормализованный вещественный параметр VALUE ($0.0 \leq \text{VALUE} \leq 1.0$) согласно типу данных и диапазону значений, определенному параметрами MIN и MAX : $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$
	<pre>out :=NORM_X(min:=_in_, value:=_in_, max:=_in_);</pre>	Нормализует параметр VALUE к диапазону значений, определенному параметрами MIN и MAX: $\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN}),$ где ($0.0 \leq \text{OUT} \leq 1.0$)

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

Таблица 8- 117 Типы данных параметров

Параметр	Тип данных ¹	Описание
MIN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Минимальное значение в диапазоне
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Значение для масштабирования или нормализации
MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Максимальное значение в диапазоне
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal NORM_X: Real, LReal	Масштабированное или нормализованное значение

¹ Для SCALE_X: Параметры MIN, MAX, и OUT должны иметь одинаковый тип данных.
Для NORM_X: Параметры MIN, VALUE, и MAX должны иметь одинаковый тип данных.

Примечание

Параметр VALUE инструкции SCALE_X должен быть ограничен диапазоном ($0.0 \leq \text{VALUE} \leq 1.0$)

Если параметр VALUE меньше 0.0 или больше 1.0:

- Операция линейного масштабирования может вернуть значения OUT, которые меньше, чем значение параметра MIN или больше чем значение параметра MAX, но которые соответствуют диапазону значений типа данных OUT. Выполнение SCALE_X устанавливает ENO = TRUE для этих случаев.
- Возможно сгенерировать масштабированные числа, которые выходят за рамки диапазона типа данных OUT. Для этих случаев значение параметра OUT устанавливается в промежуточное значение, равное младшей значащей части масштабированного вещественного числа до окончательного преобразования в тип данных OUT. Выполнение SCALE_X устанавливает в этом случае ENO = FALSE.

Параметр VALUE инструкции NORM_X должен быть ограничен диапазоном ($\text{MIN} \leq \text{VALUE} \leq \text{MAX}$)

Если значение параметра VALUE меньше MIN или больше MAX, то функции линейного масштабирования могут вернуть в параметре OUT значения, которые меньше 0.0 или больше 1.0. В этих случаях, выполнение NORM_X устанавливает ENO = TRUE.

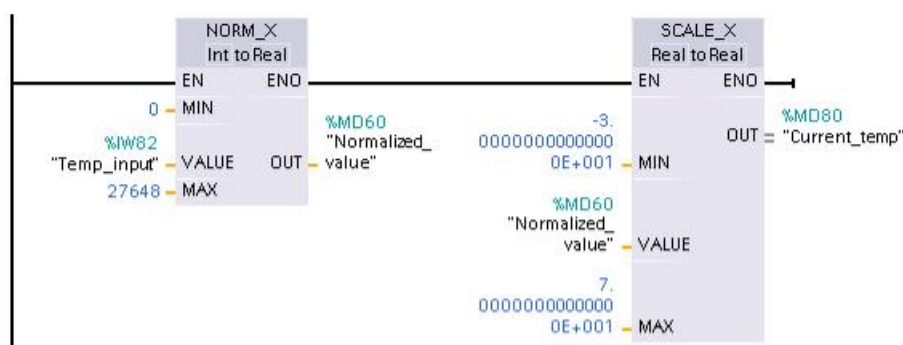
Таблица 8- 118 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Действительный результат
0	Результат превышает допустимый диапазон значений для выбранного типа данных	Промежуточный результат: младшая значащая часть вещественного числа до заключительного преобразования в тип данных.
0	Параметры $\text{MAX} \leq \text{MIN}$	SCALE_X: младшая значащая часть вещественного числа VALUE, чтобы заполнить размер OUT.
0	Параметр VALUE = +/- INF или +/- NaN	NORM_X: Значение в типе данных VALUE расширяется

Пример (LAD): нормализация и масштабирование значения аналогового входа

Аналоговый вход соответствующего сигнального модуля работает с диапазоном значений от 0 до 27648. Предположим, что сигнал на аналоговом входе представляет температуру, где значение 0 соответствует -30.0 градусам C, а 27648 соответствует 70.0 градусам C.

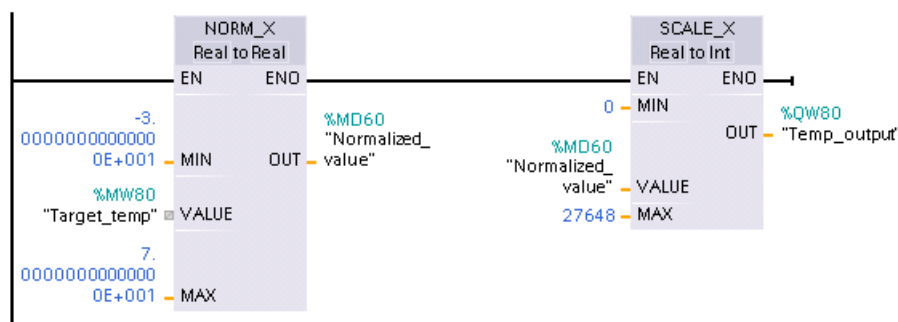
Для преобразования аналогового значения в соответствующие единицы, нормализуйте вход в значения от 0.0 до 1.0, а затем промасштабируйте в диапазон от -30.0 до 70.0. Результирующее значение является температурой, представленной аналоговым входом в градусах C:



Примите во внимание, что если аналоговый вход использует напряжение, то значение MIN для инструкции NORM_X будет равно -27648 вместо 0.

Пример (LAD): нормализация и масштабирование значения аналогового выхода

Аналоговый выход, который будет установлен в аналоговом сигнальном модуле или сигнальной плате, использующей токовый сигнал, должен быть в диапазоне от 0 до 27648 для получения допустимых значений. Предположим, что аналоговый выход представляет температурную уставку, где значение 0 на аналоговом выходе представляет -30.0 градусов C, а 27648 представляет 70.0 градусов C. Чтобы преобразовать значение температуры в памяти, находящееся между -30.0 и 70.0 в значение для аналогового выхода в диапазоне от 0 до 27648, Вы должны нормализовать значение в инженерных единицах к значению от 0.0 до 1.0, а затем промасштабировать его в диапазон аналогового выхода, от 0 до 27648:



Обратите внимание на то, что, если бы аналоговый выход был для модуля аналогового сигнала или сигнальной платы, использующих напряжение, то значение MIN для инструкции SCALE_X было бы -27648 вместо 0.

Дополнительная информация о представлениях аналогового входа (стр. 1194) и аналогового выхода (стр. 1195) как по напряжению, так и по току может быть найдена в Технических данных.

8.7.6. Инструкции преобразования типа Variant

8.7.6.1 Инструкция VARIANT_TO_DB_ANY (Преобразование из VARIANT в DB_ANY)

Вы можете воспользоваться инструкцией языка SCL "Convert VARIANT to DB_ANY" для чтения операнда в параметре IN и преобразования его в тип данных DB_ANY. Параметр IN должен иметь тип данных Variant и представлять либо экземплярный блок данных, либо блок данных с типом ARRAY. При создании программы, Вам не требуется знать какой блок соответствует параметру IN.

Инструкция считывает номер блока данных в процессе исполнения и записывает его в операнд параметра RET_VAL.

Таблица 8- 119 Инструкция VARIANT_TO_DB_ANY

LAD / FBD	SCL	Описание
Недоступно	<pre>RET_VAL := VARIANT_TO_DB_ANY(in := _variant_in_, err => _int_out_);</pre>	Считывает операнд из параметра IN с типом данных Variant и сохраняет его в DB_ANY

Таблица 8- 120 Параметры инструкции VARIANT_TO_DB_ANY

Параметр	Тип данных	Описание
IN	Variant	Variant, который представляет экземплярный блок данных или блок данных массива
RET_VAL	DB_ANY	Тип данных выхода DB_ANY, которой содержит номер преобразованного блока данных
ERR	Int	Информация об ошибке

Таблица 8- 121 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Инструкция преобразует вход с типом Variant и сохраняет его в DB_ANY
0	Состояние входа EN равно 0 или параметр IN недействительный	Инструкция ничего не делает.

Таблица 8- 122 Коды ошибок на выходе инструкции VARIANT_TO_DB_ANY

Err (W#16#...)	Описание
0000	Без ошибок
252C	Значение параметра IN типа данных Variant равно 0. ЦПУ переходит в режим STOP.
8131	Блок данных не существует или слишком мал (первый доступ).
8132	Блок данных слишком мал или не является блоком данных с типом Array (второй доступ).
8134	Блок данных защищен от записи
8150	Значение параметра IN типа данных Variant равно 0. Для получения данного сообщения об ошибке, необходимо активировать свойство блока "Handle errors within block". В противном случае CPU перейдет в режим STOP и сгенерирует код ошибки 16#252C
8154	Блок данных имеет неправильный тип данных.
* Вы можете отобразить коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.	

8.7.6.2 Инструкция DB_ANY_TO_VARIANT (Преобразование из DB_ANY в VARIANT)

Вы можете воспользоваться инструкцией языка SCL "Convert DB_ANY to VARIANT" для получения номера блока данных, который удовлетворяет следующим условиям. Операнд в параметре IN имеет тип данных DB_ANY, что означает, что Вы не должны знать во время создания программы, какой блок данных, номер которого должен быть считан, будет указан. Номер блока данных считывается во время выполнения и записывается посредством указателя VARIANT в операнд, определенный в параметре RET_VAL

Таблица 8- 123 Инструкция DB_ANY_TO_VARIANT

LAD / FBD	SCL	Описание
Недоступно	<pre>RET_VAL := DB_ANY_TO_VARIANT(in := _db_any_in_, err => _int_out_);</pre>	Считывает номер блока из параметра IN с типом данных Variant и возвращает его в качестве результата с типом данных Variant

Таблица 8- 124 Параметры инструкции DB_ANY_TO_VARIANT

Параметр	Тип данных	Описание
IN	DB_ANY	Variant, в котором находится номер блока
RET_VAL	Variant	Выход с типом данных DB_ANY, содержащий преобразованный номер блока
ERR	Int	Информация об ошибке

Таблица 8- 125 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Инструкция преобразует номер блока данных в Variant и запишет его в выход DB_ANY
0	Состояние входа EN равно 0 или параметр IN недействительный	Инструкция ничего не делает.

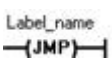
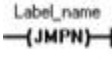
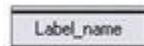
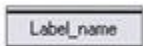
Таблица 8- 126 Коды ошибок инструкции DB_ANY_TO_VARIANT

Err (W#16#...)	Описание
0000	Без ошибок
8130	Номер блока данных равен 0.
8131	Блок данных не существует или слишком мал.
8132	Блок данных слишком мал или не является блоком данных с типом Array.
8134	Блок данных защищен от записи.
8154	Блок данных имеет неправильный тип данных.
8155	Неизвестный код типа
* Вы можете отобразить коды ошибок в программном редакторе в виде целых или шестнадцатеричных чисел.	

8.8 Операции управления обработкой программы

8.8.1. Инструкции JMP (Переход, если RLO = 1), JMPN (Переход, если RLO = 0), и Label (Метка перехода)

Таблица 8- 127 Инструкции JMP, JMPN, и LABEL

LAD	FBD	SCL	Описание
		См.оператор GOTO (стр. 304).	Переход, если RLO (результат логической операции) = 1: Если есть прохождение сигнала к катушке JMP (LAD), или активен вход блока JMP (FBD), тогда обработка программы продолжается с первой инструкции, следующей за указанной меткой.
			Переход, если RLO = 0: Если нет прохождения сигнала к катушке JMPN (LAD), или не активен вход блока JMPN (FBD), тогда обработка программы продолжается с первой инструкции, следующей за указанной меткой
			Метка для инструкции JMP или JMPN.

- ¹ Вы создаете имена меток, вводя их непосредственно в инструкции LABEL. Используйте пиктограмму помощника параметра, чтобы выбрать доступные имена меток для полей имени метки JMP и JMPN. Вы можете также ввести имя метки непосредственно в инструкцию JMP или JMPN

Таблица 8- 128 Типы данных параметров

Параметр	Тип данных	Описание
Label_name	Идентификатор метки	Идентификатор для команд перехода и соответствующей целевой программной метки перехода.

- Каждая метка должна быть уникальной в своем кодовом блоке.
- Вы можете выполнять переходы внутри одного кодового блока, переход в другие блоки невозможен.
- Переход может выполняться как вперед так и назад.
- Вы можете перейти к одной и той же метке более чем из одного места в одном и том же кодовом блоке.

8.8.2. Инструкция JMP_LIST (Определение списка переходов)

Таблица 8- 129 Инструкция JMP_LIST

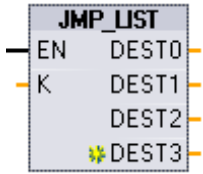
LAD / FBD	SCL	Описание
	<pre> CASE k OF 0: GOTO dest0; 1: GOTO dest1; 2: GOTO dest2; [n: GOTO destn;] END_CASE; </pre>	Инструкция JMP_LIST является распределителем переходов по программным секциям. В зависимости от значения на входе K, переход выполняется к соответствующей метке программы. Выполнение программы начинается с инструкции, следующей за меткой перехода. Если значение на входе K превышает номер метки - 1, то переход не выполняется и работа программы продолжается со следующего сегмента.

Таблица 8- 130 Типы данных параметров

Параметр	Тип данных	Описание
K	UInt	Управляющее значение распределителя переходов
DEST0, DEST1, ..., DESTn.	Программные метки	Метки перехода, соответствующие заданным значениям параметра K: Если значение K равно 0, тогда переход выполняется к программной метке, заданной на выходе DEST0. Если значение K равно 1, тогда переход произойдет к программной метке, заданной на выходе DEST1, и так далее. Если значение на входе K превышает (количество меток - 1), тогда переход не выполняется, а работа программы продолжается со следующего сегмента.

Для LAD и FBD: Если в Вашу программу впервые была добавлена инструкция JMP_LIST, то изначально у нее имеется всего два выхода. Вы можете добавить или удалить метки перехода.



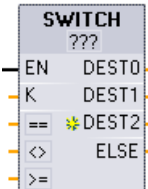
Нажмите на пиктограмму внутри блока (слева от последнего параметра DEST) для того, чтобы добавить новые метки для выполнения переходов.



- Щелкните правой кнопкой мыши на одном из выходов, после чего выберите команду "Insert output" (Добавить выход).
- Щелкните правой кнопкой мыши на одном из выходов, после чего выберите команду "Delete" (Удалить выход).

8.8.3. Инструкция SWITCH (Распределение переходов)

Таблица 8- 131 Инструкция SWITCH Инструкция

LAD / FBD	SCL	Описание
	Недоступно	Инструкция SWITCH является распределителем переходов по программным секциям. В зависимости от результатов сравнения между значением на входе K и значениями, заданными на соответствующих входах с типом сравнения, происходит переход к первой метке, соответствующей результату сравнения. Если ни одно сравнение не является истинным, то выполняется переход к метке, назначенной в параметре ELSE. Выполнение программы начинается с инструкции, следующей за меткой программы.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего списка.

² Для SCL: Используйте набор сравнений IF-THEN.

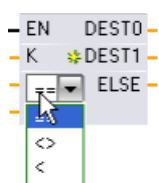
Таблица 8- 132 Типы данных параметров

Параметр	Тип данных ¹	Описание
K	UInt	Сравниваемое значение
==, <>, <, <=, >, >=	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Тип сравнения
DEST0, DEST1, ..., DESTn, ELSE	Программные метки	Целевые метки перехода, соответствующие определенным сравнениям: Вход сравнения сразу под входом K обрабатывается вначале и вызывает переход к метке, назначенной DEST0, если сравнение между значением K и этим входом истинно. Следующая сравнительная проверка использует следующий вход ниже и вызывает переход к метке, назначенной DEST1, если сравнение истинно, оставшиеся сравнения обрабатываются также, и, если ни одно из сравнений не истинно, то происходит переход к метке, назначенной выходу ELSE.

¹ Вход K и входы сравнений (==, <>, <, <=, >, >=) должны иметь одинаковый тип данных.

Добавление, удаление входов и назначение вариантов сравнения

Если в Вашу программу, на языке LAD или FBD, впервые была добавлена инструкция SWITCH, то изначально у нее имеется всего два входа для назначения вариантов сравнения. Вы можете назначить варианты сравнения и добавить переходы, как показано ниже.



Выберите оператор сравнения внутри инструкции из выпадающего списка.



Щелкните по пиктограмме создания, внутри самой инструкции (слева от последнего параметра DEST), чтобы добавить новые параметры для вставки в них операторов сравнения.



- Щелкните правой кнопкой мыши на одном из входов и выберите команду "Insert input" (Добавить вход).
- Щелкните правой кнопкой мыши на одном из входов и выберите команду "Delete" (Удалить).

Таблица 8- 133 Выбор типа данных блока SWITCH и допустимых операций сравнения

Тип данных	Сравнение	Синтаксис
Byte, Word, DWord	Равно	==
	Не равно	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Равно	==
	Не равно	<>
	Больше или равно	>=
	Меньше или равно	<=
	Больше	>
	Меньше	<

Правила добавления инструкции SWITCH

- Никакое соединение с инструкциями LAD/FBD перед входом сравнить не разрешено.
- Выход ENO отсутствует, таким образом в сегменте допустима только одна инструкция SWITCH, и она должна быть последней операцией в сегменте

8.8.4. Инструкция RET (Завершение блока)

Инструкция RET используется для завершения работы текущего блока. Если есть прохождение сигнала к катушке RET (LAD) или вход блока RET активен (FBD), то выполнение текущего блока будет завершено в данном месте программы и инструкции, следующие за RET, выполнены не будут. Если блоком является OB, то параметр "Return_Value" будет проигнорирован. Если блоком является FC или FB, значение параметра "Return_Value" передается обратно в вызывающую процедуру в качестве значения ENO, вызванной инструкции.

Вам не требуется использовать инструкцию RET в качестве последней инструкции блока; это производится автоматически за Вас. Внутри одного блока, Вы можете иметь несколько инструкций RET.

При работе с языком SCL, обратитесь к выражению RETURN (стр. 304).

Таблица 8- 134 Инструкция завершения работы блока Return_Value (RET)

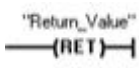
LAD	FBD	SCL	Описание
		RETURN;	Завершает выполнение работы текущего блока

Таблица 8- 135 Типы данных параметров

Параметр	Тип данных	Описание
Return_Value	Bool	Параметр "Return_value" инструкции RET назначается на выход ENO инструкции вызова в вызывающем блоке.

Простые шаги по использованию инструкции RET внутри блока FC:

1. Создайте новый проект и добавьте FC:
2. Отредактируйте FC:
 - Из дерева инструкций, добавьте функционал.
 - Добавьте инструкцию RET, в которой значение параметра "Return_Value" может быть следующим:
TRUE, FALSE, или ячейка памяти, которая определяет требуемое возвращаемое значение.
 - Добавьте другие инструкции.
3. Вызовите FC из MAIN [OB1].

Вход EN инструкции FC в блоке MAIN должен быть активным до начала выполнения FC.

Значение, определенное инструкцией RET в FC, будет присутствовать на выходе ENO блока FC в кодовом блоке MAIN после выполнения FC, для которого есть прохождение сигнала к инструкции RET.

8.8.5. Инструкция ENDIS_PW (Активировать/деактивировать пароли ЦПУ)

Таблица 8- 136 Инструкция ENDIS_PW

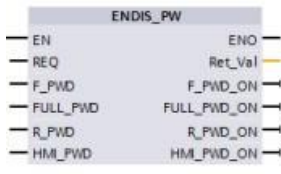
LAD / FBD	SCL	Описание
	<pre> ENDIS_PW(req:=_bool_in_, f_pwd:=_bool_in_, full_pwd:=_bool_in_, r_pwd:=_bool_in_, hmi_pwd:=_bool_in_, f_pwd_on=>_bool_out_, full_pwd_on=>_bool_out_, r_pwd_on=>_bool_out_, hmi_pwd_on=>_bool_out_); </pre>	С помощью инструкции ENDIS_PW можно разрешить или запретить клиентские соединения с S7-1200 ЦПУ, даже при корректном вводе пароля. Данная инструкция не запрещает пароли к Web серверу.

Таблица 8- 137 Типы данных параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Выполнение инструкции при REQ=1
F_PWD	IN	Bool	Fail-safe пароль: Разрешить (=1) или запретить (=0)
FULL_PWD	IN	Bool	Пароль на полный доступ: Разрешить (=1) или запретить (=0) полный доступ
R_PWD	IN	Bool	Пароль на чтение: Разрешить (=1) или запретить (=0)
HMI_PWD	IN	Bool	Пароль на доступ со стороны HMI: Разрешить (=1) или запретить (=0)
F_PWD_ON	OUT	Bool	Fail-safe состояние пароля: Разрешен (=1) или запрещен (=0)
FULL_PWD_ON	OUT	Bool	Состояние пароля с полным доступом: Разрешен (=1) или запрещен (=0)
R_PWD_ON	OUT	Bool	Состояние пароля на чтение: Разрешен (=1) или запрещен (=0)
HMI_PWD_ON	OUT	Bool	Состояние пароля на доступ со стороны HMI: Разрешен (=1) или запрещен (=0)
Ret_Val	OUT	Word	Результат функции

Вызов ENDIS_PW с REQ=1 запрещает использование типов паролей, у которых соответствующий входной параметр имеет состояние FALSE. Каждый тип пароля может быть разрешен или запрещен. Например, если fail-safe пароль разрешен, и все остальные пароли запрещены, то Вы можете запретить доступ к ЦПУ для определенной группы персонала.

ENDIS_PW выполняется синхронно в скане программы, и выходные параметры пароля всегда показывают текущее состояние допуска пароля, независимого от входного параметра REQ. Все пароли, которые Вы устанавливаете, как разрешенные, должны быть изменяемыми на запрещенные/разрешенные. Иначе, возвращается ошибка, и разрешаются все пароли, которые были разрешены до выполнения ENDIS_PW. Это означает, что в стандартном ЦПУ (где fail-safe пароль не сконфигурирован) F_PWD должен всегда устанавливаться в 1, чтобы дать возвращаемое значение 0. В этом случае F_PWD_ON всегда равен 1.

Примечание

- Выполнение ENDIS_PW может заблокировать доступ от устройств HMI, если пароль на доступ со стороны HMI был запрещен.
- Клиентские сессии, которые были авторизованы до выполнения ENDIS_PW, остаются неизменными после выполнения ENDIS_PW.

После включения питания, доступ к ЦПУ запрещен с помощью паролей, назначенных в обычной конфигурации защиты ЦПУ. Запрет пароля можно устранить, повторно выполнив инструкцию ENDIS_PW. Тем не менее, если ENDIS_PW был выполнен и требуемые пароли запрещены, то доступ из TIA portal может быть заблокирован. Вы можете воспользоваться таймерами, для установки задержки выполнения инструкции ENDIS_PW и разрешить время для ввода пароля, до того как они станут запрещены.

Примечание

Восстановление ЦПУ, который заблокировал обмен с TIA portal

Обратитесь к разделу "Восстановление потерянного пароля (стр. 144)" для получения детальной информации о том, как стереть внутреннюю загрузочную память ПЛК при помощи карты памяти.

Изменение рабочего режима в STOP, вызванное ошибками, выполнением STP или действиями в STEP 7, не отменяет защиту. Защита действительна, пока ЦПУ не будет выполнен цикл включения и выключения питания. См. следующую таблицу для получения более подробной информации

Действие	Режим работы	Состояние ENDIS_PW
После сброса памяти из STEP 7	STOP	Активна: Запрещенные пароли останутся запрещенными.
После включения питания, или при смене карты памяти	STOP	Выключена: Запрещенных паролей нет.
После выполнения ENDIS_PW в программном цикле или стартовом OB	STARTUP, RUN	Активна: Пароли запрещены в соответствии с параметрами ENDIS_PW
После смены режима работы из RUN или STARTUP в STOP из-за инструкции STP, ошибки или действий в STEP 7	STOP	Активна: Запрещенные пароли останутся запрещенными

Примечание

Пароль защищает уровни доступа к ЦПУ с помощью строгих паролей. Строгие пароли имеют длину равную, по крайней мере, десяти символам, комбинируют буквы, цифры и специальные символы, не являются словами, которые могут быть найдены в словаре и не являются именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.

Таблица 8- 138 Коды состояния

RET_VAL (W#16#...)	Описание
0000	Без ошибок
8090	Инструкция не поддерживается.
80D0	Fail-safe пароль не сконфигурирован.
80D1	Пароль на чтение/запись не сконфигурирован.
80D2	Пароль на чтение не сконфигурирован.
80D3	Пароль на доступ от HMI не сконфигурирован.

8.8.6. Инструкция RE_TRIGR (Рестарт времени контроля цикла)

Таблица 8- 139 Инструкция RE_TRIGR

LAD / FBD	SCL	Описание
	RE_TRIGR() ;	RE_TRIGR (Перезапустить сторожевой таймер времени сканирования) используется, чтобы продлить максимальное время, отсчитываемое, прежде чем сторожевой таймер цикла сканирования сгенерирует ошибку

Используйте инструкцию RE_TRIGR для перезапуска таймера контроля цикла сканирования в течение одиночного сканирования. При этом происходит продление максимально допустимого времени цикла на время одного максимального цикла сканирования с момента последнего использования инструкции RE_TRIGR.

Примечание

До версии 2.2 встроенного ПО ЦПУ S7-1200 RE_TRIGR была ограничен использованием в ОБ программного цикла и могла использоваться, чтобы продлить время сканирования ПЛК на неопределенное время. ENO = FALSE и сторожевой таймер не сбрасывается, когда RE_TRIGR выполнялся из ОБ запуска, ОБ прерывания или ОБ ошибки.

Для версии встроенного ПО 2.2 и выше RE_TRIGR может выполняться из любого ОБ (включая ОБ запуска, прерывания и ошибки). Однако сканирование ПЛК может быть продлено не более чем на 10 значений сконфигурированного максимального времени сканирования.

Установка максимального времени цикла PLC

Сконфигурируйте значение для максимального времени цикла в конфигурации устройства для "Cycle time".

Таблица 8- 140 Значение времени цикла

Наблюдение за временем цикла	Минимальное значение	Максимальное значение	По умолчанию
Максимальное время цикла	1 мс	6000 мс	150 мс

Завершение мониторинга

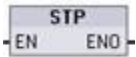
Если таймер максимального времени цикла истек до того, как была закончен цикл, то будет сгенерирована ошибка. Если в пользовательской программе имеется ОВ обработки ошибки времени выполнения (ОВ 80), то ЦПУ выполняет данный ОВ, который может содержать программную логику с соответствующей реакцией.

Если в пользовательской программе отсутствует ОВ обработки ошибки времени выполнения, то сначала ЦПУ проигнорирует данное превышение и останется в режиме RUN. При двукратном превышении максимального времени цикла, будет сгенерирована ошибка, и ЦПУ перейдет в режим STOP.

В режиме STOP, выполнение Вашей программы будет остановлено, в то время как системные коммуникации и диагностика ЦПУ продолжат работать.

8.8.7. Инструкция STP (Окончание работы программы)

Таблица 8- 141 Инструкция STP

LAD / FBD	SCL	Описание
	STP ();	STP переводит ЦПУ в режим STOP. Когда ЦПУ находится в режиме STOP, выполнение Вашей программы и обновление образа процесса остановлены.

Для получения более подробной информации обратитесь к разделу: Конфигурирование выходов при переходе из RUN в STOP (стр. 108).

Если EN = TRUE, то ЦПУ перейдет в режим STOP, выполнение программы будет остановлено, при этом значение состояния ENO не имеет значения. В противном случае, EN = ENO = 0.

8.8.8. Инструкции GET_ERROR и GET_ERROR_ID (Локальное получение ошибки и ее ID)

Необходимы для получения информации об ошибке, возникшей при выполнении блока. При добавлении в Ваш блока инструкции GET_ERROR или GET_ERROR_ID, Вы сможете обработать программные ошибки внутри Вашего программного блока.

GET_ERROR

Таблица 8- 142 Инструкция GET_ERROR

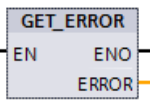
LAD / FBD	SCL	Описание
	GET_ERROR(_out_);	Указывает на то, что локальная ошибка выполнения программного блока произошла, и заполняет заранее определенную структуру данных подробной информацией об ошибке.

Таблица 8- 143 Типы данных параметров

Параметр	Тип данных	Описание
ERROR	ErrorStruct	Структура данных ошибки: Вы можете переименовать структуру, но не элементы структуры

Таблица 8- 144 Элементы структуры данных ErrorStruct

Поле		Тип данных	Описание					
ERROR_ID		Word	ID ошибки					
FLAGS		Byte	Показывает, возникла ли ошибка в процессе вызова блока. 16#01: Ошибка возникла в процессе вызова блока. 16#00: Ошибка не возникла в процессе вызова блока.					
REACTION		Byte	Реакция: 0: Игнорировать (Ошибка записи), 1: Продолжить со значением подстановки "0" (Ошибка чтения), 2: Пропустить инструкцию (Системная ошибка)					
CODE_ADDRESS		CREF	Информация об адресе и типе блока					
	BLOCK_TYPE	Byte	Тип блока, содержащий ошибку: 1: OB 2: FC 3: FB					
	CB_NUMBER	UInt	Номер кодового блока					
	OFFSET	UDInt	Указатель на внутреннюю память					
MODE		Byte	Режим доступа: В зависимости от типа доступа, можно получить следующую информацию:					
			Режим	(A)	(B)	(C)	(D)	(E)

Поле	Тип данных	Описание					
		0					
		1					Смещение
		2			Область		
		3	Место	Диапазон		Номер	
		4			Область		Смещение
		5			Область	Номер DB	Смещение
		6	Ном.ук. / Акк.		Область	Номер DB	Смещение
		7	Ном.ук. / Акк.	Ном.слота / Диапазон	Область	Номер DB	Смещение
OPERAND_NUMBER	UInt	Номер операнда машинной команды					
POINTER_NUMBER_ LOCATION	UInt	(A) Внутренний указатель					
SLOT_NUMBER_SCOPE	UInt	(B) Область хранения во внутренней памяти					
DATA_ADDRESS	NREF	Информация об адресе операнда					
	AREA	Byte	(C) Область памяти: ☐ L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE ☐ I: 16#81 ☐ Q: 16#82 ☐ M: 16#83 ☐ DB: 16#84, 85, 8A, 8B				
	DB_NUMBER	UInt	(D) Номер блока данных				
	OFFSET	UDInt	(E) Относительный адрес операнда				

GET_ERROR_ID

Таблица 8- 145 Инструкция GetErrorID

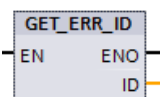
LAD / FBD	SCL	Описание
	GET_ERR_ID();	Оповещает о наличии программной ошибки и выводит ID (Идентификатор) ошибки.

Таблица 8- 146 Типы данных параметров

Параметр	Тип данных	Описание
ID	Word	Значения идентификатора ошибки для элемента ERROR_ID структуры ErrorStruct

Таблица 8- 147 Значения Error_ID

Шестнадцатеричный ERROR_ID	Десятичный ERROR_ID	Ошибка выполнения блока
0	0	Без ошибок
2520	9504	Поврежденная строка
2522	9506	Операнд вне диапазона допустимых значений при чтении
2523	9507	Операнд вне диапазона допустимых значений при записи
2524	9508	Неверная область при чтении
2525	9509	Неверная область при записи
2528	9512	Выравнивание данных при чтении (неверное выравнивание бита)
2529	9513	Выравнивание данных при записи (неверное выравнивание бита)
252C	9516	Ошибка неинициализированного указателя
2530	9520	DB защищен от записи
2533	9523	Использование недопустимого указателя
2538	9528	Ошибка доступа: DB не существует
2539	9529	Ошибка доступа: Используется неверный DB
253A	9530	Глобальный DB не существует
253C	9532	Неверная версия или FC не существует
253D	9533	Инструкция не существует
253E	9534	Неверная версия или FB не существует
253F	9535	Инструкция не существует
2550	9552	Ошибка доступа: DB не существует
2575	9589	Ошибка глубины вложения программ
2576	9590	Ошибка размещения локальных данных
2942	10562	Физический вход не существует
2943	10563	Физический выход не существует

Принцип работы

По умолчанию ЦПУ реагирует на ошибку выполнения блока, внося ошибку в диагностический буфер. Однако, если Вы помещаете одну или несколько инструкций GET_ERROR или GET_ERROR_ID в кодовый блок, этот блок теперь организован для обработки ошибки в блоке. В этом случае ЦПУ не вносит ошибку в диагностический буфер. Вместо этого информация об ошибке передается на выход инструкции GET_ERROR или GET_ERROR_ID. Вы можете считать подробную информацию об ошибке с помощью инструкции GET_ERROR или просто идентификатор ошибки с помощью инструкции GET_ERROR_ID. Обычно первая ошибка является самой важной, за которой идут ошибки являющиеся ее следствиями.

Первое выполнение инструкции GET_ERROR или GET_ERROR_ID в блоке возвращает первую ошибку, обнаруженную во время выполнения блока. Эта ошибка могла произойти где угодно между запуском блока и выполнением GET_ERROR или GET_ERROR_ID. Последующее выполнение GET_ERROR или GET_ERROR_ID возвращает первую ошибку начиная с предыдущего выполнения GET_ERROR или GET_ERROR_ID. История ошибок не сохраняется, и выполнение любой инструкции повторно активирует ПЛК систему на фиксацию следующей ошибки.

Тип данных `ErrorStruct`, используемый инструкцией `GET_ERROR`, может быть добавлен в редактор блока данных и редактор интерфейса блока, это означает, что Ваша программа сможет получать доступ к этим значениям. Из выпадающего списка с типами данных, выберите `ErrorStruct`, чтобы добавить данную структуру. Вы можете создать несколько элементов `ErrorStruct` со своими уникальными именами. Отдельные элементы `ErrorStruct` не могут быть переименованы.

Коды состояния, отображаемые в ENO

Если `EN = TRUE` и производится выполнение инструкций `GET_ERROR` или `GET_ERROR_ID`, тогда:

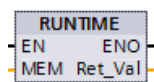
- `ENO = TRUE` – означает, что возникла ошибка выполнения блока и информация по данной ошибке представлена
- `ENO = FALSE` – означает, что ошибка выполнения блока не произошла

Вы можете соединить программную логику реакции на ошибку с `ENO`, который активируется после того, как ошибка происходит. Если ошибка существует, то выходной параметр сохранит данные об ошибке в том месте, на которое Вы сослались в нем.

`GET_ERROR` и `GET_ERROR_ID` могут использоваться, чтобы отправить информацию об ошибке из выполняющегося в настоящий момент блока (вызываемый блок) в вызывающий блок. Поместите инструкцию в последний сегмент программы вызываемого блока, чтобы сообщить о заключительном состоянии выполнения вызываемого блока.

8.8.9. Инструкция RUNTIME (Оценка работы программы)

Таблица 8- 148 Инструкция RUNTIME

LAD / FBD	SCL	Описание
	<pre>Ret_Val := RUNTIME(_lread_inout_);</pre>	Оценивает работы программы или отдельных блоков.

Если Вам необходимо измерить время выполнения Вашей целой программы, вызовите инструкцию "Measure program runtime" в OB1. При первом вызове, запустится измерение времени работы программы, и выход `RET_VAL` возвращает время выполнения программы после второго вызова. Измерение включает в себя все события ЦПУ, которые могли возникнуть в процессе работы программы, например, прерывания, вызванные высокоуровневыми событиями или коммуникацией. Инструкция "Measure program runtime" считывает внутренний счетчик ЦПУ и записывает значение в IN-OUT параметр `MEM`. Инструкция измеряет время работы программы в соответствии с внутренней частотой счетчика и производит запись в параметр `RET_VAL`.

Если Вам необходимо измерить время работы отдельных блоков, Вам потребуется три различных сегмента. Вызовите инструкцию "Measure program runtime" в отдельном сегменте Вашей программы. Этим первым вызовом, Вы определяете точку входа для начала измерения. После этого Вы вызываете необходимый блок в следующем сегменте. В другом сегменте, Вызовите инструкцию "Measure program runtime" второй раз и назначьте ту же область памяти в IN-OUT параметре `MEM`, как при первом вызове данной инструкции. Инструкция "Measure program runtime" в третьем сегменте считывает внутренний счетчик ЦПУ выполняет оценку времени работы данного блока в соответствии с внутренней частотой счетчика и производит запись в параметр `RET_VAL`.

Инструкция "The Measure program runtime" использует внутренний высокочастотный счетчик для вычисления времени. Если счетчик превысил максимальное значение, то инструкция возвращает ≤ 0.0 . Можно проигнорировать данные значения.

Примечание

ЦПУ не может точно определить время выполнения последовательности команд, поскольку порядок инструкций в командной последовательности изменяется во время оптимизированной компиляции программы

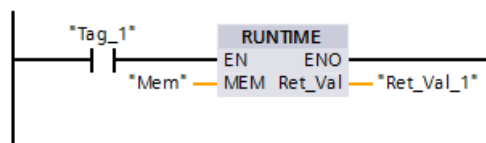
Таблица 8- 149 Типы данных параметров

Параметр	Тип данных	Описание
MEM	LReal	Начальная точка измерения времени выполнения
RET_VAL	LReal	Измеренное время выполнения в секундах

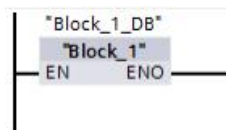
Пример: Инструкция RUNTIME

На следующем примере показано использование инструкции RUNTIME для оценки времени работы функционального блока:

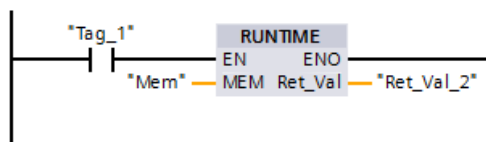
Сегмент 1:



Сегмент 2:



Сегмент 3:



Когда операнд "Tag_1" в сегменте 1 имеет состояние сигнала "1", инструкция RUNTIME выполняется. Начальная точка для измерения времени выполнения устанавливается первым вызовом инструкции и помещается как ссылка для второго вызова инструкции в операнд "Mem".

Функциональный блок FB1 выполняется в сегменте 2.

Когда программный блок FB1 завершает работу и состояние операнда "Tag_1" равно "1", тогда в сегменте 3 вновь запускается инструкция RUNTIME. Второй вызов инструкции вычисляет время работы блока и записывает результат в параметр RET_VAL_2.

8.8.10. Операторы управления программой в SCL

8.8.10.1 Обзор операторов управления программой в языке SCL

В языке SCL имеется три типа операторов для управления поведением Вашей программой:

- Операторы выбора: оператор выбора разрешает Вам направить выполнению программы в альтернативные последовательности операторов.
- Циклы: Вы можете управлять выполнением цикла, используя операторы цикла. Оператор цикла определяет, какие части программы должны быть выполнены с помощью итераций в зависимости от определенных условий.
- Программные переходы: программные переходы означают незамедлительный переход в указанное место назначения и следовательно к различным операторам в одном и том же блоке.

Эти операторы программного управления использует синтаксис языка программирования PASCAL.

Таблица 8- 150 Типы операторов программного управления в SCL

Оператор программного управления		Описание
Выбор	IF-THEN (стр. 296)	Позволяет Вам направить выполнению программы в одно из двух альтернативных ветвлений, в зависимости от условия, имеющего состояние TRUE или FALSE
	CASE (стр. 297)	Позволяет выборочное выполнение одного из n альтернативных ветвлений на основе значения переменной
Цикл	FOR (стр. 299)	Повторяет последовательность операторов столько, сколько контрольная переменная остается в указанном диапазоне значений
	WHILE-DO (стр. 300)	Повторяет последовательность операторов пока условия выполнения продолжает удовлетворяться
	REPEAT-UNTIL (стр. 301)	Повторяет последовательность операторов, пока не будет соблюдено условие окончания
Программный переход	CONTINUE (стр. 302)	Останавливает выполнение текущей итерации цикла
	EXIT (стр. 303)	Выходит из цикла в любой точке независимо от того, удовлетворено условие окончания или нет
	GOTO (стр. 304)	Заставляет программу незамедлительно перейти по указанной метке
	RETURN (стр. 304)	Заставляет программу выходить из блока, выполняемого в настоящий момент и возвращаться в вызывающий блок.

8.8.10.2 Операторы IF-THEN

Условный оператор IF-THEN обеспечивает выполнение определенной команды при условии истинности некоторого условия. Вы также можете использовать скобки для структурирования выполнения нескольких операторов IF-THEN.

Таблица 8- 151 Элементы условного оператора IF-THEN

SCL	Описание
IF "condition" THEN statement_A; statement_B; statement_C; ;	Если "condition" (условие) равно TRUE или 1, то выполняются следующие инструкции до выражения END_IF. Если "condition" (условие) равно FALSE или 0, тогда происходит переход к выражению END_IF (при отсутствующих выражениях ELSIF или ELSE).
[ELSIF "condition-n" THEN statement_N; ;]	При помощи выражения ELSEIF ¹ можно задать дополнительные условия. Например: Если "condition" (условие) в IF-THEN имеет состояние FALSE, то программа переходит к проверке "condition-n". Если "condition-n" равно TRUE, то происходит выполнение "statement_N".
[ELSE statement_X; ;]	Если состояние "condition" (условие) оператора IF-THEN равно FALSE, тогда выполняется переход к инструкциям блока ELSE.
END_IF;	Выражение END_IF завершает выполнение работы оператора IF-THEN.

¹ Вы можете добавить несколько выражений ELSIF в одном операторе IF-THEN.

Таблица 8- 152 Типы данных параметров

Переменные	Описание
"condition"	Обязательное. Результат логического выражения может быть TRUE (1) или FALSE (0).
"statement_A"	Необязательное. Одно или несколько выражений при выполнении "condition".
"condition-n"	Необязательное. Условие для проверки оператора ELSIF.
"statement_N"	Необязательное. Одно или несколько выражений при выполнении "condition-n" оператора ELSIF.
"statement_X"	Необязательное. Одно или несколько выражений, которые будут выполняться при не выполнении условия "condition" оператора IF-THEN.

Условный оператор IF выполняется при следующих условиях:

- Выполняется первое выражение, у которого логическое состояние равно TRUE. Остальные выражения выполнены не будут.
- При отсутствии состояния логического выражения Boolean равно TRUE, тогда выполняется переход к блоку ELSE (или выход из блока IF при отсутствии оператора ELSE).
- Может быть любое количество ELSIF операторов.

Примечание

Использование одного или нескольких операторов ELSIF дает преимущество в том, что после выполнения логических выражений, более не будут выполнены в программе, в отличие от условного оператора IF. Время обработки программы будет уменьшено.

8.8.10.3 Оператор выбора CASE

Таблица 8- 153 Элементы оператора CASE

SCL	Описание
CASE "Test_Value" OF "ValueList": Statement[; Statement, ...] "ValueList": Statement[; Statement, ...] [ELSE Else-statement[; Else-statement, ...] END_CASE;	Оператор выбора CASE реализует выбор одной из возможных альтернатив, зависящих от значения селектора.

Таблица 8- 154 Параметры

Параметр	Описание
"Test_Value"	Обязательный. Любое числовое значение с типом данных Int
"ValueList"	Обязательный. Одно или несколько значений или диапазонов значений. (используйте две точки для задания диапазона: 2..8). В следующем примере показано использование различных значений диапазона: 1: Statement_A; 2, 4: Statement_B; 3, 5..7,9: Statement_C;
Statement	Обязательный. Одно или несколько выражений, которые выполняются при "Test_Value" равном одному из значений в списке
Else-statement	Необязательный. Одно или несколько выражений, которые выполняются если селектор не совпал ни с одним значением из списка

Оператор выбора CASE выполняется при следующих условиях:

- Выражение Test_value должно возвращать значение с типом Int.
- При обработке значения CASE, программа проверяет имеется ли значение выражения Test_value в списке допустимых значений. Если имеется, то выполняется соответствующий элемент списка.
- Если значение не совпало ни с одним из разрешенных, то выполняется переход к оператору ELSE, при отсутствии оператора ELSE инструкции не выполняются.

Пример: Вложенные операторы выбора CASE

Оператор выбора CASE может быть вложенным. Каждый вложенный оператор выбора должен иметь соответствующий конец блока END_CASE.

```
CASE "var1" OF
    1 : #var2 := 'A';
    2 : #var2 := 'B';
ELSE
    CASE "var3" OF

        65..90: #var2 := 'UpperCase';
        97..122: #var2 := 'LowerCase';
    ELSE
        #var2:= 'SpecialCharacter';
    END_CASE;
END_CASE;
```

8.8.10.4 Цикл FOR

Таблица 8- 155 Параметры цикла FOR

SCL	Описание
<pre>FOR "control_variable" := "begin" TO "end" [BY "increment"] DO statement; ; END_FOR;</pre>	Оператор FOR используется для создания цикла со счетчиком, а именно цикл будет выполняться, пока значение параметра "control_variable" не достигнет конечного. Для объявления данного цикла необходимо задать начальные и конечные значения счетчика. Оба значения должны быть одного типа данных, так же как и сам счетчик. Цикл FOR могут быть вложенными. Выражение END_FOR объявляется для завершения работы оператора FOR.

Таблица 8- 156 Параметры

Параметр	Описание
"control_variable"	Обязательный. Целое число (Int или DInt), которое является счетчиком цикла
"begin"	Обязательный. Начальное значение
"end"	Обязательный. Конечное значение
"increment"	Необязательный. Значение, на которое счетчик "control variable" будет изменяться после каждой итерации цикла. Параметр "increment" должен быть того же типа данных, что и "control variable". Если значение "increment" не назначено, тогда после каждой итерации, счетчик будет увеличиваться на 1. Вы не можете изменить параметр "increment" в процессе выполнения оператора цикла FOR.

Работа оператора цикла FOR:

- В начале цикла, счетчик устанавливается в начальное значение (инициализация) и после каждой итерации, он будет увеличиваться или уменьшаться до тех пор пока не достигнет своего конечного значения.
- После каждой итерации цикла, происходит проверка условия (достигнуто ли конечное значение) для того чтобы определить – заканчивать цикл или нет. Если конечное значение не было достигнуто, то работа цикла продолжается, иначе производится выход из цикла а выполнение следующей инструкции, следующей за блоком FOR.

Правила объявления оператора цикла FOR:

- Тип данных счетчика может быть Int или DInt.
- Вы можете не указывать выражение BY [increment]. Тогда после каждой итерации, счетчик будет увеличиваться автоматически на +1.

Для выхода из цикла, независимо от состояния "condition", воспользуйтесь выражением EXIT (Страница 303). Выражение EXIT выполнит выход из цикла и переход к выполнению следующей инструкции за выражение END_FOR.

Используйте оператор CONTINUE (Страница 302) для перехода к следующей итерации цикла FOR.

8.8.10.5 Цикл WHILE-DO

Таблица 8- 157 Цикл WHILE

SCL	Описание
<pre>WHILE "condition" DO Statement; Statement; ...; END_WHILE;</pre>	Цикл WHILE работает пока состояние условия TRUE. Циклы WHILE могут быть вложенными. Выражение END_WHILE производит завершение цикла WHILE.

Таблица 8- 158 Параметры

Параметр	Описание
"condition"	Обязательный. Логическое выражение, состояние которого может быть TRUE или FALSE. (Условие "null" интерпретируется как FALSE.)
Statement	Необязательный. Одно или несколько выражений, которые выполняются в процессе цикла пока состояние условия TRUE.

Примечание

В цикле WHILE сначала выполняется проверка условия "condition". Если Вам необходимо, чтобы цикл выполнялся хотя бы один раз, воспользуйтесь оператором цикла REPEAT (Страница 301).

Работа оператора цикла WHILE:

- До начала выполнения инструкций в теле цикла, сначала производится проверка условия.
- Инструкции в цикле, следующие за оператором DO, будут выполняться до тех пор, пока состояние условия TRUE.
- При состоянии FALSE, производится выход из цикла и переход к выполнению следующей инструкции.

Для выхода из цикла, независимо от состояния "condition", воспользуйтесь выражением EXIT (Страница 303). Выражение EXIT выполнит выход из цикла и переход к выполнению следующей инструкции за выражение END_WHILE.

Используйте оператор CONTINUE (Страница 302) для перехода к следующей итерации цикла WHILE.

8.8.10.6 Цикл REPEAT-UNTIL

Таблица 8- 159 Цикл REPEAT

SCL	Описание
REPEAT Statement; ; UNTIL "условие" END_REPEAT;	Цикл REPEAT выполняется до тех пор, пока не достигнуто условие завершения выхода из цикла. Цикл REPEAT может быть вложенным. Выражение END_REPEAT относится к последней выполняемой инструкции данного цикла.

Таблица 8- 160 Параметры

Параметр	Описание
Statement	Необязательный. Одна или несколько инструкций выполняются пока состояние условия TRUE.
"condition"	Обязательный. Одно или несколько выражений: Числовое значение или строка, состояние которой оценивается как TRUE или FALSE. Условие "нуль" интерпретируется как FALSE.

Примечание

При первой итерации цикла, инструкции в теле REPEAT будут выполнены до проверки условия "condition" (даже если состояние "condition" FALSE). Для того чтобы, "условие" проверялось до выполнения инструкций, воспользуйтесь циклом WHILE (стр. 300).

Для выхода из цикла, независимо от "условия", воспользуйтесь выражением EXIT (Страница 303). Выражение EXIT выполнит выход из цикла и переход к выполнению следующей инструкции за выражение END_REPEAT.

Используйте оператор CONTINUE (Страница 302) для перехода к следующей итерации цикла REPEAT.

8.8.10.7 Оператор CONTINUE

Таблица 8- 161 Оператор CONTINUE

SCL	Описание
CONTINUE Statement; ;	Оператор CONTINUE позволяет пропустить текущую итерацию цикла и перейти к следующей (FOR, WHILE, REPEAT) предварительно проверив условие. Если условие TRUE, тогда выполняется следующая итерация.

Работа оператора CONTINUE:

- Немедленно прекращается выполнение инструкций тела цикла.
- Выполняется проверка условия цикла, и если результат истинный, то начинается новая итерация.
- В цикле FOR, счетчик сразу же изменяется, в зависимости от установленного значения модификатора.

Используйте CONTINUE только внутри одного тела цикла. Во вложенных циклах, CONTINUE позволяет перейти к выполнению цикла в котором данный оператор содержится. CONTINUE также широко используется с оператором IF.

Для немедленного выхода из цикла, используйте оператор EXIT.

Пример: Оператор CONTINUE

В следующем блоке показан пример использования оператора CONTINUE, для того чтобы избежать деления на 0, вычисляя процент:

```
FOR i := 0 TO 10 DO
  IF value[i] = 0 THEN CONTINUE; END_IF;
  p := part / value[i] *
    100; s :=
    INT_TO_STRING(p);
  percent := CONCAT(IN1:=s, IN2:="%");
END_FOR;
```

8.8.10.8 Оператор EXIT

Таблица 8- 162 Оператор EXIT

SCL	Описание
EXIT;	Оператор EXIT используется для немедленного выхода из цикла (FOR, WHILE или REPEAT), независимо от состояния условия.

Работа оператора EXIT:

- Данный оператор используется для немедленного выхода из цикла.
- Выполнение программы продолжается после оператора завершения цикла (например, после END_FOR).

Используйте оператор EXIT внутри цикла. Во вложенных циклах, использование оператора EXIT приведет к выполнению цикла более высокого уровня (тот, который вызвал текущий цикл).

Пример: Оператор EXIT

```
FOR i := 0 TO 10 DO
CASE value[i, 0] OF
  1..10: value [i, 1]:="A";
  11..40: value [i, 1]:="B";
  41..100: value [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

8.8.10.9 Оператор GOTO

Таблица 8- 163 Оператор GOTO

SCL	Описание
GOTO JumpLabel; Statement; ... ; JumpLabel: Statement;	Оператор GOTO позволяет выполнить переход к назначенной метке блока. Метка перехода ("JumpLabel") и оператор GOTO должны быть в одном блоке. Внутри одного блока имя метки должно быть уникальным. К одной метке могут выполнить переход несколько операторов GOTO.

Невозможно выполнить переход в секцию цикла (FOR, WHILE или REPEAT). Но возможно выйти по метке из цикла.

Пример: Оператор GOTO

В следующем примере: В зависимости от операнда "Tag_value", выполнение программы происходит с метки, на которую указывает данное значение. Если значение "Tag_value" равно 2, то выполнение программы начнется с метки "MyLabel2" и перепрыгнет метку "MyLabel1".

```

CASE "Tag_value" OF
1 : GOTO MyLabel1;
2 : GOTO
MyLabel2; ELSE
GOTO MyLabel3;
END_CASE;
MyLabel1: "Tag_1" :=
1; MyLabel2: "Tag_2"
:= 1; MyLabel3:
"Tag_4" := 1;

```

8.8.10.10 Оператор RETURN

Таблица 8- 164 Оператор RETURN

SCL	Описание
RETURN;	Оператор Return выполняет выход из блока независимо от состояний условия. Выполнение программы переходит к вызывающему блоку или к операционной системе (при выходе из ОВ).

Пример: Инструкция RETURN:

```

IF "Error" <> 0 THEN
RETURN;
END_IF;

```

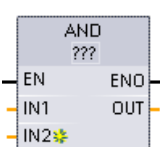
Примечание

После выполнения последней инструкции, выполнение программы автоматически переходит к вызывающему блоку. Не используйте инструкцию RETURN в конце блока.

8.9 Логические операции

8.9.1. Логические операции AND, OR, и XOR

Таблица 8- 165 Логические операции AND, OR, и XOR

LAD / FBD	SCL	Описание
	<code>out := in1 AND in2;</code>	AND: Логическое И
	<code>out := in1 OR in2;</code>	OR: Логическое ИЛИ
	<code>out := in1 XOR in2;</code>	XOR: Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.



Для добавления нового входа, щелкните левой клавишей мыши по пиктограмме "Create" или правой клавишей мыши на одном из уже существующих входов IN и нажмите "Insert input" (Добавить вход).

Для удаления входа, щелкните правой клавишей мыши на одном из уже существующих входов IN (при наличии более двух входов, созданных по умолчанию) и выберите "Delete"(Удалить).

Таблица 8- 166 Типы данных параметров

Параметр	Тип данных	Описание
IN1, IN2	Byte, Word, DWord	Логические входы
OUT	Byte, Word, DWord	Логические выходы

¹ Выбор типа данных устанавливает IN1, IN2, и OUT к одному типу.

Результат логической операции (AND, OR, XOR) параметров IN1 и IN2 передается в параметр OUT. ENO всегда TRUE при выполнении данных инструкций.

8.9.2. Инструкция INV (Инверсия)

Таблица 8- 167 Инструкция INV

LAD / FBD	SCL	Описание
	Недоступно	Инвертирует биты параметра IN. Данная инструкция инвертирует значение в параметре IN (изменяет каждый 0 на 1, а каждую 1 в 0). ENO всегда имеет состояние TRUE при выполнении данной инструкции.

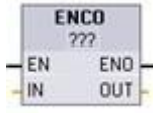
¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.

Таблица 8- 168 Типы данных параметров

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, UInt, UDInt, Byte, Word, DWord	Инвертируемый элемент
OUT	SInt, Int, DInt, UInt, UDInt, Byte, Word, DWord	Инвертированный элемент

8.9.3. Инструкции DECO (Расшифровать) и ENCO (Закодировать)

Таблица 8- 169 Инструкции ENCO и DECO

LAD / FBD	SCL	Описание
	<code>out := ENCO(_in_);</code>	Шифрует комбинацию битов в двоичное число. Инструкция ENCO преобразует параметр IN в двоичное число в соответствие с наименее значимой позицией бита в параметре IN и возвращает результат в параметр OUT. Если параметр IN равен либо 0000 0001 либо 0000 0000, тогда значение 0 будет передано в параметр OUT. Если значение параметра IN равно 0000 0000, тогда ENO будет иметь состояние FALSE.
	<code>out := DECO(_in_);</code>	Дешифрует двоичное число в комбинацию битов. Инструкция DECO дешифрует двоичное число из параметра IN, назначив соответствующую позицию бита в параметре OUT в 1 (все остальные биты устанавливаются в 0). ENO всегда будет иметь состояние TRUE при выполнении данной инструкции DECO. Примечание: По умолчанию тип данных для инструкции DECO это DWORD. В SCL, поменяйте название инструкции на DECO_BYTE или DECO_WORD для дешифровки байта или слова, и задайте параметр с соответствующим типом на выход OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.

Таблица 8- 170 Типы данных параметров

Параметр	Тип данных	Описание
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Комбинация битов для шифрования DECO: Значение для шифрования
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Зашифрованное значение DECO: Расшифрованная комбинация битов

Таблица 8- 171 Состояние ENO

ENO	Условие	Результат
1	Без ошибок	Верный результат
0	IN равен нулю	OUT равен нулю

Параметр OUT инструкции DECO с типом данных Byte, Word, или DWord запрещает допустимый диапазон значений параметра IN. Если значение параметра в IN превышает допустимый диапазон, то выполняется операция получения остатка от деления для выделения наименее значимых битов, как показано ниже.

Параметр IN инструкции DECO:

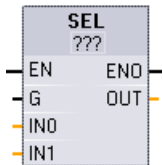
- 3 бита (значения 0-7) IN используются для установки 1 позиции бита в OUT с типом Byte
- 4 бита (значения 0-15) IN используются для установки 1 позиции бита в OUT с типом Word
- 5 бита (значения 0-31) IN используются для установки 1 позиции бита в OUT с типом DWord

Таблица 8- 172 Примеры

Значение DECO IN			Значение OUT инструкции DECO (Расшифровка одной позиции бита)
Byte OUT 8 бит	Мин. IN	0	00000001
	Макс. IN	7	10000000
Word OUT 16 бит	Мин. IN	0	0000000000000001
	Макс. IN	15	1000000000000000
DWord OUT 32 бит	Мин. IN	0	00000000000000000000000000000001
	Макс. IN	31	10000000000000000000000000000000

8.9.4. Инструкции SEL (Выбор), MUX (Мультиплексирование), и DEMUX (Демультимплексирование)

Таблица 8- 173 Инструкция SEL (выбор)

LAD / FBD	SCL	Описание
	<pre>out := SEL(g:=_bool_in, in0:=_variant_in, in1:=_variant_in);</pre>	SEL назначает одно из двух входных значений на выход OUT, в зависимости от значения G.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.

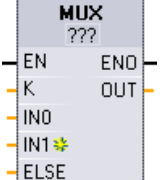
Таблица 8- 174 Типы данных для инструкции SEL

Параметр	Тип данных ¹	Описание
G	Bool	0 для выбора IN0 1 для выбора IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Входы
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выход

¹ Входные переменные должны быть с тем же типом данных, что и выходная переменная.

Коды состояния: ENO всегда имеет состояние TRUE при выполнении инструкции SEL.

Таблица 8- 175 Инструкция MUX (Мультиплексирование)

LAD / FBD	SCL	Описание
	<pre> out := MUX(k:=_unit_in, in1:=variant_in, in2:=variant_in, [...in32:=variant_in,] inelse:=variant_in); </pre>	MUX копирует одно из входных значений в параметр OUT, в зависимости от значения параметра K. Если параметр K превышает (INn - 1), тогда значение параметра ELSE копируется в OUT.

¹ Для LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.



Для добавления нового входа, щелкните левой клавишей мыши по пиктограмме "Create" или правой клавишей мыши на одном из уже существующих входов IN и нажмите "Insert input" (Добавить вход).

Для удаления входа, щелкните правой клавишей мыши на одном из уже существующих входов IN (при наличии более двух входов, созданных по умолчанию) и выберите "Delete"(Удалить).

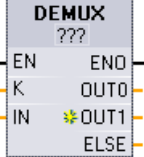
Для удаления входа, щелкните правой клавишей мыши на одном из уже существующих входов IN (при наличии более двух входов, созданных по умолчанию) и выберите "Delete"(Удалить).

Таблица 8- 176 Типы данных инструкции MUX

Параметр	Тип данных	Описание
K	UInt	0 для выбора IN1 1 для выбора IN2 - n для выбора INn
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Входы
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Заменяющее значение для входа (необязательно)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выход

¹ Входные переменные должны быть с тем же типом данных, что и выходная переменная.

Таблица 8- 177 Инструкция DEMUX (Демультимплексирование)

LAD / FBD	SCL	Описание
	<pre> DEMUX(k:=_unit_in, in:=variant_in, out1:=variant_in, out2:=variant_in, [...out32:=variant_in,] outelse:=variant_in); </pre>	DEMUX копирует значение со входа IN в один из нескольких выходов. Значение параметр K определяет какой из выходов будет выбран для значения IN. Если K больше чем количество (OUTn - 1) тогда значение в IN будет скопировано в параметр ELSE.

¹ LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.



Для добавления нового входа, щелкните левой клавишей мыши по пиктограмме "Create" или правой клавишей мыши на одном из уже существующих входов IN и нажмите "Insert input" (Добавить вход).

Для удаления входа, щелкните правой клавишей мыши на одном из уже существующих входов IN (при наличии более двух входов, созданных по умолчанию) и выберите "Delete"(Удалить).

Таблица 8- 178 Типы данных инструкций DEMUX

Параметр	Тип данных ¹	Описание
K	UInt	Селектор значений: • 0 для выбора OUT1 • 1 для выбора OUT2 • n для выбора OUTn
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Вход
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выходы
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Заменяющее значение для выхода, если K больше чем (OUTn - 1)

¹ Входные переменные должны быть с тем же типом данных, что и выходная переменная.

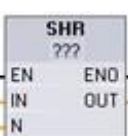
Таблица 8- 179 Состояние ENO инструкций MUX и DEMUX

ENO	Условие	Результат
1	Без ошибок	MUX: Выбранное значение в IN скопировано в OUT DEMUX: Значение в IN скопировано в выбранный OUT
0	MUX: К больше чем количество входов -1	- Условие ELSE не выполнилось: OUT остался без изменений, - Условие ELSE выполнилось, значение в ELSE записано в OUT
	DEMUX: К больше чем количество выходов -1	- Условие ELSE не выполнилось: выходы остались без изменений, - Условие ELSE выполнилось, значение в IN было скопировано в ELSE

8.10 Сдвиг и циклический сдвиг

8.10.1. Инструкции SHR (Сдвиг вправо) и SHL (Сдвиг влево)

Таблица 8- 180 Инструкции SHR и SHL

LAD / FBD	SCL	Описание
	<pre> out := SHR(in:=_variant_in_, n:=_uint_in_); out := SHL(in:=_variant_in_, n:=_uint_in_); </pre>	Используйте инструкции SHL и SHR для сдвига набора битов параметра IN. Результат сохраняется в параметр OUT. Параметр N определяет количество позиций бита для сдвига: - SHR: Сдвиг набора битов вправо - SHL: Сдвиг набора битов влево

¹ LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.

Таблица 8- 181 Типы данных параметров

Параметр	Тип данных	Описание
IN	Integers	Набор битов для сдвига
N	USInt, UDint	Количество позиций бита для сдвига
OUT	Integers	Набор битов после сдвига

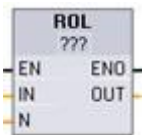
- Для N=0, сдвиг не выполняется. Значение IN сохраняется в OUT.
- Освободившиеся позиции заполняются нулями.
- Если количество позиций (N) для сдвига превышает размер типа выбранного значения (8 для Byte, 16 для Word, 32 для DWord), то все заполняется нулями (в OUT записывается ноль).
- Состояние ENO всегда TRUE при выполнении операций сдвига.

Таблица 8- 182 Пример: SHL для типа данных Word

Сдвиг битов типа Word влево, при помощи добавления нулей справа (N = 1)			
IN	1110 0010 1010 1101	Значение в OUT до первого сдвига:	1110 0010 1010 1101
		После первого сдвига влево:	1100 0101 0101 1010
		После второго сдвига влево:	1000 1010 1011 0100
		После третьего сдвига влево:	0001 0101 0110 1000

8.10.2. Инструкции ROR (Циклический сдвиг вправо) и ROL (Циклический сдвиг влево)

Таблица 8- 183 Инструкции ROR и ROL

LAD / FBD	SCL	Описание
	<pre> out := ROL(in:=_variant_in_, n:=_uint_in_); out := ROR(in:=_variant_in_, n:=_uint_in_); </pre>	Используйте инструкции ROR и ROL для выполнения циклического сдвига набора битов параметра IN. Результат записывается в параметр OUT. Параметр N определяет количество бит для циклического сдвига. - ROR: Циклический сдвиг набора битов вправо - ROL: Циклический сдвиг набора битов влево

¹ LAD и FBD: Нажмите на "???" и выберите тип данных из выпадающего меню.

Таблица 8- 181 Типы данных параметров

Параметр	Тип данных	Описание
IN	Integers	Набор битов для циклического сдвига
N	USInt, UDint	Количество позиций бита для циклического сдвига
OUT	Integers	Набор битов после сдвига

- Для N=0, циклический сдвиг не выполняется. Значение IN сохраняется в OUT.
- Битовые данные циклически сдвигаются и значения, которые выталкиваются с одной стороны вставляются с другой.
- Если количество позиций (N) для сдвига превышает размер типа выбранного значения (8 для Byte, 16 для Word, 32 для DWord), то циклический сдвиг все равно будет выполнен.
- Состояние ENO всегда TRUE при выполнении операций циклического сдвига.

Таблица 8- 185 Пример: ROR для типа данных Word

Циклический сдвиг битов с правой стороны влево (N = 1)			
IN	0100 0000 0000 0001	Значение OUT до первого циклического сдвига:	0100 0000 0000 0001
		После первого циклического сдвига:	1010 0000 0000 0000
		После второго циклического сдвига:	0101 0000 0000 0000

Расширенные инструкции

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

9.1.1. Инструкции "Date" (Дата) и "time-of-day" (Время дня)

Инструкции "date" и "time" используются для работы с форматами данных календаря и времени.

- T_CONV преобразует значения в (или из) типы данных "date" и "time" и размерные типы данных (байт, слово и двойное слово)
- T_ADD складывает значения "Time" и "DTL": (Time + Time = Time) или (DTL + Time = DTL)
- T_SUB вычитает значения "Time" и "DTL": (Time - Time = Time) или (DTL - Time = DTL)
- T_DIFF выводит разность между двумя DTL-значениями в виде значения времени "Time": DTL - DTL = Time
- T_COMBINE объединяет значения "Date" и "Time_and_Date" для формирования DTL значения

Для получения информации о формате данных "DTL" и "Time" обратитесь к разделу "Типы данных "Time" и "Date"" (Стр. 120).

Таблица 9-1 Инструкция T_CONV (Преобразование и извлечение значений времени)

LAD / FBD	Пример SCL	Описание
	<pre> out := DINT_TO_TIME(in:=_variant_in); out := TIME_TO_DINT(in:=_variant_in); </pre>	T_CONV преобразует значения в (или из) типы данных "date" и "time" и размерные типы данных (байт, слово и двойное слово)

¹ Для LAD и FBD блоков: Щелкните мышкой "???" и в выпадающем меню выберите начальный/конечный тип данных.

² Для SCL: Перетащите T_CONV из дерева инструкций в редактор программы, затем выберите начальный / конечный тип данных.

Таблица 9-2 Допустимые типы данных для T_CONV преобразований

Типы IN-данных (или OUT-данных)	Типы OUT-данных (или IN-данных)
TIME (миллисекунды)	DInt, Int, SInt, UDInt, UInt, USInt, TOD SCL только: Byte, Word, Dword
DATE (отсчет дней, начиная с 1 января 1990 года)	DInt, Int, SInt, UDInt, UInt, USInt, DTL SCL только: Byte, Word, Dword
TOD (отсчет миллисекунд, начиная с полуночи - 24:00:00.000)	DInt, Int, SInt, UDInt, UInt, USInt, TIME, DTL SCL только: Byte, Word, Dword

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Примечание

Использование T_CONV для преобразования больших объемов в меньшие объемы данных

При преобразовании типа данных, содержащего большое количество байт в тип данных с меньшим размером, значения данных могут быть испорчены. При возникновении этой ошибки ENO сбрасывается в 0.

Преобразование в(из) тип(а) данных DTL

DTL (Date and Time Long) содержит информацию о годе, месяце, дате и времени. DTL-данные могут быть преобразованы в(из) тип(а) данных DATE и TOD.

DTL-преобразование "DATE"-данных оказывает влияние только на значения года, месяца и дня. DTL-преобразование "TOD"-данных оказывает влияние только на значения часов, минут и секунд.

При преобразовании T_CONV в DTL незадействованные элементы данных в DTL-формате остаются без изменений.

Таблица 9-3 Инструкции T_ADD (Add times = время выполнения операции сложения) и T_SUB (Subtract times = время выполнения операции вычитания)

LAD / FBD	SCL	Описание
	<pre>out := T_ADD(in1:=_variant_in, in2:=_time_in);</pre>	T_ADD прибавляет значение на входе IN1 (типы данных DTL или Time) к значению времени на входе IN2. Параметр OUT выводит результат в виде значений DTL или Time. Возможные сочетания двух типов данных: - Time + Time = Time - DTL + Time = DTL
	<pre>out := T_SUB(in1:=_variant_in, in2:=_time_in);</pre>	T_SUB вычитает значение времени на входе IN2 из значения времени на входе IN1 (значения DTL или Time). Параметр OUT выводит разность значений как тип данных DTL или Time. Возможные сочетания двух типов данных: - Time - Time = Time - DTL - Time = DTL

¹ Для LAD и FBD: Щелкните мышкой на "???" и в выпадающем меню выберите тип данных.

Таблица 9-4 Типы данных для параметров T_ADD и T_SUB

Параметр и тип	Тип данных	Описание
IN1 ¹	IN	DTL, Time
IN2	IN	Time
OUT	OUT	DTL, Time

¹ Выберите тип данных IN1 из выпадающего списка под именем инструкции. Выбор типа данных IN1 также определяет тип данных параметра OUT.

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Таблица 9-5 Инструкция T_DIFF (Time difference = Разность значений времени)

LAD / FBD	SCL	Описание
	<pre>out := T_DIFF(in1:=_DTL_in, in2:=_DTL_in);</pre>	T_DIFF вычитает DTL-значение (IN2) из DTL-значения (IN1). Параметр OUT выводит разность значений в виде типа данных Time. DTL - DTL = Time

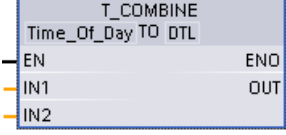
Таблица 9-6 Типы данных для параметров T_DIFF

Параметр и тип		Тип данных	Описание
IN1	IN	DTL	DTL-значение
IN2	IN	DTL	DTL-значение для вычитания
OUT	OUT	Time	Разность, как тип данных Time

Коды сообщений: ENO = 1 означает, что ошибки отсутствуют. ENO = 0 и параметр OUT = 0 сигнализируют о наличии ошибок:

- Недопустимое DTL-значение
- Недопустимое Time-значение

Таблица 9-7 Инструкция T_COMBINE (Combine times = объединение значений времени)

LAD / FBD	SCL	Описание
	<pre>out := CONCAT_DATE_TOD(In1 := _date_in, In2 := _tod_in);</pre>	T_COMBINE используется для объединения значений Date и Time_of_Day при создании DTL-значения.

¹ Помните, что инструкция T_COMBINE в расширенных инструкциях (Extended Instructions) эквивалентна функции CONCAT_DATE_TOD в SCL.

Таблица 9-8 Типы данных для параметров T_COMBINE

Параметр и тип		Тип данных	Описание
IN1	IN	Date	Значение даты, используемое для объединения, должно находиться в пределах между DATE#1990-01-01 и DATE#2089-12-31
IN2	IN	Time_of_Day	Значение времени дня (Time_of_Day), используемое для объединения
OUT	OUT	DTL	DTL-значение

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

9.1.2. Функции часов (Clock)



ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник получит доступ к Вашей сети через протокол сетевого времени Network Time Protocol (NTP), с помощью которого в Internet производится синхронизация системного времени компьютера пользователя с системным временем сервера, то он может влиять на управление Вашим процессом, сдвигая системное время CPU.

По умолчанию свойства NTP-клиента S7-1200 CPU не активированы, а при включении позволяют сконфигурировать только IP-адреса для работы в качестве NTP-сервера. По умолчанию это свойство CPU отключено, и Вам необходимо его сконфигурировать, чтобы иметь возможность корректировать системное время дистанционно-управляемого CPU.

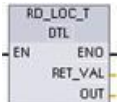
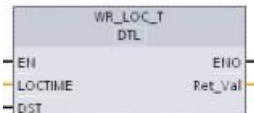
S7-1200 CPU поддерживает прерывания по времени суток ("time of day") и инструкции установки часов (clock instructions), которые зависят от точности системного времени CPU. При конфигурировании NTP и синхронизации с системным временем сервера Вам необходимо убедиться, что сервер является надежным источником. Невыполнение этого требования может стать причиной нарушения безопасности, что позволит неизвестному пользователю получить ограниченное управление Вашим процессом путем сдвига системного времени CPU.

Для получения информации и рекомендаций по обеспечению безопасности обратитесь к руководству "Operational Guidelines for Industrial Security" (Рабочие директивы по промышленной безопасности), доступному на сайте поддержки Siemens (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf).

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Использование clock-инструкций для установки и чтения системного времени CPU. Для отображения значений даты и времени используется тип данных DTL (стр. 120).

Таблица 9-9 Инструкции системного времени

LAD / FBD	SCL	Описание
	<pre>ret_val := WR_SYS_T(in:=_DTL_in_);</pre>	WR_SYS_T (Set time-of-day = установка времени суток) устанавливает время суток CPU с DTL-значением параметра IN. Это значение не учитывает локальный часовой пояс или сдвиг часов при переходе на "декретное" время.
	<pre>ret_val := RD_SYS_T(out=>_DTL_out);</pre>	RD_SYS_T (Read time-of-day = чтение времени суток) считывает фактическое системное время из CPU. Это значение не учитывает локальный часовой пояс или сдвиг часов при переходе на "декретное" время.
	<pre>ret_val := RD_LOC_T(out=>_DTL_out);</pre>	RD_LOC_T (Read local time = чтение значений локального времени) выводит фактическое время CPU в виде DTL-данных. В этом скорректированном значении отражено время локального часового пояса с учетом перехода на декретное время (если сконфигурировано).
	<pre>ret_val := WR_LOC_T(LOCTIME:=DTL_in_, DST:=_in_);</pre>	WR_LOC_T (Write local time = запись значений локального времени) устанавливает дату и время часов CPU. Информацию о дате и времени Вы устанавливаете как местное время LOCTIME с типом данных DTL. Для расчета системного времени инструкция использует "TimeTransformationRule" (стр. 322) DB-структуры. Информация о локальном времени и системном времени конкретного изделия отображается с точностью до одной миллисекунды. Входные значения параметра LOCTIME, которые меньше поддерживаемых CPU, при расчете системного времени округляются в сторону увеличения. Примечание: Для установки параметров "Time of day" (часовой пояс, активация DST, запуск DST, останов DST) Вы должны использовать конфигурацию устройств CPU. В противном случае, WR_LOC_T не сможет распознать переход на декретное время.

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Таблица 9-10 Типы данных для параметров

Параметр и тип		Тип данных	Описание
IN	IN	DTL	Время суток для установки в системных часах CPU
OUT	OUT	DTL	RD_SYS_T: Фактическое системное время CPU RD_LOC_T: Фактическое местное время, с учетом корректировки на декретное время, если сконфигурировано
LOCTIME	IN	DTL	WR_LOC_T: Местное время
DST	IN	BOOL	WR_LOC_T: Оценка перехода на декретное время (Daylight Saving Time) выполняется только во время "двойного часа" при переводе часов на декретное время. • TRUE = декретное время (первый час) • FALSE = обычное время (второй час)
RET_VAL	OUT	Int	Выполнение кода условия

- Местное время рассчитывается с учетом часового пояса и сдвига часов на декретное время, параметры которых Вы задали на вкладке "Time of day" (Время суток) конфигурации устройств.
- Конфигурирование часового пояса – это сдвиг UTC- или GMT-времени.
- При конфигурировании летнего времени задается месяц, неделя, день и час, с которых начинается декретное время.
- При конфигурировании стандартного времени также задается месяц, неделя, день и час, с которых начинается обычное время.
- Сдвиг системного времени с учетом часового пояса выполняется всегда. Сдвиг системного времени на декретное время применяется только на период действия летнего времени.

Примечание

Конфигурирование перехода на декретное время и обратно

В свойствах "Time of day" (Время суток) для "Start for daylight saving time" (Переход на декретное время) конфигурации устройств CPU device должно быть установлено местное время.

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Коды сообщений: ENO = 1 означает, что ошибки отсутствуют. ENO = 0 сигнализирует о наличии ошибки, а на выходе RET_VAL выводится код сообщения.

RET_VAL (W#16#....)	Описание
0000	Значение текущего времени соответствует стандартному времени.
0001	Декретное время сконфигурировано, и текущее время соответствует декретному времени.
8080	Местное время недоступно или недействительное значение LOCTIME.
8081	Недействительное значение года или недопустимое значение времени, назначенное параметру LOCTIME
8082	Недействительное значение месяца (байт 2 в DTL-формате)
8083	Недействительное значение дня (байт 3 в DTL-формате)
8084	Недействительное значение часов (байт 5 в DTL-формате)
8085	Недействительное значение минут (байт 6 в DTL-формате)
8086	Недействительное значение секунд (байт 7 в DTL-формате)
8087	Недействительное значение наносекунд (байты с 8 по 11 в DTL-формате)
8089	Значение времени не существует (после перехода на декретное время прошел уже час)
80B0	Сбой часов реального времени
80B1	Невозможно определить структуру "TimeTransformationRule".

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

9.1.3. Структура данных TimeTransformationRule

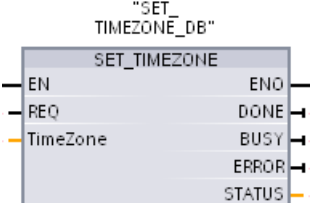
Описание

В структуре TimeTransformationRule определены правила переключения стандартного и декретного времени. Ниже приведено описание структуры:

Наименование	Тип данных	Описание
TimeTransformationRule	STRUCT	
Bias	INT	Разность между местным временем и UTC [в минутах] Диапазон значений: от -1439 до 1439
DaylightBias	INT	Разность между декретным временем и стандартным временем [в минутах] Диапазон значений: от 0 до 60
DaylightStartMonth	USINT	Месяц, в котором выполняется переход на декретное время Диапазон значений: от 1 до 12
DaylightStartWeek	USINT	Неделя перехода на декретное время 1 = Переход в течение первой недели месяца, ..., 5 = Переход в течение последней недели месяца
DaylightStartWeekday	USINT	День недели перехода на декретное время: 1 = Воскресенье
DaylightStartHour	USINT	Час перехода на декретное время: Диапазон значений: от 0 до 23
DaylightStartMinute	USINT	Минута перехода на декретное время: Диапазон значений: от 0 до 59
StandardStartMonth	USINT	Месяц перехода на обычное время Диапазон значений: от 1 до 12
StandardStartWeek	USINT	Неделя перехода на обычное время 1 = Переход в течение первой недели месяца, ..., 5 = Переход в течение последней недели месяца
StandardStartWeekday	USINT	День недели перехода на обычное время: 1 = Воскресенье
StandardStartHour	USINT	Час перехода на обычное время: Диапазон значений: от 0 до 23
StandardStartMinute	USINT	Минута перехода на обычное время: Диапазон значений: от 0 до 59
TimeZoneName	STRING[80]	Название часового пояса: "(GMT+01:00) Амстердам, Берлин, Берн, Рим, Стокгольм, Вена"

9.1.4. Инструкция SET_TIMEZONE (установка часового пояса)

Таблица 9-11 Инструкция SET_TIMEZONE

LAD / FBD	SCL	Описание
	<pre>"SET_TIMEZONE_DB"(REQ:=_bool_in, Timezone:=_struct_in, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	Установка параметров часового пояса и перехода на декретное время, которые используются для преобразования системного времени CPU в местное время.

¹ В SCL-примере, "SET_TIMEZONE_DB" – это имя экземпляра блока данных.

Таблица 9-12 Тип данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	REQ=1: выполнение функции
Timezone	IN	TimeTransformationRule	Правила перехода с системного времени на местное время
DONE	OUT	Bool	Функция выполнена
BUSY	OUT	Bool	Функция выполняется
ERROR	OUT	Bool	Обнаружена ошибка
STATUS	OUT	Word	Результат выполнения функции / сообщение об ошибке

При ручном конфигурировании параметров часового пояса для CPU используйте свойства "Time of day" (Время дня) вкладки "General" (Основные параметры) конфигурации устройств.

При программном конфигурировании параметров местного времени используйте инструкцию SET_TIMEZONE. В параметрах структуры "TimeTransformationRule (стр. 322)" задается часовой пояс и время для перехода с обычного времени на декретное время.

Коды сообщений: ENO = 1 означает, что ошибки отсутствуют. ENO = 0 сигнализирует о наличии ошибки, а на выходе STATUS выводится код сообщения.

STATUS (W#16#....)	Описание
0	Ошибки отсутствуют.
7000	Не активных заданий на обработку.
7001	Запуск задания на обработку. Параметры BUSY = 1, DONE = 0
7002	Промежуточный вызов (REQ не существует): Инструкция уже активна; BUSY имеет значение "1".
808x	Ошибка компонента "X": Например, 8084 означает, что DaylightStartWeekif это не значение от 1 до 5.

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

9.1.5. Инструкция RTM (счетчик рабочего времени)

Таблица 9-13 RTM-инструкция

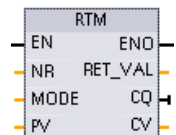
LAD / FBD	SCL	Описание
	<pre>RTM(NR:=_uint_in_, MODE:=_byte_in_, PV:=_dint_in_, CQ=>_bool_out_, CV=>_dint_out_);</pre>	Инструкция RTM (счетчик рабочего времени) может устанавливать, запускать, останавливать и считывать информацию счетчиков рабочего времени в CPU.

Таблица 9-14 Типы данных для параметров

Параметр и тип	Тип данных	Описание
NR	IN	UInt
MODE	IN	Byte
PV	IN	DInt
RET_VAL	OUT	Int
CQ	OUT	Bool
CV	OUT	DInt

CPU обрабатывает значения до 10 счетчиков рабочего времени, отслеживая длительность рабочего цикла критических подсистем управления. Для каждого таймера Вам необходимо запустить отдельный счетчик рабочего времени с помощью режима исполнения RTM-инструкции. Все счетчики рабочего времени будут остановлены при переходе CPU из RUN в STOP.

С помощью режима 2 исполнения RTM-инструкции Вы можете остановить отдельные таймеры.

При выполнении перехода CPU из STOP в RUN Вам необходимо перезапустить таймеры с одним режимом исполнения RTM-инструкции для каждого из запускаемых таймеров. При превышении значения счетчика рабочего времени более 2147483647 часов процесс счета останавливается и передается сообщение об ошибке "Overflow" (Переполнение). Для сброса таймера или изменения его значения Вам необходимо выполнить RTM-инструкцию один раз для каждого из таймеров.

При ошибке питания CPU, вызывающей периодическое отключение питания, фактические значения счетчиков рабочего времени записываются в сохраняемую область памяти. При восстановлении питания CPU в таймеры загружаются сохраненные значения счетчиков рабочего времени, т.е. суммарные значения времени наработки не теряются. Для выполнения дальнейшего суммирования рабочего времени счетчики должны быть перезапушены.

9.1 Функции "Date" (Дата), "time-of-day" (Время дня) и "clock" (часы)

Для сохранения значений счетчика рабочего времени на карте памяти Вам необходимо запрограммировать использование режима 7 исполнения RTM-инструкции. При использовании режима 7 исполнения RTM-инструкции RTM все значения будут сохранены на карте памяти. Со временем эти сохраненные значения могут стать недействительными, т.к. при запуске сеансов программы таймеры запускаются и останавливаются. Вам необходимо периодически обновлять содержимое карты памяти для фиксации последних значений рабочего времени. Преимущество от хранения RTM-значений на карте памяти состоит в том, что можно вставить карту памяти в другой CPU, в котором будет доступна Ваша пользовательская программа и сохраненные RTM-значения. Если Вы не сохранили RTM-значения на карте памяти, то значения счетчиков будут утрачены (при замене CPU).

Примечание

Не допускайте выполнения слишком большого количества операций записи на карту памяти

Минимизируйте количество операций записи на flash-память для продления времени использования карты памяти.

Таблица 9-15 Коды сообщений

RET_VAL (W#16#....)	Описание
0	Ошибки отсутствуют
8080	Некорректный номер счетчика рабочего времени
8081	В параметр PV было записано отрицательное значение
8082	Переполнение счетчика рабочего времени
8091	Входной параметр MODE содержит недопустимое значение
80B1	Значение не может быть сохранено в MC (MODE=7)

9.2 Строка и символ

9.2.1. Обзор строковых данных

Тип строковых данных

Строковые данные (String data) представляют собой заголовок размером 2 байта, за которым следует до 254 байт символов в ASCII-коде. Заголовок строки состоит из двух байт. Первый байт – это максимальная длина, которая указывается в квадратных скобках при инициализации строки, или 254 по умолчанию. Второй байт заголовка – это фактическая длина, т.е. количество допустимых символов в строке. Фактическая длина должна быть не больше максимально допустимой длины. Количество сохраненных байтов, занимаемых форматом строки (String format) на 2 байта больше, чем максимальная длина.

Инициализация Ваших строковых данных

Перед выполнением любых строковых инструкций вход и выход строковых данных должны быть инициализированы в памяти в виде действительных строк.

Действительные строковые данные

Действительная строка должна иметь длину больше чем ноль, но меньше, чем 255. Фактическая длина должна быть не больше максимальной длины.

Строки не могут быть назначены областям памяти входов или выходов (I или Q).

Дополнительная информация: Формат данных String (строка) (стр. 122).

9.2.2. Инструкция S_MOVE (Перемещение символов строки)

Таблица 9-16 Инструкция перемещения строки

LAD / FBD	SCL	Описание
	<code>out := in;</code>	Копирование исходной IN-строки в OUT-ячейку. Выполнение инструкции S_MOVE не влияет на содержимое исходной строки.

Таблица 9-17 Тип данных для параметров

Параметр	Тип данных	Описание
IN	String	Исходная строка
OUT	String	Целевой адрес

Если фактическая длина строки на входе IN превышает максимальную длину строки, сохраненную на выходе OUT, то часть IN-строки, которая не может поместиться в OUT-строку, копируется.

9.2.3. Инструкции преобразования строки

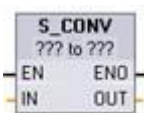
9.2.3.1. Инструкции S_CONV, STRG_VAL и VAL_STRG (Конвертирование в(из) строку символов или числовые значения)

С помощью этих инструкций Вы можете преобразовывать строки из числовых символов в числовые значения или числовые значения в строки из числовых символов:

- S_CONV преобразует числовую строку в числовое значение или числовое значение в числовую строку
- STRG_VAL преобразует числовую строку в числовое значение с опциями форматирования
- VAL_STRG преобразует числовое значение в числовую строку с опциями форматирования

S_CONV (преобразование строки символов)

Таблица 9-18 Инструкция преобразования строки

LAD / FBD	SCL	Описание
	<pre>out := <Type>_TO_<Type>(in);</pre>	Конвертирование строки символов в соответствующее значение, или значения в соответствующую строку символов. Инструкция S_CONV не содержит выходных опций форматирования. Это делает инструкцию S_CONV более простой, но менее гибкой, чем инструкции STRG_VAL и VAL_STRG.

- ¹ Для LAD / FBD: Щелкните мышкой на "???" и в выпадающем меню выберите тип данных.
- ² Для SCL: Выберите S_CONV из окна расширенных инструкций (Extended Instructions) и подтвердите преобразование типов данных. STEP 7 содержит соответствующие инструкции преобразования.

Таблица 9-19 Типы данных (строка в значение)

Параметр и тип		Тип данных	Описание
IN	IN	String, WString	Входная строка символов
OUT	OUT	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Выходная строка символов

Конвертирование параметров IN-строки начинается с первого символа и продолжается до конца строки, или пока не будет обнаружен первый символ, кроме 0...9, "+", "-" или ".". Результирующее значение выводится в ячейке, указанной в параметре OUT. Если выходное числовое значение выходит за пределы диапазона OUT-данных, то параметр OUT сбрасывается в 0, а ENO принимает значение FALSE. С другой стороны, если параметр OUT содержит достоверный результат, то ENO принимает значение TRUE.

Правила входного формата String:

- Если в IN-строке используется десятичная точка, то Вы должны использовать символ ".".
- Символ, слева от десятичной точки допускается и игнорируется при конвертации.
- Предшествующие пробелы слева игнорируются.

9.2 Строка и символ

S_CONV (конвертирование значения в строку)

Таблица 9-20 Типы данных (конвертирование значения в строку)

Параметр и тип		Тип данных	Описание
IN	IN	String, WString, Char, WChar, Sint, Int, DInt, USInt, UInt, UDInt, Real, LReal	Входное числовое значение
OUT	OUT	String, WString	Выходная строка символов

Число, целое число без знака, или IN-значение с плавающей точкой преобразуется в соответствующую строку символов в параметре OUT. Перед началом конвертирования параметр OUT должен получить ссылку на действительную строку. В первом байте действительной строки содержится информация о максимальной длине строки, во втором байте – фактическая длина строки, а в остальных байтах – текущие символы строки. Конвертированная строка заменяет символы в OUT-строке, начиная с первого символа, и корректирует фактическую длину OUT-строки в байтах. Максимальная длина OUT-строки в байтах не изменяется.

Сколько байтов будет заменено, зависит от типа данных параметра IN и числового значения. Количество заменяемых символов должно находиться в пределах длины строки параметра OUT. Максимальная длина OUT-строки (первый байт) должна быть не меньше максимального предполагаемого числа конвертированных символов. В приведенной ниже таблице приведены значения S_CONV для примеров преобразования строки.

Правила выходного формата String:

- Символ "+" не используется при записи значений в параметр OUT.
- Представление чисел с фиксированной точкой (не экспоненциальное).
- Использование символа "." для представления десятичной точки, если параметр IN имеет тип данных Real.
- Выходные символы выравниваются по правому краю, а свободные позиции слева заполняются нулями.

Таблица 9-21 Максимальная длина строки для каждого типа данных

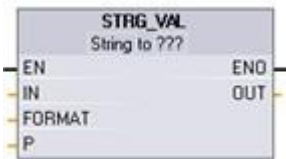
Тип IN-данных	Позиция символа, определяемая S_CONV	Пример конвертированной строки ¹	Суммарная длина строки в байтах, включая максимальную и фактическую длину строки
USInt	4	"x255"	6
SInt	4	"-128"	6
UInt	6	"x65535"	8
Int	6	"-32768"	8
UDInt	11	"x4294967295"	13
DInt	11	"-2147483648"	13

Тип IN-данных	Позиция символа, определяемая S_CONV	Пример конвертированной строки ¹	Суммарная длина строки в байтах, включая максимальную и фактическую длину строки
Real	14	"x-3.402823E+38" "x-1.175495E-38" "x+1.175495E-38" "x+3.402823E+38"	16
LReal	21	"-1.7976931348623E+308" "-2.2250738585072E-308" "+2.2250738585072E-308" "+1.7976931348623E+308"	23

¹ Символы "x" представляют собой пробелы, которыми заполняют пустые позиции при выравнивании по правому краю поля, выделенного для преобразованного значения.

STRG_VAL (преобразование символьной строки в числовое значение)

Таблица 9-22 Инструкция STRG_VAL (преобразование строки в значение)

LAD / FBD	SCL	Описание
	<pre>"STRG_VAL"(in:=string_in, format:=word_in, p:=uint_in, out=>_variant_out);</pre>	Преобразует знаки символьной строки в соответствующее целое число или число с плавающей точкой.

¹ Для LAD / FBD: Щелкните мышкой на "???" и в выпадающем меню выберите тип данных.

Таблица 9-23 Типы данных для инструкции STRG_VAL

Параметр и тип	Тип данных	Описание
IN	String, WString	Строка ASCII-символов для преобразования
FORMAT	Word	Опции формата вывода
P	UInt, Byte, USInt	IN: Индекс первого символа, с которого начнется преобразование (первый символ = 1)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Преобразованное числовое значение

Преобразование начинается с символа IN-строки, сдвинутого на "P", и продолжается до конца строки, или пока не будет обнаружен первый символ, кроме "+", "-", ".", ",", "e", "E", или от "0" до "9". Результирующее значение выводится в ячейке, указанной в параметре OUT.

До выполнения преобразования строковые данные должны быть инициализированы в памяти в виде действительной строки.

9.2 Строка и символ

Ниже приведено описание параметра FORMAT для инструкции STRG_VAL. В неиспользуемые позиции битов должны быть записаны нули.

Таблица 9-24 Формат инструкции STRG_VAL

Бит 16							Бит 8	Бит 7							Бит 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

f = Формат представления

1 = Экспоненциальное представление

0 = Представление с фиксированной точкой

r = Формат с десятичной точкой

1 = "," (символ "запятая")

0 = "." (символ "точка")

Таблица 9-25 Значения параметра FORMAT

FORMAT (W#16#)	Формат представления	Представление десятичной точки
0000 (по умолчанию)	С фиксированной точкой (Fixed point)	"."
0001		","
0002	Экспоненциальное представление (Exponential)	"."
0003		","
с 0004 по FFFF	Недействительные значения	

Правила преобразования для STRG_VAL:

- Если для отображения десятичной точки используется символ ".", то символ запятой "," слева от десятичной точки интерпретируется как разделители тысяч. Символы запятой допускаются и игнорируются.
- Если для отображения десятичной точки используется символ запятой ",", то символы "." слева от десятичной точки интерпретируется как разделители тысяч. Эти символы допускаются и игнорируются.
- Пробелы игнорируются.

VAL_STRG (преобразование числового значения в строку)

Таблица 9-26 Инструкция VAL_STRG (преобразование числового значения в строку)

LAD / FBD	SCL	Описание
	<pre>"VAL_STRG" (in:=_variant_in, size:=_usint_in, prec:=_usint_in, format:=_word_in, p:=uint_in, out=>_string_out);</pre>	Преобразование целого числа, целого числа без знака или значения с плавающей точкой в соответствующее представление в виде символьной строки.

¹ Для LAD / FBD: Щелкните мышкой на "???" и в выпадающем меню выберите тип данных.

Таблица 9-27 Типы данных для инструкции VAL_STRG

Параметр и тип		Тип данных	Описание
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Значение для преобразования
SIZE	IN	USInt	Количество символов, записанных в OUT-строку
PREC	IN	USInt	Точность или размер дробной части. Не включает в себя десятичную точку.
FORMAT	IN	Word	Опции формата вывода
P	IN	UInt, Byte, USInt	IN: Индекс первого символа OUT-строки, с которого начнется преобразование (первый символ = 1)
OUT	OUT	String, WString	Преобразованная строка

Инструкция преобразует значение, соответствующее параметру IN, в строку, соответствующую параметру OUT. Перед выполнением преобразования параметр OUT должен быть инициализирован в виде действительной строки.

Преобразованная строка заменяет символы OUT-строки, начиная с символа, определяемого сдвигом "P" до номера символа, определяемого параметром SIZE. Количество символов, определяемых параметром SIZE, должны находиться в пределах длины OUT-строки, начиная с символа, соответствующего позиции "P". Если значение параметра SIZE равно нулю, то символы перезаписываются, начиная с позиции "P" в OUT-строке без ограничения. Эта инструкция используется для вставки числовых символов в текстовую строку. Например, можете вставить числа "120" в строку "Pump pressure = 120 psi" ("Давление насоса = 120 фунтов/дюйм²")

Параметр PREC определяет точность или количество цифр в дробной части строки. Если значение параметра IN представляет собой целое число, то параметр PREC указывает расположение десятичной точки. Например, если исходное значение "123", а PREC = 1, то результат – "12.3". Максимальная поддерживаемая точность для типа данных Real – 7 знаков.

Если значение параметра "P" превышает фактический размер OUT-строки, то размер добавляется, начиная с позиции "P", а результат добавляется в конец строки. Преобразование прекращается, если достигнут максимальный размер OUT-строки.

9.2 Строка и символ

Ниже приведено описание параметра FORMAT для инструкции VAL_STRG. В неиспользуемые позиции битов должны быть записаны нули.

Таблица 9-28 Формат инструкции VAL_STRG

Бит 16							Бит 8	Бит 7							Бит 0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

s = Используемые знаковые символы 1= использование символов "+" и "-"

0 = использование только символа "-"

f = Формат представления

1= Экспоненциальное представление

0 = Представление с фиксированной точкой

r = Формат десятичной точки

1 = "," (символ "запятая")

0 = "." (символ "точка")

Таблица 9-29 Значения параметра FORMAT

FORMAT (WORD)	Используемые знаковые символы	Формат представления	Представление десятичной точки
W#16#0000	Только "-"	Фиксированная точка (Fixed point)	"."
W#16#0001			","
W#16#0002		Экспоненциальное представление (Exponential)	"."
W#16#0003			","
W#16#0004	"+" и "-"	Фиксированная точка (Fixed point)	"."
W#16#0005			","
W#16#0006		Экспоненциальное представление (Exponential)	"."
W#16#0007			","
с W#16#0008 по W#16#FFFF	Недействительные значения		

Правила форматирования параметров OUT-строки:

- Если преобразованная строка меньше заданного размера, то в левую часть строки добавляются пробелы.
- Если знаковый бит параметра FORMAT – FALSE, то значения целочисленных типов данных (со знаком или без знака) записываются в выходной буфер без ввода символа "+". Если необходимо, то используется символ "-".
<Пробелы><Цифры без добавления нулей>'.<Дробная часть (PREC)>
- Если знаковый бит параметра FORMAT – TRUE, то значения целочисленных типов данных (со знаком или без знака) записываются в выходной буфер всегда с вводом символа знака.
<Пробелы><Знак><Цифры без добавления нулей>'.<Дробная часть (PREC)>
- Если параметром FORMAT задано экспоненциальное представление, то значения типа данных Real записываются в выходной буфер в виде:
<Пробелы><Знак><Цифра>'.<Дробная часть (PREC)>'E' <Знак><Цифры без добавления нулей>
- Если параметром FORMAT задано представление с фиксированной точкой, то целые числа, целые числа без знака и значения типа данных REAL записываются в выходной буфер в виде:
<Пробелы><Знак><Цифры без добавления нулей>'.<Дробная часть (PREC)>
- Нули, вводимые слева от десятичной точки (кроме цифры, находящейся рядом с десятичной точкой), не отображаются.
- Значения, расположенные справа от десятичной точки, округляются до количества цифр справа от десятичной точки, заданного параметром PREC.
- Размер выходной строки должен быть больше количества цифр, расположенных справа от десятичной точки, не менее, чем на 3 байта.
- Значения выравниваются по правому краю выходной строки.

9.2 Строка и символ

Условия, передаваемые ENO

Если при выполнении операции преобразования возникает ошибка, то инструкция возвращает следующие результаты:

- ENO сбрасывается в 0.
- OUT устанавливается в 0, или, как показано в примере для строки, или в преобразованное значение.
- OUT остается без изменений, или, как показано в примерах, если OUT – строка.

Таблица 9-30 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Недопустимое или недействительное значение параметра; например, доступ к несуществующему DB.
0	Недопустимая максимальная длина строки (0 или 255).
0	Недействительная строка, когда фактическая длина больше максимальной длины.
0	Преобразованное значение больше значения типа данных, заданного в параметре OUT.
0	Максимальная длина строки параметра OUT должна быть достаточно большой для доступа к количеству символов, заданных в параметре SIZE, начиная с позиции символа, заданного в параметре "P".
0	Недействительное значение параметра "P", когда P=0 или P больше фактической длины строки.
0	Значение параметра SIZE должно быть больше значения параметра PREC.

Таблица 9-31 Примеры преобразования строки S_CONV в значение

Строка IN	Тип данных OUT	Значение OUT	ENO
"123"	Int или DInt	123	TRUE
"-00456"	Int или DInt	-456	TRUE
"123.45"	Int или DInt	123	TRUE
"2345"	Int или DInt	2345	TRUE
"00123AB"	Int или DInt	123	TRUE
"123"	Real	123.0	TRUE
"123.45"	Real	123.45	TRUE
"1.23e-4"	Real	1.23	TRUE
"1.23E-4"	Real	1.23	TRUE
"12,345.67"	Real	12345.67	TRUE
"3.4e39"	Real	3.4	TRUE
"-3.4e39"	Real	-3.4	TRUE
"1.17549e-38"	Real	1.17549	TRUE
"12345"	SInt	0	FALSE
"A123"	N/A	0	FALSE
""	N/A	0	FALSE
"++123"	N/A	0	FALSE
"+-123"	N/A	0	FALSE

Таблица 9-32 Примеры преобразования значения S_CONV в строку

Тип данных	Значение IN	Строка OUT ¹	ENO
UInt	123	"xxx123"	TRUE
UInt	0	"xxxxx0"	TRUE
UDInt	12345678	"xxx12345678"	TRUE
Real	+9123.456	"xx+9.123456E+3"	TRUE
LReal	+9123.4567890123	"xx+9.1234567890123 E+3"	TRUE
Real	-INF	"xxxxxxxxxxxINF"	FALSE
Real	+INF	"xxxxxxxxxxxINF"	FALSE
Real	NaN	"xxxxxxxxxxxNaN"	FALSE

¹ Символы "x" представляют собой пробелы, которые заполняют пустые позиции справа от поля, выделенного для преобразованного значения.

Таблица 9-33 Пример: Преобразование STRG_VAL

Строка IN	FORMAT (W#16#....)	Тип данных OUT	Значение OUT	ENO
"123"	0000	Int или DInt	123	TRUE
"-00456"	0000	Int или DInt	-456	TRUE
"123.45"	0000	Int или DInt	123	TRUE
" +2345"	0000	Int или DInt	2345	TRUE
"00123AB"	0000	Int или DInt	123	TRUE
"123"	0000	Real	123.0	TRUE
"-00456"	0001	Real	-456.0	TRUE
" +00456"	0001	Real	456.0	TRUE
"123.45"	0000	Real	123.45	TRUE
"123.45"	0001	Real	12345.0	TRUE
"123.45"	0000	Real	12345.0	TRUE
"123.45"	0001	Real	123.45	TRUE
".00123AB"	0001	Real	123.0	TRUE
"1.23e-4"	0000	Real	1.23	TRUE
"1.23E-4"	0000	Real	1.23	TRUE
"1.23E-4"	0002	Real	1.23E-4	TRUE
"12,345.67"	0000	Real	12345.67	TRUE
"12,345.67"	0001	Real	12.345	TRUE
"3.4e39"	0002	Real	+INF	TRUE
"-3.4e39"	0002	Real	-INF	TRUE
"1.1754943e-38" (и меньше)	0002	Real	0.0	TRUE
"12345"	N/A	SInt	0	FALSE
"A123"	N/A	N/A	0	FALSE
""	N/A	N/A	0	FALSE

9.2 Строка и символ

Строка IN	FORMAT (W#16#....)	Тип данных OUT	Значение OUT	ENO
"++123"	N/A	N/A	0	FALSE
"+-123"	N/A	N/A	0	FALSE

Следующие примеры преобразований VAL_STRG основаны на строке OUT, инициализированной следующим образом:

"Current Temp = xxxxxxxxx C",

где символы "x" представляют собой пробелы, вставляемые в конвертированное значение.

Таблица 9-34 Пример: Преобразование VAL_STRG

Тип данных	Значение IN	P	SIZE	FORMAT (W#16#....)	PREC	Строка OUT	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxx123 C	TRUE
UInt	0	16	10	0000	2	Current Temp = xxxxx0.00 C	TRUE
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	TRUE
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	TRUE
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	TRUE
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	TRUE
Real	-0.00123	16	10	0004	4	Current Temp = xxx- 0.0012 C	TRUE
Real	-0.00123	16	10	0006	4	Current Temp = - 1.2300E-3 C	TRUE
Real	-INF	16	10	N/A	4	Current Temp = xxxxxx-INF C	FALSE
Real	+INF	16	10	N/A	4	Current Temp = xxxxxx+INF C	FALSE
Real	NaN	16	10	N/A	4	Current Temp = xxxxxxNaN C	FALSE
UDInt	12345678	16	6	N/A	3	Current Temp = xxxxxxxxx C	FALSE

9.2.3.2. Инструкции Strg_TO_Chars и Chars_TO_Strg (Преобразование в(из) строку символов и массив CHAR)

Strg_TO_Chars копирует строку ASCII-символов в массив символьных байтов.

Chars_TO_Strg копирует массив символьных ASCII-байтов в символьную строку.

Примечание

Только массивы с нулевым младшим индексом (Массив [0...n] типа Char) или (Массив [0...n] типа Byte) могут быть использованы в качестве входного параметра Chars для инструкции Chars_TO_Strg или в качестве IN_OUT параметра Chars для инструкции Strg_TO_Chars.

Таблица 9-35 Инструкция Strg_TO_Chars

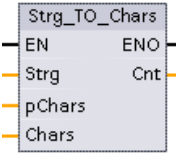
LAD / FBD	SCL	Описание
	<pre>Strg_TO_Chars(Strg:=_string_in_, pChars:=_dint_in_, Cnt=>_uint_out_, Chars:=_variant_inout_);</pre>	Копирует всю входную строку Strg в массив символов IN_OUT параметра Chars. Операция перезаписывает байты, начиная с элемента массива, заданного параметром pChars. Могут быть использованы все строки с максимальной поддерживаемой длиной (1...254). Граница разделения данных не прописывается (под Вашу ответственность). Для установки границы разделения данных после последнего записанного символа массива, используется следующий элемент массива [pChars+Cnt].

Таблица 9-36 Типы данных для параметров (Strg_TO_Chars)

Параметр и тип	Тип данных	Описание
Strg	IN	String, WString
pChars	IN	DInt
Chars	IN_OUT	Variant
Cnt	OUT	UInt

9.2 Строка и символ

Таблица 9-37 Инструкция Chars_TO_Strg

LAD / FBD	SCL	Описание
	<pre>Chars_TO_Strg(Chars:=_variant_in_, pChars:=_dint_in_, Cnt:=_uint_in_, Strg=>_string_out_);</pre>	Все части массива символов скопированы в строку. Выходная строка должна быть описана перед выполнением Chars_TO_Strg. После этого строка будет перезаписана инструкцией Chars_TO_Strg. Используются строки с поддержкой максимальной длины в диапазоне от 1 до 254. Значение максимальной длины строки не может быть изменено в процессе выполнения инструкции Chars_TO_Strg. Копирование из массива в строку прекращается при достижении максимальной длины строки. Значения нулевых символов '\$00' или 16#00 массива используются в качестве ограничителя и завершают процесс копирования символов в строку.

Таблица 9-38 Типы данных для параметров (Chars_TO_Strg)

Параметр и тип	Тип данных	Описание
Chars	IN	Variant
		Параметр Chars – это указатель на массив [0..n] с нулевым младшим индексом, преобразуемый в строку. Массив может быть описан в DB или в виде локальных переменных в интерфейсном блоке. Пример: "DB1".MyArray указатель на массив MyArray [0..10] значений элемента Char в DB1.
pChars	IN	Dint
		Количество элементов первого символа, копируемых в массив. [0] – значение элемента массива по умолчанию.
Cnt	IN	UInt
		Счетчик копируемых символов: 0 означает "все"
Strg	OUT	String, WString
		Целевая строка

Таблица 9-39 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Chars_TO_Strg: Попытка скопировать в выходную строку больше символьных байтов, чем допускается максимальной длиной описываемой строки.
0	Chars_TO_Strg: Во входном символьном байте массива было обнаружено значение нулевого символа (16#00).
0	Strg_TO_Chars: Попытка скопировать в выходную строку больше символьных байтов, чем допускается предельным количеством элементов массива.

9.2.3.3. Инструкции ATH и HTA (Преобразование в(из) ASCII-строку и шестнадцатеричное число)

Инструкции ATH (ASCII в шестнадцатеричное число) и HTA (шестнадцатеричное число в ASCII) используются для преобразования между символьными байтами ASCII (только символы с 0 по 9 и с A по F в верхнем регистре) и соответствующими 4-битными шестнадцатеричными полубайтами.

Таблица 9-40 Инструкция ATH

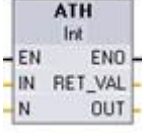
LAD / FBD	SCL	Описание
	<pre>ret_val := ATH(in:=_variant_in_, n:=_int_in_, out=>_variant_out_);</pre>	Преобразование ASCII-символов в сжатые шестнадцатеричные числа.

Таблица 9-41 Типы данных для ATH-инструкции

Параметр и тип	Тип данных	Описание
IN	Variant	Указатель на символьный ASCII-байт массива
N	UInt	Количество символьных ASCII-байтов для преобразования
RET_VAL	Word	Код состояния выполнения
OUT	Variant	Указатель на преобразованный шестнадцатеричный байт массива

Преобразование начинается с адреса, указанного в параметре IN и продолжается для N байтов. Результат записывается в ячейку, указанную в параметре OUT. Могут быть конвертированы только действительные ASCII-символы с 0 по 9, символы от "a" до "f" в нижнем регистре и символы от "A" до "F" в верхнем регистре. Любой другой символ будет преобразован в ноль.

8-битные кодированные ASCII-символы преобразуются в 4-битные шестнадцатеричные полубайты. Два ASCII-символа могут быть преобразованы в одиночный байт содержащий два 4-битных шестнадцатеричных полубайта.

Параметры IN и OUT определяют байты массивов, а не шестнадцатеричные строковые данные (String data). Конвертированные ASCII-символы выводятся на шестнадцатеричный выход в том же порядке, в котором были считаны. Если количество ASCII-символов нечетное, то в крайний правый полубайт конвертированного шестнадцатеричного числа записываются нули.

Таблица 9-42 Примеры: Преобразование ASCII-в-шестнадцатеричное число (ATH)

Символьные IN-байты	N	Значение OUT	ENO
'0a23'	4	W#16#0A23	TRUE
'123AFx1a23'	10	16#123AF01023	FALSE
'a23'	3	W#16#A230	TRUE

9.2 Строка и символ

Таблица 9-43 Инструкция HTA

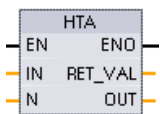
LAD / FBD	SCL	Описание
	<pre>ret_val := HTA(in:=_variant_in_, n:=_uint_in_, out=>_variant_out_);</pre>	Преобразование сжатых шестнадцатеричных чисел в их соответствующие символьные ASCII-байты.

Таблица 9-44 Типы данных для HTA-инструкции

Параметр и тип	Тип данных	Описание
IN	IN	Variant
N	IN	UInt
RET_VAL	OUT	Word
OUT	OUT	Variant

Преобразование начинается с адреса, указанного в параметре IN и продолжается для N байтов. Каждый 4-битный полубайт конвертируется в один 8-битный ASCII-символ и выводит на выход символьные байты 2N ASCII. Все выходные 2N-байты записываются в виде ASCII-символов от 0 до 9 через символы от A до F в верхнем регистре. Параметр OUT описывает байт массива, а не строку.

Каждый полубайт шестнадцатеричного байта конвертируется в символ в том же порядке, в котором был считан (первым конвертируется крайний левый полубайт шестнадцатеричного числа, затем крайний правый полубайт того же байта).

Таблица 9-45 Примеры: Преобразование "Шестнадцатеричное число – в – ASCII" (HTA)

Значение IN	N	Символьные байты OUT	ENO (ENO всегда TRUE после выполнение HTA)
W#16#0123	2	'0123'	TRUE
DW#16#123AF012	4	'123AF012'	TRUE

Таблица 9-46 Коды сообщений инструкций ATH и HTA

RET_VAL (W#16#....)	Описание	ENO
0000	Ошибки отсутствуют	TRUE
0007	Недействительный входной ATH-символ: Входной символ не соответствует ASCII-символам от 0 до 9 или от "A(a)" до "F(f)" в верхнем или нижнем регистрах	FALSE
8101	Недействительная или несуществующая входная ссылка; например, доступ к несуществующему блоку данных.	FALSE
8120	Недействительный формат входной строки, т.е., max= 0, max=255, фактическая длина больше максимальной, или длина, указанная в ссылке меньше максимальной	FALSE
8182	Недостаточный размер входного буфера для N	FALSE
8151	Недопустимый тип данных для входного буфера	FALSE
8301	Недействительная или несуществующая выходная ссылка, например, доступ к несуществующему блоку данных.	FALSE
8320	Недействительный формат выходной строки, т.е., max= 0, max=255, фактическая длина больше максимальной, или длина, указанная в ссылке меньше максимальной	FALSE
8382	Недостаточный размер выходного буфера для N	FALSE
8351	Недопустимый тип данных для выходного буфера	FALSE

9.2.4. Строковые инструкции

Для управления программой Вы можете использовать следующие строковые и символьные инструкции при создании сообщений для операторских дисплеев и процессных журналов.

9.2.4.1. Инструкция MAX_LEN (Максимальная длина символьной строки)

Таблица 9-47 Инструкция MAX_LEN

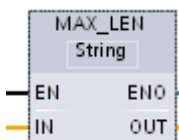
LAD / FBD	SCL	Описание
	<pre>out := MAX_LEN(in);</pre>	MAX_LEN (Maximum length of string = Максимальная длина строки) содержит максимальное значение длины, назначенное строке IN на выходе OUT. Если во время обработки инструкции возникает ошибка, то строка на выходе будет пустой (нулевой длины). Типы данных String и WString содержат информацию о двух длинах: первый байт (или слово) обозначает максимальную длину, а второй байт (или слово) - фактическую длину (фактическое количество действительных символов). - Максимальная длина символьной строки, назначенная типам данных String или WString, указывается в круглых скобках. Количество байтов, занимаемых типом данных String – на 2 байта больше максимальной длины. Количество слов, занимаемых типом данных WString – на 2 слова больше максимальной длины. - Фактическая длина представляет собой фактически используемое количество символов. Фактическая длина должна быть не больше максимальной длины. Для типа данных String фактическая длина измеряется в байтах, а для типа данных WString – в словах. Для вычисления максимальной длины строки символов используется инструкция MAX_LEN, а для вычисления фактической длины строки символов - инструкция LEN.

Таблица 9-48 Типы данных для параметров

Параметры и типы		Типы данных	Описание
IN	IN	String, WString	Входная строка
OUT	OUT	DInt	Максимальное количество символов, допустимое для IN-строки

9.2.4.2. Инструкция LEN (Определение фактической длины строки символов)

Таблица 9-49 Инструкция LEN

LAD / FBD	SCL	Описание
	<code>out := LEN(in);</code>	LEN (length = длина) описывает фактическую длину строки IN на выходе OUT. Пустая строка имеет нулевую длину.

Таблица 9-50 Типы данных для параметров

Параметр и тип		Тип данных	Описание
IN	IN	String, WString	Входная строка
OUT	OUT	Int, DInt, Real, LReal	Количество действительных символов в IN-строке

Таблица 9-51 Состояние ENO

ENO	Условие	OUT
1	Отсутствие недействительных строк	Строка допустимой длины
0	Фактическая длина IN-строки превышает максимальную длину	Фактическая длина сбрасывается в 0
	Максимальная длина IN-строки выходит за пределы выделенного диапазона памяти	
	Максимальная длина IN-строки – 255 (недопустимая длина)	

9.2.4.3. Инструкция CONCAT (Объединение строк символов)

Таблица 9-52 Инструкция объединения строк

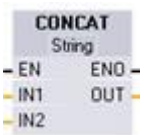
LAD / FBD	SCL	Описание
	<code>out := CONCAT(in1, in2);</code>	CONCAT (объединение строк) объединяет строковые параметры IN1 и IN2 в одну строку на выходе OUT. После объединения строка String IN1 располагается в левой части, а строка String IN2 – в правой части суммарной строки.

Таблица 9-53 Типы данных для параметров

Параметр и тип		Тип данных	Описание
IN1	IN	String, WString	Входная строка 1
IN2	IN	String, WString	Входная строка 2
OUT	OUT	String, WString	Суммарная строка (строка 1 + строка 2)

9.2 Строка и символ

Таблица 9-54 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительные
0	Результирующая строка после объединения больше максимально допустимой длины строки OUT	Символы результирующей строки копируются, пока не будет достигнута максимальная длина строки OUT
	Фактическая длина строки IN1 превышает максимально допустимую длину для строки IN1, фактическая длина строки IN2 превышает максимальную длину для строки IN2, или фактическая длина строки OUT превышает максимальную длину для строки OUT (недействительная строка)	Фактическая длина строки сбрасывается в 0
	Максимальная длина строк IN1, IN2 или OUT выходит за пределы выделенного диапазона памяти	
	Максимальная длина строк IN1 или IN2 равна 255 или максимальная длина строки OUT равна 0 или 255 (тип данных String)	
	Максимальная длина строк IN1 или IN2 равна 65534 или максимальная длина строки OUT равна 0 или 65534 (тип данных WString)	

9.2.4.4. Инструкции LEFT, RIGHT и MID (Чтение подстрок в строке символов)

Таблица 9-55 Чтение левой, правой и средней подстрок

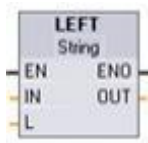
LAD / FBD	SCL	Описание
	<code>out := LEFT(in, L);</code>	LEFT (левая подстрока) представляет собой подстроку, содержащую первые L символов строкового параметра IN. - Если L больше фактической длины строки IN, то вся строка IN будет возвращена в параметр OUT. - Если на входе пустая строка, то пустая строка будет возвращена в параметр OUT.
	<code>out := MID(in, L, p);</code>	MID (средняя подстрока) представляет собой среднюю часть строки. Средняя подстрока, содержащая L символов, начинается с символа позиции P (включительно). Если сумма L и P превышает фактическую длину строкового параметра IN, то подстрока возвращается с начальной позиции символа P и продолжается до конца строки IN.
	<code>out := RIGHT(in, L);</code>	RIGHT (правая подстрока) представляет собой подстроку, содержащую последние L символов строки. - Если она больше фактической длины IN строки, то вся строка IN будет возвращена в параметр OUT. - Если на входе пустая строка, то пустая строка будет возвращена в параметр OUT.

Таблица 9-56 Типы данных для параметров

Параметр и тип		Тип данных	Описание
IN	IN	String, WString	Входная строка
L	IN	Int	Длина сформированной подстроки: • LEFT использует часть символов с левого края символьной строки • RIGHT использует часть символов с правого края символьной строки • MID использует часть символов, начиная с позиции P, в середине строки
P	IN	Int	Только для MID: Скопирована позиция первого символа подстроки P= 1, для исходной позиции символов в IN строке
OUT	OUT	String, WString	Выходная строка

Таблица 9-57 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительны
0	• L или P меньше или равны 0 • P больше максимальной длины строки IN • Фактическая длина IN строки превышает максимальную длину для IN строки или фактическая длина OUT строки превышает максимальную длину для OUT строки • Максимальная длина строк IN или OUT выходит за пределы выделенного диапазона памяти • Максимальная длина строк IN или OUT равна 0 или 255 (тип данных String) или 0 или 65534 (тип данных WString)	Фактическая длина сбрасывается в 0
	Длина скопированной подстроки (L) больше максимальной длины OUT строки.	Символы копируются, пока не будет достигнута максимальная длина OUT строки
	Только для MID: L или P меньше или равны 0	Фактическая длина сбрасывается в 0
	Только для MID: P больше максимальной длины IN строки	
	Фактическая длина IN1 превышает максимальную длину для IN1 или фактическая длина IN2 превышает максимальную длину для IN2 (недействительная строка)	Фактическая длина сбрасывается в 0
	Максимальная длина IN1, IN2 или OUT находятся вне пределов выделенного диапазона памяти	
	Недопустимая длина IN1, IN2 или OUT: 0 или 255 (тип данных String) или 0 или 65534 (тип данных WString)	

9.2 Строка и символ

9.2.4.5. Инструкция DELETE (Удаление символов из строки символов)

Таблица 9-58 Инструкция Delete

LAD / FBD	SCL	Описание
	<code>out := DELETE(in, L, p);</code>	Удаляет L символов из строки IN. Удаление начинается с позиции символа P (включительно), а остальная часть строки выводится в параметре OUT. - Если L равно нулю, то вся входная строка возвращается в параметр OUT. - Если сумма L и P больше максимальной длины входной строки, то строка удаляется до конца.

Таблица 9-59 Типы данных для параметров

Параметр и тип	Тип данных	Описание
IN	IN	String, WString
L	IN	Int
P	IN	Int
OUT	OUT	String, WString

Таблица 9-60 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительные
0	P больше фактической длины строки IN	IN копируется в OUT без удаления символов
	Результирующая строка после удаления символов больше максимальной длины OUT строки	Символы результирующей строки копируются, пока не будет достигнута максимальная длина OUT строки
	L меньше 0, или P меньше или равно 0	Фактическая длина сбрасывается в 0
	Фактическая длина IN превышает максимальную длину для IN строки или фактическая длина OUT превышает максимальную длину OUT строки	
	Максимальная длина строк IN или OUT находится вне пределов выделенного диапазона памяти	
	Максимальная длина строк IN или OUT – 0 или 255	

9.2.4.6. Инструкция INSERT (Вставка символов в строку символов)

Таблица 9-61 Инструкция Insert

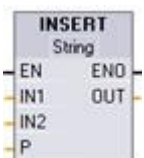
LAD / FBD	SCL	Описание
	<pre>out := INSERT(in1, in2, p);</pre>	Вставка строки IN2 в строку IN1. Вставка начинается после символа с позицией P.

Таблица 9-62 Типы данных для параметров

Параметр и тип		Тип данных	Описание
IN1	IN	String, WString	Входная строка 1
IN2	IN	String, WString	Входная строка 2
P	IN	Int	Последняя позиция символа в строке IN1 перед вставкой точки для строки IN2 Первый символ строки IN1 в позиции №1.
OUT	OUT	String, WString	Результирующая строка

Таблица 9-63 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительные
0	P больше длины IN1	Строка IN2 объединяется со строкой IN1 сразу за последним символом строки IN1
	P меньше 0	Фактическая длина сбрасывается в 0
	После вставки результирующая строка больше максимальной длины OUT строки	Символы результирующей строки копируются, пока не будет достигнута максимальная длина OUT строки
	Фактическая длина IN1 превышает максимальную длину для строки IN1, фактическая длина IN2 превышает максимальную длину для строки IN2 или фактическая длина OUT превышает максимальную длину для строки OUT (недействительная строка)	Фактическая длина сбрасывается в 0
	Максимальная длина строк IN1, IN2 или OUT находится вне пределов выделенного диапазона памяти	
	Максимальная длина строк IN1 или IN2 – 255 или максимальная длина строки OUT – 0 или 255 (тип данных String)	
	Максимальная длина строк IN1 или IN2 – 65534 или максимальная длина строки OUT – 0 или 65534 (тип данных WString)	

9.2 Строка и символ

9.2.4.7. Инструкция REPLACE (Замена символов в строке символов)

Таблица 9-64 Инструкция замены символов (Replace)

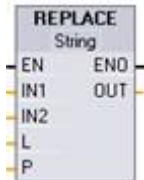
LAD / FBD	SCL	Описание
	<pre>out := REPLACE(in1:=_string_in_, in2:=_string_in_, L:=_int_in_, p:=_int_in_);</pre>	Замена L символов в строковом параметре IN1. Начало замены – с символа позиции P (включительно) строки IN1, с заменой символов, поступающих из строкового параметра IN2.

Таблица 9-65 Типы данных для параметров

Параметр и тип	Тип данных	Описание
IN1	IN	String, WString
IN2	IN	String, WString
L	IN	Int
P	IN	Int
OUT	OUT	String, WString

Если значение параметра L равно нулю, то строка IN2 будет вставлена с позиции P строки IN1 без удаления символов из строки IN1.

Если значение параметра P равно единице, то первые L символов строки IN1 будут заменены символами строки IN2.

Таблица 9-66 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительные
0	P больше длины IN1	IN2 будет "подсоединена" к IN1 сразу после последнего символа строки IN1
	P находится в пределах IN1, но меньше L символов, остающихся в строке IN1	IN2 заменяет последние символы строки IN1, начиная с позиции P
	Результирующая строка после замены больше максимальной длины строки OUT	Символы будут копироваться в результирующую строку, пока не будет достигнута максимальная длина строки OUT
	Максимальная длина IN1 равна 0	Символы строки IN2 копируются в строку OUT
	L меньше нуля или P меньше или равно нулю	Фактическая длина сбрасывается в 0
	Фактическая длина IN1 превышает максимальную длину для строки IN1, фактическая длина IN2 превышает максимальную длину для строки IN2 или фактическая длина OUT превышает максимальную длину для строки OUT	
	Максимальные длины строк IN1, IN2 или OUT находятся вне пределов выделенного диапазона памяти	
	Максимальная длина строк IN1 или IN2 равна 255 или максимальная длина строки OUT равна 0 или 255 (тип данных String)	

ENO	Условие	OUT
	Максимальная длина строк IN1 или IN2 равна 65534 или максимальная длина строки OUT равна 0 или 65534 (тип данных WString)	

9.2.4.8. Инструкция FIND (Поиск символов в строке символов)

Таблица 9-67 Инструкция поиска (Find)

LAD / FBD	SCL	Описание
	<pre>out := FIND(in1:=_string_in_, in2:=_string_in);</pre>	Содержит позицию символа подстроки, указанную для строки IN2 в пределах строки IN1. Поиск начинается слева. Позиция символа первого вхождения строки IN2 возвращается в строку OUT. Если строка IN2 не найдена в строке IN1, то возвращается нулевая строка.

Таблица 9-68 Типы данных для параметров

Параметр и тип	Тип данных	Описание
IN1	IN	String, WString
IN2	IN	String, WString
OUT	OUT	Int

Таблица 9-69 Состояние ENO

ENO	Условие	OUT
1	Ошибки не обнаружены	Все символы действительные
0	IN2 больше IN1 Фактическая длина IN1 превышает максимальную длину для строки IN1 или фактическая длина IN2 превышает максимальную длину для строки IN2 (недействительная строка) Максимальная длина строк IN1 или IN2 находится вне пределов выделенного диапазона памяти Максимальная длина строк IN1 или IN2 равна 255 (тип данных String) или 65535 (тип данных WString)	Позиция символа сбрасывается в 0

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

9.3.1. Инструкции распределенного ввода/вывода

С PROFINET, PROFIBUS или AS-интерфейсом могут быть использованы следующие инструкции распределенного ввода/вывода:

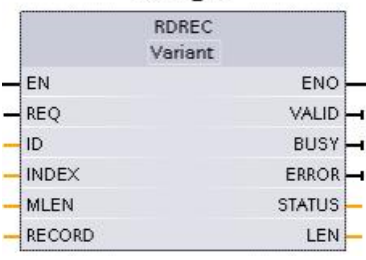
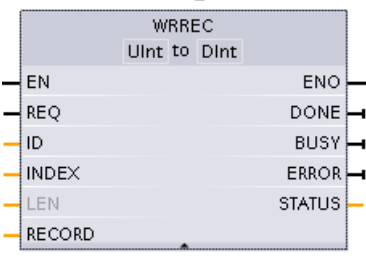
- Инструкция RDREC (страница 351): с помощью данной инструкции Вы можете считывать записи данных из модуля или устройства с числовым типом данных INDEX.
- Инструкция WRREC (страница 351): с помощью данной инструкции Вы можете передавать записи данных с числовым типом данных INDEX в модуль или устройство, определяемые идентификатором ID.
- Инструкция RALRM (страница 354): с помощью данной инструкции Вы можете принимать прерывание из модуля или устройства со всей соответствующей информацией и передавать эту информацию в соответствующие выходные параметры.
- Инструкция DPRD_DAT (страница 362): с помощью данной инструкции Вы можете считывать более 64 байт области консистентных данных модуля или устройства.
- Инструкция DPWR_DAT (страница 362): с помощью данной инструкции Вы можете записывать более 64 байт области консистентных данных модуля или устройства.

Инструкция DPNRM_DG (страница 365) может быть использована только с PROFIBUS. Текущие диагностические данные ведомого DP-устройства могут быть считаны в формате, определяемом EN 50 170 Volume 2, PROFIBUS.

9.3.2. Инструкции RDREC и WRREC (чтение/запись записей данных)

Инструкции RDREC (чтение записей данных) и WRREC (запись записей данных) могут быть использованы с PROFINET, PROFIBUS и AS-интерфейсом.

Таблица 9-70 Инструкции RDREC и WRREC

LAD / FBD	SCL	Описание
"RDREC_DB" 	<pre>"RDREC_DB"(req:=_bool_in_, ID:=_word_in_, in- dex:=_dint_in_, mlen:=_uint_in_, valid=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_dword_out_, len=>_uint_out_, rec- ord:=_variant_inout_);</pre>	Инструкция RDREC используется для чтения записей данных с числовым типом данных INDEX из компонента, с адресацией по ID, например, из центральной стойки или компонента распределенного ввода/вывода (PROFIBUS DP или PROFINET IO). Для считывания задайте максимальный размер MLEN в байтах. Выбранная длина целевой области RECORD в байтах должна быть не менее MLEN.
"WRREC_DB" 	<pre>"WRREC_DB"(req:=_bool_in_, ID:=_word_in_, in- dex:=_dint_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_dword_out_, rec- ord:=_variant_inout_);</pre>	Инструкция WRREC используется для передачи данных RECORD с числовым типом данных INDEX в ведомое DP-устройство/компонент PROFINET IO устройства с адресацией по ID, например, в центральную стойку или компонент распределенного ввода/вывода (PROFIBUS DP или PROFINET IO). Задайте длину в байтах передаваемым записям данных. Следовательно, выбранный размер исходной области RECORD, должен быть длиной как минимум в LEN- байтов.

¹ Когда Вы вставляете инструкцию, STEP 7 автоматически создает блок данных.

² В SCL-примерах, "RDREC_DB" и "WRREC_DB" имеют наименование экземплярных блоков данных DB.

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Таблица 9-71 Типы данных для параметров инструкций RDREC и WRREC

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	REQ = 1: Запись передаваемых данных
ID	IN	HW_IO (Word)	Логический адрес ведомого DP-устройства/компонента PROFINET IO (модуля или submodule): - Бит 15 должен быть установлен для модуля вывода (например, для адреса 5: ID:= DW#16#8005). - Для комбинированного модуля должен быть указан меньший из двух адресов. Примечание: Для версии V3.0 идентификатор устройства ID может быть определен двумя способами: - В окне отображения сети "Network view" выбрать следующее: - Устройство (серый блок) - Свойства устройства ("Properties") - Аппаратный идентификатор ("Hardware identifier") Примечание: Не каждое устройство может отображать свой аппаратный идентификатор "Hardware identifiers". - В окне дерева проекта ("Project tree") выбрать: - PLC-теги - Таблица тегов по умолчанию - Вкладка системных постоянных (System constants) Отображаются аппаратные конфигуризаторы всех сконфигурированных устройств. Примечание: В версии V4.0 аппаратный идентификатор интерфейсного модуля определяется переходом в таблицу тегов и локализации параметра "Device Name [HEAD]" в системных константах (System Constants).
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Количество записей данных
MLEN	IN	Byte, USInt, UInt	Выбранный максимальный размер записей данных в байтах (RDREC)
VALID	OUT	Bool	Новые достоверные принимаемые записи данных (RDREC). Бит VALID устанавливается в TRUE для цикла сканирования, выполненного без ошибок после последнего запроса.
DONE	OUT	Bool	Передаваемые записи данных (WRREC). Бит DONE устанавливается в TRUE для цикла сканирования, выполненного без ошибок после последнего запроса.
BUSY	OUT	Bool	- BUSY = 1: Процесс чтения (RDREC) или записи (WRREC) еще не закончен. - BUSY = 0: Передача записей данных выполнена.
ERROR	OUT	Bool	ERROR = 1: Произошла ошибка при выполнении операции чтения (RDREC) или записи (WRREC). Бит ERROR устанавливается в TRUE, если последнее сканирование было завершено с ошибками. Значение кода ошибки параметра STATUS действительно при одном цикле сканирования, когда ERROR = TRUE.
STATUS	OUT	DWord	Состояние блока или информация об ошибке (страница 455)

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Параметр и тип		Тип данных	Описание
LEN	OUT (RDREC) IN (WRREC)	UInt	- Размер выбранных записей данных (RDREC) - Максимальная длина переданных записей данных в байтах (WRREC)
RECORD	IN_OUT	Variant	- Целевая область для выбранных записей данных (RDREC) - Записи данных (WRREC)

Инструкции RDREC и WRREC выполняются асинхронно, т.е. обработка охватывает несколько вызовов инструкций. Обработка начинается с вызова инструкций RDREC или WRREC с помощью REQ = 1.

Состояние обработки отображается через выходной параметр BUSY и два центральных байта выходного параметра STATUS. Передача записей данных считается выполненной, когда выходной параметр BUSY будет установлен в FALSE

Значение TRUE (только для одного цикла сканирования) в выходном параметре VALID (RDREC) или DONE (WRREC) подтверждает, что записи данных были успешно переданы в целевую область RECORD (RDREC) или целевое устройство (WRREC). В случае RDREC, выходной параметр LEN содержит размер передаваемых данных в байтах.

Выходной параметр ERROR (только для одного цикла сканирования при ERROR = TRUE) показывает, что при передаче записей данных возникла ошибка. В этом случае выходной параметр STATUS (только для одного цикла сканирования при ERROR = TRUE) содержит информацию об ошибке.

Записи данных определяются производителем устройства. Для получения подробной информации о записях данных обратитесь к документации на конкретное устройство, предоставляемой его производителем.

Примечание

Если конфигурация ведомого DPV1-устройства выполнено с использованием GSD-файла (GSD rev. 3 и выше), а DP-интерфейс DP-мастера установлен в "S7 compatible", то с помощью "RDREC" пользовательской программы Вы не можете считывать любые записи данных из модулей ввода/вывода или выполнять записи в модули ввода/вывода с помощью "WRREC". В этом случае DP-мастер адресован неправильному слоту (сконфигурированный слот + 3).

Устранение неисправности: задайте интерфейс DP-мастера "DPV1".

Примечание

Интерфейсы инструкций "RDREC" и "WRREC" идентичны функциональным блокам "RDREC" и "WRREC" описанным в документации "PROFIBUS Guideline, PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

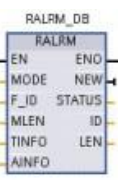
Примечание

Если для чтения и записи данных для PROFINET IO Вы используете "RDREC" или "WRREC", то отрицательные значения в параметрах INDEX, MLEN и LEN могут быть интерпретированы как беззнаковое 16-битное целое число.

9.3.3. Инструкция RALRM (Получение прерывания)

Инструкция RALRM (Чтение аварийного сообщения) может быть использована с PROFINET и PROFIBUS.

Таблица 9-72 Инструкция RALRM

LAD / FBD	SCL	Описание
	<pre> "RALRM_DB" (mode:=_int_in_, f_ID:=_word_in_, mlen:=_uint_in_, new=>_bool_out_, sta- tus=>_dword_out_, ID=>_word_out_, len=>_uint_out_, tinfo:=_variant_inout_, ainfo:=_variant_inout_); </pre>	Инструкция RALRM (чтение аварийного сообщения) используется для считывания информации о диагностическом прерывании из модулей/устройств PROFIBUS или PROFINET I/O. Выходные параметры содержат начальную информацию о вызываемом OB, а также информацию об источнике прерывания. Вызовите RALRM в OB обработки прерывания для возврата информации о событии(ях) вызвавшем(их) прерывание. В S7-1200 диагностическими OB обработки прерываний поддерживают следующие типы прерываний: состояние (Status), обновление (Update), профиль (Profile), ошибка диагностики (Diagnostic error), вставка или удаление модулей, неисправность стойки или станции.

¹ Когда Вы вставляете инструкцию, STEP 7 автоматически создает DB.

² В SCL-примере, "RALRM_DB" – это имя экземплярного блока данных DB.

Таблица 9-73 Типы данных для параметров

Тип и параметр	Тип данных	Описание
MODE	IN	Byte, USInt, SInt, Int
F_ID	IN	HW_IO (Word)
		Начальный логический адрес компонента (модуля), от которого получено прерывание Примечание: Идентификатор устройства (ID) можно определить двумя способами: - В окне отображения сети ("Network view") выбрав следующее: - Устройство (Device, серый блок) - Свойства устройства ("Properties") - Аппаратный идентификатор ("Hardware identifier") Примечание: Не каждое устройство может отображать свой аппаратный идентификатор "Hardware identifiers". - В меню дерева проекта ("Project tree") выбрав следующее: - PLC-теги - Таблица тегов по умолчанию (Default tag table) - Вкладка системных констант (System constants) - Будут отображены аппаратные идентификаторы всех сконфигурированных устройств.
MLEN	IN	Byte, USInt, UInt
		Максимальный размер полученных данных о прерывании в байтах. MLEN of 0 позволяет получить столько данных о прерывании, сколько содержится в целевой области AINFO Target Area.
NEW	OUT	Bool
STATUS	OUT	DWord
		Состояние инструкции RALRM. За дополнительной информацией обратитесь к главе "Параметр STATUS для RDREC, WRREC и RALRM" (страница 358).

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Параметр и тип		Тип данных	Описание
ID	OUT	HW_IO (Word)	Аппаратный идентификатор модуля ввода/вывода, вызвавшего диагностическое прерывание. Примечание: Для определения ID-устройства обратитесь к описанию параметра F_ID.
LEN	OUT	DWord, UInt, UInt, DInt, Real, LReal	Размер полученной информации AINFO о прерывании.
TINFO	IN_OUT	Variant	Информация о задаче: Целевой диапазон для запуска OB и информация об администрировании. Размер TINFO всегда 32 байта.
AINFO	IN_OUT	Variant	Информация о прерывании: Целевая область для основной информации и дополнительной информации о прерывании. Для AINFO обеспечьте размер как минимум в MLEN-байт (если MLEN больше 0). Размер AINFO – переменный.

Примечание

Если Вы вызываете "RALRM" в OB, который запускает событие, а не обработку прерывания ввода/вывода, то инструкция сокращает соответствующую информацию на его выходах.

Убедитесь, что при вызове "RALRM" в нескольких OB используется несколько экземплярных блоков данных. Если Вы оцениваете данные, полученные при вызове "RALRM" из соответствующего OB обработки прерывания, то необходимо использовать отдельный экземплярный блок данных DB для OB запуска события.

Примечание

Интерфейс инструкции "RALRM" идентичен "RALRM" FB, описанному в документации "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Вызов RALRM

Инструкция RALRM может быть вызвана в трех различных режимах работы (MODE).

Таблица 9-74 Режимы работы для инструкции RALRM

MODE	Описание
0	- ID содержит аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание. - Выходной параметр NEW установлен в TRUE. - LEN содержит 0 на выходе. - Информация AINFO и TINFO не обновлена.
1	- ID содержит аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание. - Выходной параметр NEW установлен в TRUE. - LEN содержит на выходе возвращаемую суммарную информацию AINFO-данных в байтах. - AINFO и TINFO обновляются с информацией о прерывании.
2	Если аппаратный идентификатор, назначенный входному параметру F_ID, вызвал прерывание, то: - ID, содержащий аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание, должен быть таким же, как значение в F_ID. - Выходной параметр NEW устанавливается в TRUE. - LEN содержит на выходе возвращаемую суммарную информацию AINFO-данных в байтах. - AINFO и TINFO обновляются с информацией о прерывании.

Примечание

Если область, назначаемая Вами для TINFO или AINFO, слишком мала, то RALRM не может вернуть всю информацию.

MLEN может ограничить объем данных, возвращаемых AINFO.

Информацию, описывающую данные TINFO и AINFO, Вы найдете в описании параметров AINFO и TINFO online-справки STEP 7.

Организационный блок данных TInfo

В приведенной ниже таблице показано распределение TInfo-данных для инструкции RALRM:

Одинаково для OB: Status (Статус), Update (Обновление), Profile (Профиль), Diagnostic error interrupt (Ошибка диагностического прерывания), Pull or plug of modules (Вставка и извлечение модулей), Rack or station failure (Неисправность стойки или станции)	0	SI_Format	OB_Class	OB_Nr	
	4	LADDR			
TI_Submodule – OB: Status, Update, Profile	4			Slot	
	8	Specifier (Признак)		0	
TI_DiagnosticInterrupt – OB: Diagnostic error interrupt	4			IO_State	
	8	Channel (Канал)		MultiError	0
TI_PlugPullModule - OB: Pull or plug of modules	4			Event_Class	Fault_ID
	8	0		0	
TI_StationFailure - OB: Rack or station failure	4			Event_Class	Fault_ID
	8	0		0	
Одинаково для OB: Status, Update, Profile, Diagnostic error interrupt, Pull or plug of modules, Rack or station failure	12	0			
	16				
	20	address (адрес)		slv_prfl	intr_type
	24	flags1 (флаг)	flags2 (Флаг)	id (идентификатор)	
	28 ¹	manufacturer (производитель)		instance (экземпляр)	

¹ Байты 28 - 31 ("manufacturer" и "instance") не используются с PROFIBUS.

Примечание

Для получения подробной информации об TINFO данных обратитесь к online-справке STEP 7.

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

9.3.4. Параметр STATUS для RDREC, WRREC и RALRM

Выходной параметр STATUS содержит информацию об ошибке, которая может быть интерпретирована как ARRAY[1...4] OF BYTE, имеющую следующую структуру:

Таблица 9-75 Выходной массив STATUS

Элемент массива	Имя	Описание
STATUS[1]	Function_Num	- В#16#00, если нет ошибок - Функция ID из DPV1-PDU: В случае возникновения ошибки, В#16#80 – OR (для чтения записей данных: В#16#DE; для записи записей данных: В#16#DF). Если используется элемент протокола не DPV1, то В#16#C0 должен быть выходом.
STATUS[2]	Error Decode	Локализация ошибки ID
STATUS[3]	Error_Code_1	Ошибка ID
STATUS[4]	Error_Code_2	Расширение ошибки ID, определяемое производителем

Таблица 9-76 Значения STATUS[2]

Error_decode (В#16#....)	Источник	Описание
с 00 по 7F	CPU	Отсутствие ошибок или предупреждений
80	DPV1	Ошибка в соответствии с IEC 61158-6
с 81 по 8F	CPU	В#16#8x отображает ошибку в вызываемом параметре "x" инструкции.
FE, FF	DP Profile	Ошибка, определяемая профилем.

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Таблица 9-77 Значения STATUS[3]

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Пояснение (DVP1)	Описание
00	00		Нет ошибок, нет предупреждений
70	00	Резерв, отказ	Начальный вызов; нет активной передачи записей данных
	01	Резерв, отказ	Начальный вызов; начало передачи записей данных
	02	Резерв, отказ	Промежуточный вызов; передача записей данных уже активна
80	90	Резерв, передача	Недействительный логический стартовый адрес
	92	Резерв, передача	Недопустимый тип для указателя "Variant"
	93	Резерв, передача	DP-компонент, адресованный через ID или F_ID, не сконфигурирован.
	96		"RALRM (страница 354)" не может предоставить OB стартовую информацию, управляющую информацию, основную информацию или дополнительную информацию о прерывании. Для следующих OB Вы можете использовать инструкцию "DPNRM_DG (стр. 365)" при чтении текущего кадра диагностических сообщений ведомого DP-устройства с асинхронной передачей данных (информация об адресации – из стартовой информации OB): • Аппаратное прерывание (стр. 90) • Status (страница 97), Update (страница 98) или Profile (стр.98) • Diagnostic error interrupt (стр.92) • Pull or plug of modules (стр.95)
	A0	Ошибка чтения	Отказ в подтверждении чтения из модуля
	A1	Ошибка записи	Отказ в подтверждении записи в модуль
	A2	Неисправность модуля	Ошибка DP-протокола на уровне 2 (например, неисправность ведомого устройства или проблемы с шиной)
	A3	Резерв, передача	• PROFIBUS DP: ошибка DP-протокола с Direct-Data-Link-Mapper или User-Interface/User • PROFINET IO: основная CM-ошибка
	A4	Резерв, передача	Нарушение связи по коммуникационной шине
	A5	Резерв, передача	-
	A7	Резерв, передача	Недоступны ведомое DP-устройство или модуль (временная ошибка).
	A8	Конфликт версий	Несовместимые версии ведомого DP-устройства или модуля
	A9	Неподдерживаемое свойство	Свойство не поддерживается ведомым DP-устройством или модулем
	от AA до AF	Определяется пользователем	Ведомое DP-устройство или модуль сообщают об ошибке их применения. Обратитесь к документации производителя о применении ведомого DP-устройства или модуля.
	B0	Недопустимый индекс	Отсутствие записей данных в модуле; недопустимое количество записей данных ≥ 256

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Пояснение (DVP1)	Описание
	B1	Запись ошибочной длины	Некорректный размер информации в параметре RECORD. - С "RALRM": Ошибка длины в AINFO Примечание: Обратитесь к информационной online-системе STEP 7 для непосредственного доступа к информации, как интерпретировать возвращаемое содержимое буфера "AINFO". - С "RDREC (страница 351)" и "WRREC (страница 351)": Ошибка длины в "MLEN"
	B2	Недействительный слот	Сконфигурированный слот не занят.
	B3	Конфликт типов	Фактический тип модуля не соответствует заданному типу.
	B4	Недействительная область	Ведомое DP-устройство или модуль информируют о доступе к недействительной области.
	B5	Конфликт состояний	Ведомое DP-устройство или модуль не готовы.
	B6	Запрет доступа	Запрещен доступ к ведомому DP-устройству или модулю.
	B7	Недопустимый диапазон	Ведомое DP-устройство или модуль информируют о недопустимом диапазоне параметра или значения.
	B8	Недопустимый параметр	Ведомое DP-устройство или модуль информируют о недопустимом параметре.
	B9	Недопустимый тип	Ведомое DP-устройство или модуль информируют о недопустимом типе: - С "RDREC (страница 351)": Буфер слишком мал (подмножества не могут быть прочитаны) - С "WRREC (страница 351)": Буфер слишком мал (подмножества не могут быть прочитаны)
	с BA по BF	Определяется пользователем	Ведомое DP-устройство или модуль информируют о ошибке, специфичной для производителя. Обратитесь к документации от производителя по применению ведомого DP-устройства или модуля.
	C0	Конфликт ограничений чтения данных	- С "WRREC (страница 351)": Данные могут быть записаны только когда CPU находится в STOP режиме. Примечание: Это означает, что данные не могут быть записаны пользовательской программой. В online-режиме Вы можете записывать данные только с помощью PG/PC. - С "RDREC (страница 351)": Модуль посылает записи данных, но эти данные или отсутствуют или они не могут быть считаны, т.к. CPU находится в STOP режиме. Примечание: Если данные могут быть считаны, когда CPU находится в STOP режиме, то их оценка пользовательской программой невозможна. В этом случае, Вы можете считывать данные только в online-режиме с помощью PG/PC.
	C1	Конфликт ограничений записи данных	Данные от предыдущего запроса на запись в модуль для той же записи данных еще не обработаны модулем.
	C2	Недостаток ресурсов	В настоящее время модуль обрабатывает максимально возможное количество заданий для CPU.
	C3	Недостаток ресурсов	Все необходимые ресурсы в настоящий момент заняты.
Error_decode (B#16#....)	Error_code_1 (B#16#....)	Пояснение (DVP1)	Описание

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

	C4		Внутренняя временная ошибка. Операция не может быть выполнена. Повторите операцию. При частом повторении ошибки проверьте Вашу установку на отсутствие источников электрических помех.
	C5		Ведомое DP-устройство или модуль недоступны.
	C6		Передача записей данных была отменена из-за приоритета операции отмены.
	C7		Операция прервана из-за "горячего" или "холодного" рес-тарта DP-мастера.
	с C8 по CF		Ведомое DP-устройство или модуль информируют об ошибке ресурсов, специфичной для производителя. Обратитесь к документации от производителя по применению ведомого DP-устройства или модуля.
	Dx	Определяется пользова-телем	Ошибка, специфичная для ведомого DP-устройства. Обратитесь к описанию на ведомое DP-устройство.
81	с 00 по FF		Ошибка в начальном вызове параметра (с "RALRM (страница 354)": MODE)
	00		Недопустимый режим работы
82	с 00 по FF		Ошибка во втором вызове параметра
88	с 00 по FF		Ошибка в восьмом вызове параметра (с "RALRM (страница 354)": TINFO) Примечание: Обратитесь к информационной online-системе STEP 7 для непосредственного доступа к информации, как интерпретировать возвращаемое содержимое буфера "TINFO".
	01		Недопустимый синтаксис ID
	23		Превышение количественной структуры или назначенная область слишком мала.
	24		Недопустимый диапазон ID.
	32		Количество DB/DI вне пользовательского диапазона.
	3A		Количество DB/DI – это NULL для области ID DB/DI, или заданные DB/DI не существуют.
89	с 00 по FF		Ошибка в девятом вызове параметра (с "RALRM (страница 354)": AINFO) Примечание: Обратитесь к информационной online-системе STEP 7 для непосредственного доступа к информации, как интерпретировать возвращаемое содержимое буфера "AINFO".
	01		Недопустимый синтаксис ID.
	23		Превышение количественной структуры или назначенная область слишком мала.
	24		Недопустимый диапазон ID.
	32		Количество DB/DI вне пользовательского диапазона.
	3A		Количество DB/DI – это NULL для области ID DB/DI, или заданные DB/DI не существуют.
8A	с 00 по FF		Ошибка в десятом вызове параметра.
8F	с 00 по FF		Ошибка в пятнадцатом вызове параметра.
FE, FF	с 00 по FF		Ошибка, специфичная для профиля.

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Элемент массива STATUS[4]

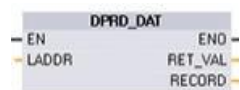
При ошибках DPV1 DP Master передает STATUS[4] в CPU и в инструкцию. При отсутствии ошибок DPV1 это значение сбрасывается в 0 со следующими исключениями для RDREC:

- STATUS[4] содержит целевую область размером, как в RECORD, если MLEN > размера области, назначенного для RECORD.
- STATUS[4]=MLEN, если фактический размер записей данных < MLEN < размера области, назначенного для RECORD.
- STATUS[4]=0, если STATUS[4] > 255; будет как в PROFINET IO, STATUS[4] имеет значение 0.

9.3.5. Инструкции DPRD_DAT и DPWR_DAT (Чтение/запись консистентных данных ведомых DP-устройств)

Инструкции DPRD_DAT (Чтение консистентных данных) и DPWR_DAT (Запись консистентных данных) могут использоваться с PROFINET и PROFIBUS.

Таблица 9-78 Инструкции DPRD_DAT и DPWR_DAT

LAD / FBD	SCL	Описание
	<pre>ret_val := DPRD_DAT(laddr:=_word_in_, record=>_variant_out_);</pre>	Инструкция DPRD_DAT используется для чтения одного или нескольких байтов данных из следующих локализаций: • Модуль в локальной базе • Стандартное ведомое DP-устройство • PROFINET I/O устройство CPU передает считанные данные консистентно. Если во время передачи данных ошибки отсутствуют, то CPU вводит считанные данные в целевую область, определяемую параметром RECORD. Целевая область должна быть одинакового размера с областью, сконфигурированной Вами в STEP 7 для выбранного модуля. Когда выполняется инструкция DPRD_DAT, то Вы имеете доступ к данным только одного модуля. Передача начинается с сконфигурированного стартового адреса.
	<pre>ret_val := DPWR_DAT(laddr:=_word_in_, record:=_variant_in_);</pre>	Инструкция DPWR_DAT используется для консистентной передачи данных параметра RECORD в следующие локализации: • Назначенному модулю в локальной базе • Стандартному ведомому DP-устройству • PROFINET I/O устройству Исходная область должна быть одинакового размера с областью, сконфигурированной Вами в STEP 7 для выбранного модуля.

- S7-1200 CPU поддерживает до 64 байт консистентных данных на локальной шине. Инструкции DPRD_DAT и DPWR_DAT используются для доступа к свыше 64 байт консистентных данных.
- PROFIBUS поддерживает до 4 байт консистентных данных. Инструкции DPRD_DAT и DPWR_DAT используются для доступа к свыше 4 байт консистентных данных.

- PROFINET поддерживает до 1472 байт консистентных данных. Нет необходимости в использовании этих инструкций для передачи консистентных данных между S7-1200 и PROFINET-устройствами.
- Эти инструкции используются для области данных в 1 или несколько байтов. Если в доступе отказано, то выводится код ошибки W#16#8090.

Примечание

Если инструкции DPRD_DAT и DPWR_DAT используются с консистентными данными, то Вам необходимо удалить эти консистентные данные из автоматического обновления образа процесса. За дополнительной информацией обратитесь к главе "PLC концепции: Выполнение пользовательской программы" (страница 79).

Таблица 9-79 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	IN	HW_IO (Word)	• Сконфигурированный стартовый адрес из "I" области модуля, для которого должны быть считаны данные (DPRD_DAT) • Сконфигурированный стартовый адрес из области вывода образа процесса модуля, для которого должны быть записаны данные (DPWR_DAT) Адреса должны вводиться в шестнадцатеричном формате (например, для входного или выходного адреса: 100 означает – LADDR:=W#16#64).
RECORD	OUT	Variant	Область назначения для считанных пользовательских данных (DPRD_DAT) или исходная область для записанных пользовательских данных (DPWR_DAT). Она должна быть такой же, как область, сконфигурированная Вами для выбранного модуля в STEP 7.
RET_VAL	OUT	Int	Если при выполнении функции возникла ошибка, то возвращаемое значение содержит код ошибки.

DPRD_DAT операции

Область назначения должна быть одинакового размера с областью, сконфигурированной для выбранного модуля в STEP 7. Если при передаче данных ошибки отсутствуют, то считанные данные вводятся в область назначения, определяемую параметром RECORD.

Если данные считываются из стандартного ведомого DP-устройства модульного исполнения или с нескольких DP-идентификаторов, то при каждом вызове инструкции DPRD_DAT Вы имеете доступ к данным только одного модуля/DP-идентификатора, определяемым сконфигурированным стартовым адресом.

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

DPWR_DAT операции

Вы передаете данные в параметр RECORD consistently с назначенным стандартным ведомым DP-устройством/PROFINET IO. Данные передаются синхронно, т.е. процесс записи прекращается после завершения инструкции.

Исходная область должна быть одинакового размера с областью, сконфигурированной Вами для выбранного модуля в STEP 7.

Если стандартное ведомое DP-устройство модульного исполнения, то Вы можете получить доступ только к одному модулю ведомого DP-устройства.

Таблица 9-80 Коды ошибок DPRD_DAT и DPWR_DAT

Код ошибки	Описание
0000	Ошибки отсутствуют
8090	Возможен один из следующих вариантов: • Вы не сконфигурировали модуль для заданного логического базового адреса. • Вы проигнорировали ограничения относительно длины консистентных данных. • Вы не ввели стартовый адрес в параметр LADDR в шестнадцатеричном формате.
8092	Параметр RECORD поддерживает следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt и массивы этих типов данных.
8093	В логическом адресе, заданном параметром LADDR, отсутствует DP-модуль/PROFINET IO устройство, из которого Вы можете считывать (DPRD_DAT) или в который Вы можете записывать (DPWR_DAT) консистентные данные.
80A0	Обнаружена ошибка доступа в тот момент, когда устройство ввода/вывода было доступно (DPRD_DAT).
80A1	Обнаружена ошибка доступа в тот момент, когда устройство ввода/вывода было доступно (DPWR_DAT).
80B0	Отказ ведомого устройства на внешнем DP-интерфейсе (DPRD_DAT) и (DPWR_DAT)
80B1	Размер области назначения (DPRD_DAT) или исходной области (DPWR_DAT) не соответствуют размеру пользовательских данных, сконфигурированных в STEP 7 Basic.
80B2	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT) и (DPWR_DAT)
80B3	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT) и (DPWR_DAT)
80C0	Данные не были считаны модулем (DPRD_DAT).
80C1	Данные от предыдущей операции записи в модуль не были обработаны модулем (DPWR_DAT).
80C2	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT) и (DPWR_DAT)
80Fx	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT) и (DPWR_DAT)
85ху	Системная ошибка с внешним интерфейсным DP-модулем (DPWR_DAT)
87ху	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT)
808х	Системная ошибка с внешним интерфейсным DP-модулем (DPRD_DAT)
8хуу	Информация об ошибке общего характера За дополнительной информацией обратитесь к главе "Коды часто встречающихся ошибок для расширенных инструкций" (страница 455).

x = номер параметра

y = номер события

Примечание

Если ведомые DPV1 устройства доступны, то информация об ошибках этих ведомых устройств может быть передана из DP-мастера в инструкцию.

9.3.6. Инструкция DPNRM_DG (чтение диагностических данных ведомого DP-устройства)

Инструкция DPNRM_DG (Чтение диагностических данных) может быть использована с PROFIBUS.

Таблица 9-81 Инструкция DPNRM_DG

LAD / FBD	SCL	Описание
	<pre>ret_val := DPNRM_DG(req:=_bool_in_, laddr:=_word_in_, rec- ord=>_variant_out_, busy=>_bool_out_);</pre>	Инструкция DPNRM_DG используется для чтения текущих диагностических данных ведомого DP-устройства в формате, определяемом EN 50 170 Volume 2, PROFIBUS. Данные, которые могут быть считаны и введены в область назначения, будут отображены в параметре RECORD при следующей передаче данных, выполненной без ошибок.

Таблица 9-82 Инструкция DPNRM_DG, типы данных для параметров

Параметр и тип	Тип данных	Описание
REQ	IN	Bool
LADDR	IN	HW_DPSLAVE
RET_VAL	OUT	Int
RECORD	OUT	Variant
BUSY	OUT	Bool

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Операция чтения запускается установкой в "1" входного параметра REQ в вызове инструкции DPNRM_DG. Операция чтения выполняется асинхронно, другими словами, ей необходимы несколько вызовов инструкции DPNRM_DG. Состояние операции отображается в выходных параметрах RET_VAL и BUSY.

Таблица 9-83 Структура диагностических данных ведомого устройства

Байт	Описание
0	Состояние станции 1
1	Состояние станции 2
2	Состояние станции 3
3	Номер мастер-станции
4	ID-производителя (старший байт)
5	ID-производителя (младший байт)
6 ...	Дополнительная диагностическая информация, определяемая ведомым устройством

Таблица 9-84 Коды ошибок инструкции DPNRM_DG

Код ошибки	Описание	Ограничение
0000	Ошибки отсутствуют	-
7000	Первый вызов с REQ=0: Нет активной передачи данных; BUSY принимает значение 0.	-
7001	Первый вызов с REQ=1: Нет активной передачи данных; BUSY принимает значение 1.	Распределенный ввод/вывод
7002	промежуточный вызов (несоответствующий REQ): Передача данных активна; BUSY принимает значение 1.	Распределенный ввод/вывод
8090	Недействительный заданный базовый логический адрес: Базовый адрес отсутствует.	-
8092	Параметр RECORD поддерживает следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt и массивы этих типов данных.	-
8093	- Эти инструкции недопустимы для модуля, определяемого LADDR (допустимы для S7-DP модулей S7-1200). - LADDR определяет устройство ввода/вывода вместо определения станции. Для задания диагностического адреса для LADDR выберите станцию (а не образ устройства) в окне отображения сети ("Network") конфигурации устройств ("Device configuration").	-
80A2	- Ошибка DP-протокола уровня 2 (например, отказ ведомого устройства или проблемы с шиной) - Для ET200S, записи данных не могут быть считаны в DPV0 режиме.	Распределенный ввод/вывод
80A3	Ошибка DP-протокола с пользовательским интерфейсом/пользователем	Распределенный ввод/вывод
80A4	Проблемы связи на коммуникационной шине	Ошибка возникает между CPU и внешним интерфейсным DP-модулем.
80B0	- Недопустимая инструкция для данного типа модуля. - Модуль не может распознать записи данных. - Недопустимый номер 241 записи данных.	-
80B1	Некорректное задание размера в параметре RECORD.	Заданный размер больше записанной длины
80B2	Сконфигурированный слот свободен.	-

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-интерфейс)

Код ошибки	Описание	Ограничение
80B3	Тип установленного модуля не соответствует заданному типу модуля.	-
80C0	Диагностическая информация отсутствует.	-
80C1	Данные от предыдущей операции записи для той же записи данных еще не обработаны модулем.	-
80C2	Модулем обрабатывается максимальное количество задач для CPU.	-
80C3	Необходимые ресурсы (память и т.д.) в данный момент заняты.	-
80C4	Внутренняя временная ошибка. Операция не может быть обработана. Повторите операцию. При частом повторении данной ошибки проверьте Вашу систему на отсутствие источника электрических помех.	-
80C5	Распределенный ввод/вывод не доступен	Распределенный ввод/вывод
80C6	Передача записей данных остановлена из-за приоритета класса прерывания (рестарт или фоновый режим)	Распределенный ввод/вывод
8хуу ¹	Коды часто встречающихся ошибок	

- ¹ За дополнительной информацией о кодах часто встречающихся ошибок обратитесь к главе "Расширенные инструкции, распределенный ввод/вывод: Информация об ошибках для RDREC, WRREC и RALRM" (страница 358).

9.4 Прерывания

9.4.1. Инструкции ATTACH и DETACH (подключение и отключение ОВ и прерывания)

С помощью инструкций ATTACH и DETACH Вы можете активировать или деактивировать прерывание управляемых событиями подпрограмм.

Таблица 9-85 Инструкции ATTACH и DETACH

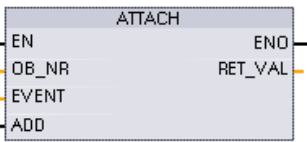
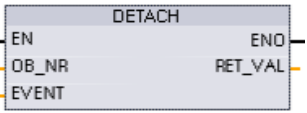
LAD / FBD	SCL	Описание
	<pre>ret_val := ATTACH(ob_nr:=_int_in_, event:=_event_att_in_, add:=_bool_in_);</pre>	Инструкция ATTACH включает ОВ обработки прерывания выполнения подпрограммы для аппаратного прерывания.
	<pre>ret_val := DETACH(ob_nr:=_int_in_, event:=_event_att_in_);</pre>	Инструкция DETACH отключает ОВ обработки прерывания выполнения подпрограммы для аппаратного прерывания.

Таблица 9-86 Типы данных для параметров

Параметр и тип		Тип данных	Описание
OB_NR	IN	OB_ATT	Идентификатор организационного блока: Выберите из доступных ОВ обработки аппаратных прерываний блок, созданный с помощью функции "Add new block" (Вставка нового блока). Дважды щелкните мышкой на поле параметров, затем щелкните на значке помощника для просмотра доступных ОВ.
EVENT	IN	EVENT_ATT	Идентификатор события: Выберите из доступных событий аппаратное прерывание, которое было включено в конфигурации устройств PLC для цифровых входов или высокоскоростных счетчиков. Дважды щелкните мышкой на поле параметров, затем щелкните на значке помощника для просмотра доступных событий.
ADD (ATTACH only)	IN	Bool	- ADD = 0 (по умолчанию): Данное событие заменяет все предыдущие события, закрепленные за данным ОВ. - ADD = 1: Данное событие добавляется к другим событиям, закрепленным за данным ОВ.
RET_VAL	OUT	Int	Выполнение кода условия

Аппаратные прерывания

CPU поддерживает следующие аппаратные прерывания:

- По переднему фронту: первые 12 дискретных входов, встроенные в CPU, (с DIa.0 по DIb.3) и все дискретные входы SB
 - Передний фронт формируется, когда состояние дискретного входа изменяется с OFF (выкл.) на ON (вкл.), в качестве реакции на изменение сигнала от полевого устройства, подключенного к этому входу.
- По заднему фронту: первые 12 дискретных входов, встроенные в CPU, (с DIa.0 по DIb.3) и все дискретные входы SB
 - Задний фронт формируется при переходе дискретного входа из ON в OFF.
- Текущее значение высокоскоростного счетчика (HSC) = заданное значение (CV = RV) событий (HSC с 1 по 6)
 - CV = RV прерывание для HSC сгенерировано, когда выполняется переход от предыдущего значения счетчика к значению, в точности соответствующему ранее заданному значению.
- Изменение направления HSC (HSC с 1 по 6)
 - Изменения направления происходит, когда HSC обнаруживает изменение счёта от увеличения к уменьшению или от уменьшения к увеличению.
- Внешний сброс HSC (HSC с 1 по 6)
 - Некоторые режимы HSC позволяют назначать дискретный вход для выполнения внешнего сброса значения HSC в 0. Для такого HSC внешний сброс происходит, когда состояние назначенного входа изменяется с OFF на ON.

Включение аппаратных прерываний в конфигурации устройства

Аппаратные прерывания могут быть включены во время конфигурирования устройства. Если Вы хотите закрепить это событие при конфигурации или во время рабочего цикла, то в окне конфигурации устройств Вам необходимо установить флажок включения опции "enable-event" (включение по событию) для канала дискретного входа или для HSC.

Опции независимого переключателя (кнопки для установки флажка) в конфигурации устройств PLC:

- Дискретный вход
 - Детектирование включения по переднему фронту
 - Детектирование включения по заднему фронту
- Высокоскоростной счетчик (High-speed counter = HSC)
 - Включение высокоскоростного счетчика.
 - Генерирование прерывания при равенстве текущего значения счетчика заданному значению.
 - Генерирование прерывания при выполнении внешнего сброса.
 - Генерирование прерывания при смене направления счёта.

Вставка новых ОВ обработки аппаратных прерываний в Вашу программу

По умолчанию, ОВ не закреплен за событием, возникающем при первом включении. Это отображено для "HW interrupt:" метка "<not connected>" конфигурации устройств. За событием, вызвавшим аппаратное прерывание, могут быть закреплены только ОВ обработки аппаратных прерываний. Все существующие ОВ обработки аппаратных прерываний отображаются в выпадающем списке "HW interrupt:". Если ОВ в списке отсутствуют, то Вам необходимо создать ОВ типа "Hardware interrupt" следующим образом. В ветке "Program blocks" (Программные блоки) дерева проекта:

1. Дважды щелкните мышкой на "Add new block" (Вставить новый блок), выберите "Organization block (OB)" (Организационный блок), а затем "Hardware interrupt" (Аппаратное прерывание).
2. Кроме того, Вы можете переименовать ОВ, выбрать язык программирования (LAD, FBD или SCL), выбрать номер блока (переключитесь в ручной режим и выберите другой номер блока вместо предложенного Вам).
3. Отредактируйте ОВ и добавьте запрограммированное действие, которое должно быть выполнено после возникновения события. Из этого ОВ Вы можете вызвать FC- и FB-блоки с максимальной глубиной вложения – 6.

Параметр OB_NR

Все имена существующих ОВ обработки аппаратных прерываний доступны в выпадающем списке конфигурации устройств "HW interrupt:" и в выпадающем списке ATTACH / DETACH (подключить, отключить) параметра OB_NR.

Параметр EVENT

При возникновении события, вызывающего аппаратное прерывание, то данному событию по умолчанию присваивается уникальное имя. Имя события Вы можете изменить, отредактировав его в окне "Event name:" (Имя события), но это имя также должно быть уникальным. Эти имена событий становятся и именами тегов в таблице тегов "Constants" (Константы), и отображаются в раскрывающемся списке параметра EVENT для инструкций ATTACH и DETACH. Значение тега – это внутренний номер, используемый для идентификации события.

Основные операции

Каждое аппаратное событие может быть вставлено в ОВ обработки аппаратного прерывания, которое будет поставлено в очередь на выполнение в случае возникновения события, вызвавшего аппаратное прерывание. Закрепление ОВ за событием может быть выполнено при конфигурировании или во время рабочего цикла.

Для закрепления ОВ за событием при конфигурировании Вам необходимо воспользоваться выпадающим списком "HW interrupt:" (Аппаратное прерывание) (нажмите на стрелку "вниз" справа) и выбрать ОВ из доступного списка ОВ обработки аппаратных прерываний. В этом списке выберите имя необходимого ОВ, или выберите "<not connected>" (не подключен) для удаления закрепления.

Для закрепления необходимого ОВ за событием или для отключения ОВ от события во время рабочего цикла Вы можете воспользоваться программными инструкциями ATTACH или DETACH (несколько раз, если необходимо). Если в настоящее время ОВ не закреплен за событием (выбрана опция "<not connected>" в конфигурации устройств или в результате выполнения инструкции DETACH), то включенное аппаратное прерывание игнорируется.

Операция DETACH

Инструкция DETACH используется для отключения отдельного события или всех событий от заданного ОБ. Если задан параметр EVENT, то только одно событие будет отключено от заданного ОБ_NR; остальные события, закрепленные в настоящий момент за ОБ_NR, остаются без изменений. Если параметр EVENT не задан, то все события, закрепленные в настоящий момент за ОБ_NR, будут отключены.

Коды ошибок

Таблица 9-87 Коды ошибок

RET_VAL (W#16#....)	ENO	Описание
0000	1	Ошибки отсутствуют
0001	1	Нечего отключать (только DETACH)
8090	0	ОБ больше не существует
8091	0	Недействительный тип ОБ
8093	0	Событие больше не существует

9.4 Прерывания

9.4.2. Циклические прерывания

9.4.2.1. Инструкция SET_CINT (Установка параметров циклических прерываний)

Таблица 9-88 Инструкция SET_CINT (Установка параметров циклических прерываний)

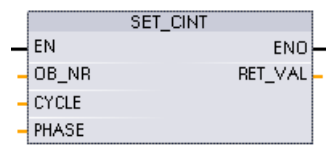
LAD / FBD	SCL	Описание
	<pre>ret_val := SET_CINT(ob_nr:=_int_in_, cycle:=_udint_in_, phase:=_udint_in_);</pre>	Установка циклически выполняемого ОВ обработки прерываний, прерывающего сканирование программы.

Таблица 9-89 Типы данных для параметров

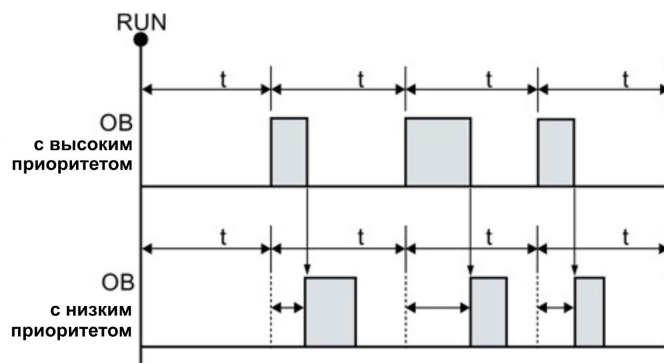
Параметр и тип		Тип данных	Описание
OB_NR	IN	OB_CYCLIC	Номер ОВ (принимает символьное имя)
CYCLE	IN	UDInt	Интервал времени, в миллисекундах
PHASE	IN	UDInt	Фазовый сдвиг, в микросекундах
RET_VAL	OUT	Int	Код состояния выполнения

Примеры: временные параметры

- Если время CYCLE = 100 мкс, то ОВ обработки прерывания, ссылающийся на OB_NR, прерывает циклическое сканирование программы каждые 100 мкс. Выполняется ОВ обработки прерывания, а затем управление возвращается программе сканирования с точки прерывания.
- Если время CYCLE = 0, то прерывание деактивировано, а ОВ обработки прерывания не выполняется.
- Время PHASE (фазовый сдвиг) – это заданная задержка времени, которая вставляется в начале интервала времени CYCLE. Вы можете использовать фазовый сдвиг для управления синхронизацией ОВ с низким приоритетом.

Если в один и тот же интервал времени вызваны ОВ с высоким и низким приоритетом, то ОВ с низким приоритетом только после завершения выполнения ОВ с высоким приоритетом. Начало времени выполнения ОВ с низким приоритетом может быть сдвинуто в зависимости от времени выполнения ОВ с высоким приоритетом.

Вызов ОВ без фазового сдвига



Если Вы хотите начать выполнение ОВ с низким приоритетом в фиксированное время цикла, то время фазового сдвига может быть больше времени выполнения ОВ с высоким приоритетом.

Вызов ОВ с фазовым сдвигом

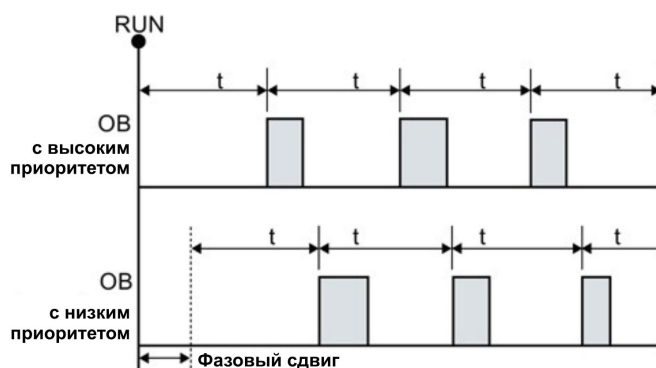


Таблица 9-90 Коды ошибок

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	ОВ больше не существует или недействительный тип
8091	Недопустимое время цикла
8092	Недопустимое время фазового сдвига
80B2	ОВ не закреплен за событием

9.4 Прерывания

9.4.2.2. Инструкция QRY_CINT (Параметры запроса циклического прерывания)

Таблица 9-91 Инструкция QRY_CINT (Запрос циклического прерывания)

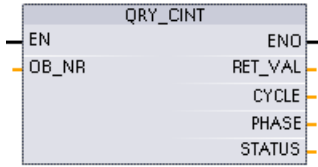
LAD / FBD	SCL	Описание
	<pre>ret_val := QRY_CINT(ob_nr:=_int_in_, cy- cle=>_uint_out_, phase=>_uint_out_ , status=>_word_out_);</pre>	Получение параметра и состояния выполнения из OB обработки циклического прерывания. Значения, возвращаемые в момент работы инструкции QRY_CINT, выполнены.

Таблица 9-92 Типы данных для параметров

Параметр и тип		Тип данных	Описание
OB_NR	IN	OB_CYCLIC	Номер OB (допускается символьное имя, например, OB_MyOBName)
RET_VAL	OUT	Int	Код состояния выполнения
CYCLE	OUT	UDInt	Интервал времени в микросекундах
PHASE	OUT	UDInt	Фазовый сдвиг в микросекундах
STATUS	OUT	Word	Код состояния циклического прерывания: - Биты с 0 по 4, смотрите расположенную ниже таблицу STATUS - Остальные биты всегда сброшены в 0

Таблица 9-93 Параметр STATUS

Бит	Значение	Описание
0	0	При RUN-режиме CPU
	1	При запуске
1	0	Прерывание включено.
	1	Прерывание отключено через инструкцию DIS_IRT.
2	0	Прерывание неактивно или закончилось.
	1	Прерывание активно.
4	0	OB, идентифицированный OB_NR, не существует.
	1	OB, идентифицированный OB_NR, существует.
Остальные биты		Всегда 0

Если произошла ошибка, RET_VAL отображает соответствующий код ошибки, а параметр STATUS= 0.

Таблица 9-94 Параметр RET_VAL

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	ОВ не существует или неправильный тип.
80B2	ОВ на закреплен за событием.

9.4.3. Прерывания по времени суток

ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник получит доступ к Вашей сети через протокол сетевого времени Network Time Protocol (NTP), с помощью которого в Internet производится синхронизация системного времени компьютера пользователя с системным временем сервера, то он может влиять на управление Вашим процессом, сдвигая системное время CPU.

По умолчанию свойства NTP-клиента S7-1200 CPU не активированы, а при включении позволяют сконфигурировать только IP-адреса для работы в качестве NTP-сервера. По умолчанию это свойство CPU отключено, и Вам необходимо его сконфигурировать, чтобы иметь возможность корректировать системное время дистанционно-управляемого CPU.

S7-1200 CPU поддерживает прерывания по времени суток ("time of day") и инструкции установки часов (clock instructions), которые зависят от точности системного времени CPU. При конфигурировании NTP и синхронизации с системным временем сервера Вам необходимо убедиться, что сервер является надежным источником. Невыполнение этого требования может стать причиной нарушения безопасности, что позволит неизвестному пользователю получить ограниченное управление Вашим процессом путем сдвига системного времени CPU.

Для получения информации и рекомендаций по обеспечению безопасности обратитесь к руководству "Operational Guidelines for Industrial Security" (Рабочие директивы по промышленной безопасности), доступному на сайте поддержки Siemens (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf).

9.4 Прерывания

9.4.3.1. SET_TINTL (установка параметров прерывания "time of day")

Таблица 9-95 SET_TINTL (Установка параметров прерывания "date and time of day" с помощью типа данных DTL)

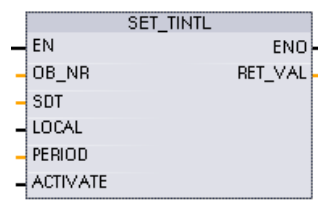
LAD / FBD	SCL	Описание
	<pre>ret_val := SET_TINTL(OB_NR:=_int_in_, SDT:=_dtl_in_, LOCAL:=_bool_in_ PERIOD:=_word_in_ ACTIVATE:=_bool_in_);</pre>	Установка даты и времени суток запуска события, вызывающего прерывание. Программа ОБ обработки прерывания может быть сконфигурирована для однократного исполнения или для исполнения, повторяющегося через заданный интервал времени.

Таблица 9-96 Тип данных для параметров

Параметр и тип	Тип данных	Описание
OB_NR	IN	OB_TOD (INT)
SDT	IN	DTL
LOCAL	IN	Bool
PERIOD	IN	Word
ACTIVATE	IN	Bool
RET_VAL	OUT	Int

В своей программе Вы можете использовать значения SET_TINTL даты и времени суток запуска события, вызывающего прерывание, которое должно быть обработано соответствующим ОБ обработки прерывания. Дата и время старта задается параметром SDT, а период времени для повторяющихся прерываний (например, ежедневно или еженедельно) задается параметром PERIOD. Если Вы выбрали ежемесячный период повторения, то день запуска события, вызывающего прерывание, Вы должны установить с 1 по 28. Дни с 29 по 31 не могут быть использованы, поскольку они не встречаются в феврале. Если Вы хотите запускать прерывание в конце каждого месяца, то используйте опцию "end of month" параметра PERIOD.

Значение "weekday" (будний день) в DTL-данных параметра SDT игнорируется. Установка фактических значений даты и времени выполняется в online-режиме CPU с помощью функции "Set time of day" (Установка времени суток) в окне "Online & diagnostics" (Online-функции и диагностика). Вы можете установить месяц, день месяца и год. STEP 7 рассчитывает периодичность выполнения события, вызывающего прерывание, на основе значений даты и часов CPU.

Примечание

При переходе с летнего времени на зимнее (декретное время) первый час дня отсутствует. Установите время старта в течение второго часа или используйте дополнительную задержку времени выполнения события, вызывающего прерывание, с учетом первого часа.

Таблица 9-97 Коды ошибок

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недействительный параметр OB_NR
8091	Недействительное значение времени запуска параметра SDT: (например, начало запуска в течение пропущенного часа в начале перехода на декретное время)
8092	Недействительное значение параметра PERIOD
80A1	Время запуска закончилось. (Этот код ошибки возникает только при PERIOD = W #16#0000.)

9.4.3.2. Инструкция CAN_TINT (Отмена прерывания "time of day")

Таблица 9-98 Инструкция CAN_TINT (Отмена даты и времени запуска события, вызывающего прерывание)

LAD / FBD	SCL	Описание
	<pre>ret_val:=CAN_TINT(_int_in);</pre>	Отмена даты и времени запуска события, вызывающего прерывание, для заданного OB обработки прерывания.

Таблица 9-99 Типы данных для параметров

Параметр и тип	Тип данных	Описание
OB_NR	IN	OB_TOD (INT)
RET_VAL	OUT	Int

9.4 Прерывания

Таблица 9-100 Коды ошибок

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недействительный параметр OB_NR
80A0	Отсутствуют заданные значения даты и времени запуска для данного OB обработки прерывания

9.4.3.3. Инструкция ACT_TINT (Активация прерывания по времени дня)

Таблица 9-101 ACT_TINT (Активация запуска прерывания по дате и времени дня)

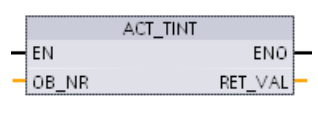
LAD / FBD	SCL	Описание
	<pre>ret_val:=ACT_TINT(_int_in_);</pre>	Активация запуска события, вызывающего прерывание, по дате и времени дня для заданного OB обработки прерывания.

Таблица 9-102 Типы данных для параметров

Параметр и тип	Тип данных	Описание
OB_NR	IN	OB_TOD (INT) Номер OB (допустимо использование символьного имени)
RET_VAL	OUT	Int Код состояния выполнения

Таблица 9-103 Коды ошибок

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недействительный параметр OB_NR
80A0	Для соответствующего OB обработки прерывания не заданы дата и время дня запуска события
80A1	Активированное время закончилось. Эта ошибка возникает, только когда выбран однократный режим (once) запуска OB обработки прерывания.

9.4.3.4. QRY_TINT (Запрос состояния прерывания по времени дня)

Таблица 9-104 QRY_TINT (Запрос даты и времени дня запуска события, вызывающего прерывание)

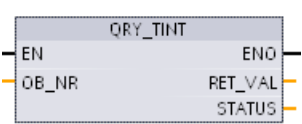
LAD / FBD	SCL	Описание
	<pre>ret_val:=QRY_TINT(OB_NR:=_int_in_, STATUS=>_word_out_);</pre>	Запрос даты и времени дня запуска события, вызывающего прерывание, для заданного OB обработки прерывания.

Таблица 9- 105 Типы данных для параметров

Параметр и тип		Тип данных	Описание
OB_NR	IN	OB_TOD (INT)	Номер запрашиваемого OB обработки прерывания (допустимо использование символьного имени)
RET_VAL	OUT	Int	Код состояния выполнения
STATUS	OUT	Word	Состояние заданного OB обработки прерывания

Таблица 9-106 Параметр STATUS

Бит	Значение	Описание
0	0	Режим Run
	1	Режим запуска (Startup)
1	0	Прерывание включено.
	1	Прерывание выключено.
2	0	Прерывание неактивно или закончилось.
	1	Прерывание активно.
4	0	Назначенный параметр OB_NR не существует.
	1	OB с назначенным параметром OB_NR существует.
6	1	Прерывание по дате и времени дня использует местное время.
	0	Прерывание по дате и времени дня использует системное время.
Остальные биты		Всегда 0

Таблица 9-107 Коды ошибок

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недействительный параметр OB_NR

9.4 Прерывания

9.4.4. Время задержки прерываний

Запуск или отмену задержки времени обработки прерывания Вы можете выполнить с помощью инструкций SRT_DINT и CAN_DINT или выполнить запрос о состоянии прерывания с помощью инструкции QRY_DINT. Каждая задержка обработки прерывания – это разовое событие, которое происходит после заданного времени задержки. Если время задержки обработки события отменено до истечения времени задержки, то прерывания выполнения программы не происходит.

Таблица 9-108 Инструкции SRT_DINT, CAN_DINT и QRY_DINT

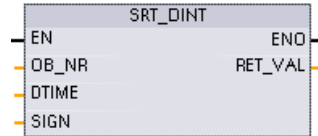
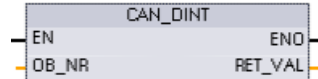
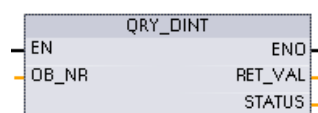
LAD / FBD	SCL	Описание
	<pre>ret_val := SRT_DINT(ob_nr:=_int_in_, dtime:=_time_in_, sign:=_word_in_);</pre>	Если время задержки, заданное параметром DTIME, истекло, то SRT_DINT запускает OB обработки прерывания.
	<pre>ret_val := CAN_DINT(ob_nr:=_int_in_);</pre>	Инструкция CAN_DINT отменяет время задержки прерывания, которое уже запущено. В этом случае время задержки OB обработки прерывания не выполняется.
	<pre>ret_val := QRY_DINT(ob_nr:=_int_in_, status=>_word_out_);</pre>	Инструкция QRY_DINT запрашивает состояние времени задержки прерывания, заданного параметром OB_NR.

Таблица 9-109 Типы данных для параметров

Параметр и тип		Тип данных	Описание
OB_NR	IN	OB_DELAY	Организационный блок (OB), запущенный после времени задержки: Выберите OB обработки time-delay прерываний, созданный с помощью свойства "Add new block" (Вставить новый блок) дерева проекта. Дважды щелкните мышкой на поле параметров, затем щелкните мышкой на значке помощника для просмотра доступных OB.
DTIME ¹	IN	Time	Значение времени задержки (от 1 до 60000 мс)
SIGN ¹	IN	Word	Не используется S7-1200: Допустимо любое значение. Значение должно быть назначено для предупреждения возникновения ошибок.
RET_VAL	OUT	Int	Код состояния выполнения
STATUS	OUT	Word	Инструкция QRY_DINT: Состояние заданного OB обработки time-delay прерывания, смотрите таблицу ниже

¹ Только для SRT_DINT

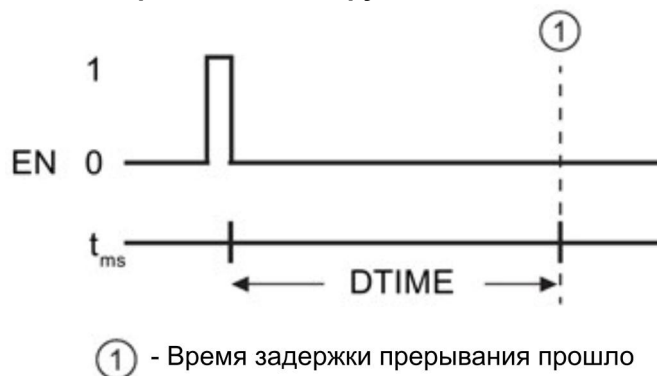
Работа

Если EN=1, то инструкция SRT_DINT запускает внутренний таймер времени задержки (DTIME). По истечении времени задержки CPU генерирует программное прерывание, чтобы активировать выполнение соответствующего OB обработки time-delay прерываний. С помощью инструкции CAN_DINT Вы можете отменить внутреннее процессорное время задержки прерывания перед началом выполнения заданного времени задержки. Суммарное количество активных задержек событий, вызывающих прерывание, не должно превышать 4.

Примечание

Если EN=1, то SRT_DINT запускает таймер времени задержки при каждом сканировании. Для запуска времени задержки используется одновибратор, а не простая подача "1" на вход EN.

Временная диаграмма для инструкции SRT_DINT:



Добавление OB обработки time-delay прерываний в Ваш проект

OB обработки time-delay прерываний Вы можете назначить только инструкциям SRT_DINT и CAN_DINT. В новом проекте OB обработки time-delay прерываний отсутствуют. OB обработки time-delay прерываний Вы можете вставить в свой проект. Для создания OB обработки time-delay прерываний выполните следующие шаги:

1. Дважды щелкните мышкой на элементе "Add new block" (Вставить новый блок) в ветви "Program blocks" (Программные блоки) дерева проекта, выберите "Organization block (OB)", затем выберите "Time delay interrupt".
2. Вы можете переименовать OB, выбрав язык программирования или номер блока. Переключитесь в режим ручной. если вы хотите назначить блоку другой номер, чем том, который был присвоен автоматически.
3. Отредактируйте подпрограмму OB обработки time-delay прерываний и запрограммируйте событие, которое должно быть выполнено по истечении времени задержки. Из OB обработки time-delay прерываний Вы можете вызвать другие FC- и FB-блоки с максимальной глубиной вложения 6.
4. Имена вновь назначенных OB обработки time-delay прерываний должны быть доступны при редактировании параметра OB_NR инструкций SRT_DINT и CAN_DINT.

9.4 Прерывания

Параметр STATUS инструкции QRY_DINT

Таблица 9-110 В случае ошибки (REL_VAL <> 0) параметр STATUS = 0.

Бит	Значение	Описание
0	0	Режим RUN
	1	Режим запуска (Startup)
1	0	Прерывание включено.
	1	Прерывание выключено.
2	0	Прерывание неактивно или закончилось.
	1	Прерывание активно.
4	0	ОВ с номером, указанным в OB_NR, не существует.
	1	ОВ с номером, указанным в OB_NR, существует.
Остальные биты		Всегда 0

Коды ошибок

Таблица 9-111 Коды ошибок для SRT_DINT, CAN_DINT и QRY_DINT

RET_VAL (W#16#...)	Описание
0000	Ошибки не обнаружены
8090	Некорректное значение параметра OB_NR
8091	Некорректное значение параметра DTIME
80A0	Время задержки прерывания не запущено.

9.4.5. Инструкции DIS_AIRT и EN_AIRT (Задержка/включение выполнения прерываний с высоким приоритетом и события, вызывающие асинхронные ошибки)

Инструкции DIS_AIRT и EN_AIRT используются для включения и отключения обработки аварийных прерываний.

Таблица 9-112 Инструкции DIS_AIRT и EN_AIRT

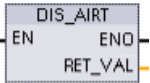
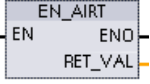
LAD / FBD	SCL	Описание
	DIS_AIRT() ;	DIS_AIRT задерживает обработку новых событий, вызывающих прерывание. Инструкция DIS_AIRT может быть выполнена в ОБ несколько раз.
	EN_AIRT() ;	EN_AIRT включает обработку событий, вызывающих прерывание и ранее отключенные Вами с помощью инструкции DIS_AIRT. Каждое выполнение инструкции DIS_AIRT может быть отменено с помощью инструкции EN_AIRT. Выполнение инструкции EN_AIRT должно быть запрограммировано в том же ОБ, или любом FC- или FB-блоке, вызываемом из этого ОБ, перед прерываниями, вновь включенными в этот ОБ.

Таблица 9-113 Типы данных для параметров

Параметр и тип	Тип данных	Описание
RET_VAL	OUT	Int
		Номер задержки = номер в последовательности выполнения инструкции DIS_AIRT.

Количество выполнений инструкций DIS_AIRT подсчитывается операционной системой. Каждое из них остается действительным, пока снова не будет отменено специальной инструкцией EN_AIRT, или пока текущий ОБ не будет полностью обработан. Например: если прерывания были отключены Вами пять раз с помощью выполнения пяти раз инструкции DIS_AIRT, то Вы должны отменить выполнение пяти EN_AIRT инструкций перед новым включением прерываний.

После нового включения событий, вызывающих прерывания, будут обработаны прерывания, происшедшие во время DIS_AIRT, или прерывания после выполнения текущего ОБ.

Параметр RET_VAL указывает на количество отключений обработки прерываний, количество которых поставлено в очередь на выполнение DIS_AIRT. Обработка прерываний будет вновь активирована, только когда параметр RET_VAL = 0.

9.5 Диагностика (PROFINET или PROFIBUS)

9.5.1. Диагностические инструкции

С PROFINET или PROFIBUS Вы можете использовать следующие диагностические инструкции:

- Инструкция LED (стр. 385): Вы можете считывать состояние светодиодных индикаторов устройств распределенного ввода/вывода.
- Инструкция DeviceStates (стр.386): Вы можете получать информацию о рабочем состоянии устройств распределенного ввода/вывода в пределах подсистемы распределенного ввода/вывода.
- Инструкция ModuleStates (стр.392): Вы можете получать информацию о рабочем состоянии модулей устройств распределенного ввода/вывода.
- Инструкция GET_DIAG (стр.398): Вы можете считывать диагностическую информацию из конкретного устройства.
- Инструкция Get_IM_Data (стр.404): Вы можете проверить диагностические данные и данные о техническом обслуживании (I&M) для конкретного модуля или submodule.

9.5.2. Диагностические события для распределенного ввода/вывода

Примечание

В PROFIBUS IO системе после загрузки программы или подачи питания CPU должен перейти в режим RUN, если совместимость оборудования допускает выполнение замены модулей (страница 162), а один или несколько модулей неисправны или для сконфигурированного модуля невозможна замена.

Как показано в приведенной таблице, CPU поддерживает диагностические данные, которые могут быть сконфигурированы для компонентов системы распределенного ввода/вывода. Каждая из этих ошибок генерирует запись, вводимую в диагностический буфер.

Таблица 9-114 Обработка диагностических событий для PROFINET и PROFIBUS

Тип ошибки	Диагностическая информация станции?	Введена в диагностический буфер?	Режим работы CPU
Диагностическая ошибка	Да	Да	Остается в режиме RUN
Неисправность стойки или станции	Да	Да	Остается в режиме RUN
Ошибка доступа к вводу/выводу ¹	Нет	Да	Остается в режиме RUN
Ошибка доступа к периферии ²	Нет	Да	Остается в режиме RUN
Вставка/извлечение модуля	Да	Да	Остается в режиме RUN

¹ Пример ошибки доступа к вводу/выводу: Модуль был удален.

² Пример ошибки доступа к периферии: Ациклическая коммуникация с submodule, который не отвечает.

Инструкция GET_DIAG (страница 398) используется для получения диагностической информации от каждой из станций. Это позволит Вам программно обрабатывать ошибки, возникающие в устройстве, и при необходимости переводить CPU в режим STOP. Этот метод требует от Вас задания конкретного аппаратного устройства, из которого необходимо считывать информацию о состоянии.

Инструкция GET_DIAG использует "L-адрес" (LADDR) для получения информации о состоянии всей станции. Этот L-адрес можно найти в окне отображения конфигурации сети (Network Configuration) и выбранной стойки станции (серая область ввода), L-адрес отображается на вкладке свойств (Properties) станции. LADDR для каждого отдельного модуля Вы можете найти или в свойствах модуля (в конфигурации устройства) или в таблице тегов по умолчанию для CPU.

9.5.3. Инструкция LED (Считывание состояния LED индикаторов)

Таблица 9-115 Инструкция LED

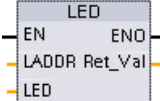
LAD / FBD	SCL	Описание
	<pre>ret_val := LED(laddr:=_word_in_, LED:=_uint_in_);</pre>	Инструкция LED используется для считывания состояния светодиодных индикаторов CPU или интерфейса. Данное состояние LED-индикаторов возвращено на выходе RET_VAL.

Таблица 9-116 Типы данных для параметров

Параметр и тип	Тип данных	Описание
LADDR	IN	HW_IO
LED	IN	UInt
		Идентификационный номер LED
		1 RUN/STOP
		2 Error (Ошибка)
		3 Maintenance (Обслуживание)
		4 Redundancy (Резервирование)
		5 Link (Канал связи)
		6 Tx/Rx (Обмен данными)
RET_VAL	OUT	Int
		Состояние LED-индикаторов

¹ Например, в выпадающем списке параметров Вы можете выбрать CPU (например "PLC_1") или PROFINET-интерфейс.

9.5 Диагностика (PROFINET или PROFIBUS)

Таблица 9-117 Состояние RET_VAL

RET_VAL (W#16#...)	Описание	
Состояние LED- индикаторов с 0 по 9	0	Индикатор отсутствует
	1	Выключен
	2	Включен цвет 1 (включен постоянно)
	3	Включен цвет 2 (включен постоянно)
	4	Включен цвет 1 (мигает с частотой 2 Гц)
	5	Включен цвет 2 (мигает с частотой 2 Гц)
	6	Индикаторы поочередно мигают цветом 1 и 2 с частотой 2 Гц
	7	Включен цвет 1 (Tx/Rx)
	8	Включен цвет 2 (Tx/Rx)
	9	Состояние LED-индикатора недоступно
8091	Устройства, идентифицируемого по LADDR, не существует	
8092	Устройство, идентифицируемое по LADDR, не содержит LED-индикаторов	
8093	LED-идентификатор не определен	
80Bx	CPU, идентифицируемый по LADDR, не поддерживает инструкцию LED	

9.5.4. Инструкция DeviceStates

Инструкция DeviceStates используется для возвращения состояния всех ведомых устройств ввода/вывода, подключенных к заданному мастер-устройству ввода/вывода.

Таблица 9-118 Инструкция DeviceStates

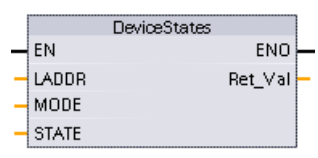
LAD / FBD	SCL	Описание
	<pre>ret_val := DeviceStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	DeviceStates возвращает рабочие состояния устройств ввода/вывода подсистемы ввода/вывода. После выполнения, параметр STATE содержит состояние ошибки каждого устройства ввода/вывода в списке битов (для назначенных LADDR и MODE). Эта информация соответствует состоянию устройства, показанному в окне отображения диагностики STEP 7. Вход LADDR инструкции DeviceStates используется для аппаратного идентификатора интерфейса распределенного ввода/вывода. В TIA portal, аппаратный идентификатор для PLC можно найти, просматривая типы данных "Hw_IoSystem" на вкладке системных констант таблицы PLC-тегов.

Таблица 9-119 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	IN	HW_IOSYSTEM	Логический адрес: (идентификатор системы ввода/вывода)
MODE	IN	UInt	Поддерживает пять рабочих режимов. Вход MODE определяет, какие из данных будут возвращены в ячейку, заданную параметром STATE. Перечень режимов: • 1: Сконфигурированное устройство активно • 2: Устройство неисправно • 3: Устройство отключено • 4: Устройство подключено • 5: Проблемы в устройстве
RET_VAL	OUT	Int	Код состояния выполнения
STATE ¹	InOut	Variant	Буфер, принимающий состояние ошибки от каждого устройства: Тип данных, выбранный Вами для параметра STATE, может быть битовыми данными различного типа (Bool, Byte, Word или DWord) или массивом битовых данных. • Бит 0 первого байта возвращенных данных STATE – это суммарный бит. Если он установлен в TRUE, то это означает, что доступны другие данные. • Данные, возвращаемые в параметре STATE, отображают взаимосвязь один-в-один между битовой ячейкой и адресом устройства распределенного ввода/вывода. Данное устройство установлено в TRUE для PROFIBUS и PROFINET. Например, бит 4 в первом байте имеет отношение к PROFIBUS адресу 4 или PROFINET устройству с номером 4.

¹ Для PROFIBUS-DP размер статусной информации составляет 128 бит. Для PROFINET I/O, этот размер – 1024 бита.

После выполнения инструкции параметр STATE содержит состояние ошибки для каждого устройства ввода/вывода в виде списка битов (для назначенных LADDR и MODE).

Таблица 9-120 Коды ошибок

RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	LADDR не подключено.
8092	LADDR не адресован системе ввода/вывода.
8093	Недействительный тип данных, назначенных параметру STATE: действительные типы данных (Bool, Byte, Word или Dword) или массивы (Bools, Bytes, Words или Dwords)
80Bx	Инструкция DeviceStates не поддерживается CPU для данного LADDR.
8452	Статусных данных слишком много для назначенного параметра STATE. Буфер STATE содержит частичный результат.

9.5 Диагностика (PROFINET или PROFIBUS)

9.5.4.1. Примеры конфигураций DeviceStates

Пример для PROFIBUS

Пример для PROFIBUS содержит:

- 16 PROFIBUS устройств, именованных с "DPSlave_10" по "DPSlave_25".
- 16 PROFIBUS устройств, использующих PROFIBUS адреса с 10 по 25, соответственно.
- Каждое ведомое устройство сконфигурировано с несколькими модулями ввода/вывода.
- Отображение первых четырех байтов данных, возвращенных параметру STATE.

Режим (MODE)	Пример 1: Нормальная работа без ошибок	Пример 2: Ведомое PROFIBUS- устройство DPSlave_12 с одним извлеченным модулем	Пример 3: Ведомое PROFIBUS- устройство DPSlave_12 отключено
1: Сконфигурированное устройство активно	0x01FC_FF03	0x01FC_FF03	0x01FC_FF03
2: Устройство неисправно	0x0000_0000	0x0110_0000	0x0110_0000
3: Устройство отключено	0x0000_0000	0x0000_0000	0x0000_0000
4: Устройство подключено	0x01FC_FF03	0x01FC_FF03	0x01EC_FF03
5: Проблемы в устройстве	0x0000_0000	0x0110_0000	0x0110_0000

В следующих четырех таблицах приведено распределение четырех байтов данных, которые должны быть проанализированы:

Таблица 9-121 Пример 1: Ошибки отсутствуют: Значение 0x01FC_FF03 возвращено для режима MODE 1 (Сконфигурированное устройство активно).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFC	Бит 15 1111-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Устройства сконфигурированы с адресами от 10 (бит 10) до 25 (бит 25).

Устройства с адресами от 1 до 9 не сконфигурированы.

Данные режима MODE 4 (Устройство подключено) соответствуют режиму MODE 1 (Сконфигурированное устройство активно), т.е. сконфигурированное устройство соответствует существующему устройству.

Таблица 9-122 Пример 2: Модуль был извлечен из ведомого PROFIBUS-устройства "DPSlave_12". Значение 0x0110_0000 возвращено для режима MODE 2 (Устройство неисправно).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x10	Бит 15 0001-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 12 (бит 12) помечено как неисправное.

Режим MODE 5 (Проблемы в устройстве) возвращает ту же информацию, что и режим MODE 2 (Устройство неисправно).

Таблица 9-123 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFIBUS-устройства "DPSlave_12". Значение 0x01FC_FF03 возвращено для режима MODE 4 (Устройство подключено).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFC	Бит 15 1111-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Даже если устройство 12 (бит 12) содержит ошибку, как показано выше в режиме MODE 2 (Устройство неисправно), устройство продолжает функционировать в сети, в которой вызван режим MODE 4 (Устройство подключено) для отображения устройства как существующего ("existing device").

Таблица 9-124 Пример 3: Ведомое PROFIBUS-устройство "DPSlave_12" было отключено (отключен кабель или напряжение питания) от PROFIBUS сети. "DPSlave_12" будет определено как неисправное устройство, а также как устройство, содержащее ошибку. Отличие в том, что "DPSlave_12" больше не определяется как существующее устройство. Значение 0x01EC_FF03 будет возвращено для режима MODE 4 (Устройство подключено).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xEC	Бит 15 1110-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Устройство 12 (бит 12) помечено как несуществующее. Кроме этого устройства, устройства с 10 по 25 остаются отмеченными как существующие.

Пример для PROFINET

Пример для PROFINET содержит:

- 16 ведомых PROFINET-устройств, именованных с "et200s_1" по "et200s_16"
- 16 PROFINET-устройств, использующих PROFINET-номера с 1 по 16, соответственно.
- Каждое ведомое устройство сконфигурировано с несколькими модулями ввода/вывода.
- Отображение первых четырех байтов данных, возвращенных параметру STATE.

MODE (Режим)	Пример 1: Нормальная работа без ошибок	Пример 2: Модуль ведомого PROFINET устройства et200s_1 извлечен	Пример 3: Ведомое PROFINET устройство et200s_1 отключено
1. Сконфигурированное устройство активно	0xFFFF_0100	0xFFFF_0100	0xFFFF_0100
2. Устройство неисправно	0x0000_0000	0x0300_0000	0x0300_0000
3. Устройство отключено	0x0000_0000	0x0000_0000	0x0000_0000
4. Устройство подключено	0xFFFF_0100	0xFFFF_0100	0xFDFF_0100
5. Проблемы в устройстве	0x0000_0000	0x0300_0000	0x0300_0000

В следующих четырех таблицах приведено распределение четырех байтов данных, которые должны быть проанализированы:

Таблица 9-125 Пример 1: Ошибки отсутствуют: Значение 0xFFFF_0100 возвращено для режима MODE 1 (Сконфигурированное устройство активно).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Адреса сконфигурированных устройств: с 1 (бит 1) по 16 (бит 16).

Устройства с адресами от 1 до 9 не сконфигурированы.

Данные режима MODE 4 (Устройство подключено) соответствуют режиму MODE 1 (Сконфигурированное устройство активно), т.е. сконфигурированное устройство соответствует существующему устройству.

Таблица 9-126 Пример 2: Модуль был извлечен из ведомого PROFINET-устройства "et200s_1". Значение 0x0300_0000 возвращено для режима MODE 2 (Устройство неисправно).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x03	Бит 7 0000-0011 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 1 (бит 1) помечено как неисправное. Т.к. устройство подключено, режим MODE 4 (Устройство подключено) отображает те же данные, что и при нормальной работе. Режим MODE 5 (Проблемы в устройстве) возвращает ту же информацию, что и режим MODE 2 (Устройство неисправно).

Таблица 9-127 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFINET-устройства "et200s_1". Значение 0xFFFF_0100 возвращено для режима MODE 4 (Устройство подключено).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Даже если устройство 1 (бит 1) содержит ошибку, как показано выше в режиме MODE 2 (Устройство неисправно), устройство, по-прежнему функционирующее в сети, вызывает режим MODE 4 (Устройство подключено), для отображения устройства как "существующее устройство".

Таблица 9-128 Пример 3: Ведомое PROFINET-устройство "et200s_1" отключено (отключен кабель или выключено питание) от сети PROFINET. Значение 0xFDFF_0100 возвращено для режима MODE 4 (Устройство подключено).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFD	Бит 7 1111-1101 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 1 (бит 1) отсутствует. Подключены устройства с 2 (бит 2) по 16 (бит 16).

9.5 Диагностика (PROFINET или PROFIBUS)

9.5.5. Инструкция ModuleStates

Инструкция ModuleStates может быть использована для возвращения состояния всех модулей в PROFIBUS или PROFINET станциях.

Таблица 9-129 Инструкция ModuleStates

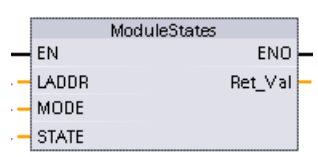
LAD / FBD	SCL	Описание
	<pre>ret_val := ModuleStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout);</pre>	Инструкция ModuleStates возвращает информацию о рабочих состояниях модулей ввода/вывода. После выполнения инструкции параметр STATE содержит информацию об ошибке каждого модуля ввода/вывода в виде списка битов (для заданных LADDR и MODE). Эта информация соответствует состоянию модуля, наблюдаемому в окне отображения диагностики STEP 7. ModuleStates использует вход LADDR в качестве аппаратного идентификатора станции распределенного ввода/вывода, а не ее главного модуля. Аппаратный идентификатор можно найти, выбрав станцию в окне отображения сети и просмотрев его в разделе аппаратных идентификаторов вкладки свойств. Его также можно найти, просмотрев типы данных "Hw_Device" и "Hw_DpSlave" вкладки системных констант таблицы PLC-тегов.

Таблица 9-130 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	IN	HW_DEVICE	Логический адрес (идентификатор модуля ввода/вывода)
MODE	IN	UInt	Вход MODE определяет, какие из данных должны быть возвращены в ячейку, определяемую параметром STATE. Поддерживаются следующие режимы работы: • 1: Сконфигурированный модуль активен • 2: Модуль неисправен • 3: Модуль выключен • 4: модуль включен • 5: Проблемы в модуле
RET_VAL	OUT	Int	Состояние (код условия)
STATE ¹	InOut	Variant	Буфер содержит информацию об ошибке каждого модуля. Используемый Вами тип данных параметра STATE может быть любым (Bool, Byte, Word или DWord) или массивом битового типа. • Бит 0 первого байта данных, возвращаемых параметру STATE, - это суммарный бит. Если он установлен в TRUE, то это означает, что данные доступны. • Данные, возвращаемые параметру STATE, являются корреляцией один-в-один между битовой ячейкой и позицией модуля. Адресация этого слота – TRUE для PROFIBUS и PROFINET. Например, для ET 200S с главным модулем, модулем питания и парой модулей ввода/вывода, бит 1 первого байта соответствует главному модулю, бит 2 – модулю питания, а биты 3 и 4 – соответственно модулям ввода/вывода.

- ¹ Может быть назначено максимум 128 бит. Количество необходимых бит зависит от количества используемых Вами модулей ввода/вывода.

Таблица 9-131 Коды ошибок

RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	Модуль, идентифицируемый для LADDR, не подключен.
8092	Модуль, идентифицируемый для LADDR, не адресован устройству ввода/вывода.
8093	Недействительный тип данных для параметра STATE: Действительные типы данных – Bool, Byte, Word или Dword; или массивы – Bools, Bytes, Words или Dwords.
80Bx	Инструкция ModuleStates не поддерживается данным CPU для данного LADDR.
8452	Слишком большое количество данных для назначенного параметра STATE. Буфер параметра STATE содержит частичный результат.

9.5 Диагностика (PROFINET или PROFIBUS)

9.5.5.1. Примеры конфигурации ModuleStates

Примеры для PROFIBUS

Примеры для PROFIBUS содержит:

- 16 PROFIBUS-устройств, именованных с "DPSlave_10" по "DPSlave_25"
- 16 PROFIBUS-устройств, использующие PROFIBUS адреса с 10 по 25, соответственно.
- Каждое ведомое устройство сконфигурировано с несколькими модулями ввода/вывода.
- В примере использован параметр LADDR ведомого PROFIBUS-устройства "DPSlave_12", содержащего главный модуль, модуль питания и два модуля ввода/вывода.
- Отображение первых четырех байтов данных, возвращенных параметру STATE.

MODE (Режим)	Пример 1: Нормальная работа без ошибок	Пример 2: Удален модуль ведомого PROFIBUS устройства DPSlave_12	Пример 3: Отключен модуль ведомого PROFIBUS устройства DPSlave_12
1: Сконфигурированный модуль активен	0x1F00_0000	0x1F00_0000	0x1F00_0000
2: Модуль неисправен	0x0000_0000	0x0900_0000	0x1F00_0000
3: Модуль отключен	0x0000_0000	0x0000_0000	0x0000_0000
4: Модуль подключен	0x1F00_0000	0x1700_0000	0x0000_0000
5: Проблемы в модуле	0x0000_0000	0x0900_0000	0x1F00_0000

Следующие четыре таблицы содержат двоичную разбивку четырех байт данных, которые должны быть проанализированы:

Таблица 9-132 Пример 1: Ошибки отсутствуют: Значение 0x1F00_0000 возвращено для режима MODE 1 (Сконфигурированный модуль активен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x1F	Бит 7 0001-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Слоты с 1 (бит 1) по 4 (бит 4) содержат модули. Слоты с 5 (бит 5) и далее не содержат модулей. Данные режима MODE 4 (Модуль подключен) соответствует режиму MODE 1 (Сконфигурированный модуль активен), т.е. сконфигурированные модули соответствуют подключенным модулям.

Таблица 9-133 Пример 2: Модуль был удален из ведомого PROFIBUS-устройства "DPSlave_12". Значение 0x0900_0000 возвращено для режима MODE 2 (Модуль неисправен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x09	Бит 7 0000-1001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Только модуль 3 (бит 3) помечен как неисправный. Все остальные модули работают.

Таблица 9-134 Пример 2 (продолжение): Модуль был удален из ведомого PROFIBUS-устройства "DPSlave_12". Значение 0x1700_0000 возвращено для режима MODE 4 (Модуль подключен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x17	Бит 7 0001-0111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модуль 3 (бит 3) показан как отсутствующий. Модули 1, 2 и 4 (биты 1, 2 и 4) показаны как подключенные.

Таблица 9-135 Пример 3: Ведомое PROFIBUS устройство "DPSlave_12" отключено (отключен кабель или напряжение питания) от сети PROFIBUS. Значение 0x1F00_0000 возвращено для режима MODE 2 (Модуль неисправен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x1F	Бит 7 0001-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модули в слотах с 1 по 4 (биты с 1 по 4) помечены как неисправные, т.к. устройство отсутствует. Режим MODE 5 (Проблемы в модуле) отображает ту же информацию, что и режим MODE 2 (Модуль неисправен).

Пример для PROFINET

Примеры для PROFINET содержат:

- 16 ведомых PROFINET устройств, именованных с "et200s_1" по "et200s_16"
- 16 PROFINET устройств, использующих номера PROFINET-устройств с 1 по 16, соответственно.
- Каждое ведомое устройство сконфигурировано с несколькими модулями ввода/вывода.
- В примере использовано ведомое PROFINET устройство "et200s_1", состоящее из главного модуля, модуля источника питания, и 18 модулей ввода/вывода.
- Отображение первых четырех байтов данных, возвращенных параметру STATE.

MODE (Режим)	Пример 1: Нормальная работа без ошибок	Пример 2: Ведомое PROFINET-устройство et200s_1 извлечено	Пример 3: Ведомое PROFINET-устройство et200s_1 отключено
1: Сконфигурированный модуль активен	0xFFFF_1F00	0xFFFF_1F00	0xFFFF_1F00
2: Модуль неисправен	0x0000_0000	0x0180_0000	0xFFFF_1F00
3: Модуль отключен	0x0000_0000	0x0000_0000	0x0000_0000
4: Модуль подключен	0xFFFF_1F00	0xFF7F_1F00	0x0000_0000
5: Проблемы в модуле	0x0000_0000	0x0180_0000	0xFFFF_1F00

В следующих четырех таблицах приведены примеры двоичного разбиения четырех байт данных, которые должны быть проанализированы:

Таблица 9-136 Пример 1: Ошибки отсутствуют: Значение 0xFFFF_1F00 возвращено для режима MODE 1 (Сконфигурированный модуль активен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x1F	Бит 23 0001-1111 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Слоты с 1 (бит 1) по 20 (бит 20) содержат модули. Слоты с 21 (бит 21) и выше не содержат модулей.

Данные режима MODE 4 (Модуль подключен) соответствуют данным режима MODE 1 (Сконфигурированный модуль активен), т.е. сконфигурированные модули соответствуют подключенным модулям.

Таблица 9-137 Пример 2: Был извлечен модуль из ведомого PROFINET устройства "et200s_1". Значение 0x0180_0000 возвращено для режима MODE 2 (Модуль неисправен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x80	Бит 15 1000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Только модуль 15 (бит 15) помечен как неисправный. Все остальные модули исправны.

Пример 9-138 Пример 2 (продолжение): Был извлечен модуль из ведомого PROFIBUS-устройства "et200s_1". Значение 0xFF7F_1F00 возвращено для режима MODE 4 (Модуль подключен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0x7F	Бит 15 0111-1111 Бит 8	
Байт 3 0x1F	Бит 23 0001-1111 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модуль 15 (бит 15) показан как отсутствующий. Модули с 1 по 14 (биты с 1 по 14) и с 16 по 20 (биты с 16 по 20) показаны как подключенные.

Таблица 9-139 Пример 3: Ведомое PROFINET-устройство "et200s_1" отключено (отключен кабель или напряжение питания) от сети PROFINET. Значение 0xFFFF_1F00 возвращено для режима MODE 2 (Модуль неисправен).

Байт со значением	Структура бита со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 действительный; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x1F	Бит 23 0001-1111 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модули в слотах с 1 по 20 (биты с 1 по 20) помечены как неисправные, т.к. устройство отсутствует. Данные режима MODE 5 (Проблемы в модуле) соответствуют информации для режима MODE 2 (Модуль неисправен).

9.5 Диагностика (PROFINET или PROFIBUS)

9.5.6. Инструкция GET_DIAG (Чтение диагностической информации)

Описание

Инструкция "GET_DIAG" используется для считывания диагностической информации устройства. Устройство выбирается с помощью параметра LADDR. С помощью параметра MODE Вы выбираете, какая диагностическая информация будет считана.

Таблица 9-140 Инструкция GET_DIAG

LAD / FBD	SCL	Описание
	<pre>ret_val := GET_DIAG(mode:=_uint_in_, laddr:=_word_in_, cnt_diag=>_uint_out_, diag:=_variant_inout_, detail:=_variant_inout_);</pre>	Чтение диагностической информации из заданного устройства.

Параметры

В следующей таблице приведены параметры инструкции "GET_DIAG":

Таблица 9-141 Типы данных для параметров

Параметр и тип		Тип данных	Описание
MODE	IN	UInt	Параметр MODE используется для выбора выводимых диагностических данных.
LADDR	IN	HW_ANY (Word)	Аппаратный идентификатор устройства
RET_VAL	OUT	Int	Состояние инструкции
CNT_DIAG	OUT	UInt	Количество выводимых диагностических команд
DIAG	InOut	Variant	Указатель области данных для хранения диагностической информации выбранного режима
DETAILS	InOut	Variant	Указатель области данных для хранения диагностических команд в соответствии с выбранным режимом

Параметр MODE

В зависимости от значения параметра MODE, различные диагностические данные выводятся в виде выходных параметров DIAG, CNT_DIAG и DETAILS:

Таблица 9-142 Параметр MODE

MODE	Описание	DIAG	CNT_DIAG	DETAILS
0	Вывод всей диагностической информации, поддерживаемой модулем, в виде DWord, где бит X=1 показывает, что режим X поддерживается.	Строка битов, поддерживаемая режимом, в виде DWord, где бит X=1 показывает, что режим X поддерживается.	0	-
1	Вывод состояния, соответствующего адресованному аппаратному объекту.	Диагностическое состояние: Вывод в соответствии с DIS-структурой. (Примечание: Обратитесь к приведенной ниже информации "DIS-структура" и примеру инструкции GET_DIAG в конце раздела.)	0	-
2	Вывод состояния всех подчиненных модулей адресованных аппаратному объекту.	Вывод диагностических данных в соответствии с DNN-структурой. (Примечание: Обратитесь к приведенной ниже информации "DNN-структура" и примеру инструкции GET_DIAG в конце раздела.)	0	-

DIS-структура

При значении параметра MODE = 1, диагностическая информация выводится в соответствии с DIS-структурой. В следующей таблице показано назначение отдельных значений параметров:

Таблица 9-143 DIS-структура (Diagnostic Information Source = DIS) (Источник диагностической информации)

Параметр	Тип данных	Значение	Описание
MaintenanceState	DWord	Enum	
		0	Обслуживания не требуется
		1	Модуль или устройство отключены.
		2	-
		3	-
		4	-
		5	необходимо обслуживание
		6	Запрос обслуживания
		7	Ошибка
		8	Состояние неизвестно / ошибка в подчиненном модуле
		9	-
		10	Входы/выходы недоступны.

9.5 Диагностика (PROFINET или PROFIBUS)

Параметр	Тип данных	Значение	Описание
Componentstate Detail	DWord	Бит массива	Состояние субмодулей: - Биты с 0 по 15: Телеграмма о состоянии модуля - Биты с 16 по 31: Телеграмма о состоянии CPU
		с 0 по 2 (enum)	Дополнительная информация: - Бит 0: Нет дополнительной информации - Бит 1: Передача не разрешена
		3	Бит 3 = 1: Как минимум один канал поддерживает диагностику.
		4	Бит 4 = 1: Необходимо обслуживание как минимум одного канала или компонента
		5	Бит 5 = 1: Зарос на обслуживание как минимум одного канала или компонента
		6	Бит 6 = 1: Ошибка как минимум в одном канале или компоненте
		с 7 по 10	Резерв (всегда = 0)
		с 11 по 14	Бит 11 = 1: PNIO – субмодуль корректный Бит 12 = 1: PNIO – замена модуля Бит 13 = 1: PNIO – некорректный модуль Бит 14 = 1: PNIO – модуль отключен
		15	Резерв (всегда = 0)
		с 16 по 31	Информация о состоянии модулей, сгенерированная для CPU: - Бит 16 = 1: Модуль отключен - Бит 17 = 1: Активен CiR-режим работы - Бит 18 = 1: Вход недоступен - Бит 19 = 1: Выход недоступен - Бит 20 = 1: переполнение диагностического буфера - Бит 21 = 1: Диагностическая информация недоступна - Бит 22 - 31: Резерв (всегда 0)
OwnState	UInt16	Enum	Значение параметра OwnState, описывающее состояние обслуживания модуля.
		0	Нет ошибок
		1	Модуль или устройство отключены.
		2	Необходимо обслуживание
		3	Запрос на обслуживание
		4	Ошибка
		5	Модуль или устройство недоступны из CPU (действительно для модулей и устройств за CPU).
		6	Входы/выходы недоступны.
IO State	UInt16	Бит массива	Состояние модулей ввода/вывода
		0	Бит 0 = 1: Обслуживания не требуется
		1	Бит 1 = 1: Модуль или устройство отключены.
		2	Бит 2 = 1: Необходимо обслуживание
		3	Бит 3 = 1: Запрос на обслуживание
		4	Бит 4 = 1: Ошибка

Параметр	Тип данных	Значение	Описание
		5	Бит 5 = 1: Модуль или устройство недоступны из CPU (действительно для модулей и устройств, расположенных за CPU).
		6	Входы/выходы недоступны.
		7	Классификатор; бит 7 = 1, если биты 0, 2 или 3 заданы
		с 8 по 15	Резерв (всегда = 0)
OperatingState	UInt16	Enum	
		0	-
		1	In STOP / обновление операционной системы
		2	In STOP / сброс памяти
		3	In STOP / автозапуск
		4	In STOP
		5	Сброс памяти
		6	In START
		7	In RUN
		8	-
		9	HOLD
		10	-
		11	-
		12	Модуль неисправен
		13	-
		14	Отсутствует напряжение питания
		15	CiR
		16	In STOP / без DIS
		17	In
		18	
		19	
		20	

9.5 Диагностика (PROFINET или PROFIBUS)

DNN-структура

При значении параметра MODE = 2 диагностическая информация выводится в соответствии с DNN-структурой. В следующей таблице приведены значения отдельных параметров:

Таблица 9-144 DNN-структура (Diagnostic Navigation Node = DNN)

Параметр	Тип данных	Значение	Описание
SubordinateState	UINT	Enum	Состояние подчиненного модуля (Смотрите параметр OwnState DIS-структуры).
SubordinateIOState	WORD	Bitarray	Состояние входов и выходов подчиненного модуля (Смотрите параметр IO State DIS-структуры).
DNNmode	WORD	Bitarray	- Бит 0 = 0: Диагностика включена - Бит 0 = 1: Диагностика выключена - Биты с 1 по 15: Резерв

Параметр RET_VAL

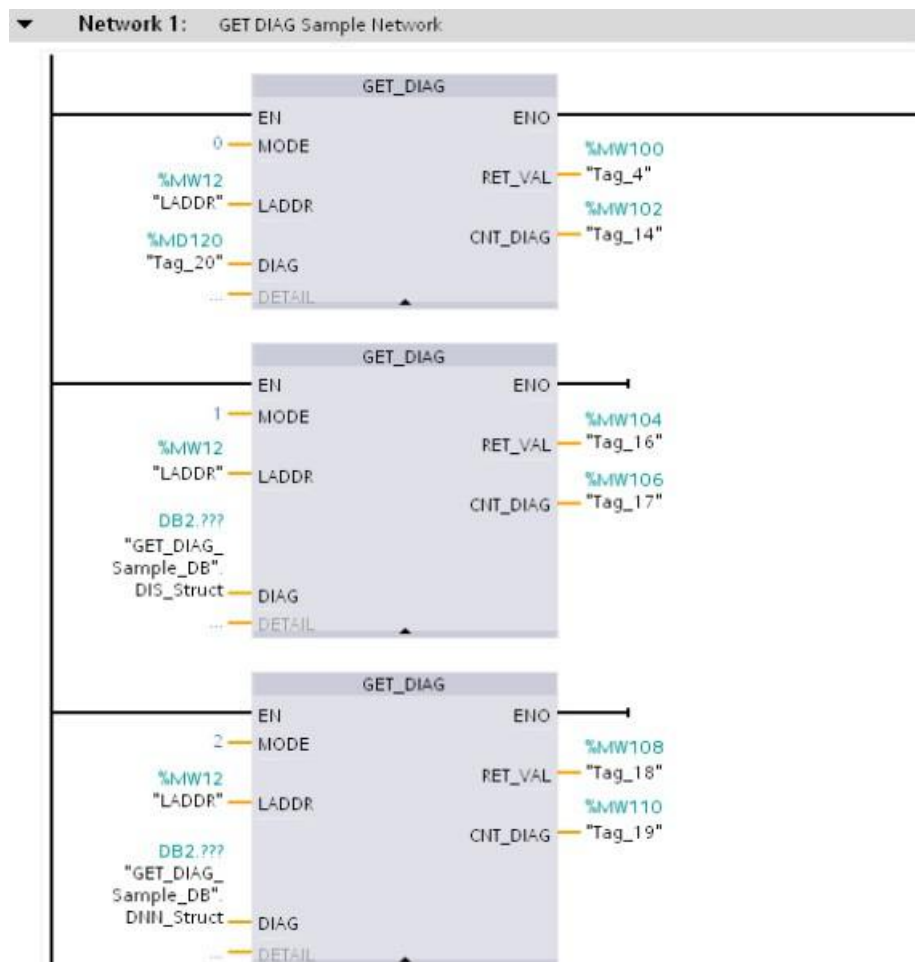
Таблица 9-145 Коды ошибок параметра RET_VAL

Код ошибки (W#16#...)	Описание
0	Ошибки отсутствуют
8080	Неподдерживаемое значение параметра MODE.
8081	Тип параметра DIAG не поддерживается в выбранном режиме (параметр MODE).
8082	Тип параметра DETAILS не поддерживается в выбранном режиме (параметр MODE).
8090	LADDR не существует.
8091	Выбранный канал параметра CHANNEL не существует.
80C1	Недостаточно ресурсов для параллельной работы

Пример

Приведенная многозвенная логическая схема сети и DB содержит примеры использования трех режимов с тремя структурами:

- DIS
- DNN



GET_DIAG_Sample_DB							
	Name	Data type	Offset	Start value	Retain	Visible in HMI	Comment
1	Static						
2	DNN_Struct	DNN	0.0		☐	☒	
3	SubordinateState	UInt	0.0	0	☐	☒	
4	SubordinateIOState	Word	2.0	0	☐	☒	
5	DNNmode	Word	4.0	0	☐	☒	
6	DIS_Struct	DIS	6.0		☐	☒	
7	MaintenanceState	DWord	0.0	0	☐	☒	
8	ComponentStateDetail	DWord	4.0	0	☐	☒	
9	OwnState	UInt	8.0	0	☐	☒	
10	IOState	Word	10.0	0	☐	☒	
11	OperatingState	UInt	12.0	0	☐	☒	

- ① DNN
② DIS

9.5 Диагностика (PROFINET или PROFIBUS)

Примечание

В DB Вам необходимо вручную ввести тип данных для доступа к каждой из трех структур; здесь отсутствует выбор в выпадающем меню. Типы данных приведены ниже:

- DNN
- DIS

9.5.7. Инструкция Get_IM_Data (Чтение идентификационных и эксплуатационных данных)

Инструкция Get_IM_Data используется для считывания идентификационных и эксплуатационных(I&M) данных заданного модуля или субмодуля.

Таблица 9-146 Инструкция Get_IM_Data

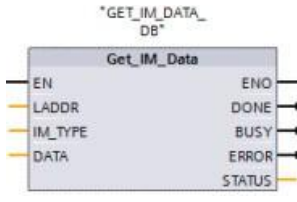
LAD / FBD	SCL	Описание
	<pre>"GET_IM_DATA_DB" (LADDR:=16#0 , IM_TYPE:=0, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	Инструкция Get_IM_Data используется для проверки идентификационных и эксплуатационных(I&M) данных заданного модуля или субмодуля.

Таблица 9-147 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	Input	HW_IO	Идентификатор модуля
IM_TYPE	Input	UInt	Номер идентификационных и эксплуатационных (I&M) данных: • 0: I&M0 (MLFB, серийный номер, версия и другая информация) • 1: I&M1 (Обозначение) • 2: I&M2 (Дата установки) • 3: I&M3 (Дескриптор) • 4: I&M4 (Сигнатура)
RET_VAL	Output	Int	Состояние (код условия)
DATA	InOut	Variant	I&M данные (STRING или массив BYTE)

Идентификационные и эксплуатационные (I&M) данные помогут Вам в проверке конфигурации системы, найти изменения в оборудовании или просмотреть данные о техническом обслуживании. Идентификационные данные модуля (I-данные) – только для чтения. Эксплуатационные данные модуля (M-данные) зависят от информации системы, например даты установки. M-данные создаются при планировании обслуживания и записаны в модуль:

- Если тип данных, используемых параметром DATA, - строка, то фактическая длина строки задана в соответствии с размером I&M данных.
- Если тип данных, используемых параметром DATA, - массив Byte или Char, то I&M-данные скопированы в виде последовательности байтов.
- Если тип данных, используемых параметром DATA, - структура (structure), то I&M-данные скопированы в виде последовательности байтов.
- Если массив типов данных byte/char параметра DATA больше запрашиваемых I&M-данных, то к значению байта добавляется 16#00.
- Остальные типы данных не поддерживаются, и возвращается код ошибки 8093.

Таблица 9-148 Коды ошибок

RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	LADDR не существует
8092	LADDR больше не адресовано HW-объекту с поддерживаемыми I&M-данными
8093	Тип данных, приведенных в параметре DATA, не поддерживается
80B1	Инструкция DATA не поддерживается CPU для данного LADDR
80B2	IM_TYPE не поддерживается CPU
8452	Полная I&M-информация не помещается в переменную, определяемую параметром DATA. Частичный результат переменной возвращается.

9.6 Импульсы

9.6.1. Инструкция CTRL_PWM (Широтно-импульсная модуляция)

Таблица 9-149 Инструкция CTRL_PWM (Широтно-импульсная модуляция)

LAD / FBD	SCL	Описание
	<pre>"CTRL_PWM_DB" (PWM:=W#16#0, enable:=FALSE, busy=>_bool_out_, status=>_word_out_);</pre>	Обеспечивает переменную скважность выходного сигнала при фиксированном времени цикла. После запуска с заданной частотой выход PWM работает непрерывно (время цикла). Длительность импульса меняется в зависимости от условий, чтобы обеспечить заданный режим управления

- 1 Когда Вы вставляете инструкцию, STEP 7 открывает диалоговое окно "Call Options" (Опции вызова) для создания соответствующих DB.
- 2 В SCL-примере, "CTRL_PWM_DB" - имя экземплярного DB.

Таблица 9-150 Типы данных для параметров

Параметр и тип		Тип данных	Описание
PWM	IN	HW_PWM (Word)	Идентификатор PWM: Имена включенных импульсных генераторов, которые станут тегами в таблице "постоянных" тегов и будут доступны для использования в качестве PWM-параметра. (Значение по умолчанию: 0)
ENABLE	IN	Bool	1 = запуск импульсного генератора 0 = останов импульсного генератора
BUSY	OUT	Bool	Функция в состоянии занятости (Значение по умолчанию: 0)
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)

Инструкция CTRL_PWM сохраняет информацию о параметре в DB. Параметры блока данных не изменяются пользователем по отдельности, а управляются инструкцией CTRL_PWM.

Укажите включенный генератор импульсов при использовании имени его тега для параметра PWM.

Когда вход EN в состоянии TRUE, то инструкция PWM_CTRL запускает или останавливает заданный режим PWM на основе значения на входе ENABLE. Длительность импульса определяется значением, связанным с выходным адресом ячейки Q-слова.

Т.к. CPU обрабатывает запросы при выполнении инструкции CTRL_PWM, то параметр BUSY всегда является FALSE. Если обнаружена ошибка, то ENO устанавливается в FALSE, а параметр STATUS содержит код ошибки.

Длительность импульса должна быть установлена в исходное значение, сконфигурированное в конфигурации устройства, при первом входе CPU в режим RUN. Вы записываете значение в ячейку Q-слова, заданную в конфигурации устройства ("Output addresses" / "Start address:"), как изменяемую длительность импульса. Вы можете использовать инструкции, например, "move" (перемещение), "convert" (преобразование), математические инструкции, или окно параметров PID-регулятора для записи различных длительностей импульса в соответствующую ячейку Q-слова. Вам необходимо использовать действительный диапазон значений для Q-слова (проценты, тысячные части, десятитысячные части или аналоговый S7-формат).

Примечание

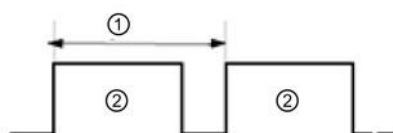
Точки дискретного ввода/вывода, назначенные PWM и PTO, не могут быть изменены с помощью инструкции "force"

Точки дискретного ввода/вывода, используемые для широтно-импульсной модуляции (PWM) и устройств вывода последовательности импульсов (PTO) заданы при конфигурировании устройства. Если этим устройствам назначены адреса дискретного ввода/вывода, то значения, назначенные адресам ввода/вывода, не могут быть изменены для таблицы наблюдений (Watch table) с помощью функции "force".

Таблица 9-151 Значение параметра STATUS

STATUS	Описание
0	Ошибки отсутствуют
80A1	Идентификатор PWM не адресован действительной PWM.

9.6.2. Работа импульсных выходов



- ① Время цикла
- ② Длительность импульса

Длительность импульса может быть выражена в виде сотых долей времени цикла (от 0 до 100), в виде тысячных долей (от 0 до 1000), в виде десятитысячных долей (от 0 до 10000) или в виде аналогового S7 формата.

Длительность импульса может изменяться от 0 (нет импульса, всегда выключен) до полного масштаба (нет импульса, всегда включен).

Т.к. выход PWM может изменяться от 0 до полного масштаба, то она (PWM) формирует дискретный выход, во многом совпадающий с аналоговым выходом. Например, выход PWM может быть использован для управления скоростью двигателя от состояния останова до максимальной скорости, или для управления положением клапана от закрытого состояния до полностью открытого.

Для управления функциями высокоскоростных импульсных выходов доступны четыре импульсных генератора: PWM и PTO. PTO используется для инструкций управления перемещением. Импульсный генератор Вы можете назначить PWM или PTO, но не обоим одновременно.

Вы можете использовать выходы, встроенные в CPU, или выходы дополнительных сигнальных плат. В следующей таблице отображены номера точек вывода (при настройках по умолчанию конфигурации вывода). Если Вами изменена нумерация точек вывода, то номерами точек вывода будут те, которые Вы назначили. Помните, что для PWM необходим только один выход, в то время, как PTO может использовать два выхода на канал. Если для импульсной функции выход не требуется, то он может быть использован для других целей. Для назначения ввода/вывода обратитесь к нижней таблице.

9.6 Импульсы

В таблице показано назначение ввода/вывода по умолчанию; однако, четыре импульсных генератора могут быть сконфигурированы для любого встроенного CPU или дискретного выхода SB. Различные точки вывода поддерживают различные напряжения и скорости, что необходимо учитывать при назначении локализации PWM/PTO.

Примечание

Выход последовательности импульсов не может быть использован другими инструкциями пользовательской программы.

При конфигурировании выходов CPU или сигнальной платы в качестве импульсных генераторов (для использования с PWM или с PTO-инструкциями управления перемещением), соответствующие адреса выходов будут удалены из Q-области памяти и не смогут быть использованы в пользовательской программе для других целей. Если Ваша пользовательская программа записывает значение на выход, используемый в качестве генератора импульсов, то CPU не записывает это значение на физический выход.

Примечание

PTO-выходы управления перемещением могут быть освобождены для использования в другом месте пользовательской программы.

Каждой PTO-инструкции необходимо назначение двух выходов: один – в качестве импульсного выхода, а другой – в качестве выхода направления. Вы можете использовать только импульсный выход, а не выход направления. Вы можете освободить выход направления для использования в другом месте Вашей пользовательской программы.

Таблица 9-152 Назначение выходов для импульсных генераторов (по умолчанию)³

Описание	Импульсный выход	Выход направления
PTO1		
Встроенный ввод/вывод	Q0.0	Q0.1
SB I ввод/вывод	Q4.0	Q4.1
PWM1		
Встроенные выходы	Q0.0	-
SB выходы	Q4.0	-
PTO2		
Встроенный ввод/вывод	Q0.2	Q0.3
SB ввод/вывод	Q4.2	Q4.3
PWM2		
Встроенные выходы	Q0.2	-
SB выходы	Q4.2	-
PTO3		
Встроенный ввод/вывод	Q0.4 ¹	Q0.5 ¹
SB ввод/вывод	Q4.0	Q4.1
PWM3		
Встроенные выходы	Q0.4 ¹	-
SB выходы	Q4.1	-
PTO4		
Встроенный ввод/вывод	Q0.6 ²	Q0.7 ²
SB ввод/вывод	Q4.2	Q4.3

Описание	Импульсный выход	Выход направления
PWM4		
Встроенные выходы	Q0.6 ²	-
SB выходы	Q4.3	-

- ¹ CPU 1211C не содержит выходов Q0.4, Q0.5, Q0.6 или Q0.7. Следовательно, эти выходы не могут быть использованы в CPU 1211C.
- ² CPU 1212C не содержит выходов Q0.6 или Q0.7. Следовательно, эти выходы не могут быть использованы в CPU 1212C.
- ³ В таблице представлены PTO/PWM-функции для CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, и CPU 1217C.

9.6.3. Конфигурирование импульсного канала для PWM

Для подготовки PWM к работе, сначала сконфигурируйте импульсный канал в конфигурации устройств для выбранного CPU, затем генератор импульсов (Pulse Generator) (PTO/PWM), и сделайте выбор от PWM1 до PWM4. Включите генератор импульсов (установите флажок). Если генератор импульсов включен, то ему по умолчанию назначается уникальное имя. Это имя Вы можете изменить, редактируя его в поле ввода "Name:", но в любом случае оно должно оставаться уникальным. Именам включенным генераторам импульсов в таблице тегов "constant" должны быть установлены в соответствие теги, которые должны быть доступны для использования в качестве PWM-параметров инструкции CTRL_PWM.

Таблица 9-153 Выход CPU: максимальная частота

CPU	Выходной канал CPU	Импульсный выход и выход направления	A/B, квадратура (сдвиг по фазе на 90°), вверх/вниз и импульс/направление
1211C	с Qa.0 по Qa.3	100 кГц	100 кГц
1212C	с Qa.0 по Qa.3	100 кГц	100 кГц
	Qa.4, Qa.5	20 кГц	20 кГц
1214C and 1215C	с Qa.0 по Qa.4	100 кГц	100 кГц
	с Qa.5 по Qb.1	20 кГц	20 кГц
1217C	с DQa.0 по DQa.3 (с .0+, .0- по .3+, .3-)	1 кГц	1 кГц
	с DQa.4 по DQb.1	100 кГц	100 кГц

Таблица 9-154 Выход сигнальной платы SB: максимальная частота (дополнительная плата)

Сигнальная плата SB	Выходной канал SB	Импульсный выход и выход направления	A/B, квадратура (сдвиг по фазе на 90°), вверх/вниз и импульс/направление
SB 1222, 200 кГц	с DQe.0 по DQe.3	200 кГц	200 кГц
SB 1223, 200 кГц	DQe.0, DQe.1	200 кГц	200 кГц
SB 1223	DQe.0, DQe.1	20 кГц	20 кГц

Примечание

Максимальная частота на выходе генератора импульсов составляет 1 МГц для CPU 1217C и 100 кГц для CPU 1211C, 1212C, 1214C и 1215C; 20 кГц (для стандартной SB); или 200 кГц (для высокоскоростной SB). Однако, STEP 7 не предупредит Вас, если Вы сконфигурируете ось со значениями максимальной скорости или частоты, превышающие аппаратные ограничения. Это может вызвать проблемы при выполнении Вашего приложения, поэтому убедитесь, что Вы не превысили аппаратные ограничения частоты импульсов.

Вы можете переименовать генератор импульсов, добавить комментарий и назначить параметры следующим образом:

- Выбор генератора импульсов: PWM или PTO (выберите PWM)
- Выбор выхода: встроенный в CPU или в сигнальную плату SB
- Масштаб времени: миллисекунды или микросекунды
- Формат длительности импульса:
 - Сотни (от 0 до 100)
 - Тысячи (от 0 до 1000)
 - Десятки тысяч (от 0 до 10000)
 - Аналоговый формат S7 (от 0 до 27648)
- Время цикла (диапазон от 0 до 16,777,215): Введите Ваше значение времени цикла. Это значение можно изменить только в конфигурации устройств (Device configuration).
- Начальная длительность импульса: Введите Ваше значение начальной длительности импульса. Это значение можно изменить во время рабочего цикла.

Введите начальный адрес для конфигурирования выходных адресов. Введите адрес Q-word address для локализации значения длительности импульса.

Примечание**Выход последовательности импульсов не может быть использован другими инструкциями пользовательской программы**

Когда Вы конфигурируете выхода CPU или сигнальной платы в качестве генератора импульсов (для использования с PWM или с инструкциями управления перемещением), то адреса соответствующих выходов будут удалены из Q-области памяти и не могут быть использованы для выполнения других задач в Вашей пользовательской программе. Если Ваша пользовательская программа записывает значение на выход, используемый в качестве генератора импульсов, то CPU не записывает это значение на физический выход.

По умолчанию используются следующие локализации длительности импульсов:

- • PWM1: QW1000
- • PWM2: QW1002
- • PWM3: QW1004
- • PWM4: QW1006

Значение каждой локализации управляет длительностью импульса и обозначается как "начальное значение длительности импульса" (Initial pulse width), задаваемое при каждом переходе CPU из режима STOP в режим RUN. Для изменения длительности импульса Вы можете изменить это значение Q-word во время рабочего цикла.

9.7 Рецептуры и записи данных

9.7.1. Рецептуры

9.7.1.1. Обзор рецептов

Сохранение рецептов данных

- Блок данных рецептов, который Вы создаете в своем проекте, должен быть сохранен в **загрузочной** области памяти CPU. Может быть использована встроенная в CPU память или внешняя карта памяти "Program".
- Другой блок данных, который Вам необходимо создать, - это блок данных активных рецептов. Этот блок данных должен быть в **оперативной** памяти, в которой с помощью логической программы считывается или записывается один активный рецепт.

Управление рецептурами данных

Рецептурный DB содержит массив записей рецептов продукта. Каждый элемент массива представляет собой отдельный рецепт, основанный на использовании общего набора компонентов.

- Вы можете создавать типы PLC-данных или структуры, которые определяют все компоненты в одной рецептурной записи. Шаблоны с этими типами данных могут быть использованы для всех рецептурных записей. Рецептуры продукта отличаются начальными значениями, заданными компонентам рецептов.
- С помощью инструкции READ_DBL одна из рецептов может быть передана в любой момент времени из рецептурного DB (все рецепты в загрузочной области памяти) в DB активных рецептов (одна рецептура в оперативной памяти). После записи рецептура будет передана в оперативную память, где пользовательской программой будут считаны значения компонентов и начнется процесс обработки. Эта передача использует минимум оперативной памяти CPU, необходимой для обработки рецептурных данных.
- Если значения компонентов активной рецептуры изменены HMI-устройством в процессе производства, то Вам необходимо записать измененные значения обратно в рецептурный DB, используя инструкцию WRIT_DBL.

Экспорт рецептов (из рецептурного DB в CSV-файл)

С помощью инструкции RecipeExport все параметры записей рецептов могут быть сгенерированы в виде CSV-файла. Неиспользуемые записи рецептов также будут экспортированы.

9.7 Рецептуры и записи данных

Импорт рецептов (из CSV-файла в рецептурный DB)

Когда операция по экспорту одиночного рецепта будет завершена, Вы можете использовать сгенерированный CSV-файл в качестве шаблона структуры данных.

1. Используйте web-сервер CPU для загрузки существующего рецептурного CSV-файла из ЦПУ в ПК.
2. С помощью текстового ASCII-редактора отредактируйте рецептурный CSV-файл. Вы можете изменить стартовые значения, назначенные компонентам, но не типы данных или структуру данных
3. Загрузите измененный CSV-файл из PC обратно в CPU. Прежний CSV-файл, расположенный в оперативной памяти CPU и имеющий то же имя должен быть удален или переименован перед выполнением операции обновления Web-сервером CPU.
4. После того, как измененный CSV-файл будет загружен в CPU, Вы можете воспользоваться инструкцией RecipeImport для передачи новых начальных значений из измененного CSV-файла (из оперативной памяти CPU) в рецептурный DB (в загрузочную область памяти CPU).

9.7.1.2. Примеры рецептов

Примеры рецептов

В приведенной ниже таблице показано, как подготовить рецептурные данные для использования в рецептурном DB. Пример рецептурного DB содержит пять записей, три из которых используются. Четвертая и пятая записи доступны для использования другими приложениями. Каждая строка таблицы представляет собой одну запись, в которой сохранены имя рецепта, типы данных компонентов и значения компонентов.

productname (имя продукта)	water (вода)	barley (яч- мень)	wheat (пшеница)	hops (хмель)	yeast (дрожжи)	waterTmp	mashTmp	mashTime	QTest
Pils (пильзен- ское пиво)	10	9	3	280	39	40	30	100	0
Lager (светлое пиво)	10	9	3	150	33	50	30	120	0
BlackBeer (темное пиво)	10	9	3	410	47	60	30	90	1
Not used	0	0	0	0	0	0	0	0	0
Not used	0	0	0	0	0	0	0	0	0

Создание рецептурного блока данных

Примечание

Правила для создания рецептурного блока данных

- Рецептурный DB должен содержать единую размерность массива типов PLC-данных или структур. В приведенном примере рецептов показано, как создать рецептурный DB с типом PLC-данных.
- В приведенном примере типом данных для всех составляющих частей компонента является тип данных UINT. Тип данных компонента может представлять собой сочетание различных типов данных, за исключением структур. В элементе массива рецептурного DB не допускается использование структур в типах PLC-данных или структур в структурах.

Первое: создание нового типа PLC-данных

Добавьте новый тип PLC-данных, имя которого – тип рецептуры. На следующем рисунке, "Beer_Recipe" – это новый тип комплексных PLC-данных, в котором сохранена последовательность стандартного типа данных. Тип PLC-данных "Beer_Recipe" – это шаблон данных, используемый при каждой записи в рецептурный DB, а также в активном рецептурном DB. Введите имена компонентов и типов данных, которые будут общими для всех примеров рецептов. Значения отдельных компонентов в дальнейшем будут добавлены в рецептурный DB.

Beer_Recipe			
	Name	Data type	Default value
1	productname	String[20]	'Beer_Recipe'
2	water	UInt	0
3	barley	UInt	0
4	wheat	UInt	0
5	hops	UInt	0
6	yeast	UInt	0
7	waterTmp	UInt	0
8	mashTmp	UInt	0
9	mashTime	UInt	0
10	QTest	UInt	0

Второе: создание рецептурного блока данных

- Создайте рецептурный DB в виде глобального блока данных с активированной опцией DB "Only store in load memory" (Сохранять только в загрузочной области памяти).
- Имя используемого рецептурного блока данных – это имя соответствующего CSV-файла. Символы, задаваемые Вами в качестве имени DB, должны учитывать ограничения, используемые при задании имен в файловой системе Windows. Не допускается использование символов \ / : * ? " < > | и "пробел".
- Задайте массив рецептов "Products" с элементами массива от 1 до 5 и типом данных "Beer_Recipe". Максимальное количество используемых элементов массива: 5.
- Значения компонентов рецептов вставьте в DB в качестве начальных значений.

На следующем рисунке приведена расширенная рецептура "BlackBeer", чтобы показать все компоненты записей рецептов.

Recipe_DB				
	Name	Data type	Offset	Start value
1	Static			
2	Products	Array [1 .. 5] of "Beer_Recipe"	...	
3	Products[1]	"Beer_Recipe"	...	
4	Products[2]	"Beer_Recipe"	...	
5	Products[3]	"Beer_Recipe"	...	
6	productname	String[20]	...	'BlackBeer'
7	water	UInt	...	10
8	barley	UInt	...	9
9	wheat	UInt	...	3
10	hops	UInt	...	410
11	yeast	UInt	...	47
12	waterTmp	UInt	...	60
13	mashTmp	UInt	...	30
14	mashTime	UInt	...	90
15	QTest	UInt	...	1
16	Products[4]	"Beer_Recipe"	...	
17	Products[5]	"Beer_Recipe"	...	

Экспорт рецептуры (из рецептурного DB в CSV-файл)

Инструкция "RecipeExport (страница 415)" выполняет передачу данных рецептурного DB в CSV-файл, как показано в следующем текстовом файле.

Recipe_DB.csv

```
index,productname,water,barley,wheat,hops,yeast,waterTmp,
mashTmp,mashTime,QTest 1,"Pils",10,9,3,280,39,40,30,100,0
2,"Lager",10,9,3,150,33,50,30,120,0
3,"BlackBeer",10,9,3,410,47,60,30,90,1
4 "Not_used",0,0,0,0,0,0,0,0,0
5 "Not_used",0,0,0,0,0,0,0,0,0
```

Импорт рецептуры (из CSV-файла в рецептурный DB)

1. Используйте web-сервер CPU для загрузки существующего рецептурного CSV-файла из загрузочной области памяти CPU в PC.
2. С помощью текстового редактора ASCII измените рецептуру CSV. Вы можете изменить начальные значения, назначенные компонентам, но не типы данных или структуру данных
3. Загрузите измененный CSV-файл из PC обратно в CPU. Кроме того, перед выполнением операции загрузки через Web-сервер CPU старый CSV-файл должен быть удален из загрузочной области памяти CPU (под тем же именем) или переименован.
4. После того, как измененный CSV-файл будет загружен в CPU, Вы можете использовать инструкцию RecipeImport для передачи новых начальных значений из измененного CSV-файла (в загрузочной области памяти CPU) в рецептурный DB (в загрузочной области памяти CPU).

CSV-файлы должны в точности соответствовать структуре рецептурного DB

- Значения в CSV-файле могут быть изменены, но изменения структуры недопустимы. Для инструкции RecipeImport необходимо точное соответствие количества записей и компонентов назначенной структуре рецептурного DB. В противном случае, инструкция RecipeImport выполняется с ошибкой. Например, если в рецептурном DB определены 10 рецептов, но только 6 из них используются, то строки с 7 по 10 CSV-файла также будут переданы в DB. Вам необходимо учитывать, действительны ли эти данные или нет. Например, Вы можете назначить переменную "Not_used" для имени продукта в неиспользуемых записях рецептов.
- Если Вы вставляете записи данных в текстовый файл и импортируете измененный файл, что предельного значение массива рецептурного DB достаточно для назначения элементов всем рецептурным записям.
- При экспорте CSV-файла автоматически генерируется индексный номер. Если Вы создаете дополнительные записи данных, то вставляйте соответствующие индексные номера.
- Инструкция RecipeImport выполняет проверку данных CSV-файла на соответствие структуре, и соответствуют ли значения типов данных заданным значениям в соответствующем рецептурном DB. Например, тип данных "Bool" не может быть сохранен в виде целого значения и инструкция RecipeImport выполняется с ошибкой.

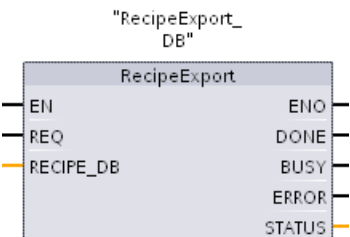
Отображение рецептурных данных CSV-файла в Excel

Для удобства чтения и редактирования CSV-файл может быть открыт в Excel. Если запятые не распознаются как десятичные разделители, то используйте функцию импорта Excel для вывода данных в структурированном виде.

	A	B	C	D	E	F	G	H	I	J	K
1	index	product	water	barley	wheat	hops	yeast	waterTmp	mashTmp	mashTime	QTest
2	1	"Pils"	10	9	3	280	39	40	30	100	0
3	2	"Lager"	10	9	3	150	33	50	30	120	0
4	3	"BlackBeer"	10	9	3	410	47	60	30	90	1
5	4	"Not_used"	0	0	0	0	0	0	0	0	0
6	5	"Not_used"	0	0	0	0	0	0	0	0	0

9.7.1.3. Инструкция RecipeExport (Экспорт рецептуры) для передачи рецептурных данных

Таблица 9-155 Инструкция RecipeExport

LAD/FBD	SCL	Описание
	<pre>"RecipeExport_DB"(req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Recipe_DB:=_variant_inout_);</pre>	Инструкция "RecipeExport" выполняет экспорт всех рецептурных записей из рецептурного блока данных в формат CSV-файла. CSV-файл содержит имена продуктов, имена компонентов и начальные значения. CSV-файл сохраняется в загрузочной области встроенной памяти или в загрузочной области внешнего носителя, если установлена дополнительная внешняя карта памяти. Выполнение экспорта запускается параметром "REQ". При обработке экспорта параметр "BUSY" устанавливается в "1". После завершения выполнения инструкции RecipeExport, параметр "BUSY" сбрасывается в "0", а завершение операции по экспорту отображается "1" параметра "DONE". Если в процессе выполнения инструкции произошла ошибка, то результат отображается параметрами ERROR и STATUS.

По возможности, рецептурный DB должен быть создан перед экспортом рецептов. Имя рецептурного блока данных будет использовано как имя нового CSV-файла. Если CSV-файл с таким именем уже существует, то он будет перезаписан во время операции экспорта.

9.7 Рецептуры и записи данных

Вы можете использовать страницу File Browser (страница 810) встроенного в CPU Web-сервера для доступа к рецептурному CSV-файлу. Файл помещается в папку рецептов в корневом каталоге загрузочной области памяти CPU.

Таблица 9-156 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQUEST: активирует процесс экспорта по положительному фронту.
RECIPE_DB	In/Out	Variant	Указатель рецептурного блока данных. За дополнительной информацией обратитесь к подразделу "Пример рецептурного DB" (страница 412)". При вводе символов в качестве имени DB необходимо учитывать ограничения операционной системы Windows при задании имен файлов. Символы \ / : * ? " < > и "пробел" недопустимы для использования.
DONE	OUT	Bool	После последнего запроса, выполненного без ошибок, бит DONE устанавливается в TRUE для одного цикла сканирования (Значение по умолчанию: False).
BUSY	OUT	Bool	Выполнение инструкции RecipeExport 0: Инструкция не выполняется 1: Инструкция выполняется
ERROR	OUT	Bool	После последнего цикла сканирования, заверщенного ошибкой, бит ERROR устанавливается в TRUE для одного цикла сканирования. Значение кода ошибки в параметре STATUS действительно только для одного цикла сканирования, когда ERROR = TRUE. 0: Нет ошибок или предупреждений 1: Произошла ошибка. Параметр STATUS содержит информацию о типе ошибки.
STATUS	OUT	Word	Код состояния выполнения

Таблица 9-157 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Отсутствие вызова по фронту REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту REQ (работа): BUSY = 1, DONE = 0
0	7002	N-й вызов (работа): BUSY = 1, DONE = 0
1	8070	Вся память экземпляра использована.
1	8090	Имя файла содержит недопустимые символы.
1	8091	Структура рецептурного блока данных RECIPE_DB не может быть обработана.
1	8092	Структура данных, заданная в RECIPE_DB, превышает 5000 байт
1	80B3	Недостаточный объем памяти на внешней карте MC или встроенной памяти.
1	80B4	Карта памяти MC защищена от записи.
1	80B6	Флажок рецептурного DB "Only store in load memory" (Сохранять только в загрузочной области памяти) не установлен.
1	80C0	CSV-файл временно недоступен.
1	80C1	DB временно недоступен.

Инструкция RecipeImport (Импорт рецептов)

Таблица 9-158 Инструкция RecipeImport

LAD/FBD	SCL	Описание
"RecipeImport_DB" 	<pre>"RecipeImport_DB"(req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Recipe_DB:=_variant_inout_);</pre>	Инструкция "RecipeImport" импортирует рецептурные данные из CSV-файла в загрузочной области памяти CPU в рецептурный блок данных заданный для параметра RECIPE_DB. При выполнении операции импорта начальные значения рецептурного блока данных будут перезаписаны. Операция импорта запускается параметром "REQ". Параметр BUSY во время импорта устанавливается в "1". После завершения выполнения инструкции RecipeImport, параметр BUSY сбрасывается в "0", а завершение операции отображается состоянием "1" параметра DONE. Если при выполнении инструкции произошла ошибка, то результат записывается в параметры ERROR и STATUS.

Таблица 9-159 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQUEST: Активирует процесс импорта по переднему фронту.
RECIPE_DB	In/Out	Variant	Указатель блока рецептурных данных. За дополнительной информацией обратитесь к подразделу "Пример рецептурного DB (стр. 412)". При вводе символов в качестве имени DB необходимо учитывать ограничения операционной системы Windows при задании имен файлов. Использование символов \ / : * ? " < > и "пробел" недопустимо.
DONE	OUT	Bool	После последнего запроса, выполненного без ошибок, бит DONE устанавливается в TRUE для одного цикла сканирования (Значение по умолчанию: False).
BUSY	OUT	Bool	0 – Инструкция выполняется 1 – Инструкция не выполняется
ERROR	OUT	Bool	После последнего цикла сканирования, заверщенного ошибкой, бит ERROR устанавливается в TRUE для одного цикла сканирования. Значение кода ошибка в параметре STATUS действительно только для одного цикла сканирования, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)

9.7 Рецептуры и записи данных

Перед выполнением операции импорта необходимо создать рецептурный блок данных, содержащий структуру, консистентную структуре данных CSV-файла.

Требования к CSV-файлу:

- CSV-файл должен быть расположен в корневой директории папки "Recipes" (Рецептуры) загрузочной области встроенной памяти или загрузочной области внешней карты памяти, если дополнительная внешняя карта памяти установлена.
- Имя CSV-файла должно совпадать с именем блока данных в параметре RECIPE_DB.
- Первая строка (заголовок) CSV-файла содержит имена компонентов рецептов. Во время импорта первая строка будет проигнорирована. Имена компонентов рецептов в CSV-файле и в блоке данных во время импорта не учитываются.
- В любом случае, первое значение каждой строки CSV-файла – это индексный номер рецептуры. Отдельные рецепты будут импортированы в соответствии с их индексными номерами. Для этого, в CSV-файле индексы должны быть расположены в порядке возрастания и не должны содержать пробелов (если это не выполняется, то параметр STATUS будет содержать код ошибки 80B0).
- CSV-файл не может содержать больше записей данных рецептов, чем предусмотрено для рецептурного блока данных. Максимальное количество записей данных ограничено предельными значениями массива в блоке данных.

Таблица 9-160 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Отсутствие вызова по фронту REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту REQ (работа): BUSY = 1, DONE = 0
0	7002	N-й вызов (работа): BUSY = 1, DONE = 0
1	8070	Вся память экземпляра использована.
1	8090	Имя файла содержит недопустимые символы.
1	8092	Не найден соответствующий CSV-файл для импорта. Возможная причина: Имя CSV-файла не соответствует имени рецептурного DB.
1	80C0	CSV-файл временно недоступен.
1	80C1	Блок данных временно недоступен.
1	80B0	Нумерация индексов в CSV-файле не является непрерывной, индексы расположены не по возрастанию или превышено их максимальное количество (предельное значение массива) в блоке данных.
1	80B1	Структуры рецептурного блока данных и CSV-файла не соответствуют: CSV-файл содержит слишком много полей.
1	80B2	Структуры рецептурного блока данных и CSV-файла не соответствуют: CSV-файл содержит слишком мало полей.
1	80B6	Флажок рецептурного DB "Only store in load memory" (Сохранять только в загрузочной области памяти) не установлен..
1	80D0 +n	Структуры рецептурного блока данных и CSV-файла не соответствуют: Несовместимость типа данных в поле "n" (n<=46).
1	80FF	Структуры рецептурного блока данных и CSV-файла не соответствуют: Несовместимость типа данных в поле "n" (n>46).

9.7.1.4. Пример программы для работы с рецептурами

Необходимые условия для примера программы работы с рецептурами

- Рецептурный DB должен содержать все записи рецептов. Рецептурный DB хранится в загрузочной области памяти.
- Активный рецептурный DB хранит копию одной рецептуры в оперативной памяти.

За дополнительной информацией о рецептурном DB и соответствующем CSV-файле обратитесь к подразделу "Пример рецептурного DB" (стр. 412)".

Создание активного рецептурного блока данных DB

В окне "Add new block" (Вставить новый блок):

- Щелкните мышкой на кнопке "Data block" (Блок данных) в окне "Add new block" (Вставить новый блок)
- В выпадающем меню "Type" (Тип) выберите тип PLC-данных "Beer_recipe", предварительно созданный Вами.

Начальные значения не требуются. Значения данных DB устанавливаются при передаче одной рецептуры из рецептурного DB в активный рецептурный DB. В приведенном примере, активный рецептурный DB – это получатель данных READ_DBL и содержащий исходные данные для WRITE_DBL. На следующем рисунке приведен пример Active_Recipe DB.

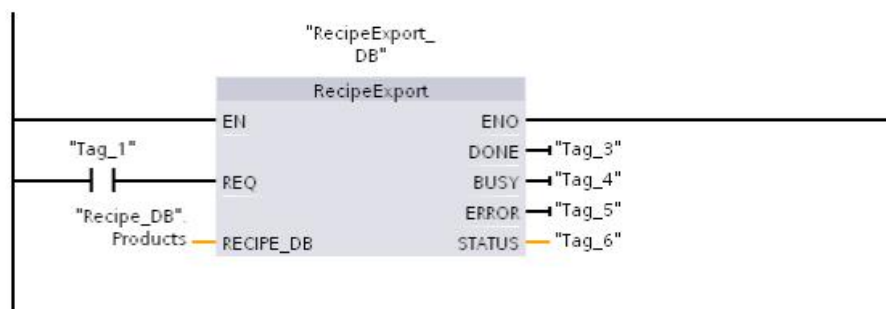
Active_Recipe			
	Name	Data type	Start value
1	Static		
2	productname	String[20]	'Beer_Recipe'
3	water	UInt	0
4	barley	UInt	0
5	wheat	UInt	0
6	hops	UInt	0
7	yeast	UInt	0
8	waterTmp	UInt	0
9	mashTmp	UInt	0
10	mashTime	UInt	0
11	QTest	UInt	0

Экземплярные блоки данных

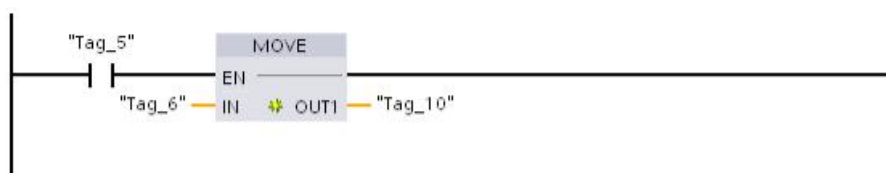
Экземплярные блоки данных используются инструкциями RecipeExport ("RecipeExport_DB") и RecipeImport ("RecipeImport_DB") и создаются автоматически, когда Вы вставляете инструкцию в Вашу программу. Экземплярные блоки данных используются для управления выполнением инструкций и не относятся к программной логике.

Пример программы для работы с рецептурами

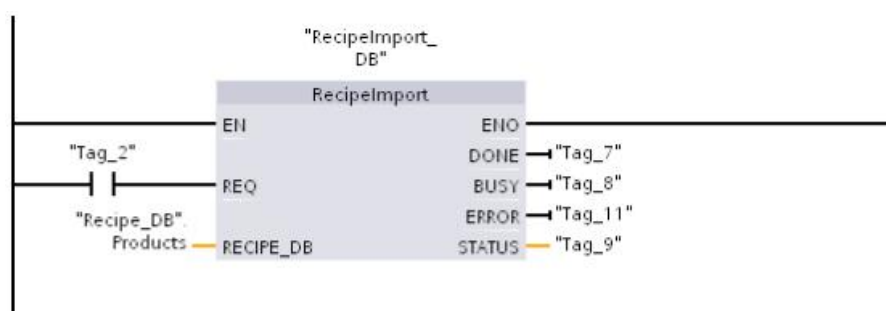
Сегмент 1. Нарастающий фронт на входе REQ запускает процедуру экспорта. CSV-файл сгенерирован для рецептурного DB и расположен в папке рецептов в памяти CPU.



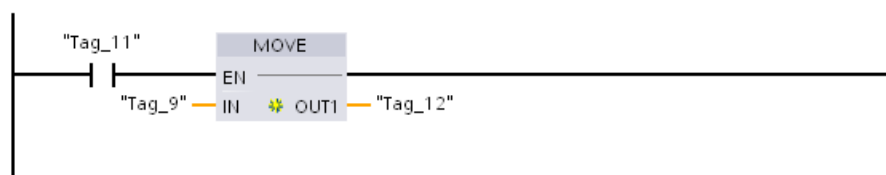
Сегмент 2. Фиксация состояния выхода STATUS после выполнения инструкции RecipeExport, т.к. оно действительно только для одного цикла сканирования.



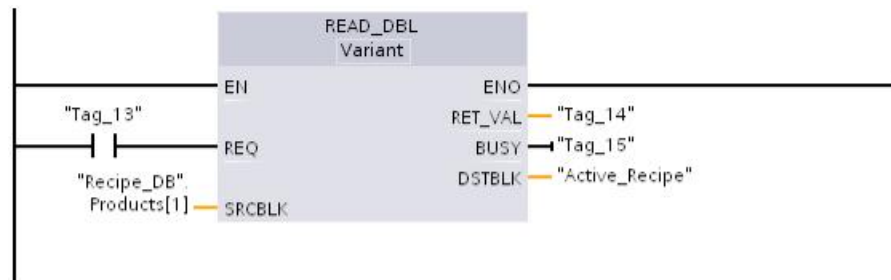
Сегмент 3. Нарастающий фронт на входе REQ запускает процедуру импорта. Существующий рецептурный DB загружен со всеми рецептурными данными, считанными из соответствующего CSV-файла, который расположен в папке рецептов в памяти CPU.



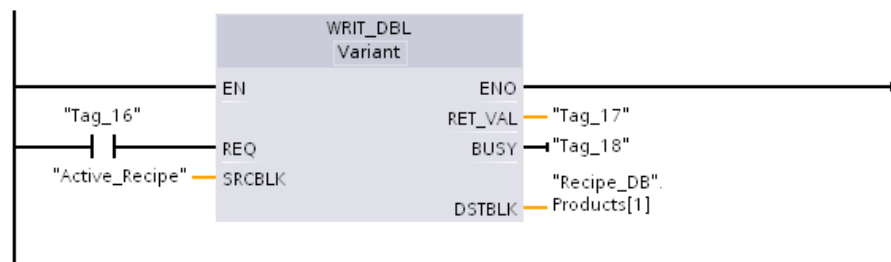
Сегмент 4. Фиксация состояния выхода STATUS после выполнения инструкции RecipeImport, т.к. оно действительно только для одного цикла сканирования.



Сегмент 5. Инструкция READ_DBL копирует начальные значения из рецептурного блока данных "Recipe_DB". Products[1] (в загрузочной области памяти CPU) в фактические значения активного рецептурного блока данных Active_Recipe DB (в оперативной памяти CPU). После выполнения инструкции READ_DBL, Ваша программа может получить доступ к значениям компонентов рецептов за счет адресации ячеек в Active_Recipe DB. Например, символьная адресация ("Active_Recipe".productname) и ("Active_Recipe.water) содержит Вашу логическую программу с именем текущей рецептуры и количеством воды.



Сегмент 6. Во время рабочего цикла, с помощью HMI-устройства можно изменить значения компонентов, сохраненные в Active_Recipe DB. Измененные рецептурные данные можно сохранить, выполнив инструкцию WRIT_DBL. В приведенном примере, все начальные значения Recipe_DB для рецептуры "Recipe_DB". Products[1] перезаписываются фактическими значениями из "Active_Recipe" DB.



9.7.2. Записи данных (Data log)

В своей программе Вы можете использовать инструкции записи данных для сохранения данных рабочего цикла в непрерывных log-файлах. Данные log-файлов хранятся во флэш-памяти (CPU или карте памяти). Данные log-файлов сохранены в стандартном CSV-формате (Comma Separated Value = Значения, разделенные запятыми). Записи данных организованы в виде циклически обновляемого log-файла предварительно заданного размера.

Инструкции записи данных могут быть использованы в Вашей программе для создания, открытия, записи и закрытия log-файлов. Вы выбираете, какие значения программы записываются с использованием буфера данных, а какие с использованием единственной записи в журнал. Буфер данных используется как временное хранилище для новых записей. Новые фактические значения должны программно перемещаться в буфер во время рабочего цикла. Когда все фактические значения обновлены, необходимо выполнить инструкцию DataLogWrite для передачи данных из буфера в журнал записи данных.

В окне проводника ("File Browser") Web-сервера Вы можете открывать, редактировать, сохранять, переименовывать и удалять файлы записей данных. У Вас должны быть соответствующие права для просмотра файлов в проводнике и для редактирования, удаления или переименования файлов записей данных.

9.7.2.1 Структура журнала записей данных

Параметры DATA и HEADER инструкции DataLogCreate задают тип данных и описания заголовков колонок для всех элементов и записей данных.

Параметр DATA для инструкции DataLogCreate

Параметр DATA указывает на область памяти, используемую в качестве временного буфера для новых записей данных, и назначенную ячейкам M или DB.

Вы можете задать весь DB (производные от типа PLC-данных, которые Вы назначаете при создании DB) или часть DB (заданный элемент DB может содержать любые типы данных, структуру типов данных, тип PLC-данных или данные массива).

Структура типов данных ограничена одним уровнем вложенности. Общее количество объявленных элементов данных, должно соответствовать количеству столбцов, указанных в заголовке параметра. Максимальное количество назначаемых Вами элементов данных – 253 (с метками времени) или 255 (без меток времени). Это ограничение позволяет Вам записывать данные в 256 столбцов Excel.

Параметр DATA может назначать сохраняемые или несохраняемые элементы данных в стандартном ("Standard") (совместимом с S7-300/400) или оптимизированном ("Optimized") типе DB.

Для того чтобы выполнить запись данных (Data log) Вы сначала должны загрузить временную запись данных с новыми процессными значениями, а затем запустить выполнение инструкции DataLogWrite, чтобы сохранить новое значение записи данных в файле Datalog.

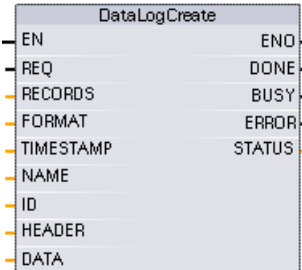
Параметр HEADER для инструкции DataLogCreate

Параметр HEADER указывает на имена столбцов заголовка для верхней строки матрицы данных, закодированной в виде CSV-файла. Данные HEADER должны быть расположены в DB или M-области памяти, а символы должны соответствовать правилам стандартного CSV-формата с запятыми, разделяющими имя каждого столбца. Данные могут быть типов "string", "byte array" или "character array". Массивы символов/байтов позволяют увеличить размер, в отличие от строк, ограниченных максимум 255 байтами. Параметр HEADER - дополнительный. Если HEADER не задан, то строка заголовка будет создана в файле записи данных (Data log).

9.7.2.2 Программные инструкции для управления записями данных

Инструкция DataLogCreate (Создание записи данных)

Таблица 9-161 Инструкция DataLogCreate

LAD/FBD	SCL	Описание
	<pre>"DataLogCreate_DB"(req:=_bool_in_, rec- ords:=_udint_in_, for- mat:=_uint_in_, timestamp:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, name:=_string_inout_, ID:=_dword_inout_, head- er:=_variant_inout_, data:=_variant_inout_);</pre>	Создание и инициализация файла записей данных (Data log). Файлы, созданные в папке PLC\DataLogs, отсортированы по имени параметра, и косвенно открыты для операции записи. Инструкция "Data log" может быть использована для программного сохранения процессных данных рабочего цикла в флэш-памяти CPU. STEP 7 автоматически создает соответствующие экземплярные блоки данных, которые Вы вставляете в инструкцию.

¹ В SCL-примере, "DataLogCreate_DB" – это имя экземплярного блока данных.

Таблица 9-162 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Запуск выполнения по сигналу высокого или низкого уровня (положительный фронт). (По умолчанию: False)
RECORDS	IN	UDint	Максимальное количество циклически обновляемых записей данных, которые могут содержаться в прежней записи перед заменой: Запись заголовка не включена. Для успешного создания записей данных необходим достаточный объем загрузочной области памяти PLC (По умолчанию - 1).
FORMAT	IN	UInt	Формат записей данных: 0 – Внутренний формат (не поддерживается) 1 – Запятая, разделяющая значения "csv-eng" (Значение по умолчанию)
TIMESTAMP	IN	UInt	Формат метки времени данных: Заголовки столбцов для данных и поля ввода параметров времени не требуются. Метки времени используют системное время (Coordinated Universal Time – UTC? универсальное синхронизированное время), а не локальное. 0 – Без меток времени 1 – Метки даты и времени (По умолчанию)

9.7 Рецептury и записи данных

Параметр и тип		Тип данных	Описание
ID	In/Out	DWord	Цифровой идентификатор записи данных: Это сгенерированное значение Вы сохраняете для использования с инструкциями записи данных. Параметр ID используется в качестве выходного только с инструкцией DataLogCreate. (По умолчанию: 0). Для этого параметра доступ к символьному имени недоступен.
HEADER	In/Out	Variant	Указатель имен столбцов записей данных для верхней строки матрицы данных, кодированной в виде CSV- файла. (По умолчанию: 0). Данные HEADER должны быть расположены в DB или M-области памяти. Символы должны соответствовать стандартному CSV-формату с разделением запятыми имени каждого столбца. Данные могут быть типа "string", "byte array" или "character array". Массивы символов/байтов позволяют увеличить размер, в отличие от строк, ограниченных максимум 255 байт. Параметр HEADER является дополнительным. Если HEADER не параметризован, то заголовок строки будет создан в файле записей данных (Data log).
DATA	In/Out	Variant	Указатель структуры записи данных пользовательского типа (UDT), или массива. Запись данных должна быть расположена в DB или M-области памяти. Параметр DATA определяет отдельные элементы данных (столбцов) журнала записей данных и их тип. Структура типов данных ограничена одной глубиной вложенности. Количество описываемых элементов данных должно соответствовать количеству столбцов, определяемых параметром "header". Максимальное количество назначаемых Вами элементов данных – 253 (с метками времени) или 255 (без меток времени). Это ограничение позволяет Вам выполнять запись в пределах 256 столбцов формата Excel.
DONE	OUT	Bool	После последнего запроса бит DONE устанавливается в TRUE для одного цикла сканирования, выполненного без ошибок. (Значение по умолчанию: False)
BUSY	OUT	Bool	0 – Не выполняется 1 – Выполняется
ERROR	OUT	Bool	После последнего запроса бит ERROR устанавливается в TRUE для одного цикла сканирования, заверщенного с ошибкой. Значение кода ошибки параметра STATUS действительно только в течение одного цикла сканирования, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения (По умолчанию: 0)

Файл записей данных создается фиксированного размера на основе параметров RECORDS и DATA. Записи данных организованы в виде циклически обновляемого файла журнала. Новые записи добавляются в файл журнала, пока не будет достигнуто их максимальное число, определяемое параметром RECORDS. Очередная запись перезаписывает самую старшую по времени создания запись.

Следующие операции записи будут перезаписывать предыдущую запись данных и т.д.

Использование ресурсов памяти:

- Для записей данных используется только загрузочная область памяти.
- Нет установленного лимита на общее количество записей данных. Суммарный размер записей данных ограничивается доступными ресурсами загрузочной области памяти. Одновременно могут быть открыты только восемь записей данных.
- Максимально доступное количество записей, определяемых параметром RECORDS, ограничивается числом для UDint (4,294,967,295). Фактическое ограничение для параметра RECORD зависит от размера одиночной записи, размера других записей данных и доступных ресурсов загрузочной области памяти. Кроме того, Excel ограничивает количество строк, допустимых для Excel-страницы.

Примечание

Создание журнала записей данных должно быть завершено до запуска операции записи данных

- Формирование журналов записей данных DataLogCreate и DataLogNewFile продолжатся в течение многих циклов сканирования программы. Фактическое время для создания файла журнала зависит от структуры и количества записей. Ваша программа должна отслеживать и фиксировать переход битов DONE в состояние TRUE, которое означает завершение создания файла журнала записей данных. Если инструкция DataLogWrite будет выполнена перед завершением создания журнала записей данных, то операция записи новых данных будет завершена с ошибкой.
- В некоторых случаях, когда программа сканирования работает очень быстро, формирование журнала записей данных может занять продолжительное время. Если формирование журнала записей данных происходит слишком медленно, то Вам необходимо убедиться, что флажок "Enable minimum cycle time for cyclic OBs" (Включение минимального времени цикла для циклически выполняемых ОБ) активен, а установленное минимальное время цикла равно 1 мс или выше. За дополнительной информацией обратитесь к подразделу "Конфигурирование контроля времени цикла и конфигурационной нагрузки" (страница 101).

Примечание

Инструкция DataLogNewFile копирует структуру существующих записей данных

Если Вам необходимо предотвратить перезапись любых записей данных, то Вам необходимо воспользоваться инструкцией DataLogNewFile для создания нового журнала, основанного на текущих записях данных, после того, как в текущем журнале будет сохранено максимальное количество записей данных. Новые записи данных будут сохранены в новом журнале. Файл старого журнала и записи данных будут сохранены в флэш-памяти.

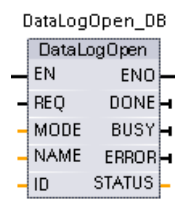
9.7 Рецептуры и записи данных

Таблица 9-163 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту, заданному в REQ (обработка): BUSY = 1, DONE = 0
0	7002	N-й вызов (обработка): BUSY = 1, DONE = 0
1	8070	Вся строенная память использована.
1	807F	Внутренняя ошибка
1	8090	Недопустимое имя файла
1	8091	Имя параметра не соответствует "String".
1	8093	Запись данных с этим именем уже существует. Используйте другое имя, убедитесь, что существующий CSV-файл журнала записей данных не открыт, а затем воспользуйтесь проводником Web-сервера (стр. 810) для удаления существующего журнала записей данных.
1	8097	Запрашиваемая длина файла превышает максимально допустимый размер.
1	80B3	Недостаточный объем загрузочной области памяти.
1	80B4	Карта памяти MC (Memory Cartridge) защищена от записи.
1	80C1	Слишком много открытых файлов: Допускается не более 8 открытых файлов записей данных.
1	8253	Недействительный подсчет записей
1	8353	Недопустимый выбранный формат
1	8453	Недопустимая выбранная метка времени
1	8B24	Недопустимая область назначения параметра HEADER: Например, указывает на локальную область памяти.
1	8B51	Недопустимый тип данных для параметра HEADER.
1	8B52	Слишком много элементов данных параметра HEADER
1	8C24	Недопустимая область назначения параметра DATA: Например, указывает на локальную область памяти.
1	8C51	Недопустимый тип данных для параметра DATA.
1	8C52	Слишком много элементов данных параметра DATA.

Инструкция DataLogOpen (Открытие журнала записей данных)

Таблица 9-164 Инструкция DataLogOpen

LAD / FBD	SCL	Описание
	<pre>"DataLogOpen_DB"(req:=_bool_in_, mode:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, name:=_string_inout_, ID:=_dword_inout_);</pre>	Открывает существующий файл журнала записей данных. Журнал записей данных должен быть открыт перед тем, как Вы добавите новую запись в журнал. Журналы записей данных могут быть открыты и закрыты по отдельности. Максимальное количество открытых файлов записей данных не должно превышать 8. При запуске инструкции STEP 7 автоматически создает соответствующий экземплярный блок данных.

1 В SCL-примере, "DataLogOpen_DB" – это имя экземплярного блока данных.

Таблица 9-165 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Запуск обработки по положительному фронту сигнала. (Значение по умолчанию: False)
MODE	IN	UInt	Режим работы: 0 – Добавить к существующим данным (Значение по умолчанию) 1 – Удалить все существующий записи
NAME	IN	Variant	Имя файла существующего журнала записей данных: Поддерживается только тип данных "String" и может быть расположен в локальной памяти, DB или M-области памяти. (Значение по умолчанию: ' ')
ID	In/Out	DWord	Цифровой идентификатор журнала записей данных. (Значение по умолчанию: 0) Примечание: Доступ к символьному имени для этого параметра недоступен.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE для одного цикла сканирования, после последнего запроса, выполненного с ошибкой. (Значение по умолчанию: False)
BUSY	OUT	Bool	0 – Не выполняется 1 – Выполняется
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE для одного цикла сканирования, после последнего запроса, выполненного с ошибкой. Значение кода ошибки в параметре STATUS действительно только для одного цикла сканирования, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения (По умолчанию: 0)

Вы можете задать или параметр NAME или параметр ID (параметр ID в качестве входа) ранее существующего журнала записей данных. Если Вы задаете оба параметра, а действующий ID не соответствует журналу записей данных NAME, то параметр ID будет использован, а параметр NAME проигнорирован.

Параметр NAME должен соответствовать журналу записей данных, созданному для инструкции DataLogCreate. Если задано только ИМЯ (NAME) и оно ссылается на действительно существующий архив данных, то выводится идентификатор этого архива в качестве выходного значения (ID).

Примечание

Основные правила использования файлов журнала записей данных

- Журнал записей данных автоматически открывается после выполнения инструкций DataLogCreate и DataLogNewFile.
- Журнал записей данных автоматически закрывается после перехода PLC в состояние STOP или при выключении PLC и включении его снова.
- Если возможно, то журнал записей данных должен быть открыт перед очередным выполнением инструкции DataLogWrite.
- Одновременно могут быть открыты не более восьми файлов журналов данных. Могут существовать более восьми файлов журналов данных, но некоторые из них должны быть закрыты (открыты не более восьми).

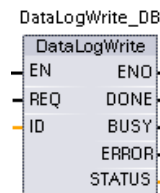
9.7 Рецептуры и записи данных

Таблица 9-166 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#)	Описание
0	0000	Ошибки отсутствуют
0	0002	Предупреждение: Файл журнала данных для этой прикладной программы уже открыт
0	7000	Вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту, заданному в REQ (обработка): BUSY = 1, DONE = 0
0	7002	N-й вызов (обработка): BUSY = 1, DONE = 0
1	8070	Вся внутренняя память использована.
1	8090	Определение журнала данных несовместимо с существующим файлом журнала записей данных.
1	8091	Имя параметра не соответствует формату "String".
1	8092	Запись данных отсутствует.
1	80C0	Файл журнала записей данных заблокирован.
1	80C1	Открыто слишком много файлов: Может быть открыто не более восьми файлов журналов данных.

Инструкция DataLogWrite (Запись журнала данных)

Таблица 9-167 Инструкция DataLogWrite

LAD / FBD	SCL	Описание
	<pre>"DataLogWrite_DB"(req:= bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	Выполняет запись данных в соответствующий журнал данных. До выполнения инструкции DataLogWrite должен быть открыт ранее созданный журнал записей данных. Перед выполнением инструкции STEP 7 автоматически создает соответствующий экземплярный DB.

1 В SCL-примере, "DataLogWrite_DB" – это имя экземплярного блока данных.

Таблица 9-168 Типы данных для параметров

Параметр и тип	Тип данных	Описание
REQ	IN	Bool Выполнение инструкции запускается по переднему фронту сигнала. (По умолчанию: False)
ID	In/Out	DWord Цифровой идентификатор журнала данных. Используется только в качестве входного параметра для инструкции DataLogWrite. (По умолчанию: 0) Примечание: Доступ к символьному имени для данного параметра недопустим.
DONE	OUT	Bool Бит DONE устанавливается в TRUE для одного цикла сканирования, если последний запрос был выполнен без ошибок.
BUSY	OUT	Bool 0 – Выполняется 1 – Не выполняется
ERROR	OUT	Bool Бит ERROR устанавливается в TRUE для одного цикла сканирования, если последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только в течение одного цикла сканирования, когда ERROR = TRUE.
STATUS	OUT	Word Код состояния выполнения (По умолчанию: 0)

Адресация памяти и структура данных буфера записи сконфигурированы для параметра DATA инструкции DataLogCreate. Вам необходимо программно загрузить буфер записи текущими процессными значениями рабочего цикла, а затем запустить выполнение инструкции DataLogWrite для перемещения новых записей данных из буфера в журнал записей данных.

Параметр ID используется для идентификации журнала данных и конфигурации записей данных. Номер ID сгенерирован при создании журнала записей данных.

Если в циклически обновляемом файле журнала есть пустые записи, то следующая запись будет выполнена на свободное место. Если все записи заполнены, то старые записи будут перезаписаны.

ЗАМЕЧАНИЕ

Процедура создания журнала должно быть завершена перед началом операции записи данных в журнал

Процедура создания файлов журналов DataLogCreate и DataLogNewFile длится в течение нескольких циклов сканирования программы. Фактическое время необходимое для создания файла журнала зависит от структуры и количества записей. Ваша программа должна контролировать и фиксировать переключение битов DONE в состояние TRUE, сигнализирующем о завершении создания файла журнала. Если инструкция DataLogWrite была выполнена перед завершением процедуры создания файла журнала, то операция записи новых данных в журнал не будет выполнена, как ожидалось.

Примечание

Влияние записей журнала данных на использования встроенной в CPU памяти

Каждая запись в журнале данных занимает минимум 2 кБ памяти. Если Ваша программа выполняет частые записи небольшого количества данных, то каждая из записей будет занимать не менее 2 кБ памяти. Лучше собирать небольшие объемы данных в блоках данных, а затем блоки данных записывать в журнал с менее частыми интервалами.

Если ваша программа выполняет множество записей в журнал данных с высокой частотой, то необходимо рассмотреть вопрос об использовании сменной карты памяти SD.

ЗАМЕЧАНИЕ

Потенциальная потеря записей в журнале данных из-за сбоя питания ЦПУ

Если во время выполнения инструкции DataLogWrite произошел сбой по питанию, то записи данных, передаваемые в журнал, могут быть потеряны.

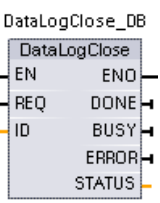
9.7 Рецептуры и записи данных

Таблица 9-169 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#)	Описание
0	0000	Ошибки отсутствуют
0	0001	Показывает, что журнал записей данных заполнен: Каждый журнал данных создается с указанием максимального количества записей. Запись с наибольшим (максимальным) номером выполняется последней. Затем выполняется перезапись наиболее старшей записи данных.
0	7000	Вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту, заданному в REQ (обработка): BUSY = 1, DONE = 0
0	7002	N-й вызов (обработка): BUSY = 1, DONE = 0
1	8070	Вся внутренняя память использована.
1	8092	Запись данных отсутствует.
1	80B0	Файл журнала записей данных не открыт.

Инструкция DataLogClose (Закрытие журнала записей данных)

Таблица 9-170 Инструкция DataLogClose

LAD / FBD	SCL	Описание
	<pre>"DataLogClose_DB"(req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_) ;</pre>	Закрывает файл открытого журнала данных. Инструкция DataLogWrite используется для закрытия результирующего журнала данных в случае ошибки. Пока выполняется инструкция DataLogOpen, не разрешены никакие операции записи в данный журнал. Переход в режим STOP закрывает все открытые файлы журналов записей данных. При выполнении инструкции STEP 7 автоматически создает соответствующий экземплярный блок данных.

1 В SCL-примере, "DataLogClose_DB" – это имя экземплярного блока данных.

Таблица 9-171 Типы данных для параметров

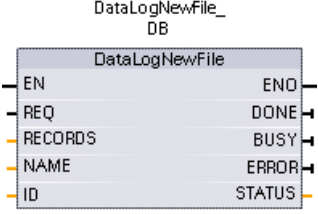
Параметр и тип	Тип данных	Описание
REQ	IN	Bool Запуск выполнения по переднему фронту. (По умолчанию: False)
ID	In/Out	DWord Цифровой идентификатор журнала данных. Используется только в качестве входного параметра для инструкции DataLogWrite. (Значение по умолчанию: 0) Примечание: Доступ к символьному имени для данного параметра недопустим.
DONE	OUT	Bool Бит DONE устанавливается в TRUE для одного цикла сканирования, если последний запрос был выполнен без ошибок.
BUSY	OUT	Bool 0 – Выполняется 1 – Не выполняется
ERROR	OUT	Bool Бит ERROR устанавливается в TRUE для одного цикла сканирования, если последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только в течение одного цикла сканирования, когда ERROR = TRUE.
STATUS	OUT	Word Код состояния выполнения (По умолчанию: 0)

Таблица 9-172 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#)	Описание
0	0000	Ошибки отсутствуют
0	0001	Журнал данных не открыт
0	7000	Вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту, заданному в REQ (обработка): BUSY = 1, DONE = 0
0	7002	N-й вызов (обработка): BUSY = 1, DONE = 0
1	8092	Журнал записей данных отсутствует.

Инструкция DataLogNewFile (Создание нового файла записей данных)

Таблица 9-173 Инструкция DataLogNewFile

LAD / FBD	SCL	Описание
	<pre>"DataLogNewFile_DB"(req:=_bool_in_, rec- ords:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_DataLog_out_, ID:=_dword_inout_);</pre>	Позволяет Вашей программе создавать новый файл журнала записей данных на базе существующего файла журнала данных. При выполнении инструкции STEP 7 автоматически создает соответствующий экземплярный блок данных.

1 В SCL-примере, "DataLogNewFile_DB" – это имя экземплярного блока данных.

9.7 Рецептуры и записи данных

Таблица 9-174 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Запуск обработки по положительному фронту сигнала. (Значение по умолчанию: False)
RECORDS	IN	UDInt	Максимальное количество записей данных, которые может содержать циклически обновляемый журнал перед перезаписью старых значений. (По умолчанию: 1) Заголовок записей данных не обрабатывается. Для создания журнала записей данных необходим достаточный объем загрузочной области памяти CPU.
NAME	IN	Variant	Наименование журнала данных: Вы должны задать имя. Поддерживается только тип данных "String", которые могут быть расположены только в локальном стеке, DB или M-области памяти. (Значение по умолчанию: ' ') В качестве имени журнала также может быть использована ссылка на строку. При вводе символов в качестве имени необходимо учитывать ограничения операционной системы Windows при задании имен файлов. Использование символов \ / : * ? " < > и "пробел" недопустимо.
ID	In/Out	DWord	Цифровой идентификатор журнала записей данных (Значение по умолчанию: 0): • При обработке: вход ID идентифицирует действительный журнал записей данных. Конфигурация нового журнала скопирована из этого журнала данных. • После обработки: ID становится выходным параметром, возвращающим ID вновь созданному журналу записей данных. Примечание: Доступ к символьному имени для этого параметра недопустим.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE для одного цикла сканирования, если последний запрос был завершен без ошибок.
BUSY	OUT	Bool	• 0 – Не выполняется • 1 – Выполняется
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE для одного цикла сканирования, если последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только для одного цикла сканирования при ERROR = TRUE.
STATUS	OUT	Word	Код выполнения условия (Значение по умолчанию: 0)

Вы можете использовать инструкцию DataLogNewFile для формирования нового журнала записей данных, если текущий журнал переполнен или считается завершенным, а Вы не хотите терять данные, сохраненные в этом журнале. Новый файл пустого журнала записей данных может быть создан на основе структуры заполненного журнала записей данных. Заголовок записи данных будет продублирован из оригинального журнала записей данных со всеми его свойствами (DATA – буфером записи данных, форматом данных и настройками меток времени). Файл оригинального журнала записей данных закроется, а файл нового журнала будет открыт.

Переключатель параметра DataLogWrite: Ваша программа должна контролировать параметры ERROR и STATUS при каждом выполнении инструкции DataLogWrite. Когда будет выполнена последняя запись, и журнал записей данных заполнится, то бит ERROR инструкции DataLogWrite устанавливается в 1, а слово параметра STATUS инструкции DataLogWrite установится в 1. Эти значения параметров ERROR и STATUS действительны только для одного цикла сканирования, т.е. Ваша программа мониторинга использует ERROR = 1 в качестве стробирующего сигнала для фиксации значения STATUS, и последующей проверки STATUS = 1 (журнал записей данных заполнен).

Инструкция DataLogNewFile: Когда Ваша программа получает сигнал о заполнении журнала записей данных, то это активирует выполнение инструкции DataLogNewFile. Вы должны запустить выполнение инструкции DataLogNewFile с уже существующим ID журнала (обычно заполнен) и открыть журнал данных, а не новый уникальный параметр NAME. После завершения выполнения инструкции DataLogNewFile, значение ID нового журнала записей данных будет возвращено (в качестве выходного параметра), что соответствует имени нового журнала записей данных. Файл нового журнала будет открыт и готов к сохранению новых записей данных. Инструкция DataLogWrite обращается к файлу нового журнала записей данных и использует значение ID возвращаемое инструкцией DataLogNewFile.

ЗАМЕЧАНИЕ
Создание журнала должно быть завершено перед началом операции записи данных в журнал Процедура создания файлов журналов DataLogCreate и DataLogNewFile длится в течение нескольких циклов сканирования программы. Фактическое время необходимое для создания файла журнала зависит от структуры и количества записей. Ваша программа должна контролировать и фиксировать переключение битов DONE в состояние TRUE, сигнализирующем о завершении создания файла журнала. Если инструкция DataLogWrite была выполнена перед завершением процедуры создания файла журнала, то операция записи новых данных в журнал не будет выполнена, как ожидалось.

Таблица 9-175 Значения параметров ERROR и STATUS

ERROR	STATUS (W#16#)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0
0	7001	Первый вызов по фронту, заданному в REQ: BUSY = 0, DONE = 0 (обработка): BUSY = 1, DONE = 0
0	7002	N-й вызов (обработка): BUSY = 1, DONE = 0
1	8070	Вся встроенная память использована.
1	8090	Недопустимое имя файла
1	8091	Имя параметра не соответствует формату "String".
1	8092	Журнал записей данных отсутствует.
1	8093	Журнал записей данных существует.
1	8097	Запрашиваемый размер файла превышает максимально допустимый размер для файловой системы.
1	80B3	Недостаточный объем доступной загрузочной области памяти.
1	80B4	Карта памяти защищена от записи.
1	80C1	Слишком много открытых файлов.

9.7 Рецептury и записи данных

9.7.2.3 Работа с журналами записей данных

Файлы журналов сохранены в виде CSV-файлов (*.csv) во встроенной флэш-памяти. Файлы журналов данных Вы можете просмотреть, используя свойства Web-сервера PLC Web, или вставив карту памяти в стандартное устройство для чтения карт в PLC.

Просмотр журналов записей данных с помощью свойств Web-сервера PLC

Если PROFINET-порт PLC и PC подключены к сети, то Вы можете использовать web-браузер PC, например Microsoft Internet Explorer или Mozilla Firefox, для доступа к встроенному в PLC Web-серверу. При работе с Web-сервером PLC может быть в режимах "run" или "stop". Если PLC находится в режиме "run", то Ваша управляющая программа продолжает выполняться, даже при передаче журналов записей данных по сети с помощью Web-сервера PLC.

Доступ к Web-серверу:

1. Включите Web-сервер в конфигурации устройств (Device Configuration) для конкретного CPU (страница 787).
2. Подключите Ваш PC к PLC через PROFINET-сеть (страница 791).
3. Получите доступ к CPU через встроенный Web-сервер (страница 794).
4. Загружайте, редактируйте и удаляйте файлы журналов записей данных с помощью стандартной Web-страницы "File Browser" (стр. 810).
5. Откройте csv-файл, например, с помощью Microsoft Excel.

Примечание

Управление журналом данных

Храните не более 1000 записей в файле журнала. При превышении данного значения Web-серверу может не хватить ресурсов CPU для отображения записей данных.

Если Вы обнаружите, что в проводнике (File Browser) Web-страницы не отображаются записи данных, то Вам необходимо перевести CPU в режим STOP для отображения содержимого журнала и удаления записей данных.

Управляйте Вашими записями данных, чтобы их количество не превышало максимально допустимого значения, равного 1000 записей.

Просмотр записей данных на PLC-карте памяти

Если S7-1200 CPU содержит встроенную карту памяти типа "Program", то Вы можете извлечь эту карту памяти и вставить ее в стандартный слот SD (Secure Digital) или MMC (MultiMediaCard) в PC или PG. Если карта памяти извлечена и Ваша программа не выполняется, то PLC находится в режиме "stop".

Используйте проводник Windows для перехода к папке \DataLog карты памяти. В этой папке находятся все файлы Ваших журналов записей данных (*.csv).

Скопируйте файлы журналов записей данных на локальный жесткий диск Вашего PC. Теперь с помощью программы Excel открывайте копии csv-файлов, а не оригинальные файлы, сохраненные на карте памяти.

ЗАМЕЧАНИЕ

С помощью устройства чтения карт памяти Вы можете копировать, но не изменять или удалять файлы журналов записей данных с S7-1200 карты памяти

В качестве рекомендованного инструмента для просмотра, загрузки (копирования) и удаления файлов журналов записей данных можно использовать стандартную страницу проводника Web-сервера.

При непосредственном просмотре содержимого карты памяти в проводнике Windows есть риск, что Вы можете случайно удалить (изменить) записи данных или другие системные файлы, повредив файл журнала или сделав непригодной карту памяти.

ЗАМЕЧАНИЕ

Эффективная работа с картами памяти

Для обеспечения наибольшей производительности и надежности вашей системы, ограничить скорость записи данных на карту памяти, не быстрее 200 мс.

9.7.2.4 Ограничение размера файлов журналов записей данных

Файлы журналов записей данных занимают место в загрузочной области памяти PLC, наряду с программами, программными данными, конфигурационными данными, пользовательскими Web-страницами и системными PLC-данными. Большой программе необходим большой объем загрузочной области памяти. Для файлов журналов может не хватить свободной памяти. В этом случае Вы можете использовать "Program card" (стр. 138) для увеличения размера загрузочной области памяти. S7-1200 CPU могут использовать или внутреннюю или внешнюю память, но не обе одновременно.

Условие максимального размера файлов журнала записей данных

Максимальный размер одного файла журнала записей данных ограничен свободной загрузочной областью памяти или 500 мегабайтами, в зависимости от того, что из них меньше. Размер в 500 мегабайт в данном случае имеет отношение к структуре мегабайта, например, максимальный размер файла журнала записей данных в 500,000,000 байт соответствует 500 x 10002 байт.

Таблица 9-176 Размер загрузочной области памяти

Область памяти	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C, CPU 1217C	Хранение данных
Загрузочная область во встроенной флэш- памяти	1 МБ	1 МБ	4 МБ	4 МБ	Пользовательская программа и про- граммные данные, конфигурационные данные, журналы записей данных, пользовательские Web-страницы и системные PLC- данные
Загрузочная область на внешнем носителе Дополнительная флэш-память "Program card"	4 МБ, 12 МБ, 24 МБ, 256 МБ, 2 ГБ или 32 ГБ в зависимости от объема памяти SD-карты				

Определение размера свободной загрузочной области памяти

Свободный размер загрузочной области памяти изменяется при выполнении различных операций, т.к. операционная система периодически занимает и освобождает память. Для определения размера свободной загрузочной области памяти выполните следующие шаги.

1. Установите online-подключение между STEP 7 и конечным S7-1200 PLC.
2. Загрузите программу управления Вашими журналами записей данных.
3. Создайте дополнительные пользовательские Web-страницы, которые будут Вам необходимы. Стандартные Web-страницы доступа к файлам журнала записей данных сохранены в операционной системе PLC и не используют загрузочную область памяти.
4. Для просмотра занято и свободно объема загрузочной области памяти воспользуйтесь инструментом "Online and diagnostic" (Online-функции и диагностика) (стр. 1065).

Вычисление размера файла журнала (все записи данных)

Для созданного файла журнала записей данных выделяется максимальный объем памяти. В дополнение к объему памяти, необходимой для всех записей данных, Вам необходимо выделить область памяти для хранения заголовка журнала (если используется), заголовка меток времени (если используются), заголовка индекса записи и минимальный размер блока для распределения памяти.

Для определения размера файлов Ваших журналов записей данных используйте приведенную ниже формулу и убедитесь, что Вами не нарушено правило по ограничению максимального размера файлов.

Размер записей данных в байтах = (размер одной записи данных в байтах + байты меток времени + 12 байт) * количество записей данных

Заголовок

Размер заголовка записи данных в байтах = размер символов заголовка в байтах + 2 байта

Размер символов заголовка в байтах

- Без заголовка данных и без меток времени = 7 байт
- Без заголовка данных и с заголовком меток времени = 21 байт
- С заголовком данных и без заголовка меток времени = размер символов заголовка в байтах во всех столбцах, включая разделители в виде запятых
- С заголовком данных и с заголовком меток времени = размер символов заголовка в байтах во всех столбцах, включая разделители в виде запятых + 21 байт

DATA (Данные)

Размер записей данных в байтах = (размер одной записи данных в байтах + байты меток времени + 12 байт) * количество записей данных

Размер одной записи данных в байтах

Параметр DATA инструкции DataLogCreate указывает на структуру, которая определяет количество полей ввода данных и тип данных каждого поля одной записи данных.

Умножьте количество событий для данного типа данных на необходимое количество байтов. Повторите процесс для каждого типа записи данных, сложите все результаты в байтах для получения суммарного размера всех элементов для одной записи.

Размер отдельных элементов данных

Записи данных сохранены в виде символьных байтов в формате CSV-файла. В приведенной ниже таблице показано количество байт, необходимых для сохранения каждого элемента данных.

Тип данных	Количество байтов (байты данных + один байт для запятой)
Bool	2
Byte	5
Word	7
DWord	12
Char	4

9.7 Рецептуры и записи данных

String	Пример 1: Строка MyString [10] Максимальный размер заданной строки – 10 символов. Текстовые символы + автоматическое заполнение пустыми символами = 10 байт Двойные кавычки открытия и закрытия + символы запятой = 3 байта Суммарный размер: 10 + 3 = 13 байт Пример 2: Строка Mystring 2 Если размер не задан в квадратных скобках, то по умолчанию выделяется 254 байта. Текстовые символы + автоматическое заполнение пустыми символами = 254 байта Двойные кавычки открытия и закрытия + символы запятой = 3 байта Суммарный размер: 254 + 3 = 257 байт
USInt	5
UInt	7
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Количество записей данных в файле журнала

Параметр RECORDS инструкции DataLogCreate устанавливает максимальное количество записей данных в файле журнала.

Размер меток времени в байтах для одной записи данных

- Без меток времени = 0 байт
- С метками времени = 20 байт

9.7.2.5 Пример программы записи данных

Этот пример программы записи данных не содержит всех программных компонентов, необходимых для получения выборочных значений из динамического процесса программы, но приведены основные операции, выполняемые инструкциями записи данных (Data log). Структура и количество файлов журналов, которые Вам необходимы, зависит от требований управления Вашим процессом.

Примечание

Основное использование файлов журналов записей данных (Data log)

- Файлы журналов автоматически открываются после каждого выполнения инструкций DataLogCreate и DataLogNewFile.
- Файлы журналов автоматически закрываются после перехода PLC в режим "stop" или при выключении питания PLC.
- Файлы журналов записей данных (Data log) могут быть открыты перед выполнением инструкции DataLogWrite.
- Одновременно могут быть открыты до восьми файлов журналов записей данных. Может существовать более восьми файлов журналов, но некоторые из них должны быть закрыты, так как допускается открывать не более восьми журналов.

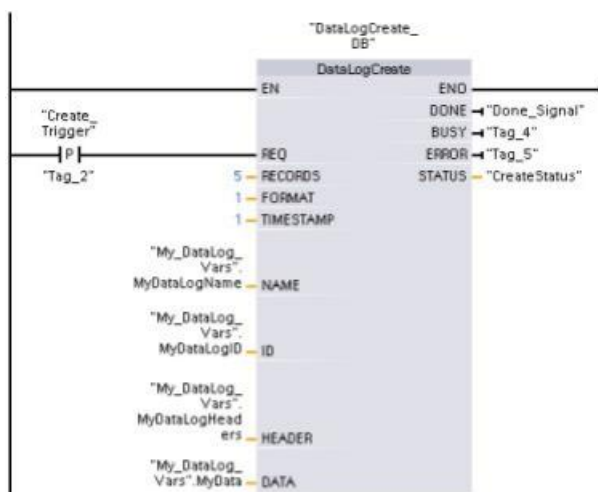
Пример программы записи данных

Пример наименований записей данных, текстовых заголовков и структуры MyData, созданной в блоке данных. Новые выборочные значения могут быть сохранены в трех временных переменных MyData. Процессные выборочные значения, находящиеся в DB, будут переданы в файл журнала записей данных после выполнения инструкции DataLogWrite.

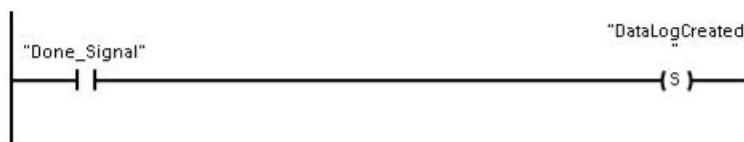
My_Datalog_Vars				
	Name	Data type	Start value	
1	Static			
2	MyNewDataLogName	String	'MyNEWDatLog'	
3	MyDataLogName	String	'MyDataLog'	
4	MyDataLogID	DWord	0	
5	MyDataLogHeaders	String	'Count,Temperature,Pressure'	
6	MyData	Struct		
7	MyCount	Int	0	
8	MyTemperature	Real	0.0	
9	MyPressure	Real	0.0	

9.7 Рецептуры и записи данных

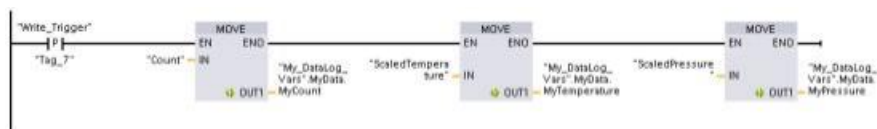
Сегмент 1: Положительный фронт на входе REQ запускает процесс создания файла журнала записей данных.



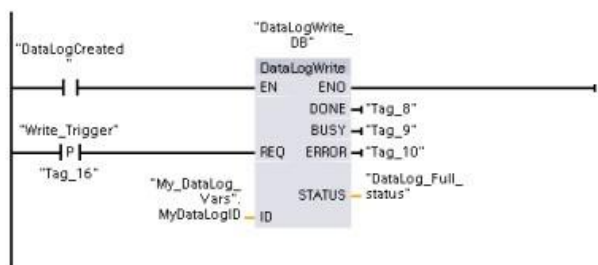
Сегмент 2: Фиксация значения на выходе DONE после выполнения инструкции DataLogCreate, т.к. оно действительно только для одного цикла сканирования.



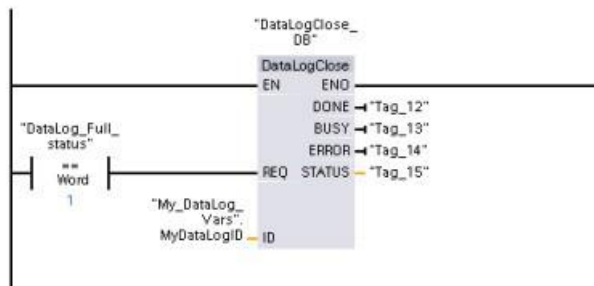
Сегмент 3: Положительный фронт сигнала включает сохранение новых процессных значений в структуре MyData.



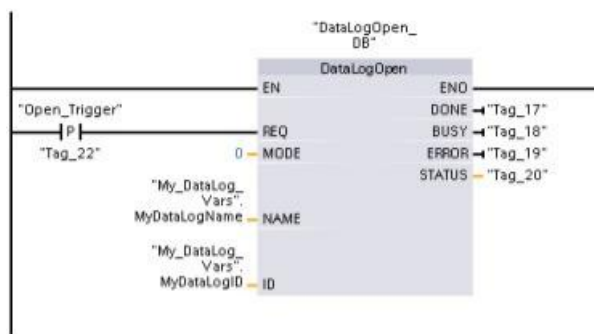
Сегмент 4: Состояние сигнала на входе EN изменяется по завершении выполнения инструкции DataLogCreate. Операции создания распространяется на множество циклов сканирования и должна быть завершена до выполнения операции записи. Положительный фронт сигнала на входе REQ вызывает включение операции записи.



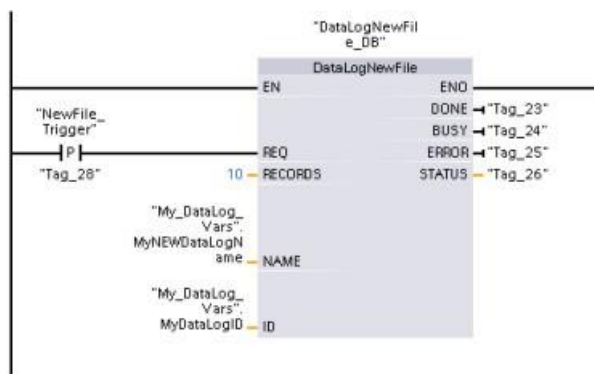
Сегмент 5: Заккрытие журнала после выполнения последней записи данных. После выполнения инструкции DataLogWrite выполняется последняя запись в журнал, файл журнала считается заполненным и после выполнения инструкции DataLogWrite значение параметра STATUS = 1.



Сегмент 6: Положительный фронт сигнала DataLogOpen на входе REQ имитирует нажатие кнопки на HMI пользователем, открывающей файл журнала записей данных. Если вы открыли файл журнала (Data log), содержащий все записи процессных данных, то при следующем выполнении инструкции DataLogWrite старые записи будут перезаписаны. Вы можете сохранить старый журнал записей данных, а вместо него создать новый, как показано в сегменте 7.

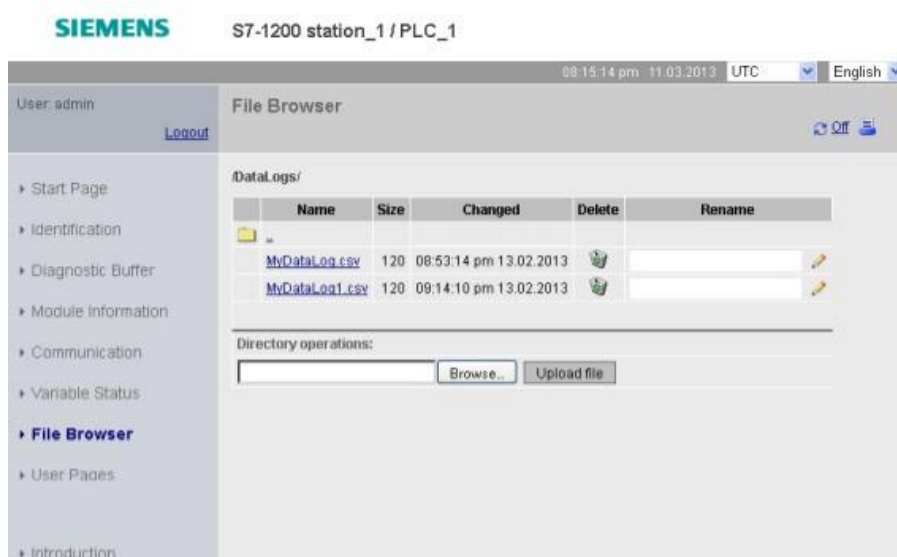


Сегмент 7: Тип параметра ID – IN/OUT. Во-первых, Вы задаете ID-значение существующему файлу журнала записей данных, структуру которого Вы хотите скопировать. После завершения выполнения инструкции DataLogNewFile, новое и уникальное ID-значение для нового файла журнала записей данных записывается обратно в ячейку, заданную для ID. Фиксированное значение бита DONE = TRUE не отображено, обратитесь к сегментам 1, 2 и 4 для просмотра примеров бита DONE.



9.7 Рецептуры и записи данных

Файлы журналов записей данных, созданные на примере программы, отображаемой с использованием Веб-сервера S7-1200 CPU



- ① Опция "Delete" (Удалить) недоступна, если Вы не обладаете соответствующими правами доступа.
- ② Опция "Rename" (Переименовать) недоступна, если Вы не обладаете соответствующими правами доступа.

Таблица 9-177 Примеры загруженных csv-файлов, просмотренных с помощью программы Excel

Две записи, записанные максимум в 5 файлов		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00
	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00
	4						
	5						
Пять записей, записанные максимум в 5 файлов		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
	7						
После выполнения дополнительной записи в уже заполненный файл, шестая операция перезаписывает наиболее старую из пяти запись. После следующей операции записи будут перезаписаны уже две записи из семи и т.д.		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
	7						
	8						

Примечание

Неиспользуемые записи данных отмечены маркером //END, показывающем завершение журнала записи данных, который еще не заполнен. До версии V4.1 S7-1200 CPU, записи данных не использовали маркер //END при неполном заполнении файла журнала.

9.8 Управление блоком данных

9.8.1 Инструкции READ_DBL и WRIT_DBL (Чтение запись блока данных в загрузочной области памяти)

Таблица 9-178 Инструкции READ_DBL и WRIT_DBL

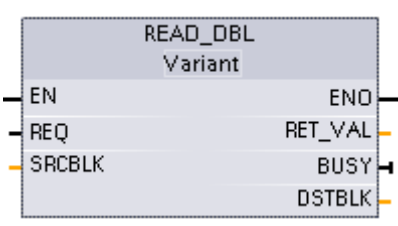
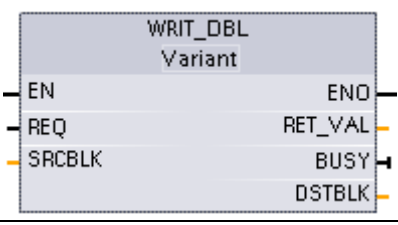
LAD / FBD	SCL	Описание
	<pre>READ_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_);</pre>	Копирует начальные значения DB или часть значений из загрузочной области памяти в целевой DB в оперативной памяти. Содержимое загрузочной области памяти не изменяется во время процесса копирования.
	<pre>WRIT_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_);</pre>	Копирует фактические значения DB или части значений из оперативной памяти в целевой DB в загрузочной области памяти. Содержимое оперативной памяти не изменяется во время процесса копирования.

Таблица 9-179 Типы данных для параметров

Параметр и тип	Тип данных	Описание
REQ	IN	BOOL
SRCBLK	IN	VARIANT
RET_VAL	OUT	INT
BUSY	OUT	BOOL
DSTBLK	OUT	VARIANT

9.8 Управление блоком данных

Обычно блоки данных (DB) и в загрузочной области памяти (flash-память) и в оперативной памяти (RAM). Стартовые значения (исходные значения) всегда сохраняются в загрузочной области памяти, а текущие значения всегда сохраняются в оперативной памяти. Инструкция READ_DBL может быть использована для копирования стартовых значений из загрузочной области памяти в текущие значения DB в оперативной памяти, чтобы использовать их в Вашей программе. Инструкция WRIT_DBL может быть использована для обновления стартовых значений, сохраненной в загрузочной области встроенной памяти или карты памяти текущими значениями из оперативной памяти.

Примечание**Влияние использования инструкций WRIT_DBL и READ_DBL на flash-память**

Инструкция WRIT_DBL выполняет операции записи во flash-память (загрузочную область встроенной памяти или карты памяти). Чтобы продлить срок службы flash-памяти, используйте инструкцию WRIT_DBL для нечастых обновлений, например, выполнение записей в процессе производства. По тем же причинам сократите количество вызовов инструкции READ_DBL для операций чтения.

Блоки данных для READ_DBL и WRIT_DBL Вы можете создавать перед вызовом этих инструкций в STEP 7. Если созданный Вами исходный блок данных стандартного ("standard") типа, то и целевой блок данных также должен быть стандартного ("standard") типа. Если созданный Вами исходный блок данных оптимизированного ("optimized") типа, то и целевой блок данных также должен быть оптимизированного ("optimized") типа.

Если блоки данных стандартные, то Вы можете задать имя тега или значение P#. Значение P# позволяет Вам задавать и копировать любое количество элементов заданного размера (Byte, Word или DWord). Т.е., Вы можете копировать части блоков данных или целые блоки DB. Если блоки данных оптимизированные, то Вы можете задавать только имя тега; Вы не можете изменять значение оператора P#. Если Вы задаете имя тега для стандартных или оптимизированных блоков данных (или для других типов в оперативной памяти), то инструкция копирует данные, т.е. ссылки на имена тегов. Это могут быть определяемые пользователем тип, массив или основной элемент. Если блок данных стандартный (не оптимизированный), то с этими инструкциями Вы можете использовать только тип "Struct" (Структура). Вы должны использовать тип, определяемый пользователем (UDT), если это структура в оптимизированной памяти. Только определяемый пользователем тип данных гарантирует идентичность типов данных исходной и целевой структур.

Примечание**Использование структуры (типа данных "Struct") в "оптимизированном" блоке данных**

Для использования типов данных "Struct" в оптимизированных блоках данных Вы сначала должны создать пользовательский тип данных (UDT) для "Struct". Затем Вам необходимо сконфигурировать исходный и целевой блоки данных с UDT. UDT гарантирует, что тип данных в пределах структуры (Struct) остается консистентным для обоих блоков данных.

Для "стандартных" блоков данных Вы можете использовать структуру (Struct) без создания пользовательского типа данных UDT.

Инструкции READ_DBL и WRIT_DBL выполняются асинхронно с циклической программой сканирования. Обработка продолжится в течение нескольких вызовов READ_DBL и WRIT_DBL. Вы запускаете обработку вызываемого DB с помощью REQ = 1, а затем контролируете выходные значения параметров BUSY и RET_VAL, чтобы определить, когда передача данных будет завершена без ошибок.

Примечание

Влияние инструкций WRIT_DBL и READ_DBL на коммуникационную нагрузку

Если инструкции WRIT_DBL или READ_DBL выполняются непрерывно, то увеличение потребляемых коммуникационных ресурсов может привести к потере связи STEP 7 с CPU. В этом случае предпочтительнее использовать положительный фронт сигнала на входе (стр. 215) для параметра REQ, чем нормально открытый или закрытый вход (стр. 209), чтобы он оставался включенным (высоким уровнем сигнала) для нескольких циклов сканирования.

Для обеспечения консистентности данных, не изменяйте параметры целевой области при обработке инструкции READ_DBL или параметры исходной области при обработке инструкции WRIT_DBL (т.е. до тех пор, пока параметр BUSY = TRUE).

Предварительные условия для параметров SRCBLK и DSTBLK:

- Блок данных должен быть создан перед тем, как к нему можно обращаться.
- Размер указателя VARIANT типа BOOL должен быть кратным 8.
- Размер указателя VARIANT типа STRING должен быть одним и тем же для исходного и целевого указателей.

Рецептуры и параметры настроек оборудования

Инструкции READ_DBL и WRIT_DBL могут быть использованы для управления рецептурами или параметрами настроек оборудования. Это, по сути, является еще одним способом сохранения значений, которые изменяются не так часто, для ограничения количества записей и предотвращения преждевременного выхода из строя flash-памяти. Это позволяет эффективней использовать сохраняемую область памяти и, кроме того, поддерживается сохранение данных при выключении питания, по крайней мере, для значений, которые не меняются не так часто. С помощью инструкции WRIT_DBL Вы можете записывать в загрузочную область памяти данные рецептов или параметры настроек оборудования из оперативной памяти, а с помощью инструкции READ_DBL Вы можете копировать эту информацию из загрузочной области памяти в оперативную память.

9.8 Управление блоком данных

Таблица 9-180 Коды ошибок

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
0081	Предупреждение: Исходная область меньше целевой области. Исходные данные полностью скопированы с дополнительными байтами в целевую область без изменений.
7000	Вызов при REQ = 0: BUSY = 0
7001	Первый вызов при REQ = 1 (обработка): BUSY = 1
7002	N-й вызов (обработка): BUSY = 1
8051	Неправильный тип блока данных.
8081	Исходная область больше целевой области. Целевая область заполняется полностью, а остальные исходные байты игнорируются.
8251	Неправильный тип исходного блока данных.
82B1	Исходный блок данных отсутствует.
82C0	Исходный блок данных редактируется другим оператором или коммуникационной функцией.
8551	Неправильный тип целевого блока данных.
85B1	Отсутствует целевой блок данных.
85C0	Целевой блок данных редактируется другим приложением или коммуникационной функцией.
80C3	В настоящий момент выполняется более 50 READ_DBL или 50 WRIT_DBL операторов.

Смотрите также "Рецептуры" (стр. 411).

9.9 Обработка адресов

9.9.1. Инструкция GEO2LOG (Определение аппаратного идентификатора слота)

Инструкция GEO2LOG может быть использована для определения аппаратного идентификатора, основанного на информации слота.

Таблица 9-181 Инструкция GEO2LOG

LAD / FBD	SCL	Описание
	<pre>ret_val := GEO2LOG(GEOADDR:=_variant_in_out_, laddr:=_word_out_);</pre>	Инструкция GEO2LOG может быть использована для определения аппаратного идентификатора, основанного на информации слота.

Инструкция GEO2LOG определяет аппаратный идентификатор на основе информации слота, который Вы задаете, используя системный тип данных GEOADDR:

В зависимости от типа оборудования Вы определяете для параметра HWTYPE следующую информацию на основе оценки других GEOADDR-параметров:

- При HWTYPE = 1 (PROFINET IO система):
 - Оценивается только IOSYSTEM. Остальные GEOADDR-параметры не учитываются.
 - Аппаратный идентификатор PROFINET IO системы – выходной параметр.
- При HWTYPE = 2 (PROFINET IO устройство):
 - Оцениваются IOSYSTEM и STATION. Остальные GEOADDR-параметры не учитываются.
 - Аппаратный идентификатор PROFINET IO устройства – выходной параметр.
- При HWTYPE = 3 (Стойка):
 - Оцениваются только IOSYSTEM и STATION. Остальные GEOADDR-параметры не учитываются.
 - Аппаратный идентификатор стойки – выходной параметр.
- При HWTYPE = 4 (Модуль):
 - Оцениваются IOSYSTEM, STATION и SLOT. Параметр SUBSLOT GEOADDR не учитывается.
 - Аппаратный идентификатор модуля – выходной параметр.
- При HWTYPE = 5 (Субмодуль):
 - Оцениваются все GEOADDR-параметры.
 - Аппаратный идентификатор субмодуля – выходной параметр.

9.9 Обработка адресов

Параметр AREA системного типа данных GEOADDR не учитывается.

Таблица 9-182 Типы данных для параметров

Параметр и тип		Тип данных	Описание
GEOADDR	IN/OUT или IN?	Variant	Указатель структуры системного типа данных GEOADDR. Системный тип данных GEOADDR содержит информацию слота, аппаратный идентификатор которого был определен. За дополнительной информацией обратитесь к странице 453 "Системный тип данных GEOADDR".
RET_VAL	OUT или RETURN?	Int	Вывод информации об ошибки.
LADDR	OUT	HW_ANY	Аппаратный идентификатор сборки или модуля. Номер будет назначен автоматически и будет сохранен в свойствах аппаратной конфигурации.

За дополнительной информацией о поддерживаемых типах данных обратитесь к online-справке STEP 7 "Overview of the valid data types" (Обзор поддерживаемых типов данных).

Таблица 9-183 Коды ошибок

RET_VAL* (W#16#...)	Пояснение
0	Ошибки отсутствуют.
8091	Недопустимое значение GEOADDR для HWTYPE.
8094	Недопустимое значение GEOADDR для IOSYSTEM.
8095	Недопустимое значение GEOADDR для STATION.
8096	Недопустимое значение GEOADDR для SLOT.
8097	Недопустимое значение GEOADDR для SUBSLOT.
* Коды ошибок могут отображаться в редакторе программы в виде целых или шестнадцатеричных значений.	

9.9.2. Инструкция LOG2GEO (Определение слота по аппаратному идентификатору)

Инструкция LOG2GEO используется для определения географического адреса (слота модуля) по логическому адресу, содержащемуся в аппаратном идентификаторе.

Таблица 9-184 Инструкция LOG2GEO

LAD / FBD	SCL	Описание
	<pre>ret_val := LOG2GEO(laddr:=_word_in_, GEOADDR:=_variant_in_out_);</pre>	Инструкция LOG2GEO используется для определения географического адреса (слота модуля) по логическому адресу, содержащемуся в аппаратном идентификаторе.

Инструкция LOG2GEO определяет географический адрес логического адреса по аппаратному идентификатору:

- Параметр LADDR используется для выбора логического адреса по аппаратному идентификатору.
- GEOADDR содержит географический адрес логического адреса заданного для входа LADDR.

Примечание

В случае, если компонент не поддерживается оборудованием, возвращается номер субслота для модуля 0.

Если вход LADDR не адресован аппаратному объекту, то выводится сообщение об ошибке.

9.9 Обработка адресов

Таблица 9-185 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	IN	HW_ANY	Аппаратный идентификатор IO-системы или модуля. Номер назначается автоматически и сохраняется в свойствах CPU или интерфейсе аппаратной конфигурации.
RET_VAL	OUT	Int	Код ошибки при выполнении инструкции.
GEOADDR	IN_OUT	Variant	Указатель системного типа данных GEOADDR. Тип системных данных GEOADDR содержит информацию слота. За дополнительной информацией обратитесь к подразделу "Тип системных данных GEOADDR" стр.453.

За дополнительной информацией о поддерживаемых типах данных обратитесь к online-справке STEP 7 "Overview of the valid data types" (Обзор поддерживаемых типов данных).

Таблица 9-186 Коды ошибок

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый адрес, заданный параметру LADDR.
* Коды ошибок могут отображаться в редакторе программы в виде целых или шестнадцатеричных значений.	

9.9.3. Инструкция IO2MOD (Определение аппаратного идентификатора по адресу ввода/вывода)

Инструкция IO2MOD используется для определения аппаратного идентификатора модуля по адресу ввода/вывода (суб)модуля.

Таблица 9-187 Инструкция IO2MOD

LAD / FBD	SCL	Описание
	<pre>ret_val := IO2MOD(ADDR:=_word_in_, LADDR:=_word_out_);</pre>	Инструкцию IO2MOD Вы можете использовать для определения слота модуля по аппаратному идентификатору.

Инструкция IO2MOD определяет аппаратный идентификатор модуля по адресу ввода/вывода (I, Q, PI, PQ) (суб)модуля.

Введите адрес ввода/вывода в параметр ADDR. Если в этом параметре используется последовательность адресов ввода/вывода, то для определения аппаратного идентификатора будет оцениваться только первый адрес. Если первый адрес задан корректно, то размер списка адресов в параметре ADDR не имеет значения. Если используемая адресная область включает в себя несколько модулей или неиспользуемых адресов, то будет определен аппаратный идентификатор первого модуля.

Если параметр ADDR не содержит заданных адресов ввода/вывода (суб)модуля, то в качестве выходного значения параметра RET_VAL выводится код ошибки "8090".

Примечание

Ввод адресов ввода/вывода в SCL

Вы можете программировать, используя идентификатор "%QWx:P" доступа к вводу/выводу в SCL. В этом случае используйте символическое имя тега или абсолютный адрес образа процесса.

Таблица 9- 188 Типы данных для параметров

Параметр	Обозначение	Тип данных	Область памяти	Описание
ADDR	IN или IN/OUT ?	Variant	I, Q, M, D, L	Адреса ввода/вывода (I, Q, PI, PQ) в пределах (суб)модуля. Убедитесь, что для параметра ADDR не используется частичный доступ. В противном случае, в параметре LADDR выводятся недействительные значения.
RET_VAL	OUT или RETURN ?	Int	I, Q, M, D, L	Код ошибки при выполнении инструкции.
LADDR	OUT	HW_IO	I, Q, M, D, L	Определенный аппаратный идентификатор (логический адрес) (суб)модуля ввода/вывода.

9.9 Обработка адресов

За дополнительной информацией о поддерживаемых типах данных обратитесь к online-справке STEP 7 "Overview of the valid data types" (Обзор поддерживаемых типов данных).

Таблица 9-189 Коды ошибок

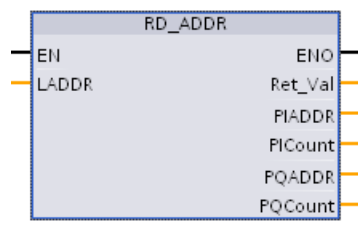
RET_VAL* (W#16#...)	Описание
0	Ошибки отсутствуют.
8090	Адрес ввода/вывода, заданный в параметре ADDR, не используется никаким аппаратным компонентом.

* Коды ошибок могут отображаться в редакторе программы в виде целых или шестнадцатеричных значений.

9.9.4. Инструкция RD_ADDR (Определение адреса ввода/вывода по аппаратному идентификатору)

Инструкция RD_ADDR используется для получения адреса ввода/вывода субмодуля.

Таблица 9-190 Инструкция RD_ADDR

LAD / FBD	SCL	Описание
	<pre>ret_val := RD_ADDR(laddr:=_word_in_, PIADDR=>_uint_out_, PICount=>_uint_out_, PQADDR=>_uint_out_, PQCount=>_uint_out_,);</pre>	Инструкция RD_ADDR используется для получения адреса ввода/вывода субмодуля.

Инструкция RD_ADDR определяет размер и стартовый адрес входов или выходов по аппаратному идентификатору субмодуля:

- Параметр LADDR используется для выбора входа или выхода модуля по его аппаратному идентификатору.
- В зависимости от типа модуля (модуль ввода или вывода) используются следующие выходные параметры:
 - Для модуля ввода: определенные значения выводятся в виде параметров PIADDR и PICOUNT.
 - Для модуля вывода: определенные значения выводятся в виде параметров PQADDR и PQCOUNT.
- Каждый из параметров PIADDR и PQADDR содержит начальный адрес диапазона ввода-вывода модуля.
- Каждый из параметров PICOUNT и PQCOUNT содержит байты входов и выходов (1 байт – для 8 входов/выходов, 2 байта – для 16 входов/выходов).

Таблица 9-191 Типы данных для параметров

Параметр и тип		Тип данных	Описание
LADDR	IN	HW_IO	Аппаратный идентификатор (суб)модуля
RET_VAL	OUT	Int	Код ошибки при выполнении инструкции
PIADDR	OUT	UDInt	Стартовый адрес модуля ввода
PICOUNT	OUT	UInt	Количество байт входов
PQADDR	OUT	UDInt	Стартовый адрес модуля вывода
PQCOUNT	OUT	UInt	Количество байт выходов

За дополнительной информацией о поддерживаемых типах данных обратитесь к online-справке STEP 7 "Overview of the valid data types" (Обзор поддерживаемых типов данных).

Таблица 9-192 Коды ошибок

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый аппаратный идентификатор модуля в параметре LADDR.
* Коды ошибок могут отображаться в редакторе программы в виде целых или шестнадцатеричных значений.	

9.9.5. Тип системных данных GEOADDR

Географический адрес

Системный тип данных GEOADDR содержит географический адрес модуля (или информацию слота).

- Географический адрес для PROFINET IO:
Географический адрес PROFINET IO состоит из идентификатора PROFINET IO системы, номера устройства, номера слота и субмодуля (если субмодуль используется).
- Географический адрес для PROFIBUS DP:
Географический адрес PROFIBUS DP состоит из идентификатора DP мастер-системы, номера станции и номера слота.

Информацию слота для каждого из модулей можно найти в аппаратной конфигурации.

Структура системного типа данных GEOADDR

Структура GEOADDR создается автоматически, когда Вы вводите тип данных "GEOADDR" в блок данных.

Наименование параметра		Тип данных	Описание
GEOADDR		STRUCT	
	HWTYPE	UINT	Тип аппаратного компонента: • 1: Система ввода/вывода (PROFINET/PROFIBUS) • 2: IO устройство/DP ведомое устройство • 3: Стойка • 4: Модуль • 5: Субмодуль Если тип аппаратного компонента не поддерживается инструкцией, параметр HWTYPE выдаёт значение "0".
	AREA	UINT	Идентификатор области: • 0 = CPU • 1 = PROFINET IO • 2 = PROFIBUS DP • 3 = AS-интерфейс
	IOSYSTEM	UINT	PROFINET IO система (0 = центральный модуль или стойка)
	STATION	UINT	• Номер стойки, если идентификатор области AREA = 0 (центральный модуль). • Номер станции, если идентификатор области AREA > 0.
	SLOT	UINT	Номер слота
	SUBSLOT	UINT	Номер субмодуля. Этот параметр принимает значение "0", если субмодуль недоступен или может быть подключен

9.10 Основные коды ошибок для "Расширенных" инструкций

Таблица 9-193 Основные коды ошибок для расширенных инструкций

Код ошибки (W#16#....) ¹	Описание
8x22 ²	Область ввода слишком мала
8x23	Область вывода слишком мала
8x24	Недопустимая входная область
8x25	Недопустимая выходная область
8x28	Недопустимое назначение входного бита
8x29	Недопустимое назначение выходного бита
8x30	Выходная область является защищённым от записи DB.
8x3A	DB не загружен.

¹ Если возникает одна из этих ошибок, когда выполняется кодовый блок, то процессор остается в Run (по умолчанию) или можно сконфигурировать переход в режим STOP. При желании, Вы можете использовать GetError или GetErrorID инструкции внутри этого блока для обработки ошибки локально (CPU остается в Run), и создать программную реакцию на ошибку..

² "x" представляет собой номер параметра с ошибкой. Номера параметров начинаются с 1.

Технологические инструкции

10.1 Высокоскоростной счетчик

10.1.1. Инструкция CTRL_HSC (Управление высокоскоростным счетчиком)

Таблица 10- 1 Инструкция CTRL_HSC (Для стандартного счета)

LAD / FBD	SCL	Описание
	<pre>"CTRL_HSC_1_DB" (hsc:=W#16#0, dir:=False, cv:=False, rv:=False, peri- od:=False, new_dir:=0, new_cv:=L#0, new_rv:=L#0, new_period:=0, busy=>_bool_out_, status=>_word_out_);</pre>	Каждая инструкция CTRL_HSC (Управление высокоскоростным счетчиком) использует структуру, сохраненную в DB, чтобы сохранять данные счетчика. Вы назначаете DB, когда инструкция CTRL_HSC помещена в редакторе.

¹ Когда Вы вставляете инструкцию, STEP 7 отображает диалоговое окно "Call Options" для создания связанного DB.

² В примере на SCL, "CTRL_HSC_1_DB" является именем экземплярного DB.

Таблица 10- 2 Типы данных для параметров

Параметр и тип		Тип данных	Описание
HSC	IN	HW_HSC	Идентификатор в/с счетчика
DIR ^{1, 2}	IN	Bool	1 = Запрос нового направления
CV ¹	IN	Bool	1 = Запросе установки нового значения счета
RV ¹	IN	Bool	1 = Запрос установки нового ссылочное значение
PERIOD ¹	IN	Bool	1 = Запросите установить нового контрольного значения (только для режима измерения частоты)
NEW_DIR	IN	Int	Новое направление: 1 = вперед, -1 = назад
NEW_CV	IN	DInt	Новое значение счета
NEW_RV	IN	DInt	Новое контрольное значение
NEW_PERIOD	IN	Int	Новое значение периода в секундах (только для режима измерения частоты): 1= 1 с 2 = .1 с 3 = 0.1 с

Параметр и тип		Тип данных	Описание
BUSY ³	OUT	Bool	Функция занята
STATUS	OUT	Word	Код состояния выполнения

- ¹ Если обновление значения параметра не требуется, то соответствующие входные значения игнорируются.
- ² Параметр DIR допустим только, если сконфигурированное направление счета установлено на "User program (internal direction control)". Вы определяете, как использовать этот параметр в конфигурации устройства в/с счетчика.
- ³ Для в/с счетчика в ЦПУ или на СП, параметр BUSY всегда имеет значение 0.

Вы конфигурируете параметры для каждого в/с счетчика в конфигурации устройства для ЦПУ для функции счета/частоты, опций сброса, конфигурации событий прерывания, аппаратного ввода-вывода и адреса значения счета.

Некоторые параметры для в/с счетчика могут быть изменены Вашей пользовательской программой, чтобы обеспечить программное управление процессом счета:

- Установка направления счета в значение NEW_DIR
- Установка текущего значения счета в значение NEW_CV
- Установка контрольного значения в значение NEW_RV
- Установка значения периода (для режима измерения частоты) в значению NEW_PERIOD

Если следующие булевы флаговые значения установлены в 1, когда выполняется инструкция CTRL_HSC, то соответствующее значение NEW_xxx загружается в счетчик. Многократные запросы (более одного флага установлено одновременно) обрабатываются в одиночном выполнении инструкции CTRL_HSC.

- DIR = 1 является запросом загрузки значения NEW_DIR, 0 = без изменений
- CV = 1 является запросом загрузки значения NEW_CV, 0 = без изменений
- RV = 1 является запросом загрузки значения NEW_RV, 0 = без изменений
- PERIOD = 1 является запросом загрузки значения NEW_PERIOD, 0 = без изменений

Инструкция CTRL_HSC обычно помещается в ОБ аппаратного прерывания, который выполняется, когда инициировано событие аппаратного прерывания от датчика счета. Например, если событие CV=RV инициировало прерывание счета, то ОБ аппаратного прерывания выполняет инструкцию CTRL_HSC и может изменить контрольное значение, загружая значение NEW_RV.

Текущее значение счета недоступно в параметрах CTRL_HSC. Адрес образа процесса, который хранит текущее значение счета, назначается во время аппаратного конфигурирования высокоскоростного счетчика. Вы можете использовать логику программы, чтобы непосредственно считывать значение счета. Значение, возвращенное в Вашу программу, будет корректным количеством в момент, когда счетчик был считан. Счетчик продолжит считать высокоскоростные события. Поэтому, фактическое значение счета могло измениться, прежде чем Ваша программа завершает процесс, используя старое значение счета.

Текущее значение в/с счетчика: Программный доступ, диапазон значений и динамическая смена направления

ЦПУ хранит текущее значение каждого в/с счетчика по входному (I) адресу. Следующая таблица показывает адреса по умолчанию, назначенные текущему значению каждого в/с счетчика. Вы можете изменить, I адрес для текущего значения, изменяя свойства ЦПУ в Конфигурации устройства.

Высокоскоростные счетчики используют значение DInt, чтобы сохранять текущее значение счета. Значение DInt обладает диапазоном от -2147483648 до +2147483647. Счетчик переворачивается из максимального положительного значения в максимальное отрицательное значение при счете вверх, и из максимального отрицательного значения в максимальное положительное значение при счете вниз.

В/с счетчик	Тип данных текущего значения	Адрес текущего значения по умолчанию
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

Если происходит ошибка, ENO сбрасывается в 0, и выход STATUS указывает код состояния.

Таблица 10- 3 Коды состояния выполнения

STATUS (W#16#)	Описание
0	Без ошибок
80A1	Идентификатор не адресует в/с счетчик
80B1	Недопустимое значение в NEW_DIR
80B2	Недопустимое значение в NEW_CV
80B3	Недопустимое значение в NEW_RV
80B4	Недопустимое значение в NEW_PERIOD
80C0	Множественный доступ к высокоскоростному счетчику
80D0	Высокоскоростной счетчик (HSC) не активирован в аппаратной конфигурации ЦПУ

10.1.2. Инструкция CTRL_HSC_EXT (Управление высокоскоростным счетчиком (расширенное))

Инструкция CTRL_HSC_EXT обеспечивает программный доступ к числу входных импульсов назначенного в/с счетчика за указанный период времени. Эта инструкция позволяет программе определять период между входными импульсами с хорошим наносекундным разрешением.

Чтобы использовать инструкцию CTRL_HSC_EXT, выполните шаги ниже:

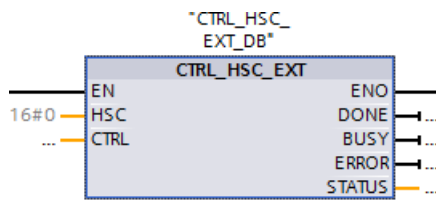
- 1. Сконфигурируйте связанный в/с счетчик в режим периода. Выберите нужную рабочую фазу. Если Вы выбираете внутреннее управление направлением, Вы освобождаете вход направления для другого использования.
- 2. Поместите CTRL_HSC_EXT в LAD-сегмент, что также создает экземплярный блок данных CTRL_HSC_EXT_DB.
- 3. Создайте User Global_DB=ex: "MYDB", который является входным параметром CTRL_HSC_EXT. Этот DB содержит информацию, необходимую SFB.
- 4. В MYDB определите местоположение пустой строки и добавьте переменную Name=Ex: "My period".
- 5. Добавьте тип данных, вводя в "HSC_Period" <enter> (выпадающий список в настоящее время не содержит эту опцию). Вы должны ввести это имя точно как показано.
- 6. Проверьте, что переменная "MyPeriod" теперь является расширяемой коммуникационной структурой данных.
- 7. Подключите в LAD-инструкции CTRL_HSC_EXT вывод "CTRL" к переменной DB "MYDB".MyPeriod.

Примечание

Настройка времени фильтра цифрового входа

Для цифровых входов в/с счетчика используйте наименьшую ожидаемую длительность импульса для настройки фильтра связанного цифрового входа.

Таблица 10- 4 Инструкция CTRL_HSC_EXT

LAD / FBD	SCL	Описание
	<pre>"CTRL_HSC_1_DB" (hsc:=_hw_hsc_in_ done:=_done_out_ busy:=_busy_out_ error:=_error_out_ status:=_status_out_ ctrl:=MYDB.MyPeriod);</pre>	Каждая инструкция CTRL_HSC_EXT (Управление высокоскоростным счетчиком) использует определенную системой структуру данных, сохраняемую в пользовательском глобальном DB для хранения данных счета. Тип данных HSC_Period назначается как входной параметр CTRL_HSC_EXT.

1 STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

2 В примере на SCL, "CTRL_HSC_1_DB" является именем экземплярного DB.

Таблица 10- 5 Типы данных для параметров

Параметр и тип		Тип данных	Описание
HSC	IN	HW_HSC	Идентификатор в/с счетчика
CTRL	IN	HSC_Period	Вход SFB и данные возврата (см. таблицу "Структура HSC_Period"),
DONE	OUT	Bool	1 = Указывает, что SFB закончен. Всегда 1, потому что SFB является синхронным
BUSY	OUT	Bool	Всегда 0, функция никогда не занята
ERROR	OUT	Bool	1 = Указывает ошибку
STATUS	OUT	Word	Код состояния выполнения (см. таблицу "Коды состояния выполнения")

Таблица 10- 6 Структура HSC_Period

Элемент структуры		Тип данных	Описание
ElapsedTime	OUT	UDINT	Время прошедшее между последними входными импульсами последовательных интервалов.
EdgeCount	OUT	UDINT	Возвращает число входных импульсов последнего завершенного периода.
EnHSC	IN	Bool	Разрешает в/с счетчик: 1 = разрешает SFB, 0 = отключает SFB
EnPeriod	IN	Bool	Позволяет обновить Период: 1 = Изменяет период SFB на NewPeriod
NewPeriod	IN	INT	NewPeriod определяет интервал измерения периода (время, затраченное на выполнение измерения периода). Разрешенными значениями являются 10, 100 или 1000 миллисекунд.

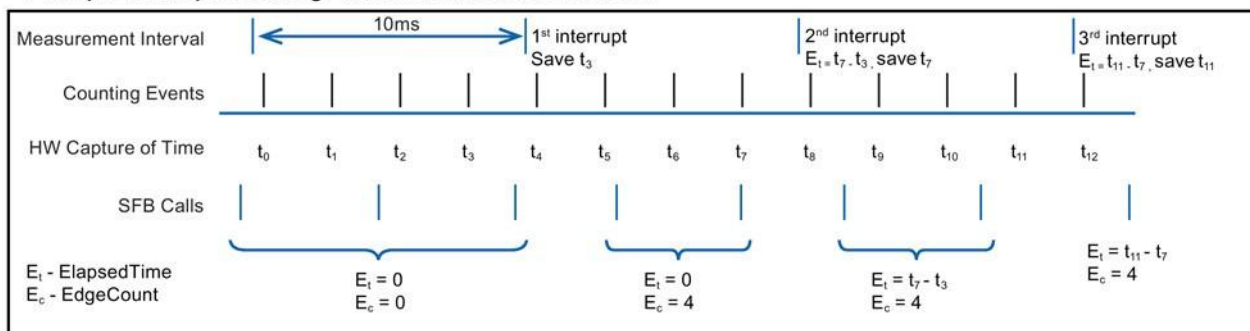
Определения элементов структуры HSC_Period

- ElapsedTime возвращает время в наносекундах, прошедшее между последним импульсом в текущем интервале измерения периода и последним импульсом в предыдущем интервале измерения периода.
Если EdgeCount = 0, то ElapsedTime является суммарным временем с момента последнего импульса. ElapsedTime обладает диапазоном от 0 до 4,294,967,280 наносекунд (от 0x0000 0000 до 0xFFFF FFF0). Переполнение периода обозначается возвращаемым значением 4 294 96 295 (0xFFFF FFFF. Значения от 0xFFFF FFF1 до 0xFFFF FFFE зарезервированы.

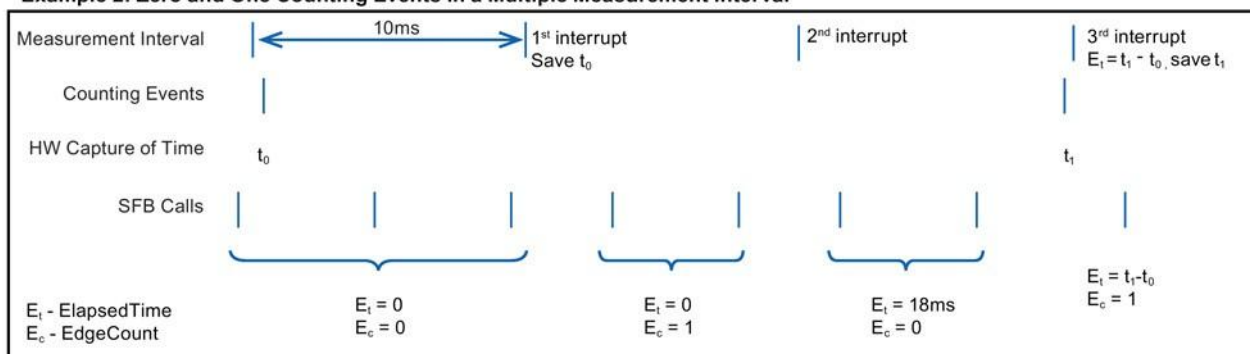
Если ElapsedTime либо равно 0 (нет принятых входных импульсов) или 0xFFFF FFFF (переполнение периода), то EdgeCount недопустим.
- EdgeCount возвращает число входных импульсов нового измерительного интервала. EdgeCount должен быть равным "1" или больше, чтобы существовала возможность вычислить период между импульсами. Период вычисляется по формуле: Период = ElapsedTime/EdgeCount.

Следующие примеры показывают, как инструкция измеряет периоды.

Example 1: Multiple Counting Events in a Measurement Interval



Example 2: Zero and One Counting Events in a Multiple Measurement Interval



Rules:

1. If $E_1 = 0$, the period is invalid
2. Else, $\text{Period} = E_1 / E_c$

ЦПУ хранит текущее значение каждого в/с счетчика по входному (I) адресу. Следующая таблица показывает адреса по умолчанию, назначенные текущему значению каждого в/с счетчика. Вы можете изменить, I адрес для текущего значения, изменяя свойства ЦПУ в Конфигурации устройства.

Высокоскоростные счетчики используют значение DInt, чтобы сохранять текущее значение счета. Значение DInt обладает диапазоном от -2147483648 до +2147483647. Счетчик переворачивается из максимального положительного значения в максимальное отрицательное значение при счете вверх, и из максимального отрицательного значения в максимальное положительное значение при счете вниз.

Таблица 10- 7 Адреса по умолчанию, назначенные текущим значениям каждого в/с счетчика

В/с счетчик	Тип данных текущего значения	Адрес текущего значения по умолчанию
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

Если происходит ошибка, ENO сбрасывается в 0, и выход STATUS указывает код состояния.

Таблица 10- 8 Коды состояния выполнения

STATUS (W#16#)	Описание
0	Без ошибок
80A1	Идентификатор не адресует в/с счетчик
80D0	SBF 124 недоступен
80B5	Недопустимое значение для NewPeriod

10.1.3. Работа высокоскоростного счетчика

Высокоскоростные счетчики (HSC) могут считать события, которые происходят быстрее, чем скорость выполнения циклического ОБ. Если события, которые будут подсчитываться, происходят медленнее, чем скорость выполнения ОБ, Вы можете использовать стандартные CTU, CTD или CTUD счетные инструкции. Если события происходят быстрее, чем скорость выполнения ОБ, используйте более быстрое устройство в/с счета. Инструкция CTRL_HSC позволяет Вашей программе изменять некоторые параметры в/с счетчика.

Например: Вы можете использовать в/с счетчик в качестве входа для инкрементного датчика положения вала. Датчик положения вала обеспечивает определенное количество импульсов на оборот и импульс сброса, который происходит один раз за оборот. Синхроимпульс(-ы) и импульс сброса от датчика положения вала являются входами для в/с счетчика.

В в/с счетчик загружается первая из нескольких предварительных установок, и выходы активируются в течение времени, когда текущее значение меньше, чем текущая предварительная установка. В/с счетчик обеспечивает прерывание, когда текущее количество равно предварительной установке, когда происходит сброс, а также когда выполняется изменение направления.

Как только происходит каждое событие прерывания по равенству текущего значения предварительной установке, новая предварительная установка загружается, и устанавливается следующее состояние для выходов. Когда имеет место событие прерывания сброса, первая предварительная установка и первые состояния вывода устанавливаются, и цикл повторяется.

Так как прерывания происходят гораздо реже, чем скорость счета, точное управление высокоскоростными операциями может быть реализовано с относительно незначительным влиянием на цикл сканирования ЦПУ. Метод привязки к прерыванию позволяет каждой загрузке новой предварительной установки быть выполненной в отдельной подпрограмме прерывания для простого контроля состояния. Альтернативно, все события прерывания могут быть обработаны в единственной подпрограмме прерывания.

Выбор входного канала в/с счетчика

Используйте следующую таблицу и гарантируйте, что входные каналы ЦПУ и СП, которые Вы подключаете, могут поддерживать максимальные частоты следования импульсов в Ваших технологических сигналах.

Примечание

Входные каналы ЦПУ и СП (с V4 или более поздней версией ПО) обладают конфигурируемым временем фильтра входа

Более ранние версии ПО обладали фиксированными входными каналами в/с счета и фиксированным временем фильтра, которые не могли быть изменены.

V4 или более поздние версии позволяют Вам назначать время фильтра и входные каналы. Настройка фильтра входа по умолчанию 6.4 мс может быть слишком медленной для Ваших технологических сигналов. Вы должны оптимизировать времена фильтра цифровых входов (стр. 164) для входов в/с счета Вашего приложения.

Таблица 10- 9 Вход ЦПУ: максимальная частота

ЦПУ	Входной канал ЦПУ	1 или 2 фазный режим	А/В квадратурно-фазный режим
1211C	от Ia.0 до Ia.5	100 кГц	80 кГц
1212C	от Ia.0 до Ia.5	100 кГц	80 кГц
	Ia.6, Ia.7	30 кГц	20 кГц
1214C и 1215C	от Ia.0 до Ia.5	100 кГц	80 кГц
	от Ia.6 до Ib.5	30 кГц	20 кГц
1217C	от Ia.0 до Ia.5	100 кГц	80 кГц
	от Ia.6 до Ib.1	30 кГц	20 кГц
	от Ib.2 до Ib.5 (от .2+, .2- до .5+, .5-)	1 МГц	1 МГц

Таблица 10- 10 Вход сигнальной платы (SB): максимальная частота (дополнительная плата)

Сигнальная плата (SB)	Входной канал SB	1 или 2 фазный режим	А/В квадратурно-фазный режим
SB 1221, 200 кГц	от Ie.0 до Ie.3	200 кГц	160 кГц
SB 1223, 200 кГц	Ie.0, Ie.1	200 кГц	160 кГц
SB 1223	Ie.0, Ie.1	30 кГц	20 кГц

Выбор функциональности в/с счетчика

Все в/с счетчики функционируют одинаково в одном и том же режиме счета. Режим счета, управление направлением и начальное направление назначены в конфигурации устройства ЦПУ для свойств функции в/с счета.

Существует четыре основных типа в/с счетчиков:

- Однофазный счетчик с внутренним управлением направлением
- Однофазный счетчик с внешним управлением направлением
- Двухфазный счетчик с 2 входами синхронизации
- Счетчик со сдвигом по фазе на девяносто градусов A/B

Вы можете использовать каждый тип в/с счетчика со входом сброса или без него. Когда Вы активируете вход сброса (с некоторыми ограничениями, см. следующую таблицу), текущее значение сбрасывается и сохраняется таковой, пока Вы не деактивировали вход сброса.

- Функция частоты: Некоторые режимы позволяют конфигурирование в/с счетчика (Тип счета), чтобы сообщать о частоте вместо текущего количества импульсов. Доступны три различных периода измерения частоты: 0.01, 0.1 или 1.0 секунды.

Период измерения частоты определяет, как часто в/с счетчик вычисляет и сообщает новое значение частоты. Выводимая частота является средним значением, определенным по общему количеству импульсов за прошедший период измерения. Если частота быстро изменяется, то выводимое значение будет промежуточным значением между наибольшей и наименьшей частотами, имеющими место в течение периода измерения. Частота всегда выводится в Герцах (импульсах в секунду) независимо от настройки периода измерения частоты.

- Режимы счета и входы: следующая таблица показывает входы, используемые для синхронизации, управления направлением и функций сброса, связанных с в/с счетчиком.
- Функция измерения периода: измерение периода выполняется по сконфигурированному измерительному интервалу (10 мс, 100 мс или 1000 мс). Системный тип данных HSC_Period возвращает результаты измерения периода и обеспечивает измерения в качестве двух значений: ElapsedTime и EdgeCount. Входы в/с счетчика от ID1000 до ID1020 не используются в измерении периода:
 - ElapsedTime является двойным целым значением без знака в наносекундах, представляя время от первого до последнего события счета в измерительном интервале. Если EdgeCount = 0, то ElapsedTime представляет время, начиная с последнего события счета в предшествующем интервале. ElapsedTime обладает диапазоном от 0 до 4,294,967,280 нс (от 0x0000 0000 до 0xFFFF FFF0). Переполнение обозначено значением 4,294,967,295 (0xFFFF FFFF). Значения от 0xFFFF FFF1 до 0xFFFF FFFE зарезервированы.
 - EdgeCount - двойное целое значение без знака, представляющее число событий счета в измерительном интервале.

Один и тот же вход не может использоваться для двух различных функций, но любой вход, не используемый существующим режимом его в/с счетчика может использоваться для другой цели. Например, если HSC1 находится в режиме, который использует два встроенных входа, но не использует третий внешний вход сброса (назначение по умолчанию I0.3), тогда I0.3 может использоваться для прерываний по фронту или для HSC 2.

Таблица 10- 11 Режимы счета для в/с счетчика

Тип	Вход 1	Вход 2	Вход 3	Функция
Однофазный счетчик с внутренним управлением направлением	Синхроимпульс	-	-	Подсчет частоты
			Сброс	Счет
Однофазный счетчик с внешним управлением направлением	Синхроимпульс	Направление	-	Подсчет частоты
			Сброс	Счет
Двухфазный счетчик с 2 входами синхронизации	Синхроимпульс вверх	Синхроимпульс вниз	-	Подсчет частоты
			Сброс	Счет
Квадратурный A/B-фазный счетчик	Фаза А	Фаза В	-	Подсчет частоты
			Сброс ¹	Счет

¹ Для энкодера: Фаза Z, Начальное положение

Адреса входов для в/с счетчиков

Когда Вы конфигурируете ЦПУ, у Вас есть опция разрешить и сконфигурировать "Аппаратные входы" для каждого в/с счетчика.

Все входы в/с счетчиков должны быть соединены с терминалами на модуле ЦПУ или дополнительной сигнальной плате, которая вставляется в переднюю часть модуля ЦПУ.

Примечание

Как показано в последующих таблицах, назначения по умолчанию для дополнительных сигналов для различных в/с счетчиков перекрываются. Например, дополнительный внешний сброс для HSC 1 использует тот же вход, что используется в качестве одного из входов для HSC 2.

Для ЦПУ V4 или более поздней, Вы можете переназначить входы в/с счетчика во время конфигурирования ЦПУ. Вы не обязаны использовать входные назначения по умолчанию.

Всегда гарантируйте, что Вы сконфигурировали свои счетчики так, чтобы любой из входов не использовался двумя в/с счетчиками.

Следующие таблицы показывают назначения входов в/с счетчиков по умолчанию для встроенного ввода-вывода ЦПУ и дополнительной SB (Если у выбранной модели SB есть только 2 входа, доступны только входы 4.0 и 4.1)

Определения таблицы входов в/с счетчиков

- **Однофазный:** С является входом **синхроимпульса**, [d] является входом (дополнительным) **направления**, а [R] является входом (дополнительным) **внешнего сброса** (Сброс доступен только для режима "Счет")
- **Двухфазный:** CU является входом **синхроимпульса вверх**, CD является входом **синхроимпульса вниз**, а [[R] является входом (дополнительным) **внешнего сброса** (Сброс доступен только для режима "Счет")
- **AB-фазный со сдвигом на 90°:** А является входом **синхроимпульса А**, В является входом **синхроимпульса В**, а [[R] является входом (дополнительным) **внешнего сброса** (Сброс доступен только для режима "Счет")

Таблица 10- 12 ЦПУ 1211С: Назначения адресов по умолчанию для в/с счетчика

Режим в/с счетчика		Встроенный вход ЦПУ (по умолчанию 0.x)						Дополнительный вход SB (по умолчанию 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
HSC 1	1-фазный	C	[d]		[R]			C	[d]		[R]
	2-фазный	CU	CD		[R]			CU	CD		[R]
	AB-фазный	A	B		[R]			A	B		[R]
HSC 2	1-фазный		[R]	C	[d]				[R]	C	[d]
	2-фазный		[R]	CU	CD				[R]	CU	CD
	AB-фазный		[R]	A	B				[R]	A	B
HSC 3	1-фазный					C	[d]	C	[d]		R]
	2-фазный										
	AB-фазный										
HSC4	1-фазный					C	[d]	C	[d]		R]
	2-фазный					CU	CD				
	AB-фазный					A	B				
HSC 5	1-фазный							C	[d]		[R]
	2-фазный							CU	CD		[R]
	AB-фазный							A	B		[R]
HSC 6	1-фазный								[R]	C	[d]
	2-фазный								[R]	CU	CD
	AB-фазный								[R]	A	B

¹ SB только с 2 цифровыми входами обеспечивает только входы 4.0 и 4.1.

Таблица 10- 13 ЦПУ 1212С: Назначения адресов по умолчанию для высокоскоростных счетчиков

Режим в/с счетчика		Встроенный вход ЦПУ (по умолчанию 0.x)								Дополнительный вход SB (по умол- чанию 4.x) ¹			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1	1-фазный	C	[d]		[R]					C	[d]		[R]
	2-фазный	CU	CD		[R]					CU	CD		[R]
	AB-фазный	A	B		[R]					A	B		[R]
HSC 2	1-фазный		[R]	C	[d]						[R]	C	[d]
	2-фазный		[R]	CU	CD						[R]	CU	CD
	AB-фазный		[R]	A	B						[R]	A	B
HSC 3	1-фазный					C	[d]		[R]	C	[d]		[R]
	2-фазный					CU	CD		[R]				
	AB-фазный					A	B		[R]				
HSC 4	1-фазный						[R]	C	[d]	C	[d]		[R]
	2-фазный						[R]	CU	CD				
	AB-фазный						[R]	A	B				

Режим в/с счетчика		Встроенный вход ЦПУ (по умолчанию 0.x)								Дополнительный вход SB (по умолчанию 4.x) ¹			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 5	1-фазный									C	[d]		[R]
	2-фазный									CU	CD		[R]
	AB-фазный									A	B		[R]
HSC 6	1-фазный										[R]	C	[d]
	2-фазный										[R]	CU	CD
	AB-фазный										[R]	A	B

¹ SB только с 2 цифровыми входами обеспечивает только входы 4.0 и 4.1.

Таблица 10- 14 ЦПУ 1214С, ЦПУ 1215С и ЦПУ 1217С:

Назначения адресов по умолчанию для высокоскоростных счетчиков
(только встроенные входы, смотри следующую таблицу для адресов дополнительных SB)

Режим в/с счетчика		Цифровой входной байт 0 (по умолчанию 0.x)								Цифровой входной байт 1 (по умолчанию: 1.x)					
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1	1-фазный	C	[d]		[R]										
	2-фазный	CU	CD		[R]										
	AB-фазный	A	B		[R]										
HSC 2	1-фазный		[R]	C	[d]										
	2-фазный		[R]	CU	CD										
	AB-фазный		[R]	A	B										
HSC 3	1-фазный					C	[d]		[R]						
	2-фазный					CU	CD		[R]						
	AB-фазный					A	B		[R]						
HSC 4	1-фазный						[R]	C	[d]						
	2-фазный						[R]	CU	CD						
	AB-фазный						[R]	A	B						
HSC 5	1-фазный									C	[d]	[R]			
	2-фазный									CU	CD	[R]			
	AB-фазный									A	B	[R]			
HSC 6	1-фазный												C	[d]	[R]
	2-фазный												CU	CD	[R]
	AB-фазный												A	B	[R]

Таблица 10- 15 Дополнительная SB в ЦПУ из вышеупомянутой таблицы: назначение адреса по умолчанию для в/с счетчика

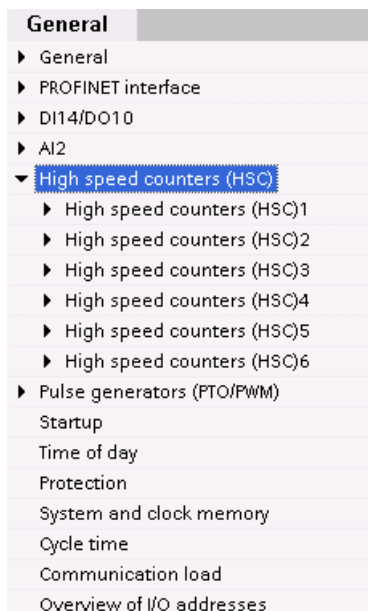
В/с счетчик		Входы дополнительной SB (по умолчанию: 4.x) ¹			
		0	1	2	3
HSC 1	1-фазный	C	[d]		[R]
	2-фазный	CU	CD		[R]
	AB-фазный	A	B		[R]
HSC 2	1-фазный		[R]	C	[d]
	2-фазный		[R]	CU	CD
	AB-фазный		[R]	A	B
HSC 5	1-фазный	C	[d]		[R]
	2-фазный	CU	CD		[R]
	AB-фазный	A	B		[R]
HSC 6	1-фазный		[R]	C	[d]
	2-фазный		[R]	CU	CD
	AB-фазный		[R]	A	B

¹ SB только с 2 цифровыми входами обеспечивает только входы 4.0 и 4.1.

Примечание

Цифровые каналы ввода-вывода, используемые высокоскоростными счетными устройствами, назначаются во время конфигурации устройства ЦПУ. Когда адреса цифровых каналов ввода-вывода назначаются в/с счетным устройствам, значения назначенных адресов каналов ввода-вывода не могут быть изменены функцией форсирования в таблице наблюдения.

10.1.4. Конфигурирование высокоскоростного счетчика



Вы можете сконфигурировать до 6 высокоскоростных счетчиков. Отредактируйте конфигурацию устройства ЦПУ и назначьте свойства каждого отдельного в/с счетчика.

Включите в/с счетчик, выбрав опцию "Enable" для него.

Используйте инструкции CTRL_HSC и/или CTRL_HSC_EXT в Вашей пользовательской программе, чтобы управлять работой в/с счетчика.



ПРЕДУПРЕЖДЕНИЕ

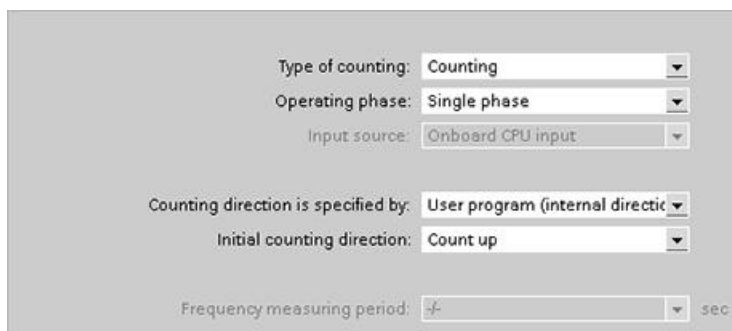
Риски, связанные с изменением настройки времени фильтра для цифровых входных каналов

Если предыдущая настройка времени фильтра для цифрового входного канала изменена, то, возможно, новое входное значение "0" уровня должно было бы быть представлено в течение накопленного времени до максимум 20.0 мс, прежде чем фильтр станет полностью чувствительным к новым входным значениям. В течение этого времени короткие "0" импульсные события длительностью меньше 20.0 мс не могут обнаруживаться или подсчитываться.

Это изменение времен фильтра может привести к неожиданной работе машины или процесса, что может вызвать смерть или серьезную травму персонала, и/или повредить оборудование.

Чтобы гарантировать, что новое время фильтра сразу вступает в силу, выполните цикл выключения и включения питания ЦПУ.

После включения в/с счетчика сконфигурируйте другие параметры, такие как функция счета, начальные значения, опции сброса и события прерывания.



The image shows a configuration window for a high-speed counter. It contains several dropdown menus and a text input field. The settings are as follows:

- Type of counting: Counting
- Operating phase: Single phase
- Input source: Onboard CPU input
- Counting direction is specified by: User program (internal directio
- Initial counting direction: Count up
- Frequency measuring period: 1 sec

Для получения дополнительной информации о конфигурировании в/с счетчика обратитесь к разделу по конфигурированию ЦПУ (стр. 162).

10.2 ПИД-регулирование

STEP 7 предлагает следующие ПИД-инструкции для S7-1200 ЦПУ:

- Инструкция PID_Compact используется, чтобы управлять техническими процессами с непрерывными переменными ввода и вывода.
- Инструкция PID_3Step используется, чтобы управлять приводимыми в действие двигателями устройствами, такими как клапаны, которые требуют дискретных сигналов для открытия и закрытия.
- Инструкция PID_Temp обеспечивает универсальный ПИД-регулятор, который позволяет обрабатывать специфические требования управления по температуре.

Примечание

Изменения, которые Вы вносите в конфигурацию ПИД и загружаете в режиме RUN, не вступают в силу до перехода ЦПУ из режима STOP в режим RUN. Изменения, которые Вы вносите в диалоговом окне "PID parameters", используя "Start value control" вступают в силу незамедлительно.

Все три ПИД-инструкции (PID_Compact, PID_3Step и PID_Temp) могут вычислить P-, I- и D-компоненты во время запуска (если сконфигурированы для "предварительной настройки"). Вы можете также сконфигурировать инструкцию для "точной настройки", чтобы позволить Вам оптимизировать параметры. Вы не должны определять параметры вручную.

Примечание

Выполняйте ПИД-инструкцию с постоянным интервалом времени (предпочтительно в циклическом ОВ).

Поскольку ПИД-контур требует определенное время, чтобы отреагировать на изменения значения управления, не вычисляйте выходное значение в каждом цикле. Не выполняйте ПИД-инструкцию в ОВ основного программного цикла (таком как ОВ 1).

Время выборки для ПИД-алгоритма представляет собой время между двумя вычислениями выходного значения (значение управления). Выходное значение вычисляется во время самонастройки и округляется до значения, кратного времени цикла. Все другие функции ПИД-инструкции выполняются в каждом вызове.

ПИД-алгоритм

ПИД (Пропорциональный/Интегральный/Дифференциальный) регулятор измеряет временной интервал между двумя вызовами и затем оценивает результаты для контроля времени выборки. Среднее значение времени выборки генерируется при каждом переключении режима и во время начального запуска. Это значение используется в качестве опорного для контролирующей функции и используется для вычисления. Контроль включает в себя текущее время измерения между двумя вызовами и среднее значение определенного времени выборки контроллера.

Выходное значение для ПИД-регулятора состоит из трех компонентов:

- Р (пропорциональный): При расчете с компонентом "Р", выходное значение пропорционально различию между уставкой и значением процесса (входное значение).
- I (интегральный): При расчете с компонентом "I", выходное значение увеличивается в пропорции к продолжительности различия между уставкой и значением процесса (входное значение), чтобы в заключение исправить различие.
- D (дифференциальный): При расчете с компонентом "D", выходное значение увеличивается как функция скорости увеличения различия между уставкой и значением процесса (входное значение). Выходное значение корректируется к уставке как можно быстрее.

ПИД-регулятор использует следующую формулу, чтобы рассчитать выходное значение для инструкции PID_Compact.

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Выходное значение	x	Значение процесса
w	Значение уставки	s	Оператор Лапласа
K _p	Пропорциональное усиление (Р-компонент)	a	Коэффициент задержки дифференцирования (D-компонент)
T _i	Время интегрирования (I-компонент)	b	Весовой коэффициент пропорционального действия (Р-компонент)
T _d	Время упреждения ()	c	Весовой коэффициент дифференциального действия (D-компонент)

ПИД-регулятор использует следующую формулу, чтобы рассчитать выходное значение для инструкции PID_3Step.

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Выходное значение	x	Значение процесса
w	Значение уставки	s	Оператор Лапласа
K _p	Пропорциональное усиление (Р-компонент)	a	Коэффициент задержки дифференцирования (D-компонент)
T _i	Время интегрирования (I-компонент)	b	Весовой коэффициент пропорционального действия (Р-компонент)
T _d	Время упреждения ()	c	Весовой коэффициент дифференциального действия (D-компонент)

10.2.1. Вставка ПИД-инструкции и технологического объекта

STEP 7 предлагает две инструкции для ПИД-управления:

- Инструкция PID_Compact и ее связанный технологический объект обеспечивают настройку универсального ПИД-контроллера. Технологический объект содержит все настройки для контура управления.
- Инструкция PID_3Step и ее связанный технологический объект обеспечивают определенные настройки ПИД-контроллера для моторизованных клапанов. Технологический объект содержит все настройки для контура управления. Контроллер PID_3Step обеспечивает два дополнительных булевых выхода.

После создания технологического объекта Вы должны сконфигурировать параметры (стр. 505). Вы также корректируете параметры автоматической настройки ("предварительная настройка" во время запуска или ручная "точная настройка"), чтобы ввести в эксплуатацию ПИД-контроллер (стр. 522).

Таблица 10- 16 Вставка ПИД-инструкции и технологического объекта

Когда Вы вставляете ПИД-инструкцию в свою пользовательскую программу, STEP 7 автоматически создает технологический объект и экземплярный DB для инструкции. Экземплярный DB содержит все параметры, которые используются ПИД-инструкцией. Каждая ПИД-инструкция для надлежащей работы должна обладать своим собственным уникальным экземпляром DB.

После вставки ПИД-инструкции и создания технологического объекта и экземплярного DB, Вы конфигурируете параметры для технологического объекта (стр. 505).

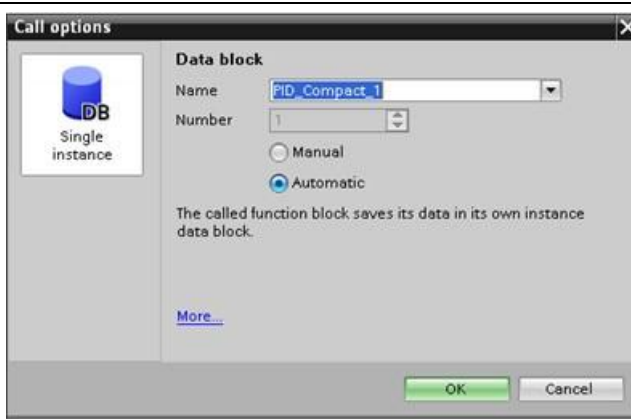
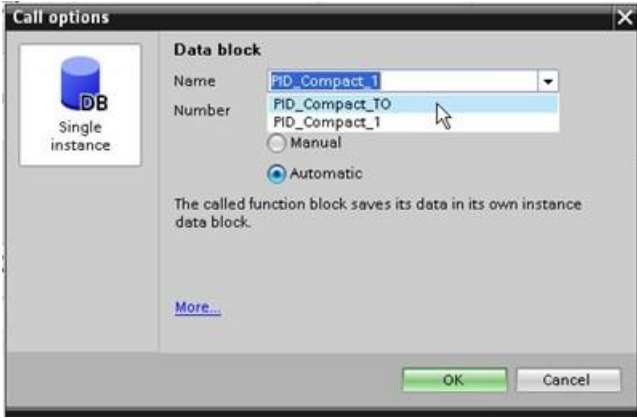
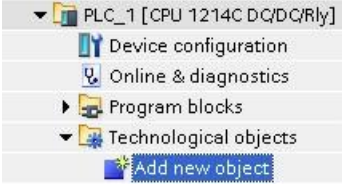
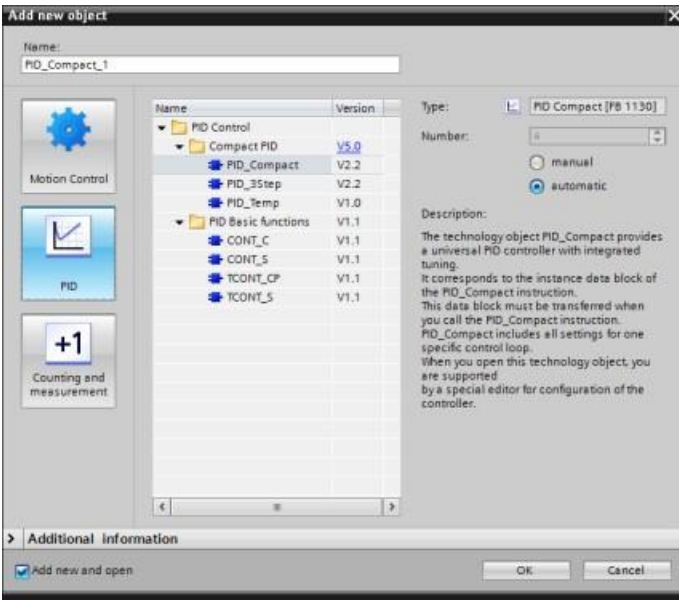


Таблица 10- 17 (Дополнительно) Создание технологического объекта из навигатора проекта

Вы можете также создать технологические объекты для своего проекта прежде, чем вставлять ПИД-инструкцию. Создавая технологический объект до вставки ПИД-инструкции в Вашу пользовательскую программу, Вы можете выбирать технологический объект, когда Вы вставляете ПИД-инструкцию.	
Чтобы создать технологический объект, дважды щелкните по значку "Add new object" в навигаторе проекта.	
Щелкните по значку "Control" и выберите технологический объект для типа ПИД-регулятора (PID_Compect или PID_3Step). Вы можете создать дополнительное имя для технологического объекта. Нажмите "OK", чтобы создать технологический объект.	

10.2.2. Инструкция PID_Compact

Инструкция PID_Compact предоставляет универсальный ПИД-регулятор с интегрированной самонастройкой для автоматического и ручного режима.

Таблица 10- 18 Инструкция PID_Compact

LAD / FBD	SCL	Описание
%DB2 "PID_Compact_1" PID_Compact EN Setpoint Input Input_PER Disturbance ManualEnable ManualValue ErrorAck Reset ModeActivate Mode ENO ScaledInput Output Output_PER Output_PWM SetpointLimit_H SetpointLimit_L InputWarning_H InputWarning_L State Error ErrorBits	<pre>"PID_Compact_1"(Setpoint:=_real_in_ Input:=_real_in_, In- put_PER:=_word_in_ Disturbance:=_real_in_ ManualEnable:=_bool_in_ ManualValue:=_real_in_ ErrorAck:=_bool_in_ Reset:=_bool_in_ ModeActivate:=_bool_in_ Mode:=_int_in_ ScaledInput=>_real_out_ Output=>_real_out_ Output_PER=>_word_out_ Output_PWM=>_bool_out_ SetpointLimit_H=>_bool_out_ SetpointLimit_L=>_bool_out_ InputWarn- ing_H=>_bool_out_ InputWarn- ing_L=>_bool_out_ State=>_int_out_ Error=>_bool_out_ ErrorBits=>_dword_out_);</pre>	PID_Compact предоставляет универсаль- ный ПИД-регулятор с интегрированной самонастройкой для автоматического и ручного режима. PID_Compact представ- ляет собой PID T1 регулятор с предот- вращением интегрального насыщения и выбором весовых коэффициентов Р-и D- компонента.

- 1

STEP 7 автоматически создает технологический объект и экземплярный DB, когда Вы вставляете инструкцию. Экземплярный DB содержит параметры технологического объекта.
- 2

В примере на SCL, "PID_Compact_1" является именем экземплярного DB.

Таблица 10- 19 Типы данных для параметров

Параметр и тип		Тип данных	Описание
Setpoint	IN	Real	Уставка ПИД-регулятора в автоматическом режиме. (Значение по умолчанию: 0.0)
Input	IN	Real	Тег пользовательской программы используется в качестве источника значения процесса. (Значение по умолчанию: 0.0) Если Вы используете параметр Input, Вы должны установить Config.InputPerOn = FALSE.
Input_PER	IN	Word	Аналоговый вход используется в качестве источника значения процесса. (Значение по умолчанию: W#16#0) Если Вы используете параметр Input_PER, Вы должны установить Config.InputPerOn = TRUE.
Disturbance	IN	Real	Переменная возмущения или значение пред-управления
ManualEnable	IN	Bool	Включает или отключает ручной режим работы. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE активирует "ручной режим", при этом если State = 4, режим остается неизменным. Пока ManualEnable = TRUE, Вы не можете изменить рабочий режим, используя нарастающий фронт в ModeActivate или использовать диалоговое окно ввода в эксплуатацию. - Фронт от TRUE к FALSE активирует рабочий режим, который назначен посредством Mode. Примечание: Мы рекомендуем, чтобы Вы изменяли рабочий режим, только используя ModeActivate.
ManualValue	IN	Real	Выходное значение для ручного режима. (Значение по умолчанию: 0.0) Вы можете использовать значения от Config.OutputLowerLimit до Config.OutputUpperLimit.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выходы предупреждения. Фронт от FALSE к TRUE.
Reset	IN	Bool	Перезапускает регулятор. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE: - Переключается в "неактивный" режим - Сбрасывает ErrorBits и выходы предупреждения - Очищает интегральное действие - Сохраняет параметры ПИД - Пока Reset = TRUE, PID_Compact остается в режиме "неактивный" (State = 0). - Фронт от TRUE к FALSE: - PID_Compact переключается в рабочий режим, который сохранен в параметре Mode.
ModeActivate	IN	Bool	PID_Compact переключается в рабочий режим, который сохранен в параметре Mode. Фронт от FALSE к TRUE:
Mode	IN	Int	Желаемый режим ПИД; Активируется по нарастающему фронту на входе ModeActivate.
ScaledInput	OUT	Real	Приведенное значение процесса. (Значение по умолчанию: 0.0)
Output ¹	OUT	Real	Выходное значение в формате REAL. (Значение по умолчанию: 0.0)
Output_PER ¹	OUT	Word	Аналоговое выходное значение. (Значение по умолчанию: W#16#0)

Параметр и тип		Тип данных	Описание
Output_PWM ¹	OUT	Bool	Выходное значение для широтно-импульсной модуляции. (Значение по умолчанию: FALSE) Времена включения и выключения формируют выходное значение.
SetpointLimit_H	OUT	Bool	Верхний предел уставки. (Значение по умолчанию: FALSE) Если SetpointLimit_H = TRUE, абсолютный верхний предел уставки достигнут (Setpoint ≥ Config.SetpointUpperLimit). Уставка ограничена значением Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Нижний предел уставки. (Значение по умолчанию: FALSE) Если SetpointLimit_L = TRUE, абсолютный нижний предел уставки достигнут (Setpoint ≤ Config.SetpointLowerLimit). Уставка ограничена значением Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Если InputWarning_H = TRUE, значение процесса достигло или превысило верхний предупреждающий предел. (Значение по умолчанию: FALSE)
InputWarning_L	OUT	Bool	Если InputWarning_L = TRUE, значение процесса достигло или опустилось ниже нижнего предупреждающего предела. (Значение по умолчанию: FALSE)
State	OUT	Int	Текущий рабочий режим ПИД-регулятора. (По умолчанию: 0) Вы можете изменить рабочий режим, используя входной параметр Mode и нарастающий фронт в ModeActivate: - State = 0: Неактивный - State = 1: Предварительная настройка - State = 2: Ручная точная настройка - State = 3: Автоматический режим - State = 4: Ручной режим - State = 5: Подстановка выходного значения с контролем ошибок
Error	OUT	Bool	Если Error = TRUE, по крайней мере одно сообщение об ошибке присутствует в этом цикле. (Значение по умолчанию: FALSE) Примечание: Параметр Error в ПИД V1.x был полем ErrorBits, которое содержало коды ошибок. Теперь это булевый флаг, указывающий на то, что ошибка произошла.
ErrorBits	OUT	DWord	Таблица параметров ErrorBits инструкции PID_Compact (стр. 480) определяет сообщения об имеющихся ошибках. (Значение по умолчанию: DW#16#0000 (ошибок нет)). ErrorBits сохраняемый и сбрасывается по нарастающему фронту на Reset или ErrorAck. Примечание: В V1.x, параметр ErrorBits был определен как Error и отсутствовал.

¹ Вы можете использовать выходные значения параметров Output, Output_PER и Output_PWM параллельно.

Работа регулятора PID_Compact

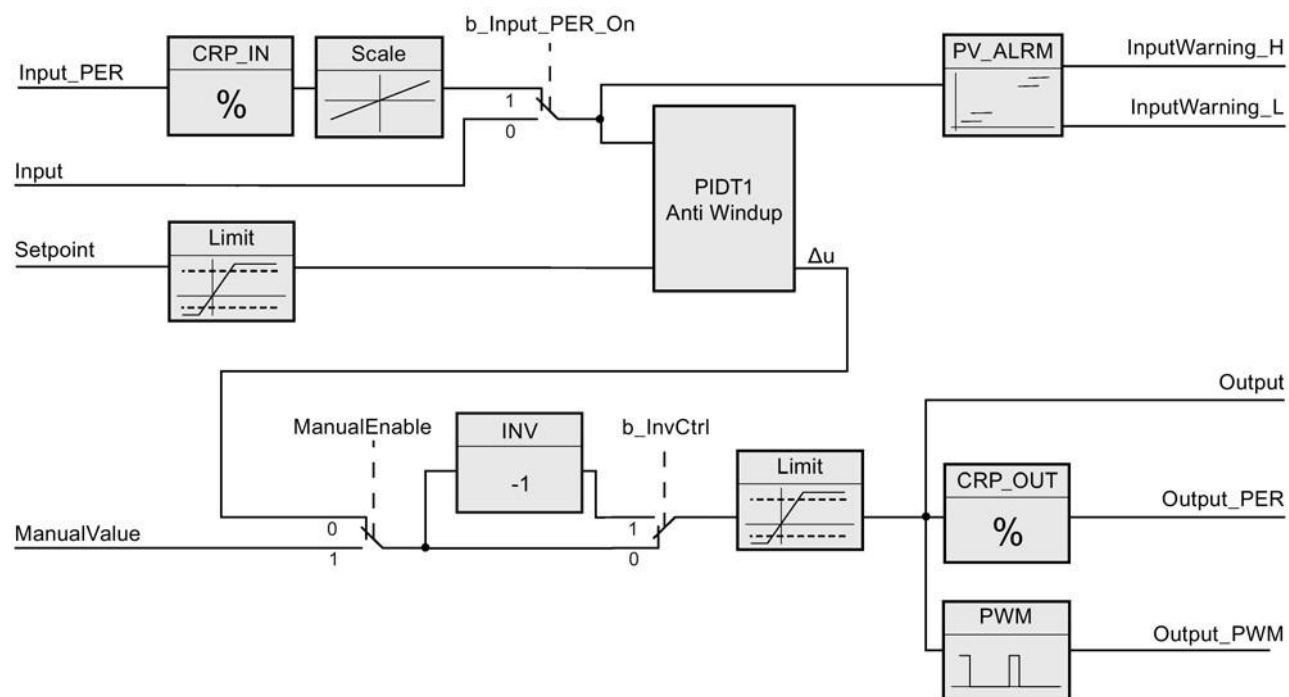


Рисунок 10-1 Работа регулятора PID_Compact

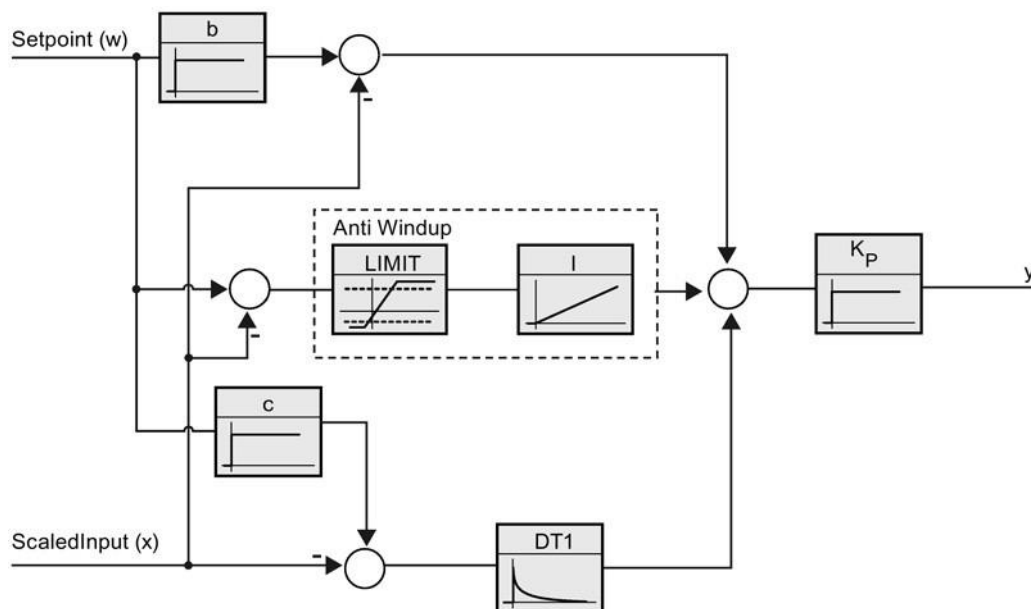


Рисунок 10-2 Работа регулятора PID_Compact, как PIDT1 регулятора с предотвращением интегрального насыщения

10.2.3. Параметр ErrorBit инструкции PID_Compact

Если присутствует несколько ошибок, то значения кодов ошибок выводятся посредством сложения в двоичном коде. Отображение кода ошибки 0003, например, указывает, на то, что ошибки 0001 и 0002 также активны.

Таблица 10- 20 Параметры ErrorBit инструкции PID_Compact

ErrorBit (DW#16#...)	Описание
0000	Ошибок нет
0001 ^{1, 2}	Параметр Input вне пределов значения процесса. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Недопустимое значение в параметре Input_PER. Проверьте наличие ошибки на аналоговом входе.
0004 ⁴	Ошибка во время точной настройки. Колебания значения процесса нельзя скомпенсировать.
0008 ⁴	Ошибка в начале предварительной настройки. Значение процесса слишком близко к уставке. Запустите точную настройку.
0010 ⁴	Уставка была изменена во время настройки. Примечание: Вы можете установить разрешенное колебание уставки в теге CancelTuningLevel.
0020	Предварительная настройка недопустима во время точной настройки. Примечание: Если ActivateRecoverMode = TRUE до возникновения ошибки, PID_Compact остается в режиме точной настройки.
0080 ⁴	Ошибка во время предварительной настройки. Неправильная конфигурация пределов выходного значения. Проверьте, правильно ли сконфигурированы пределы выходного значения и согласуйте управляющую логику.
0100 ⁴	Ошибка во время точной настройки привела к недопустимым параметрам.
0200 ^{2, 3}	Недопустимое значение в параметре Input: Значение имеет неверный числовой формат.
0400 ^{2, 3}	Расчет выходного значения не удался. Проверьте ПИД-параметры.
0800 ^{1, 2}	Ошибка времени выборки: PID_Compact не вызывается в течение времени выборки ОВ циклического прерывания.
1000 ^{2, 3}	Недопустимое значение в параметре Setpoint: Значение имеет неверный числовой формат.
10000	Недопустимое значение в параметре ManualValue: Значение имеет неверный числовой формат. Примечание: Если ActivateRecoverMode = TRUE до возникновения ошибки, PID_Compact использует SubstituteOutput в качестве выходного значения. Как только Вы назначаете допустимое значение в параметре ManualValue, PID_Compact использует его в качестве выходного значения.

ErrorBit (DW#16#...)	Описание
20000	Недопустимое значение в tere SubstituteValue: Значение имеет неверный числовой формат. PID_Compact использует нижний предел выходного значения в качестве выходного значения. Примечание: Если автоматический режим был активен, прежде чем ошибка произошла, ActivateRecoverMode = TRUE, и ошибка больше не рассматривается, PID_Compact переключается назад в автоматический режим.
40000	Недопустимое значение в параметре Disturbance: Значение имеет неверный числовой формат. Примечание: Если автоматический режим был активен и ActivateRecoverMode = FALSE, прежде чем возникла ошибка, Disturbance сбрасывается в ноль. PID_Compact остается в автоматическом режиме. Примечание: Если режимы предварительной настройки или точной настройки были активны и ActivateRecoverMode = TRUE до возникновения ошибки, PID_Compact переключается в рабочий режим, который сохранен в параметре Mode. Если Disturbance на текущем этапе не имеет никакого влияния на выходное значение, настройка не отменяется.

- ¹ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_Compact остается в автоматическом режиме.
- ² Примечание: Если режимы предварительной настройки или точной настройки были активны, прежде чем произошла ошибка, и ActivateRecoverMode = TRUE, PID_Compact переключается в рабочий режим, который сохранен в параметре Mode.
- ³ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_Compact выдает сконфигурированное подстановочное выходное значение. Как только ошибка перестает быть активной, PID_Compact переключается назад в автоматический режим.
- ⁴ Примечание: Если ActivateRecoverMode = TRUE прежде, чем возникла ошибка, PID_Compact отменяет настройку и переключается в рабочий режим, который сохранен в параметре Mode.

10.2.4. Параметр Warning инструкции PID_Compact

Если у ПИД-регулятора присутствует несколько предупреждений, он выводит значения кодов посредством сложения в двоичной системе. Отображение кода 0003, например, указывает, что коды 0001 и 0002 также активны.

Таблица 10- 21 Параметр Warning инструкции PID_Compact

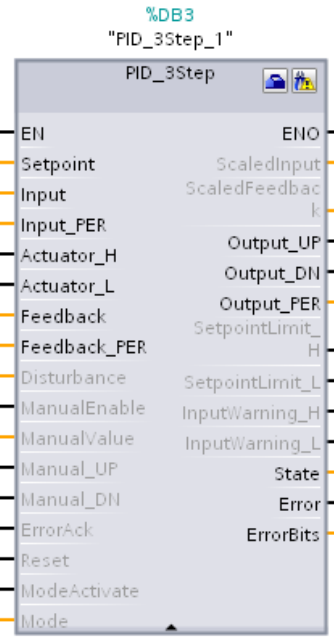
Warning (DW#16#...)	Описание
0000	Нет активных предупреждений.
0001 ¹	Точка перегиба не была найдена во время предварительной настройки.
0002	Качение было предпринято во время "настройки в режиме выполнения". (Параметр "Warning" подавляет это предупреждение, и это видно только в параметре "WarningInternal" в диагностических целях.)
0004 ¹	Уставка была ограничена до сконфигурированных пределов.
0008 ¹	Не все необходимые управляемые системные свойства были определены для выбранного метода вычисления. Вместо этого параметры ПИД были рассчитаны, используя метод TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Рабочий режим не мог быть изменен так как Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызова ОВ ограничивает время выборки ПИД-алгоритма. Улучшите результаты посредством более короткого времени цикла ОВ.
0040 ¹	Значение процесса превысило один из своих предупреждающих пределов.
0080	Недопустимое значение в Mode. Рабочий режим не переключается.
0100 ¹	Ручное значение было ограничено пределами выхода контроллера.
0200	Указанное правило для настройки не поддерживается. Никакие параметры ПИД не рассчитаны.
1000	Выходное значение подстановки не может быть достигнуто, так как находится вне пределов выходного значения.

¹ Примечание: ПИД-регулятор удалил следующие предупреждения автоматически, так как причина устранена или действие пользователя повторено с допустимыми параметрами: 0001, 0004, 0008, 0040, и 0100.

10.2.5. Инструкция PID_3Step

Инструкция PID_3Step конфигурирует ПИД-регулятор с возможностями самонастройки, который был оптимизирован для управляемых моторизованными клапанами и исполнительными устройствами.

Таблица 10- 22 Инструкция PID_3Step

LAD / FBD	SCL	Описание
	<pre> "PID_3Step_1"(SetpoInt:=_real_in_, Input:=_real_in_, ManualValue:=_real_in_, Feedback:=_real_in_, InputPer:=_word_in_, FeedbackPer:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualUP:=_bool_in_, ManualDN:=_bool_in_, ActuatorH:=_bool_in_, ActuatorL:=_bool_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, ScaledFeedback=>_real_out_, ErrorBits=>_dword_out_, OutputPer=>_word_out_, State=>_int_out_, OutputUP=>_bool_out_, OutputDN=>_bool_out_, SetpoIntLimitH=>_bool_out_, SetpoIntLimitL=>_bool_out_, InputWarningH=>_bool_out_, InputWarningL=>_bool_out_, Error=>_bool_out_, ErrorBits=>_dword_out_); </pre>	PID_3Step конфигурирует ПИД-регулятор с возможностями самонастройки, который был оптимизирован для моторизованных клапанов и исполнительных устройств. Он обеспечивает два булевых выхода. PID_3Step является PID T1-регулятором с предотвращением интегрального насыщения и выбором весовых коэффициентов P-и D-компонента.

- STEP 7 автоматически создает технологический объект и экземплярный DB, когда Вы вставляете инструкцию. Экземплярный DB содержит параметры технологического объекта.
- В примере на SCL, "PID_3Step_1" является именем экземплярного DB.

Таблица 10- 23 Типы данных для параметров

Параметр и тип		Тип данных	Описание
Setpoint	IN	Real	Уставка ПИД-регулятора в автоматическом режиме. (Значение по умолчанию: 0.0)
Input	IN	Real	Тег пользовательской программы используется в качестве источника значения процесса. (Значение по умолчанию: 0.0) Если Вы используете параметр Input, Вы должны установить Config.InputPerOn = FALSE.
Input_PER	IN	Word	Аналоговый вход используется в качестве источника значения процесса. (Значение по умолчанию: W#16#0) Если Вы используете параметр Input_PER, Вы должны установить Config.InputPerOn = TRUE.
Actuator_H	IN	Bool	Цифровая обратная связь позиции клапана для остановки в верхней точке. Если Actuator_H = TRUE, клапан находится в верхней конечной точке и больше не перемещается в этом направлении. (Значение по умолчанию: FALSE)
Actuator_L	IN	Bool	Цифровая обратная связь позиции клапана для остановки в нижней точке Если Actuator_L = TRUE, клапан находится в нижней конечной точке и больше не перемещается в этом направлении. (Значение по умолчанию: FALSE)
Feedback	IN	Real	Обратная связь позиции клапана. (Значение по умолчанию: 0.0) Если Вы используете параметр Feedback, Вы должны установить Config.FeedbackPerOn = FALSE.
Feedback_PER	IN	Int	Аналоговая обратная связь позиции клапана. (Значение по умолчанию: W#16#0) Если Вы используете параметр Feedback_PER, Вы должны установить Config.FeedbackPerOn = TRUE. Feedback_PER масштабируется на основании на следующих тегов: - Config.FeedbackScaling.LowerPointIn - Config.FeedbackScaling.UpperPointIn - Config.FeedbackScaling.LowerPointOut - Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	Переменная возмущения или значение пред-управления
ManualEnable	IN	Bool	Включает или отключает ручной режим работы. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE активирует "ручной режим", при этом если State = 4, режим остается неизменным. Пока ManualEnable = TRUE, Вы не можете изменить рабочий режим, используя нарастающий фронт в ModeActivate или использовать диалоговое окно ввода в эксплуатацию. - Фронт от TRUE к FALSE активирует рабочий режим, который назначен посредством Mode. Примечание: Мы рекомендуем, чтобы Вы изменяли рабочий режим, только используя ModeActivate.

Параметр и тип		Тип данных	Описание
ManualValue	IN	Real	Значение процесса для ручного режима. (Значение по умолчанию: 0.0) В ручном режиме Вы определяете абсолютное положение клапана. ManualValue оценивается только, если Вы используете OutputPer, или доступна обратная связь позиции.
ManualUP	IN	Bool	- Manual_UP = TRUE: - Клапан открыт, даже если Вы используете Output_PER или обратную связь позиции. Клапан больше не перемещается, если была достигнута верхняя конечная точка. - См. также Config.VirtualActuatorLimit - Manual_UP = FALSE: - Если Вы используете Output_PER или обратную связь позиции, клапан перемещается в ManualValue. В противном случае, клапан больше не перемещается. Примечание: Если Manual_UP и Manual_DN установлены в TRUE одновременно, клапан не перемещается.
ManualDN	IN	Bool	- Manual_DN = TRUE: - Клапан открыт, даже если Вы используете Output_PER или обратную связь позиции. Клапан больше не перемещается, если была достигнута верхняя конечная точка. - См. также Config.VirtualActuatorLimit - Manual_DN = FALSE: - Если Вы используете Output_PER или обратную связь позиции, клапан перемещается в ManualValue. В противном случае, клапан больше не перемещается.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выходы предупреждения. Фронт от FALSE к TRUE.
Reset	IN	Bool	Перезапускает регулятор. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE: - Переключается в "неактивный" режим - Сбрасывает ErrorBits и выходы предупреждения - Очищает интегральное действие - Сохраняет параметры ПИД - Пока Reset = TRUE, PID_3Step остается в режиме "неактивный" (State = 0). - Фронт от TRUE к FALSE: - PID_3Step переключается в рабочий режим, который сохранен в параметре Mode.
ModeActivate	IN	Bool	PID_3Step переключается в рабочий режим, который сохранен в параметре Mode. Фронт от FALSE к TRUE:
Mode	IN	Int	Желаемый режим ПИД; Активируется по нарастающему фронту на входе ModeActivate.
ScaledInput	OUT	Real	Приведенное значение процесса.
ScaledFeedback	OUT	Real	Приведенная обратная связь позиции клапана Примечание: Для привода без обратной связи позиции, позиция привода, выраженная в ScaledFeedback, очень неточна. ScaledFeedback может использоваться в этом случае только для грубой оценки текущей позиции.

Системное руководство, 01/2015, A5E02486680-AN

Работа регулятора PID_3Step

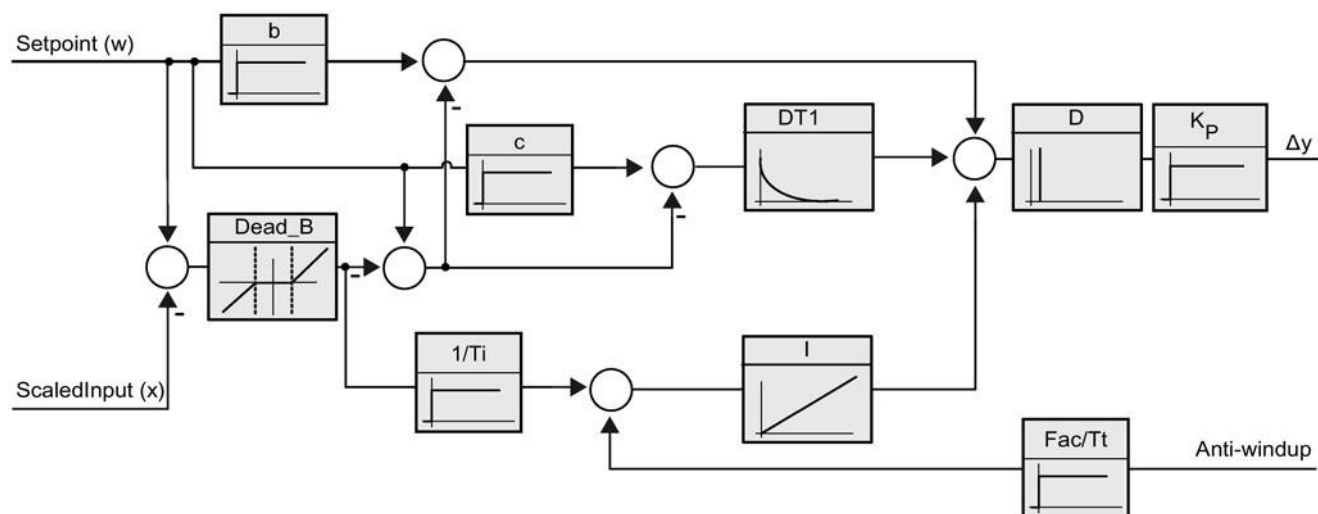


Рисунок 10-3 Работа регулятора PID_3Step, как PIDT1 регулятора с предотвращением интегрального насыщения

у

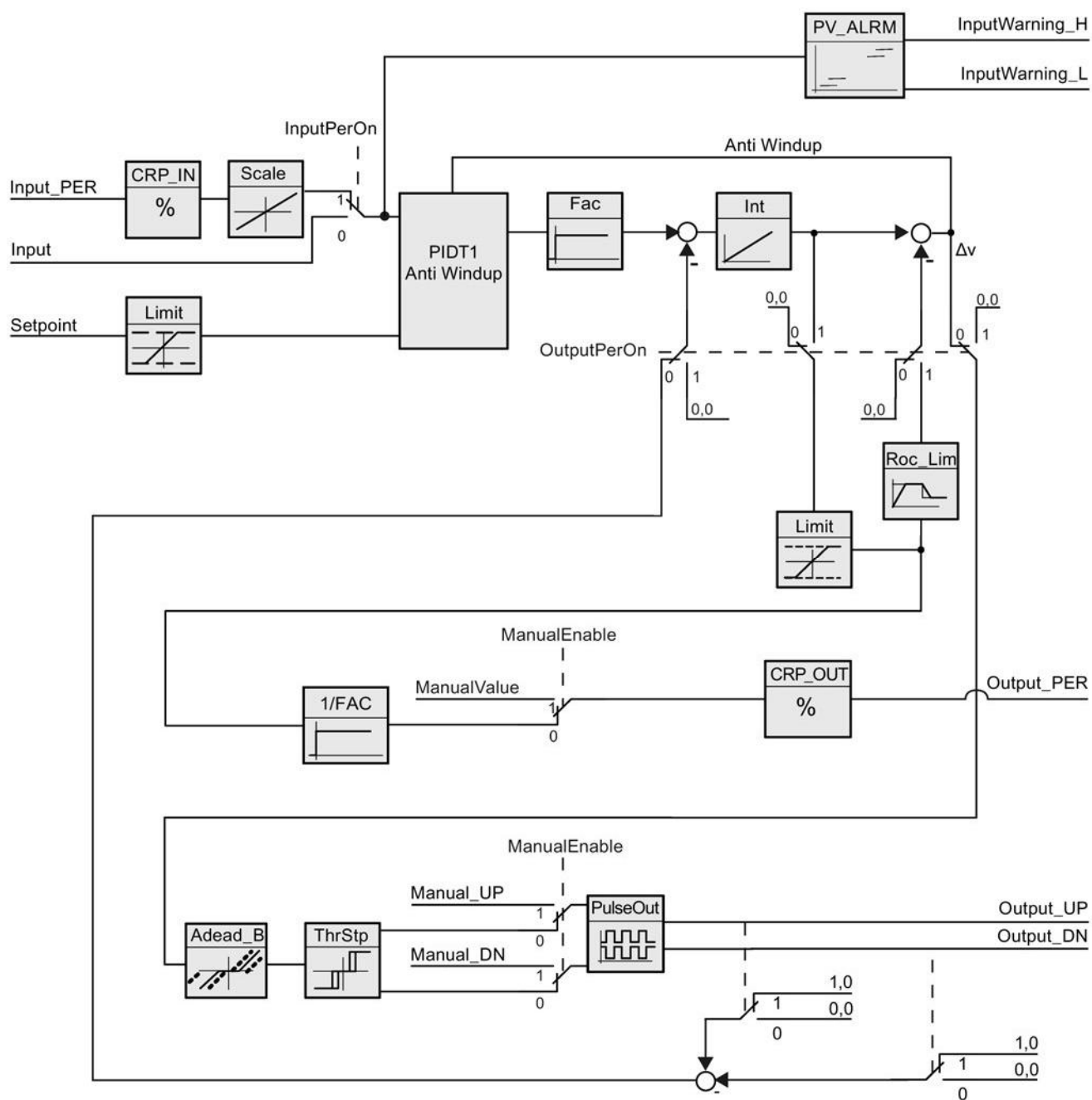


Рисунок 10-4 Работа регулятора PID_3Step без обратной связи позиции

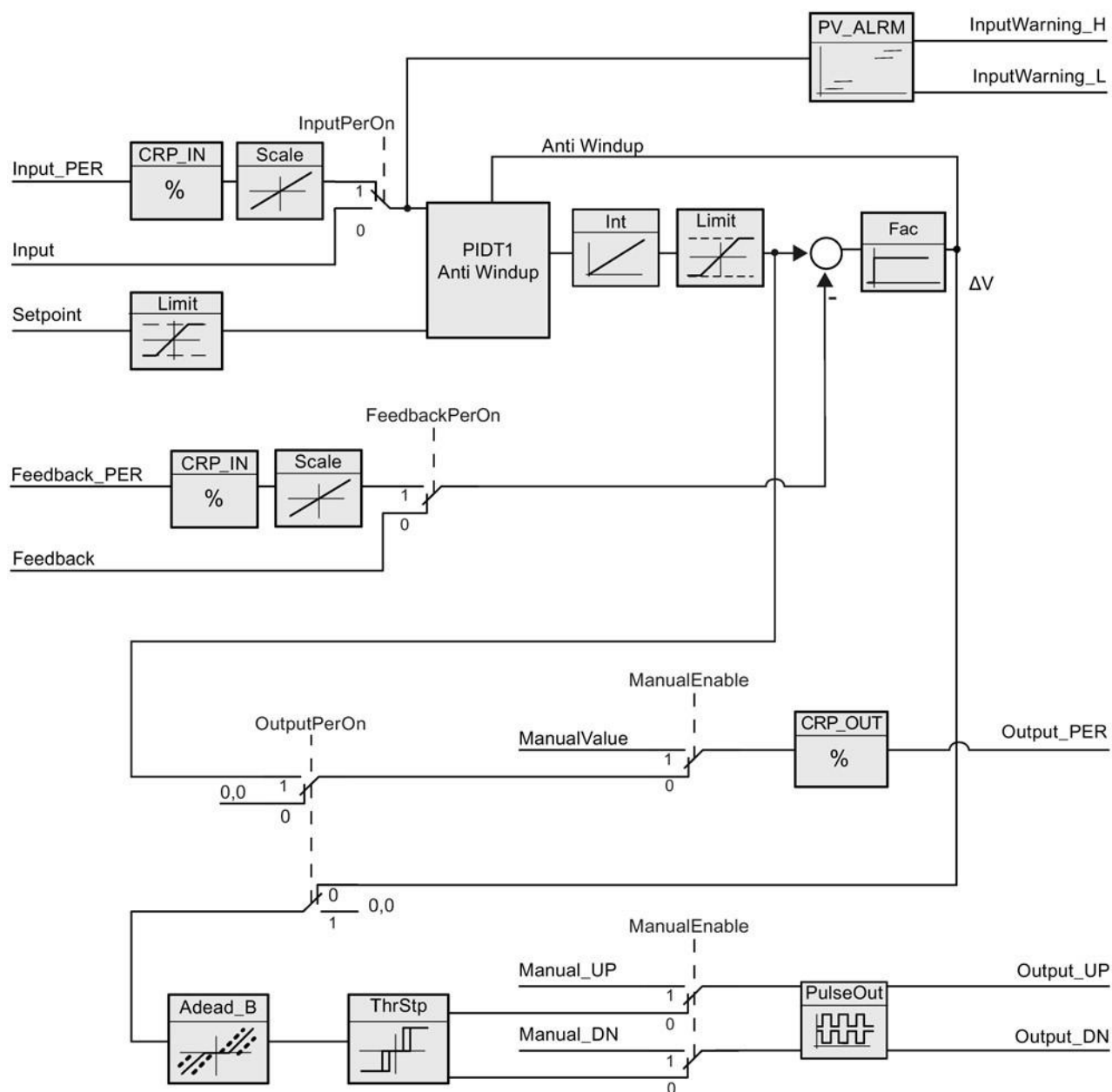


Рисунок 10-5 Работа регулятора PID_3Step с разрешенной обратной связью позиции

10.2.6. Параметр ErrorBit инструкции PID_3Step

Если присутствует несколько ошибок, то значения кодов ошибок выводятся посредством сложения в двоичном коде. Отображение кода ошибки 0003, например, указывает, на то, что ошибки 0001 и 0002 также активны.

Таблица 10- 24 Параметры ErrorBit инструкции PID_3STEP

ErrorBit (DW#16#...)	Описание
0000	Ошибок нет
0001 ^{1, 2}	Параметр Input вне пределов значения процесса. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Недопустимое значение в параметре Input_PER. Проверьте наличие ошибки на аналоговом входе.
0004 ⁴	Ошибка во время точной настройки. Колебания значения процесса нельзя скомпенсировать.
0010 ⁴	Уставка была изменена во время настройки. Примечание: Вы можете установить разрешенное колебание уставки в теге CancelTuningLevel.
0020	Предварительная настройка недопустима во время точной настройки. Примечание: Если ActivateRecoverMode = TRUE до возникновения ошибки, PID_3Step остается в режиме точной настройки.
0080 ⁴	Ошибка во время предварительной настройки. Неправильная конфигурация пределов выходного значения. Проверьте, правильно ли сконфигурированы пределы выходного значения и согласуйте управляющую логику.
0100 ⁴	Ошибка во время точной настройки привела к недопустимым параметрам.
0200 ^{2, 3}	Недопустимое значение в параметре Input: Значение имеет неверный числовой формат.
0400 ^{2, 3}	Расчет выходного значения не удался. Проверьте ПИД-параметры.
0800 ^{1, 2}	Ошибка времени выборки: PID_3Step не вызывается в течение времени выборки ОВ циклического прерывания.
1000 ^{2, 3}	Недопустимое значение в параметре Setpoint: Значение имеет неверный числовой формат.
2000 ^{1, 2, 5}	Недопустимое значение в параметре Feedback_PER. Проверьте наличие ошибки на аналоговом входе.
4000 ^{1, 2, 5}	Недопустимое значение в параметре Feedback: Значение имеет неверный числовой формат..
8000 ^{1, 2}	Ошибка цифровой обратной связи позиции. Actuator_H = TRUE и Actuator_L = TRUE. Привод не может быть перемещен в выходное подстановочное значение и остается в текущей позиции. Ручной режим невозможен в этом состоянии. Чтобы переместить привод из этого состояния, Вы должны деактивировать "Конечное положение привода" (Config.ActuatorEndStopOn = FALSE) или переключиться в ручной режим без сигналов конечных точек (Mode = 10).

ErrorBit (DW#16#...)	Описание
10000	Недопустимое значение в параметре ManualValue: Значение имеет неверный числовой формат. Привод не может быть перемещен в ручное значение и остается в текущей позиции. Назначьте допустимое значение в ManualValue или переместите привод в ручном режиме с помощью Manual_UP и Manual_DN.
20000	Недопустимое значение в теге SavePosition: Значение имеет неверный числовой формат. Привод не может быть перемещен в выходное подстановочное значение и остается в текущей позиции.
40000	Недопустимое значение в параметре Disturbance: Значение имеет неверный числовой формат. Примечание: Если автоматический режим был активен и ActivateRecoverMode = FALSE, прежде чем возникла ошибка, Disturbance сбрасывается в ноль. PID_3Step остается в автоматическом режиме. Примечание: Если режимы предварительной настройки или точной настройки были активны и ActivateRecoverMode = TRUE до возникновения ошибки, PID_3Step переключается в рабочий режим, который сохранен в параметре Mode. Если Disturbance на текущем этапе не имеет никакого влияния на выходное значение, настройка не отменяется. Ошибка не имеет значения во время измерения времени переключения.

- ¹ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_3Step остается в автоматическом режиме.
- ² Примечание: Если режимы предварительной настройки или точной настройки были активны, прежде чем произошла ошибка, и ActivateRecoverMode = TRUE, PID_3Step переключается в рабочий режим, который сохранен в параметре Mode.
- ³ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_3Step переключается в режим "Подход подстановкой выходного значения с контролем ошибок" или "Контроль ошибок". Как только ошибка перестает быть активной, PID_3Step переключается назад в автоматический режим.
- ⁴ Примечание: Если ActivateRecoverMode = TRUE прежде, чем возникла ошибка, PID_3Step отменяет настройку и переключается в рабочий режим, который сохранен в параметре Mode.
- ⁵ Привод не может быть перемещен в выходное подстановочное значение и остается в его текущей позиции. В ручном режиме Вы можете изменить положение привода только с помощью Manual_UP и Manual_DN, но не с помощью ManualValue.

10.2.7. Параметр Warning инструкции PID_3Step

Если у ПИД-регулятора присутствует несколько предупреждений, он выводит значения кодов посредством сложения в двоичной системе. Отображение кода 0003, например, указывает на то, что коды 0001 и 0002 также активны.

Таблица 10- 25 Параметр Warning инструкции PID_3Step

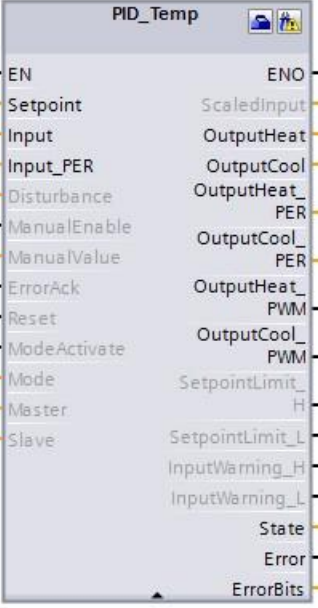
Warning (DW#16#...)	Описание
0000	Нет активных предупреждений.
0001 ¹	Точка перегиба не была найдена во время предварительной настройки.
0002	Качание было предпринято во время "настройки в режиме выполнения". (Параметр "Warning" подавляет это предупреждение, и это видно только в параметре "WarningInternal" в диагностических целях.)
0004 ¹	Уставка была ограничена до сконфигурированных пределов.
0008 ¹	Не все необходимые управляемые системные свойства были определены для выбранного метода вычисления. Вместо этого параметры ПИД были рассчитаны, используя метод TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Рабочий режим не мог быть изменен так как Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызова ОВ ограничивает время выборки ПИД-алгоритма. Улучшите результаты посредством более короткого времени цикла ОВ.
0040 ¹	Значение процесса превысило один из своих предупреждающих пределов.
0080	Недопустимое значение в Mode. Рабочий режим не переключается.
0100 ¹	Ручное значение было ограничено пределами выхода контроллера.
0200	Указанное правило для настройки не поддерживается. Никакие параметры ПИД не рассчитаны.
1000	Выходное значение подстановки не может быть достигнуто, так как находится вне пределов выходного значения.

¹ Примечание: ПИД-регулятор удалил следующие предупреждения автоматически, так как причина устранена или действие пользователя повторено с допустимыми параметрами: 0001, 0004, 0008, 0040, и 0100.

10.2.8. Инструкция PID_Temp

Инструкция PID_Temp обеспечивает универсальный ПИД-регулятор, который позволяет обработку специфических требования по управлению температурой.

Таблица 10- 26 Инструкция PID_Temp

LAD / FBD	SCL	Описание
	<pre> "PID_Temp_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_int_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, Master:=_dword_in Save:=_dword_in ScaledInput=>_real_out_, OutputHeat=>_real_out_, OutputCool=>_real_out_, OutputHeat_PER=>_int_out_, OutputCool_PER=>_int_out_, Out-putHeat_PWM=>_bool_out_, Out-putCool_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, InputWarning_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_); </pre>	PID_Temp предоставляет следующие возможности: • Нагрев и охлаждение процесса с помощью различных исполнительных устройств • Встроенная автонастройка для обработки температурных процессов • Каскадирование для обработки больше чем одного значения температуры, которые зависят от того же исполнительного механизма

¹ STEP 7 автоматически создает технологический объект и экземплярный DB, когда Вы вставляете инструкцию. Экземплярный DB содержит параметры технологического объекта.

² В примере на SCL, "PID_Temp_1" является именем экземплярного DB.

Таблица 10- 27 Типы данных для параметров

Параметр и тип		Тип данных	Описание
Setpoint	IN	Real	Уставка ПИД-регулятора в автоматическом режиме. (Значение по умолчанию: 0.0)
Input	IN	Real	Тег пользовательской программы используется в качестве источника значения процесса. (Значение по умолчанию: 0.0) Если Вы используете параметр Input, Вы должны установить Config.InputPerOn = FALSE.
Input_PER	IN	Word	Аналоговый вход используется в качестве источника значения процесса. (Значение по умолчанию: W#16#0) Если Вы используете параметр Input_PER, Вы должны установить Config.InputPerOn = TRUE.
Disturbance	IN	Real	Переменная возмущения или значение пред-управления
ManualEnable	IN	Bool	Включает или отключает ручной режим работы. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE активирует "ручной режим", при этом если State = 4, режим остается неизменным. Пока ManualEnable = TRUE, Вы не можете изменить рабочий режим, используя нарастающий фронт в ModeActivate или использовать диалоговое окно ввода в эксплуатацию. - Фронт от TRUE к FALSE активирует рабочий режим, который назначен посредством Mode. Примечание: Мы рекомендуем, чтобы Вы изменяли рабочий режим, только используя ModeActivate.
ManualValue	IN	Real	Выходное значение для ручного режима. (Значение по умолчанию: 0.0) Вы можете использовать значения от Config.OutputLowerLimit до Config.OutputUpperLimit.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выходы предупреждения по фронту от FALSE к TRUE. (Значение по умолчанию: FALSE)
Reset	IN	Bool	Перезапускает регулятор. (Значение по умолчанию: FALSE): - Фронт от FALSE к TRUE: - Переключается в "неактивный" режим - Сбрасывает ErrorBits и выходы предупреждения - Очищает интегральное действие - Сохраняет параметры ПИД - Пока Reset = TRUE, PID_Temp остается в режиме "неактивный" (State = 0). - Фронт от TRUE к FALSE: - PID_Temp переключается в рабочий режим, который сохранен в параметре Mode.
ModeActivate	IN	Bool	PID_Temp переключается в рабочий режим, который сохранен в параметре Mode по фронту от FALSE к TRUE. (Значение по умолчанию: FALSE)

Параметр и тип		Тип данных	Описание
Mode	IN/OUT	Int	Активируется по переднему фронту входа ModeActivate. Выбор рабочего режима (Значение по умолчанию: 0.0): - Mode = 0: Неактивный - Mode = 1: Предварительная настройка - Mode = 2: Точная настройка - Mode = 3: Автоматический режим - Mode = 4: Ручной режим "Подстановка выходного значения с контролем ошибок" (State = 5). Он не может быть активирован пользователем; является лишь автоматической реакцией на ошибку.
Master	IN/OUT	DWord	Каскадное соединение с ведущим устройством (AntiWindUp и условия настройки). (Значение по умолчанию: DW#16#0000) - Биты 0 - 15: Не используются в инструкции PID_Temp - Биты 16 - 23: Предельный счетчик: Ведомый постепенно увеличивает это значение, если достигает своего ограничения. Число ведомых в состоянии ограничения обрабатывается для "противо-интегрального насыщения" (Обратитесь к параметру Config.Cascade.AntiWindUpMode.) - Бит 24: IsAutomatic: Этот бит устанавливается в "1", если все ведомые этого контроллера находятся в автоматическом режиме и обрабатываются для проверки условий для настройки каскада. Этот бит идентичен параметру AllSlaveAutomaticState. - Бит 25: "IsReplacement-Setpoint": Этот бит установлен в "1", если ведомый этого контроллера имеет активированную "замещающую уставку" и обрабатывается для проверки условий для настройки каскада. Инвертированное значение сохранено в параметре NoSlaveReplacementSetpoint.
Slave	IN/OUT	DWord	
ScaledInput	OUT	Real	Приведенное значение процесса. (По умолчанию: 0.0)
OutputHeat ¹	OUT	Real	Выходное значение для нагрева в формате REAL. (Значение по умолчанию: 0.0) Это выходное значение вычисляется, независимо от выбора выхода, используя параметр Config.Output.Heat.Select.
OutputCool ¹	OUT	Real	Выходное значение для охлаждения в формате REAL. (По умолчанию: 0.0) Это выходное значение вычисляется, независимо от выбора выхода, используя параметр Config.Output.Cool.Select
OutputHeat_PER ¹	OUT	Int	Выходное значение для нагрева в дополнительном формате (Значение по умолчанию: 0) Это выходное значение рассчитывается только, когда выбрано используя параметр Config.Output.Heat.Select = 2. Если не выбрано, этот выход всегда в "0".
OutputCool_PER ¹	OUT	Int	Выходное значение для охлаждения в дополнительном формате (Значение по умолчанию: 0) Это выходное значение рассчитывается только, когда выбрано используя параметр Config.Output.Cool.Select = 2. Если не выбрано, этот выход всегда в "0".
OutputHeat_PWM ¹	OUT	Bool	Выходное значение с широтно-импульсной модуляцией для нагрева. (Значение по умолчанию: FALSE) Это выходное значение рассчитывается только, когда выбрано используя параметр Config.Output.Heat.Select = 1 (значение по умолчанию) . Если не выбрано, то этот выход всегда в состоянии FALSE.

Параметр и тип		Тип данных	Описание
OutputCool_PWM ¹	OUT	Bool	Выходное значение с широтно-импульсной модуляцией для охлаждения. (Значение по умолчанию: FALSE) Это выходное значение рассчитывается только, когда выбрано используя параметр Config.Output.Cool.Select = 1 (значение по умолчанию). Если не выбрано, то этот выход всегда в состоянии FALSE.
SetpointLimit_H	OUT	Bool	Верхний предел уставки. (Значение по умолчанию: FALSE) Если SetpointLimit_H = TRUE, абсолютный верхний предел уставки достигнут (Setpoint ≥ Config.SetpointUpperLimit). Уставка ограничена значением Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Нижний предел уставки. (Значение по умолчанию: FALSE) Если SetpointLimit_L = TRUE, абсолютный нижний предел уставки достигнут (Setpoint ≤ Config.SetpointLowerLimit). Уставка ограничена значением Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Если InputWarning_H = TRUE, значение процесса достигло или превысило верхний предупреждающий предел. (Значение по умолчанию: FALSE)
InputWarning_L	OUT	Bool	Если InputWarning_L = TRUE, значение процесса достигло или опустилось ниже нижнего предупреждающего предела. (Значение по умолчанию: FALSE)
State	OUT	Int	Текущий рабочий режим ПИД-регулятора. (По умолчанию: 0) Вы можете изменить рабочий режим, используя входной параметр Mode и нарастающий фронт в ModeActivate: - State = 0: Неактивный - State = 1: Предварительная настройка - State = 2: Ручная точная настройка - State = 3: Автоматический режим - State = 4: Ручной режим - State = 5: Подстановка выходного значения с контролем ошибок
Error	OUT	Bool	Если Error = TRUE, по крайней мере одно сообщение об ошибке присутствует в этом цикле. (Значение по умолчанию: FALSE) Примечание: Параметр Error в ПИД V1.x был полем ErrorBits, которое содержало коды ошибок. Теперь это булевый флаг, указывающий на то, что ошибка произошла.
ErrorBits	OUT	DWord	Таблица параметров ErrorBits инструкции PID_Temp (стр. 502) определяет сообщения об имеющихся ошибках. (Значение по умолчанию: DW#16#0000 (ошибок нет)). ErrorBits сохраняемый и сбрасывается по нарастающему фронту на Reset или ErrorAck. Примечание: В V1.x, параметр ErrorBits был определен как Error и отсутствовал.
Warning	OUT	DWord	Таблица значений параметра Warning инструкции PID_Temp (стр. 504) определяет активные предупреждающие сообщения важные для пользователя. (По умолчанию: DW#16#0000 (нет предупреждений)).
WarningInternal	OUT	DWord	Таблица значений параметра WarningInternal инструкции PID_Temp определяет активные предупреждающие внутренние сообщения (включает все предупреждения). (По умолчанию: DW#16#0000 (нет внутренних предупреждений)).

¹ Вы можете использовать выходные значения параметров Output, Output_PER и Output_PWM параллельно.

10.2.8.1. Работа PID_Temp регулятора

Выбор управления нагревом и / или охлаждением

Вы вначале должны выбрать, необходимо ли Вам в дополнение к выходу нагрева устройство охлаждения в параметре "ActivateCooling". Впоследствии, Вы должны определить, если Вы хотите использовать два набора ПИД-параметров (расширенный режим) или только один набор ПИД-параметров с дополнительным коэффициентом нагрев / охлаждение в параметре "AdvancedCooling".

Использование CoolFactor

В случае, если Вы хотите применить коэффициент нагрев / охлаждение, Вы должны определить значение вручную. Вы должны идентифицировать значение из технических данных Вашего приложения (отношение пропорционального усиления исполнительных устройств (например, отношение максимальных нагревательной и охлаждающей способностей исполнительных устройств) и назначить его параметру "CoolFactor". Коэффициент нагрев / охлаждение 2.0 означает, что устройство нагрева в два раза более эффективно, чем устройство охлаждения. Если Вы используете коэффициент охлаждения, PID_Temp вычисляет выходной сигнал и, в зависимости от его знака, умножает выходной сигнал на коэффициент нагрев / охлаждение (когда знак отрицателен), или нет (когда знак положителен).

Использование двух наборов ПИД-параметров

Различные наборы ПИД-параметров для нагрева и охлаждения могут быть автоматически обнаружены во время ввода в эксплуатацию. Вы можете ожидать лучшие характеристики управления по сравнению с коэффициентом нагрев / охлаждение, так как в дополнение к различным пропорциональным усилениям Вы можете принимать в расчет различные времена задержки с помощью двух наборов параметров. Однако недостаток состоит в том, что настройка процесса может занять больше времени. Если активировано переключение ПИД-параметра (Config.AdvancedCooling = TRUE), регулятор PID_Temp обнаруживает в "автоматическом режиме" (управление активно) необходимость в данный момент нагрева или охлаждения и использует наборы ПИД-параметров для управления.

ControlZone

С регулятором PID_Temp Вы можете определить зону контроля для каждого набора параметров в параметре "ControlZone". Если отклонение управления (уставка – вход) находится в зоне контроля, PID_Temp использует ПИД-алгоритм, чтобы вычислить выходные сигналы. Однако, если отклонение управления выходит из определенного диапазона, выход устанавливается в значение максимального нагрева или охлаждения (выход охлаждения активирован) / значение минимального нагрева (выход охлаждения деактивирован). Вы можете использовать эту функциональность, чтобы получить нужную уставку быстрее, особенно для начального нагрева медленных температурных процессов.

DeadZone

В параметре "DeadZone" Вы можете определить величину рассогласования для нагрева и охлаждения, которой пренебрегает ПИД-алгоритм. Это означает, что рассогласование в этом диапазоне подавляется, и регулятор PID_Temp ведет себя как будто уставка и значения процесса идентичны. Таким образом Вы можете устранить излишнее вмешательство регулятора вблизи уставки и сберечь исполнительный механизм. Если Вы хотите применить DeadZone, Вы должны определить значение вручную. Автоматическая настройка не устанавливает значение DeadZone. DeadZone симметричен (между -Retain.CtrlParams.Heat.DeadZone и +Retain.CtrlParams.Heat.DeadZone) для регуляторов нагрева без охлаждения или регуляторов нагрева/охлаждения, использующих CoolFactor. DeadZone может быть асимметричным (между -Retain.CtrlParams.Cool.DeadZone и +Retain.CtrlParams.Heat.DeadZone) регуляторов нагрева/охлаждения, использующих два набора ПИД-параметров.

Работа контроллера PID_Temp

Следующие блок-схемы иллюстрируют стандартную и каскадную работу инструкцию PID_Temp:

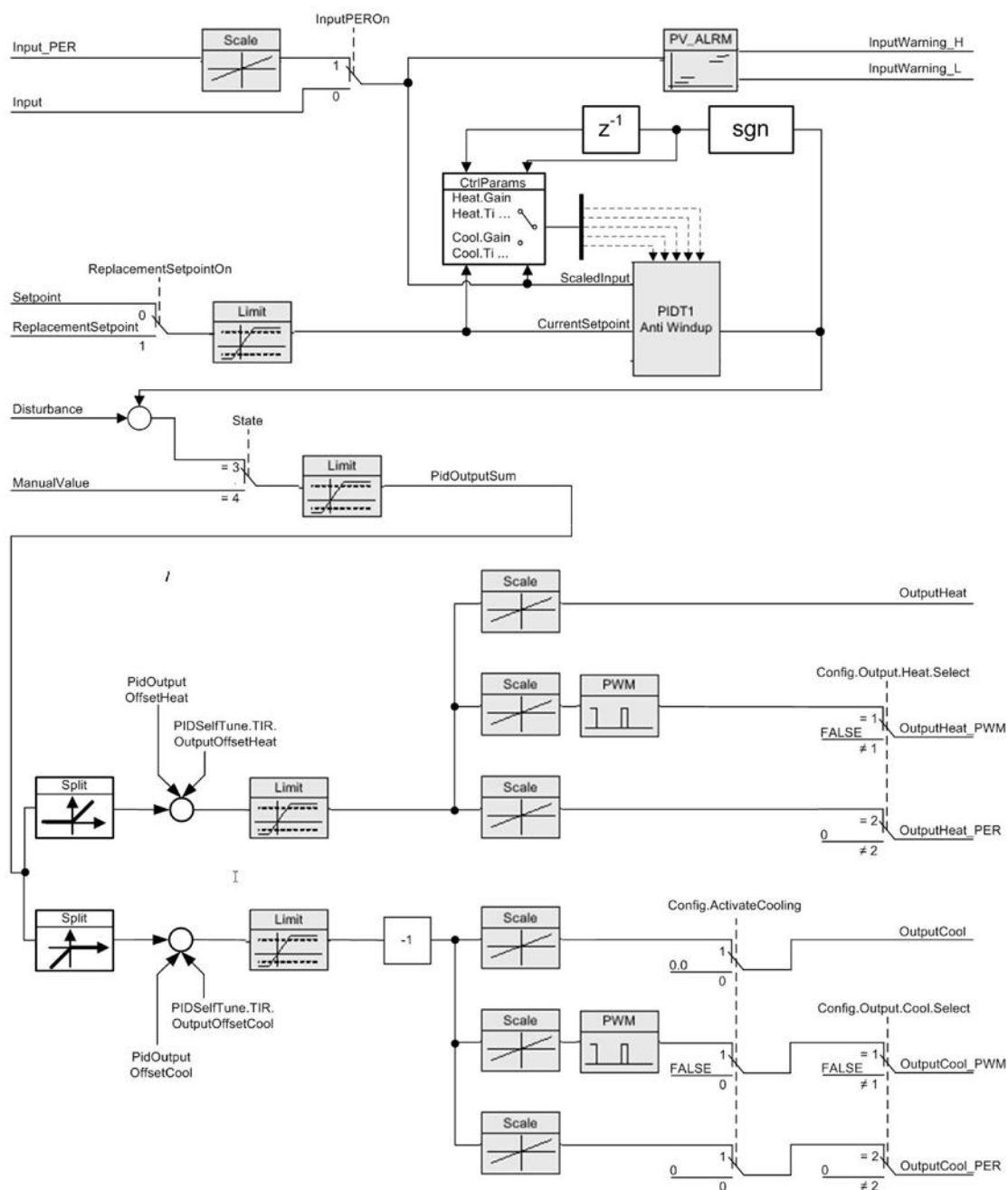


Рисунок 10-6 Функциональная диаграмма работы PID_Temp

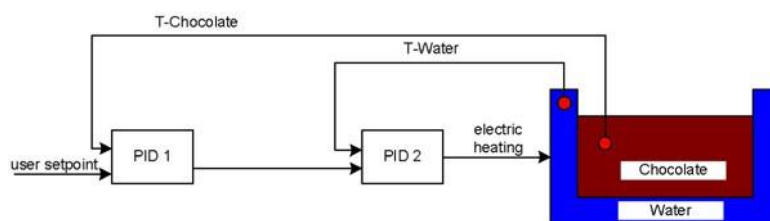


Рисунок 10-7 Функциональная диаграмма каскадной работы PID_Temp

10.2.8.2. Каскадное включение регуляторов

Вы можете включать температурные ПИД-регуляторы каскадно, чтобы обработать несколько значений температуры, зависящих от одного и того же привода.

Порядок вызова

Вы должны вызвать включенные каскадно ПИД-регуляторы в том же циклическом ОВ. В начале Вы должны вызвать ведущий регулятор, затем последующий(-ие) ведомый(-ые) в потоке управляющего сигнала, и наконец последний ведомый регулятор в каскаде. Инструкция PID_Temp не осуществляет автоматическую проверку порядка вызовов.

Коммуникационные соединения

При каскадировании регуляторов, Вы должны соединить ведущий и ведомый так, чтобы они могли делиться информацией друг с другом. Вы должны соединить IN/OUT параметр "Master" ведомого с IN/OUT параметром "Slave" ведущего в направлении потока сигналов.

Здесь показано каскадное соединение PID_Temp регуляторов в с двумя подкаскадами: "PID_Temp1" обеспечивает уставку. Конфигурация соединяет выходы "PID_Temp2", "PID_Temp3", "PID_Temp5", "PID_Temp6" и "PID_Temp8" с процессом:

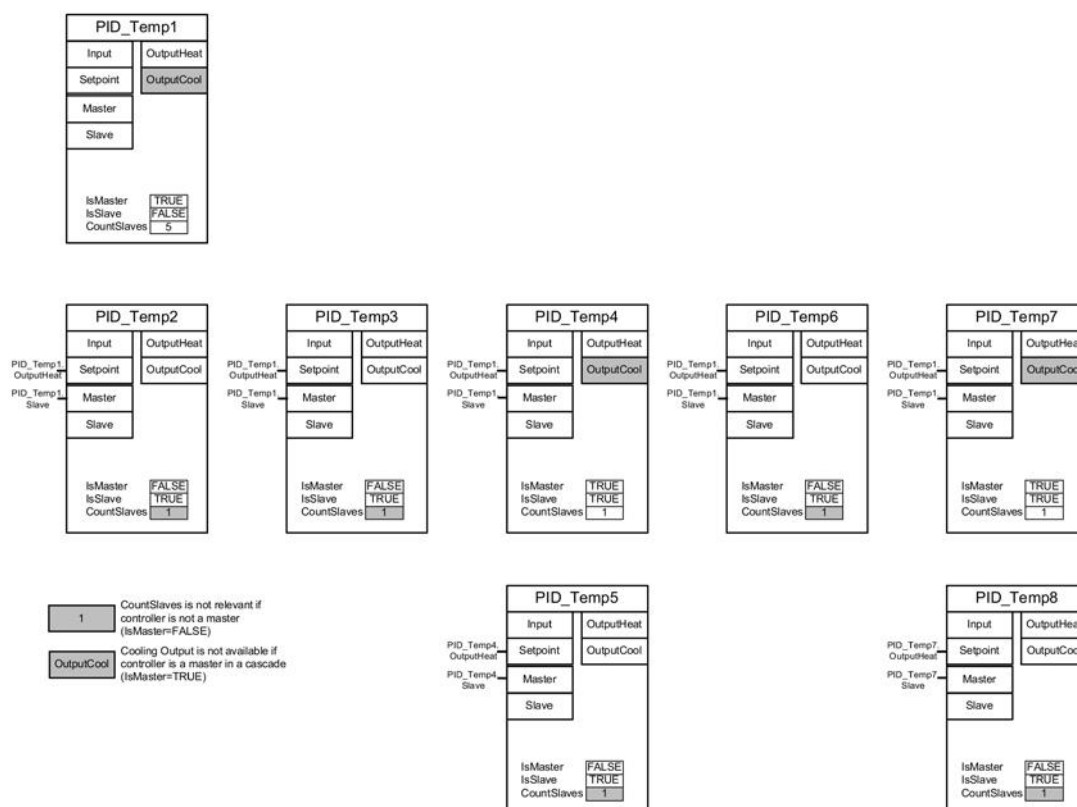


Рисунок 10-8 Коммуникационные соединения при каскадировании PID_Temp

Замещающая уставка

Инструкция PID_Temp обладает вторым входом уставки в параметре "ReplacementSetpoint", который Вы можете активировать, установив параметр "ReplacementSetpointOn" = TRUE. Вы можете использовать "ReplacementSetpoint" в качестве своего входа уставки во время ввода в эксплуатацию или настройки ведомого регулятора без необходимости разъединять соединение выхода к уставке между ведущим и ведомым. Это соединение необходимо для нормального функционирования каскада.

Таким образом Вы не должны изменять свою программу и загружать ее, если Вы хотите временно разделить ведомого от ведущего. Вы только должны активировать "ReplacementSetpoint" и деактивировать его вновь при завершении. Действующее значение уставки для ПИД-алгоритма Вы видите в параметре "CurrentSetpoint".

Автонастройка

Автоматическая настройка для ведущего регулятора каскадного соединения должна удовлетворить следующим требованиям:

- Вводиться в эксплуатацию от внутреннего ведомого до первого ведущего.
- Все ведомые ведущего устройства должны находиться в "автоматическом режиме".
- Выход ведущего должен служить уставкой для ведомых.

Инструкция PID_Temp предоставит следующую поддержку для автоматической настройки в каскаде:

- Если Вы начинаете автоматическую настройку ведущего регулятора, он проверяет находятся ли все ведомые в "автоматическом режиме" и деактивацию функции замещающей уставки для всех ведомых ("ReplacementSetpointOn" = FALSE). Если Вы не удовлетворяете этим условиям, Вы не можете автоматически настроить ведущее устройство. Ведущее устройство отменяет настройку, переходит в режим "неактивный" (если "ActivateRecoverMode" = FALSE) или назад в режим, сохраненный в параметре "Mode" (если "ActivateRecoverMode" = TRUE). Ведущее устройство выдает сообщение об ошибке 200000hex ("Ошибка с ведущего устройства в каскаде. Ведомые не находятся в автоматическом режиме или активировали замещающую уставку и препятствуют настройке ведущего устройства").
- Когда все ведомые находятся в "автоматическом режиме", система устанавливает параметр "AllSlaveAutomaticState" = TRUE. Вы можете применить этот параметр в своих программах или локализовать причину ошибки 200000hex.
- Когда "ReplacementSetpoint" деактивирован для всех ведомых, система устанавливает параметр "NoSlaveReplacementSetpoint" = TRUE. Вы можете применить этот параметр в своих программах или локализовать причину ошибки 200000hex.

При использовании диалога ввода в эксплуатацию инструкции PID_Temp, Вы обладаете дополнительной поддержкой каскадной настройки (стр. 524).

Режимы работы и обработка ошибок

Регулятор PID_Temp не позволяет переключаться рабочим режимам его ведущих или ведомых устройств. Это означает, что ведущее устройство в каскаде остается в текущем режиме, когда ведомое вызывает ошибку. Преимуществом является ситуация, когда два или более параллельных ведомых работают с этим основным регулятором; ошибка в одной цепочке не отключает параллельную цепочку.

Точно так же ведомое устройство в каскаде остается в своем текущем режиме работы, если у ведущего устройства присутствует ошибка. Однако дальнейшая работа ведомого в этом случае зависит от конфигурации ведущего устройства, потому что уставкой ведомого является выход ведущего устройства. Это означает, что, если Вы конфигурируете ведущее устройство с "ActivateRecoverMode" = TRUE и происходит ошибка, ведущее устройство выдает последнее допустимое или подстановочное выходное значение в качестве уставки для ведомого. Если Вы конфигурируете ведущее устройство с "ActivateRecoverMode" = FALSE, ведущее устройство переключается в "неактивный режим", и устанавливает все выходы в "0.0" так, что ведомое устройство использует "0.0" в качестве уставки.

Поскольку только у ведомых регуляторов есть прямой доступ к исполнительным устройствам, и они остаются в их рабочем режиме в случае основной ошибки, Вы можете избежать повреждения процесса. Например, для устройств обработки пластмасс фатальной ситуацией для ведомых устройств является прекращение работы, закрытие приводов, что позволяет пластмассе затвердевать в устройстве исключительно потому, что основной регулятор вызвал ошибку.

Анти-интегральное насыщение

Ведомое устройство в каскаде получает уставку от выхода ведущего устройства. Если ведомое устройство достигает предела собственного выходного значения, в то время как ведущее устройство все еще видит рассогласование (уставка – вход), ведущее устройство замораживает или уменьшает свою интегрирующий вклад, чтобы предотвратить так называемое "насыщение". В случае "насыщения" ведущее устройство увеличивает свой интегрирующий вклад до очень большого значения и должно вначале уменьшить его, прежде чем регулятор сможет снова нормально реагировать. Подобное "насыщение" негативно влияет на динамику управления. PID_Temp предлагает способы предотвращения этого эффекта в каскаде, путем конфигурирования параметра "Config.Cascade.AntiWindUpMode" в основном регуляторе:

Значение	Описание
0	Деактивирует функцию анти-насыщения.
1	Уменьшает интегрирующий вклад ведущего регулятора в соотношении "ведомые в ограничении" к "существующим ведомым" (параметр "CountSlaves").
2	Замораживает интегрирующий вклад ведущего устройства, как только одно ведомое устройство достигает ограничения. Действенно только, если "Config.Cascade.IsMaster" = TRUE.

10.2.9. Параметр ErrorBit инструкции PID_Temp

Если присутствует несколько ошибок, то значения кодов ошибок выводятся посредством сложения в двоичном коде. Отображение кода ошибки 0003, например, указывает, на то, что ошибки 0001 и 0002 также активны.

Таблица 10- 28 Параметры ErrorBit инструкции PID_Temp

ErrorBit (DW#16#...)	Описание
0000	Ошибок нет
0001 ^{1, 2}	Параметр Input вне пределов значения процесса. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Недопустимое значение в параметре Input_PER. Проверьте наличие ошибки на аналоговом входе.
0004 ⁴	Ошибка во время точной настройки. Колебания значения процесса нельзя скомпенсировать.
0008 ⁴	Ошибка в начале предварительной настройки. Значение процесса слишком близко к уставке. Запустите точную настройку.
0010 ⁴	Уставка была изменена во время настройки. Примечание: Вы можете установить разрешенное колебание уставки в теге CancelTuningLevel.
0020	Предварительная настройка недопустима во время точной настройки. Примечание: Если ActivateRecoverMode = TRUE до возникновения ошибки, PID_Temp остается в режиме точной настройки.
0040 ⁴	Ошибка во время предварительной настройки. Охлаждение не может уменьшить значение процесса.
0080 ⁴	Ошибка во время предварительной настройки. Неправильная конфигурация пределов выходного значения. Проверьте, правильно ли сконфигурированы пределы выходного значения и согласуйте управляющую логику.

ErrorBit (DW#16#...)	Описание
0100 ⁴	Ошибка во время точной настройки привела к недопустимым параметрам.
0200 ^{2, 3}	Недопустимое значение в параметре Input: Значение имеет неверный числовой формат.
0400 ^{2, 3}	Расчет выходного значения не удался. Проверьте ПИД-параметры.
0800 ^{1, 2}	Ошибка времени выборки: PID_Temp не вызывается в течение времени выборки ОВ циклического прерывания.
1000 ^{2, 3}	Недопустимое значение в параметре Setpoint: Значение имеет неверный числовой формат.
10000	Недопустимое значение в параметре ManualValue: Значение имеет неверный числовой формат. Примечание: Если ActivateRecoverMode = TRUE перед возникновением ошибки, PID_Temp использует SubstituteOutput в качестве выходного значения. Как только Вы назначаете допустимое значение в параметре ManualValue, PID_Temp использует его в качестве выходного значения.
20000	Недопустимое значение в тере SubstituteValue: Значение имеет неверный числовой формат. PID_Temp использует нижний предел в качестве выходного значения. Примечание: Если автоматический режим был активен, прежде чем произошла ошибка, ActivateRecoverMode = TRUE, и ошибка больше не активна, PID_Temp переключается назад в автоматический режим.
40000	Недопустимое значение в параметре Disturbance: Значение имеет неверный числовой формат. Примечание: Если автоматический режим был активен и ActivateRecoverMode = FALSE, прежде чем возникла ошибка, Disturbance сбрасывается в ноль. PID_Temp остается в автоматическом режиме. Примечание: Если режимы предварительной настройки или точной настройки были активны и ActivateRecoverMode = TRUE до возникновения ошибки, PID_Temp переключается в рабочий режим, который сохранен в параметре Mode. Если Disturbance на текущем этапе не имеет никакого влияния на выходное значение, настройка не отменяется.
200000	Ошибка ведущего устройства в каскаде. Ведомые не находятся в автоматическом режиме или активировали замещающую уставку, мешая настройке ведущего устройства.
400000	ПИД-регулятор не разрешает предварительную настройку для нагрева, пока активно охлаждение.
800000	Значение процесса должно быть близко к уставке, чтобы начать предварительную настройку для охлаждения.
1000000	Ошибочная начальная настройка. "Heat.EnableTuning" и "Cool.EnableTuning" не установлены или не соответствуют конфигурации.
2000000	Предварительная настройка для охлаждения требует успешной предварительной настройки для нагрева.

ErrorBit (DW#16#...)	Описание
4000000	Ошибка начала точной настройки. "Heat.EnableTuning" и "Cool.EnableTuning" не могут быть установлены одновременно.
8000000	Ошибка во время вычисления ПИД-параметра привела к недопустимым параметрам (например, отрицательное усиление; текущие ПИД-параметры остаются неизменными, и настройка не дает никакого результата).

- ¹ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_Temp остается в автоматическом режиме.
- ² Примечание: Если режимы предварительной настройки или точной настройки были активны, прежде чем произошла ошибка, и ActivateRecoverMode = TRUE, PID_Temp переключается в рабочий режим, который сохранен в параметре Mode.
- ³ Примечание: Если автоматический режим был активен, прежде чем произошла ошибка и ActivateRecoverMode = TRUE, PID_Temp выводит сконфигурированное подстановочное значение на выходе. Как только ошибка перестает быть активной, PID_Temp переключается назад в автоматический режим.
- ⁴ Примечание: Если ActivateRecoverMode = TRUE прежде, чем возникла ошибка, PID_Temp отменяет настройку и переключается в рабочий режим, который сохранен в параметре Mode.

10.2.10. Параметр Warning инструкции PID_Temp

Если у ПИД-регулятора присутствует несколько предупреждений, он выводит значения кодов посредством сложения в двоичной системе. Отображение кода 0003, например, указывает на то, что коды 0001 и 0002 также активны.

Таблица 10- 29 Параметр Warning инструкции PID_Temp

Warning (DW#16#...)	Описание
0000	Нет активных предупреждений.
0001 ¹	Точка перегиба не была найдена во время предварительной настройки.
0002	Качение было предпринято во время "настройки в режиме выполнения". (Параметр "Warning" подавляет это предупреждение, и это видно только в параметре "WarningInternal" в диагностических целях.)
0004 ¹	Уставка была ограничена до сконфигурированных пределов.
0008 ¹	Не все необходимые управляемые системные свойства были определены для выбранного метода вычисления. Вместо этого параметры ПИД были рассчитаны, используя метод TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Рабочий режим не мог быть изменен так как Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызова ОВ ограничивает время выборки ПИД-алгоритма. Улучшите результаты посредством более короткого времени цикла ОВ.
0040 ¹	Значение процесса превысило один из своих предупреждающих пределов.
0080	Недопустимое значение в Mode. Рабочий режим не переключается.
0100 ¹	Ручное значение было ограничено пределами выхода контроллера.
0200	Указанное правило для настройки не поддерживается. Никакие параметры ПИД не рассчитаны.
1000	Выходное значение подстановки не может быть достигнуто, так как находится вне пределов выходного значения.

Warning (DW#16#...)	Описание
4000	Указанное значение выхода для нагрева и/или охлаждения не поддерживается. Активны только OutputHeat и OutputCool.
8000	Указанное значение для параметра PIDSelfTune.SUT.AdaptDelayTime не поддерживается, следовательно используется значение по умолчанию "0".
10000	Указанное значение для параметра PIDSelfTune.SUT.CoolingMode не поддерживается, следовательно используется значение по умолчанию "0".

- ¹ Примечание: ПИД-регулятор удалил следующие предупреждения автоматически, так как причина устранена или действие пользователя повторено с допустимыми параметрами: 0001, 0004, 0008, 0040, и 0100.

10.2.11. Конфигурирование регуляторов PID_Compact и PID_3Step

Параметры технологического объекта определяют работу ПИД-регулятора. Используйте значок , чтобы открыть редактор конфигурации.

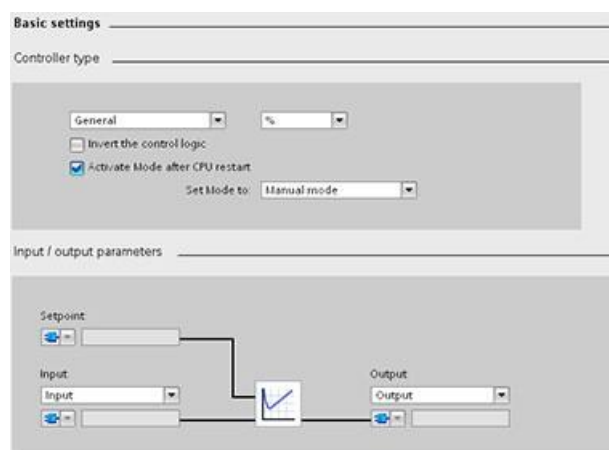


Таблица 10- 30 Пример конфигурационных настроек для инструкции PID_Compact

Настройки		Описание
Basic	Controller type	Выбирает технические единицы.
	Invert the control logic	Позволяет выбор ПИД-контура обратного действия. - Если не выбрано, ПИД-контур находится в режиме прямого действия, и выход ПИД-контура увеличивается если входное значение < уставка. - Если выбрано, выход ПИД-контура увеличивается если входное значение > уставка.
	Enable last mode after CPU restart	Перезапускает ПИД-контур после того, как он сброшен, или, если входной предел был превышен, и выполнен возврат в допустимый диапазон.
	Input	Выбирает либо параметр Input, либо параметр Input_PER (для аналогового) в качестве значения процесса. Input_PER может поступать непосредственно из модуля аналогового ввода.
	Output	Выбирает либо параметр Output, либо параметр Output_PER (для аналогового) в качестве выходного значения. Output_PER может подаваться непосредственно на модуль аналогового вывода.

Настройки	Описание
Process value	Масштабирует как диапазон, так и пределы для значения процесса. Если значение процесса заходит ниже нижнего предела или выше верхнего предела, ПИД-контур переходит в неактивный режим и устанавливает выходное значение в 0. Чтобы использовать Input_PER, Вы должны масштабировать аналоговое значение процесса (входное значение).

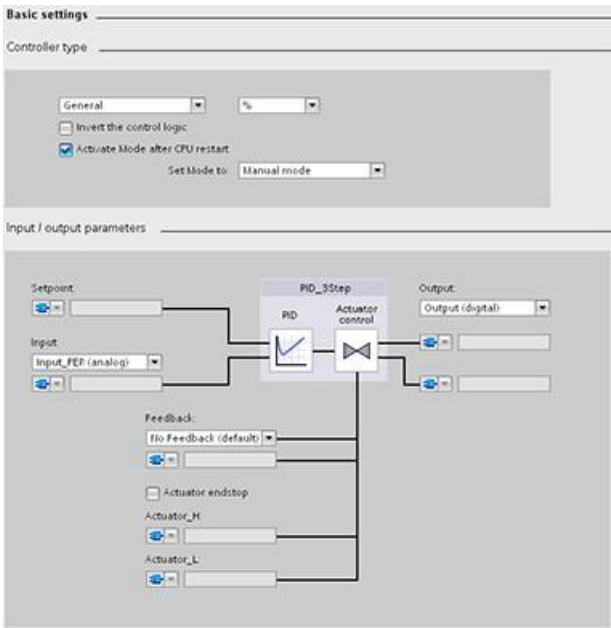


Таблица 10- 31 Пример конфигурационных настроек для инструкции PID_3Step

Настройки		Описание
Basic	Controller type	Выбирает технические единицы.
	Invert the control logic	Позволяет выбор ПИД-контура обратного действия. • Если не выбрано, ПИД-контур находится в режиме прямого действия, и выход ПИД-контура увеличивается если входное значение < уставка. • Если выбрано, выход ПИД-контура увеличивается если входное значение > уставка.
	Activate mode after CPU restart	Перезапускает ПИД-контур после того, как он сброшен, или, если входной предел был превышен, и выполнен возврат в допустимый диапазон. Set Mode to: Определяет режим, в который пользователь хочет перевести ПИД после перезапуска.
	Input	Выбирает либо параметр Input, либо параметр Input_PER (для аналогового) в качестве значения процесса. Input_PER может поступать непосредственно из модуля аналогового ввода.
	Output	Выбирает использовать ли цифровые выходы (Output_UP и Output_DN), либо аналоговый выход (Output_PER) для выходного значения.
	Feedback	Выбирает тип состояния устройства, возвращаемого в ПИД-контур: • Без обратной связи (по умолчанию) • Feedback • Feedback_PER
Process value	Масштабирует как диапазон, так и пределы для значения процесса. Если значение процесса заходит ниже нижнего предела или выше верхнего предела, ПИД-контур переходит в неактивный режим и устанавливает выходное значение в 0. Чтобы использовать Input_PER, Вы должны масштабировать аналоговое значение процесса (входное значение).	

Настройки		Описание
Actuator	Motor transition time	Устанавливает время хода от открытого до закрытого состояния для клапана. (Найдите это значение в паспорте или шильдике клапана.)
	Minimum ON time	Устанавливает минимальное время перемещения для клапана. (Найдите это значение в паспорте или шильдике клапана.)
	Minimum OFF time	Устанавливает минимальное время паузы для клапана. (Найдите это значение в паспорте или шильдике клапана.)
	Reaction to error	Определяет поведение клапана, когда обнаружена ошибка или когда ПИД-контур сброшен. Если Вы выбираете использование позицию подстановки, введите "безопасную позицию". Для аналоговой обратной связи или аналогового выхода, выберите значение между верхним или нижним пределом для выхода. Для цифровых выходов Вы можете выбрать только 0% (закрыт) или 100% (открыт).
	Scale Position Feedback ¹	"High end stop" и "Lower end stop" определяют максимальное положительное положение (полностью открыт) и максимальное отрицательное положение (полностью закрыт). Значение для "High end stop" должно быть больше, чем значение для "Lower end stop". "High limit process value" и "Low limit process value", определяют верхнее и нижнее положения клапана во время настройки и автоматического режима. "FeedbackPER" ("Low" и "High ") определяет аналоговую обратную связь положения клапана. "FeedbackPER High" должно быть больше, чем "FeedbackPER Low".
Advanced	Monitoring process value	Устанавливает предупреждающие верхний и нижний пределы для значения процесса.
	PID parameters	Если пользователь желает, он может ввести свои собственные параметры настройки ПИД в этом окне. Флажок "Enable Manual Entry" должен быть установлен, чтобы позволить это.

¹ "Scale Position Feedback" доступно для редактирования, только если Вы разрешили "Feedback" в настройках "Basic".

10.2.12. Конфигурирование регулятора PID_Temp

Параметры технологического объекта определяют работу ПИД-регулятора. Используйте значок , чтобы открыть редактор конфигурации.

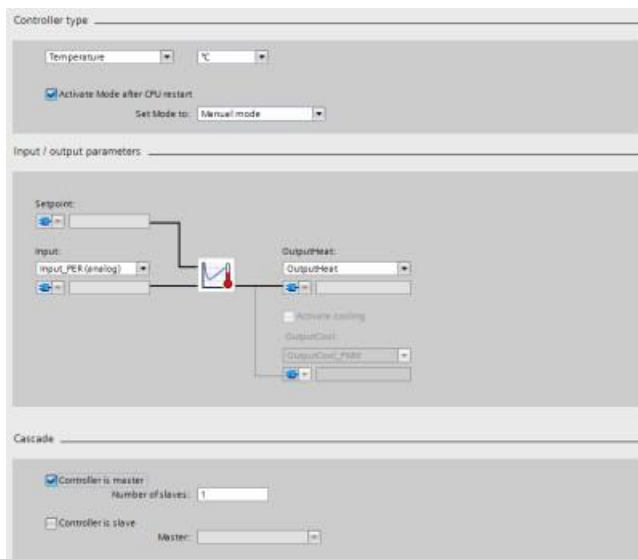


Таблица 10- 32 Пример конфигурационных настроек для инструкции PID_Temp

Настройки		Описание
Basic	Controller type	Выбирает технические единицы.
	Activate mode after CPU restart	Перезапускает ПИД-контур после того, как он сброшен, или, если входной предел был превышен, и выполнен возврат в допустимый диапазон. Set Mode to: Определяет режим, в который пользователь хочет перевести ПИД после перезапуска.
	Input	Выбирает либо параметр Input, либо параметр Input_PER (для аналогового) в качестве значения процесса. Input_PER может поступать непосредственно из модуля аналогового ввода.
	Output Heat	Выбирает использовать ли цифровые выходы (OutputHeat и OutputHeat_PWM), либо аналоговый выход (OutputHeat_PER (аналоговый)) для выходного значения.
	Output Cool	Выбирает использовать ли цифровые выходы (OutputCool и OutputCool_PWM), либо аналоговый выход (OutputCool_PER (аналоговый)) для выходного значения.
Process value		Масштабирует как диапазон, так и пределы для значения процесса. Если значение процесса заходит ниже нижнего предела или выше верхнего предела, ПИД-контур переходит в неактивный режим и устанавливает выходное значение в 0. Чтобы использовать Input_PER, Вы должны масштабировать аналоговое значение процесса (входное значение).
Cascade	Controller is master	Определяет регулятор как ведущее устройство и выбирает число ведомых устройств.
	Controller is slave	Определяет регулятор как ведомое устройство и выбирает число ведущих устройств.

Тип контроллера

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Physical quantity	"PhysicalQuantity"	Int (Enum)	- General - Temperature (=default)	Предварительный выбор значения физической единицы Без многозначного управления и недоступно для редактирования в он-лайн режиме функционального представления.
Unit of measurement	"PhysicalUnit"	Int (Enum)	- General: Units = % - Temperature: Units (возможн. значения) = °C (по ум.) °F °K	Пользовательский выбор единиц возвращается к "0", если Вы изменяете физическую величину.
Activate mode after CPU restart	"RunModeByStartup"	Bool	Checkbox	Если установлен в TRUE (=по умолч.), регулятор переключается в состояние, которое сохранено в переменной "Mode" после цикла включения и выключения питания (Питание вкл.- выкл.- вкл.), или переход ПЛК из STOP в RUN. Иначе, PID_Temp остается в режиме "неактивный".
Set mode to	"Mode"	Int (Enum)	Modes (возможн. варианты): 0: Неактивный 1: Преднастройка 2: Точ.настройка 3: Авто 4: Ручной (=по умолч.)	Инженерная станция (ES) устанавливает стартовое значение переменной "Mode" согласно выбору пользователя. Значением по умолчанию для Mode (сохранено в TO-DB) является Ручной режим.

Входные / выходные параметры

Настройка	ТО-DB параметр	Тип данных	Диапазон значений	Описание
Setpoint	Setpoint	Real	Real	Доступно только на странице свойств. Без многозначного управления и недоступно для редактирования в он-лайн режиме функционального представления.
Selection input	"Config.InputPerOn"	Bool (Enum)	Bool	Выбирает, какой тип входа использовать. Возможные варианты: - FALSE: "Input" (Real) - TRUE: "Input_PER (аналоговый)"
Input	Input или Input_PER	Real или Int	Real или Int	Доступно только на странице свойств. Без многозначного управления и недоступно для редактирования в он-лайн режиме функционального представления.
Selection Output (heating)	"Config.Output.Heat.Select"	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	у: "OutputHeat" (Real) "OutputHeat_PWM" (Bool) (=по умолч.) "OutputHeat_PER (аналог.)" (Word) Устанавливается в "OutputHeat" после установки пользователем флажка "This controller is a master" в разделе "Cascade".
Output (heating)	OutputHeat, OutputHeat_PER или OutputHeat_PWM	Real или Int или Bool	Real, Int или Bool	Доступно только на странице свойств. Без многозначного управления и недоступно для редактирования в он-лайн режиме функционального представления.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Activate output (cooling)	"Config.ActivateCooling"	Bool	Bool	Установка этого флажка: • Сразу устанавливает "Config.Output.Heat.PidLowerLimit" = 0.0. • Устанавливает параметр "Config.ActivateCooling" в TRUE, вместо FALSE если не выбрано (=по умолчанию). • Активирует все другие органы управления "выходом (охлаждения)" (в "базовых настройках" и других представлениях). • Изменяет линию от символа ПИД к органам управления от серого к черному. • Флажок "This controller is a master" в разделе "Cascade" отключен. Примечание: Доступно только, если Вы не конфигурируете регулятор как ведущее устройство для каскада (флажок "This controller is a master" в разделе "Cascade" деактивирован; "Config.Cascade.IsMaster" = FALSE).

Настройка	ТО-DB параметр	Тип данных	Диапазон значений	Описание
Selection Output (cooling)	"Config.Output.Cool.Select"	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	Выбирает, какой тип выхода использовать для охлаждения. Возможные варианты: • "OutputCool" (Real) • "OutputCool_PWM" (Bool) (=по умолчанию) • "OutputCool_PER (аналог.)" (Word) Доступно только, если Вы выбираете опцию "Activate output (cooling)"; (Config.ActivateCooling = TRUE).
Output (cooling)	OutputCool, OutputCool_PER или OutputCool_PWM	Real или Int или Bool	Real, Int или Bool	Доступно только на странице свойств. Без многозначного управления и недоступно для редактирования в он-лайн режиме функционального представления.

Каскадные параметры

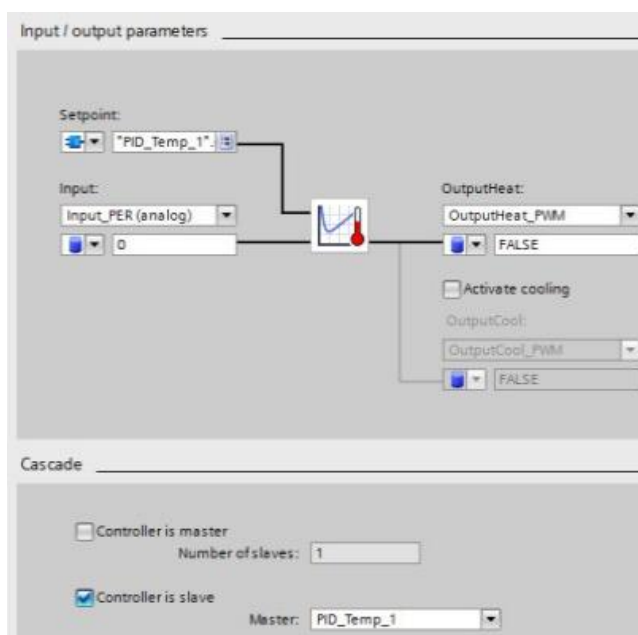
Следующие параметры позволяют Вам выбрать регуляторы в качестве ведущих или ведомых устройств и определить число ведомых регуляторов, которые получают уставку непосредственно от основного регулятора:

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
This controller is a master	"Config.Cascade.IsMaster"	Bool	Bool	Показывает, является ли этот регулятор ведущим устройством в каскаде. Когда Вы устанавливаете этот флажок, Вы выполняете следующее: - Устанавливаете параметр "Config.Cascade.IsMaster" в TRUE, вместо FALSE, если не выбрано (=по умолчанию). - Сразу устанавливаете "Selection Output (heating)" в разделе "Входных / выходных параметров" на "OutputHeat" (Config.Output.Heat.Select = 0). - Активируете поле ввода "Number of Slaves". - Снимаете флажок "Activate output (cooling)" в разделе "Входных / выходных параметров". Примечание: Доступно только, если выход охлаждения этого регулятора деактивирован (флажок "Activate output (cooling)" в разделе "Входные / выходные параметры" деактивирован (Config.ActivateCooling = FALSE).

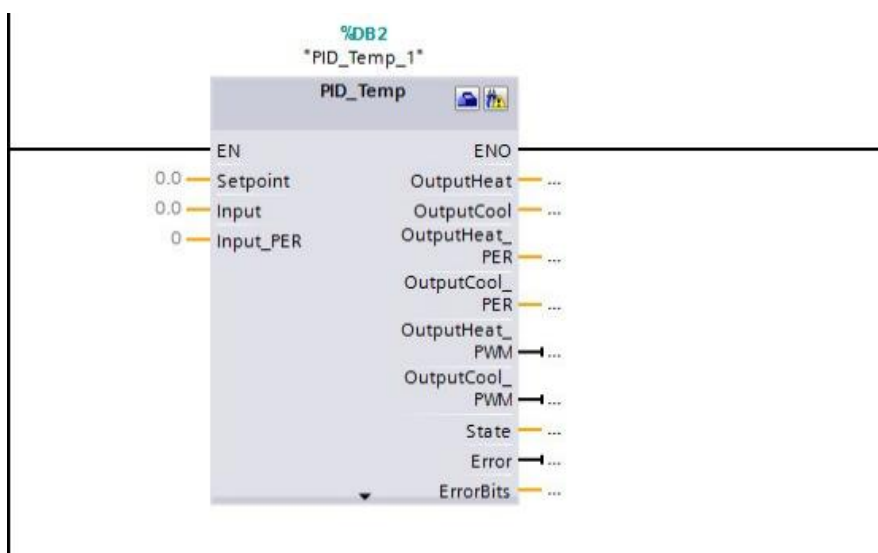
Настройка	ТО-DB параметр	Тип данных	Диапазон значений	Описание
Number of slaves	"Config.Cascade.CountSlaves"	Int	255 >= Config.Cascade.CountSlaves >= 1	Число ведомых регуляторов, которые получают уставку непосредственно от этого основного регулятора. Инструкция PID_Temp использует это значение наряду с другими для функции анти-насыщения. "Number of slaves" доступно только, если флажок "This controller is a master" установлен (Config.Cascade.IsMaster = TRUE).
This controller is a slave	"Config.Cascade.IsSlave"	Bool	Bool	Показывает, является ли этот регулятор ведомым в каскаде. Когда Вы устанавливаете этот флажок, Вы устанавливаете параметр "Config.Cascade.IsSlave" в TRUE, вместо FALSE, если не выбрано (=по умолчанию). Вы должны установить этот флажок на странице свойств, чтобы разрешить выпадающий список "SelectionMaster".

Пример: Каскадное включение регуляторов

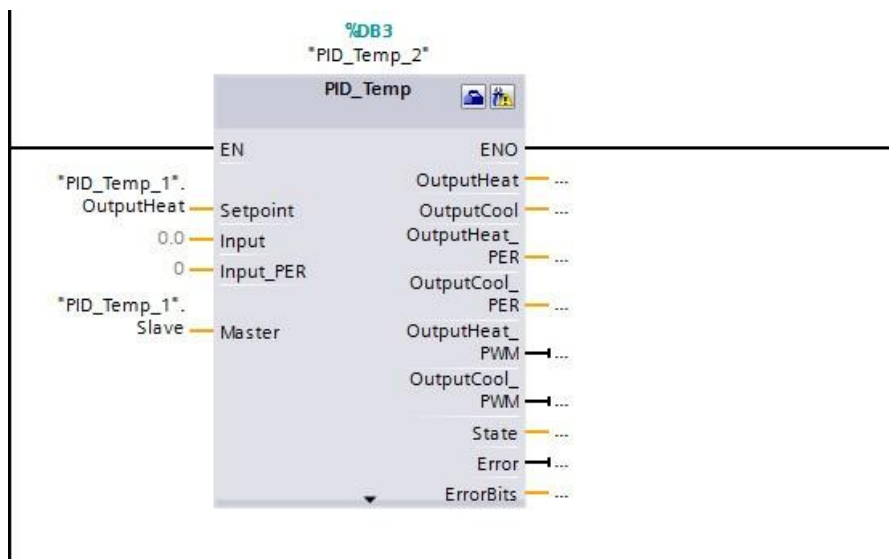
В диалоговом окне “Basic settings” ниже, Вы видите раздел “Input / output parameters” и раздел “Cascade” для ведомого регулятора “PID_Temp_2” после выбора “PID_Temp_1” в качестве ведущего устройства. Вы выполняете соединения между основным и ведомым регулятором:



Сегмент 1: В этих сегментах Вы выполняете соединения между ведущим “PID_Temp_1” и ведомым устройствами “PID_Temp_2” в программном редакторе:



Сегмент 2: Вы выполняете соединения между "OutputHeat" и "Slave" параметрами ведущего устройства "PID_Temp_1" с "Setpoint" и "Master" параметрами ведомого устройства "PID_Temp_2", соответственно:



Автонастройка температурных процессов

Инструкция PID_Temp предлагает два режима автонастройки:

- "Предварительная настройка" (параметр "Mode" = 1)
- "Точная настройка" (параметр "Mode" = 2)

В зависимости от конфигурации регулятора доступны различные варианты этих методов настройки:

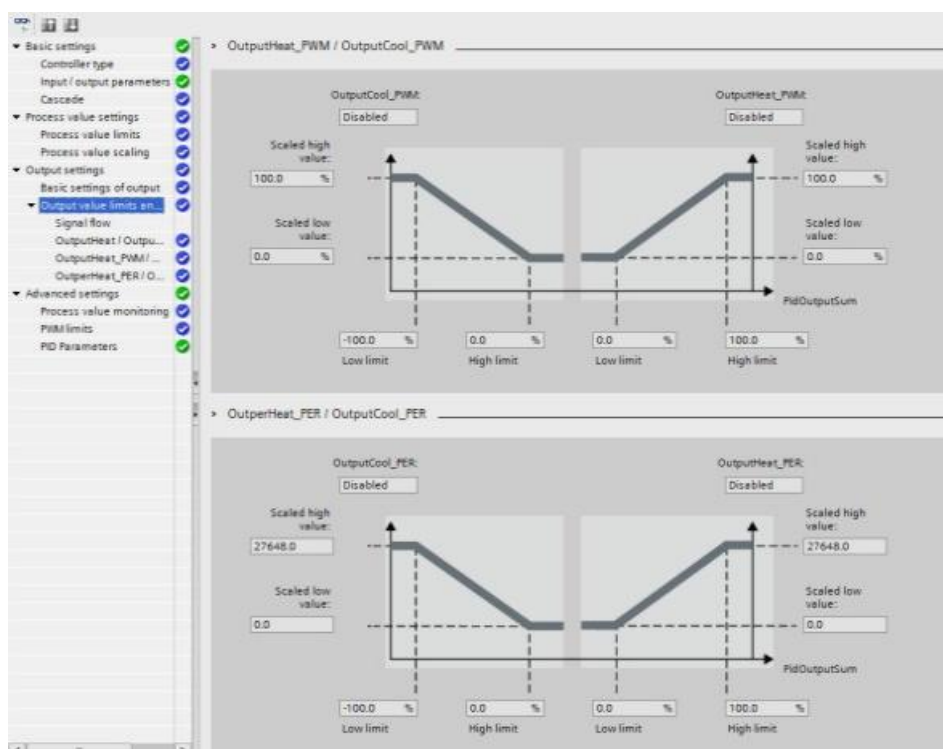
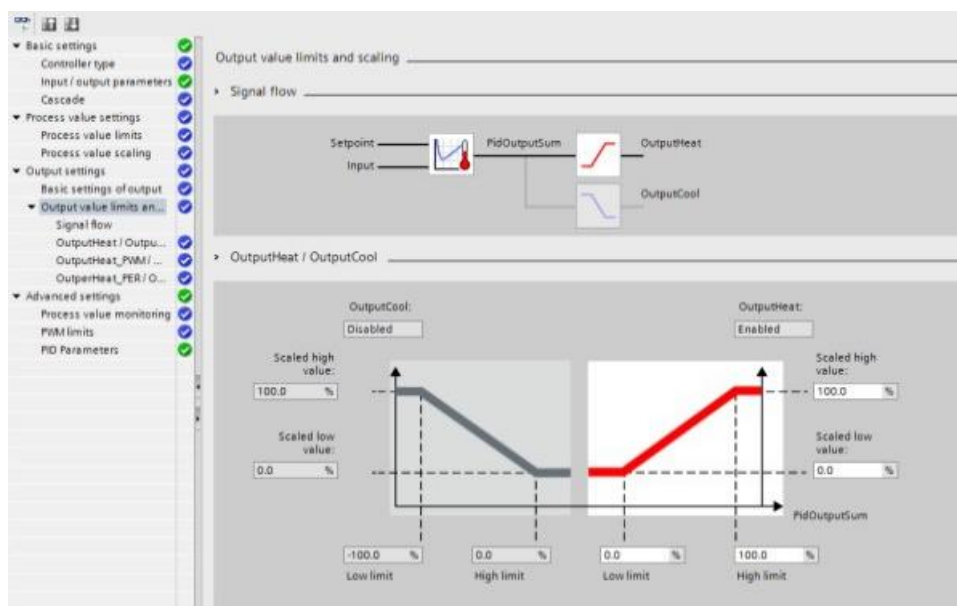
Конфигурация	Регулятор с выходом нагрева	Регулятор с выходом нагрева и охлаждения, использующий коэффициент охлаждения	Регулятор с выходом нагрева и охлаждения, использующий два набора ПИД-параметров
Значения в связанном TO-DB	- Config.ActivateCooling = FALSE - Config.AdvancedCooling = значение не важно	- Config.ActivateCooling = TRUE - Config.AdvancedCooling = FALSE	- Config.ActivateCooling = TRUE - Config.AdvancedCooling = TRUE
Доступные методы настройки	“Предварительная настройка нагрева” “Точная настройка нагрева” (смещение охлаждения не может использоваться)	“Предварительная настройка нагрева” “Точная настройка нагрева” (может использоваться смещение охлаждения)	“Предварительная настройка нагрева и охлаждения” “Предварительная настройка нагрева” “Предварительная настройка охлаждения” - Точная настройка нагрева” (может использоваться смещение охлаждения) - Точная настройка охлаждения” (может использоваться смещение нагрева)

Пределы и масштабирование выходного значения

Активация охлаждения отключена

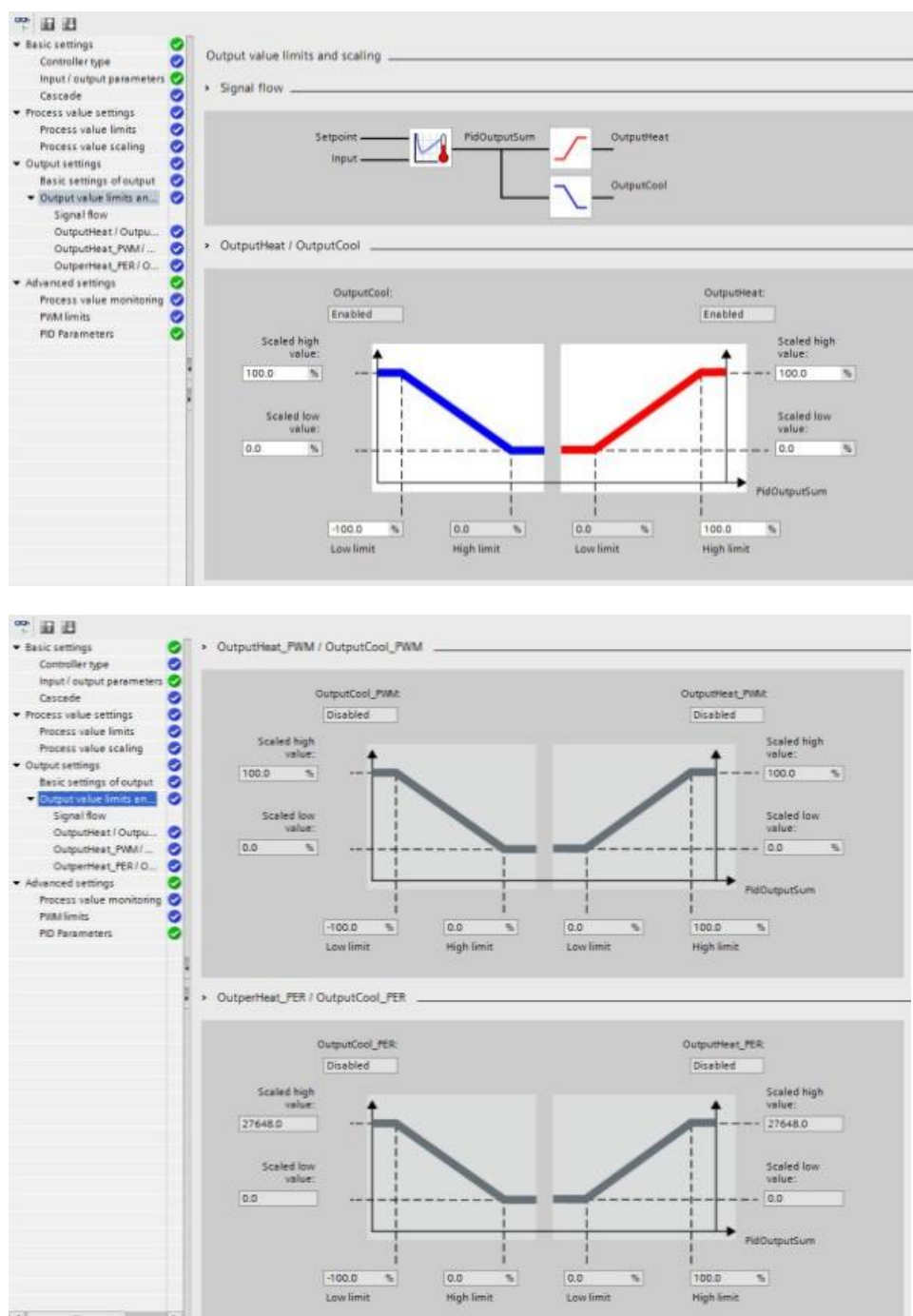
Если Вы конфигурируете инструкцию PID_Temp в качестве ведущего устройства для каскада, то флажок “Activate output (cooling)” в представлении “Basic settings” снят и деактивирован, все настройки в представлении “Output settings”, которые зависят от активации охлаждения также отключены.

На рисунке ниже показан раздел “Output value limits and scaling” с деактивированным охлаждением в представлении “Output settings” (OutputHeat_PWM выбран в представлении “Input / output parameters” и OutputHeat постоянно активирован):



Активация охлаждения разрешена

На рисунке ниже показан раздел "Output value limits and scaling" с активированным охлаждением в представлении "Output settings" (OutputCool_PER и OutputHeat_PWM выбраны в представлении "Input / output parameters"; OutputCool и OutputHeat активированы постоянно):



Рабочие режимы

Чтобы изменить режим работы вручную, пользователь должен установить in-out параметр регулятора "Mode" и активировать режим, переведя "ModeActivate" из FALSE в TRUE (инициированный нарастающий фронт). Вы должны сбросить "ModeActivate" перед следующим изменением режима; он не сбрасывается автоматически.

Выходной параметр "State" показывает текущий рабочий режим и установлен на требуемый "Режим", если это возможно. Параметр "State" не может быть изменен непосредственно; он изменяется только через параметр "Mode" или после автоматического изменения рабочего режима регулятором.

"Mode" / "State"	Название	Описание
0	Неактивный	Инструкция PID_Temp: - Деактивирует ПИД-алгоритм и широтно-импульсную модуляцию - Устанавливает в "0" (FALSE) все выходы контроллера (OutputHeat, OutputCool, OutputHeat_PWM, OutputCool_PWM, OutputHeat_PER, OutputCool_PER), независимо от сконфигурированных выходных пределов или смещений. Вы можете перейти в этот режим, установив "Mode" = 0, "Reset" = TRUE или при ошибке.
1	Предварительная настройка (настройка при запуске / SUT)	Этот режим определяет параметры во время первого запуска контроллера. В отличие от PID_Compact для PID_Temp Вы должны выбрать, нужна ли Вам настройка нагрева, настройка охлаждения или обе из них с помощью параметров "Heat.EnableTuning" and "Cool.EnableTuning". Вы можете активировать "предварительную настройку" из Неактивного, Автоматического или Ручного режимов. Если настройка прошла успешно, PID_Temp переключается в Автоматический режим. Если настройка прошла неудачно, переключение рабочего режима зависит от "ActivateRecoverMode".
2	Предварительная настройка (настройка в режиме выполнения / TIR)	Этот режим определяет оптимальную параметризацию ПИД-регулятора по уставке. В отличие от PID_Compact для PID_Temp Вы должны выбрать, нужна ли Вам настройка нагрева, настройка охлаждения или обе из них с помощью параметров "Heat.EnableTuning" and "Cool.EnableTuning". Вы можете активировать "Точную настройку" из Неактивного, Автоматического или Ручного режимов. Если настройка прошла успешно, PID_Temp переключается в Автоматический режим. Если настройка прошла неудачно, переключение рабочего режима зависит от "ActivateRecoverMode".
3	Автоматический режим	В Автоматическом режиме (стандартный режим ПИД-управления) результат ПИД-алгоритма определяет выходные значения. PID_Temp переключается в Неактивный режим, если происходит ошибка и "ActivateRecoverMode" = FALSE. Если ошибка происходит и "ActivateRecoverMode" = TRUE, переключение рабочего режима зависит от ошибки. Обратитесь к параметрам ErrorBit инструкции PID_Temp (стр. 502) для получения дополнительной информации.

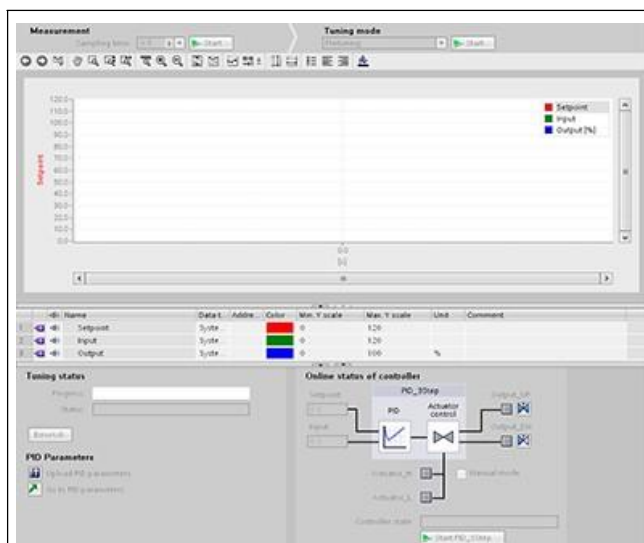
"Mode" / "State"	Название	Описание
4	Manual mode	В этом режиме ПИД-регулятор масштабирует, ограничивает и передает значение параметра "ManualValue" на выходы. ПИД-регулятор назначает "ManualValue" пропорционально ПИД-алгоритму (как "PidOutputSum"), так что его значение определяет будет ли активирован выход нагрева или охлаждения. Вы можете перейти в этот режим установкой "Mode" = 4 или "ManualEnable" = TRUE.
5	Substitute output value with error monitoring (Recover mode)	Вы можете активировать этот режим, установив "Mode" = 5. Режим представляет собой автоматическую реакцию регулятора на ошибку, если Автоматический режим активен в момент возникновения ошибки: • SetSubstituteOutput = FALSE (Последнее действительное выходное значение) • SetSubstituteOutput = TRUE (Значение, сохраненное в параметре "SubstituteOutput") Когда PID_Temp находится в "Автоматическом режиме" и параметр "ActivateRecoverMode" = TRUE, PID_Temp переходит в этот режим в случае следующих ошибок: • "Недопустимое значение в параметре "Input_PER". Проверьте наличие ошибки на аналоговом входе (например, обрыв провода)". (ErrorBits = DW#16#0002) • "Недопустимое значение в параметре "Input". Значение не является числом". (ErrorBits = DW#16#0200) • "Рассчитать выходное значения не удалось. Проверьте параметры ПИД". (ErrorBits = DW#16#0400) • "Недопустимое значение в параметре "Setpoint". Значение не является числом" (ErrorBits = DW#16#1000) Если ошибка больше не активна, то PID_Temp переключается назад в Автоматический режим.

10.2.13. Ввод в эксплуатацию регуляторов PID_Compact и PID_3Step

Используйте редактор ввода в эксплуатацию, чтобы сконфигурировать ПИД-регулятор для автоматической настройки при запуске и для автоматической настройки во время работы. Чтобы открыть редактор ввода в эксплуатацию, щелкните по значку 

Либо по инструкции в навигаторе проекта. 

Таблица 10- 33 Образец экрана ввода в эксплуатацию (PID_3Step)



- **Measurement:** Чтобы вывести на экран уставку, значение процесса (входное значение) и выходное значение в тренде реального времени, введите время выборки и нажмите кнопку "Start".
- **Tuning mode:** Чтобы настроить ПИД-контроль, выберите "Pretuning" или "Fine tuning" (ручная) и нажмите кнопку "Start". ПИД-регулятор выполняет несколько шагов, чтобы вычислить реакцию системы и время обновления. Соответствующие параметры настройки рассчитываются из этих значений.

После завершения процесса настройки Вы можете сохранить новые параметры, нажав кнопку "Upload PID parameters" в разделе "PID Parameters" редактора ввода в эксплуатацию.

Если во время настройки происходит ошибка, выходное значение ПИД переходит в 0. Режим ПИД в этом случае устанавливается в "неактивный". Состояние указывает на ошибку.

Управление стартовыми значениями ПИД

Вы можете отредактировать фактические значения параметров конфигурации ПИД так, чтобы поведение ПИД-регулятора могло быть оптимизировано в режиме онлайн.

Откройте "Technology objects" для своего ПИД-регулятора и его объект "Configuration". Чтобы получить доступ к управлению стартовыми значениями, нажмите "значок очков" в левом верхнем углу диалогового окна:



Вы можете теперь изменить значение любого из Ваших параметров конфигурации ПИД-регулятора, как показано на рисунке ниже.

Вы можете сравнить фактическое значение со стартовым значением в проекте (оффлайн) и стартовым значением ПЛК(онлайн) каждого параметра. Это необходимо, чтобы выявить он-лайн/офф-лайн различия Блока данных технологического объекта (ТО-DB) и быть проинформированным о значениях, которые будут использоваться в качестве текущих значений при следующем переходе ПЛК из Stop в Run. Кроме того, значок сравнения предоставляет визуальную индикацию, чтобы облегчить идентификацию он-лайн/офф-лайне различий:

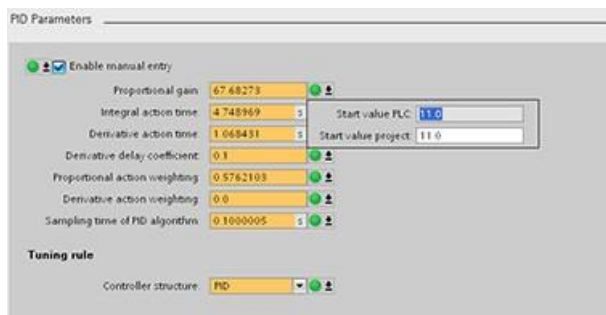
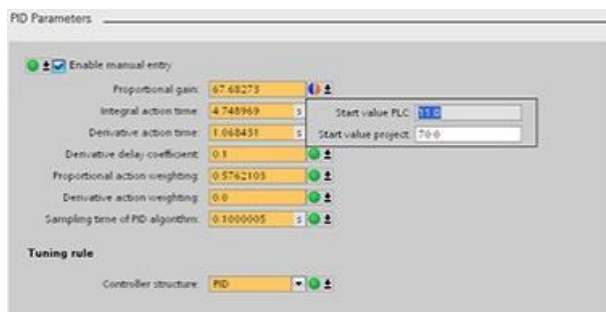


Рисунок выше представляет экран параметров ПИД со значками сравнения, показывающими, какие из значений различаются в он-лайн и офф-лайн проектах. Зеленый значок указывает на то, что значения одинаковы; синий / оранжевый значок указывает на то, что значения отличаются.

Дополнительно, нажмите кнопку параметра с направленной вниз стрелкой, чтобы открыть окно, которое отображает стартовое значение проекта (оффлайн) и стартовое значение ПЛК (онлайн) каждого параметра:

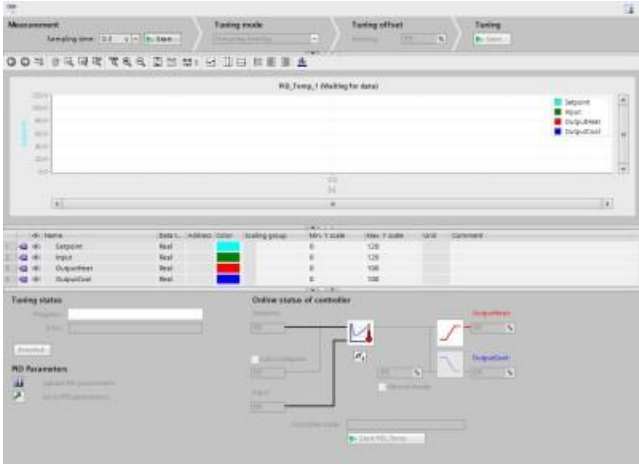


10.2.14. Ввод в эксплуатацию регулятора PID_Temp

Используйте редактор ввода в эксплуатацию, чтобы сконфигурировать ПИД-регулятор для автоматической настройки при запуске и для автоматической настройки во время работы. Чтобы открыть редактор ввода в эксплуатацию, щелкните по значку 

Либо по инструкции в навигаторе проекта. 

Таблица 10- 34 Образец экрана ввода в эксплуатацию (PID_Temp)

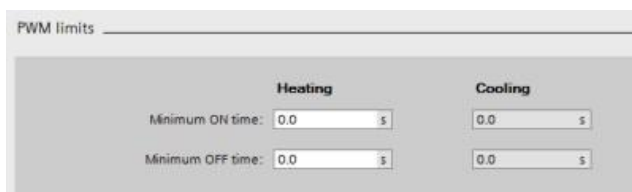
	Measurement: Чтобы вывести на экран уставку, значение процесса (входное значение) и выходное значение в тренде реального времени, введите время выборки и нажмите кнопку "Start". Tuning mode: Чтобы настроить контур PID_Temp, выберите "Pretuning" или "Fine tuning" (ручная) и нажмите кнопку "Start". ПИД-регулятор выполняет несколько шагов, чтобы вычислить реакцию системы и время обновления. Соответствующие параметры настройки рассчитываются из этих значений. После завершения процесса настройки Вы можете сохранить новые параметры, нажав кнопку "Upload PID parameters" в разделе "PID Parameters" редактора ввода в эксплуатацию. Если во время настройки происходит ошибка, выходное значение ПИД переходит в 0. Режим ПИД в этом случае устанавливается в "неактивный". Состояние указывает на ошибку.
--	---

PWM пределы

Исполнительные механизмы, которыми управляют с помощью программной ШИМ-функции регулятора PID_Temp, возможно, должны быть защищены от слишком короткой длительности импульса (например, тиристорный переключатель должен быть включен больше чем на 20 мс, прежде чем сможет вообще отреагировать); Вы назначаете минимальное время включения. Привод может также пренебречь короткими импульсами и поэтому негативно повлиять на качество управления. Может оказаться необходимым минимальное время выключения (например, чтобы предотвратить перегрев).

Чтобы отобразить пределы ШИМ, Вы должны открыть функциональное представление в конфигурации Технологических объектов (ТО) и выбрать "PWM limits" из узла "Advanced settings" в дереве навигации.

Если Вы открываете представление "PWM limits" в функциональном представлении и активируете контроль (кнопка "очки"), все средства управления отображают он-лайн значение из ТО-DB с оранжевым фоном и многозначное управление, и Вы можете отредактировать значения (если условия конфигурации выполнены; обратитесь к таблице ниже).



Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Minimum on time (heating) ^{1,2}	"Config.Output.Heat.MinimumOnTime"	Real	100000.0 ≥ "Config.Output.Heat.MinimumOnTime" ≥ 0.0	Длительность включения на "OutputHeat_PWM" никогда не короче, чем это значение.
Minimum off time (heating) ^{1,2}	"Config.Output.Heat.MinimumOffTime"	Real	100000.0 ≥ "Config.Output.Heat.MinimumOffTime" ≥ 0.0	Длительность выключения на OutputHeat_PWM никогда не короче, чем это значение.
Minimum on time (cooling) ^{1,3,4}	"Config.Output.Cool.MinimumOnTime"	Real	100000.0 ≥ Config.Output.Cool.MinimumOnTime ≥ 0.0	Длительность включения на OutputCool_PWM никогда не короче, чем это значение.
Minimum off time (cooling) ^{1,3,4}	"Config.Output.Cool.MinimumOffTime"	Real	100000.0 ≥ Config.Output.Cool.MinimumOffTime ≥ 0.0	Длительность выключения на OutputCool_PWM никогда не короче, чем это значение

- ¹ Поле выводит на экран "s" (секунды) в качестве единиц измерения времени.
- ² Если настройкой для Output (heating) в представлении "Basic settings" не является "OutputHeat_PWM" (Config.Output.Heat.Select = TRUE), Вы должны установить это значение в "0.0".
- ³ Если настройкой для Output (cooling) в представлении "Basic settings" не является "OutputCool_PWM" (Config.Output.Cool.Select = TRUE), Вы должны установить это значение в "0.0".
- ⁴ Доступно только, если Вы устанавливаете опцию "Activate output (cooling)" в представлении "Basic settings" (Config.ActivateCooling = TRUE).

Параметры ПИД

Представление “Advanced settings”, раздел “PID Parameters” показано ниже с деактивированной функцией охлаждения и/или “PID parameterswitchover”.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Enable manual entry	"Retain.CtrlParams.SetByUser"	Bool	Bool	Вы должны установить этот флажок, чтобы ввести параметры ПИД вручную.
Proportional gain (heating) ²	"Retain.CtrlParams.Heat.Gain"	Real	Усиление ≥ 0.0	Пропорциональное усиление ПИД для нагрева
Integral action time (heating) ^{1,2}	"Retain.CtrlParams.Heat.Ti"	Real	100000.0 $\geq T_i \geq 0.0$	Интегральное действие ПИД для нагрева.
Derivative action time (heating) ^{1,2}	"Retain.CtrlParams.Heat.Td"	Real	100000.0 $\geq T_d \geq 0.0$	Дифференциальное действие ПИД для нагрева.
Derivative delay coefficient (heating) ²	"Retain.CtrlParams.Heat.TdFiltRatio"	Real	$T_dFiltRatio \geq 0.0$	Коэффициент задержки дифференциального действия ПИД для нагрева, определяющий время запаздывания дифференциального действия как коэффициент от времени дифференцирования ПИД.
Proportional action weighting (heating) ²	"Retain.CtrlParams.Heat.PWeighting"	Real	$1.0 \geq PWeighting \geq 0.0$	Вес пропорционального усиления ПИД для нагрева в прямом, либо в замкнутом контуре управления.
Derivative action weighting (heating) ²	"Retain.CtrlParams.Heat.DWeighting"	Real	$1.0 \geq DWeighting \geq 0.0$	Вес дифференциального усиления ПИД для нагрева в прямом, либо в замкнутом контуре управления.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Sampling time of PID algorithm (heating) ^{1,2}	"Retain.CtrlParams. Heat.Cycle"	Real	100000.0 ≥ Cycle > 0.0	Внутренний цикл вызова ПИД-регулятора для нагрева. Округлен до целого кратного числу времени цикла вызова FB.
Deadband width(heating) ^{2,3}	"Retain.CtrlParams. Heat.DeadZone"	Real	DeadZone ≥ 0.0	Ширина мертвой зоны для рассогласования управления нагревом.
Control Zone (heating) ^{2,3}	"Retain.CtrlParams. Heat.ControlZone"	Real	ControlZone > 0.0	Ширина зоны рассогласования управления для нагрева, в которой ПИД-управление активно. Если отклонение управления выходит за этот диапазон, выход переключается на максимальные выходные значения. Значение по умолчанию "MaxReal", таким образом, зона контроля деактивирована, пока автоматическая настройка не выполняется. Значение "0.0" запрещено для зоны контроля; со значением "0.0", PID_Temp ведет себя как двухпозиционный регулятор, который всегда нагревается или охлаждается на полной мощности.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Controller structure (heating)	"PIDSelfTune.SUT.TuneRuleHeat", "PIDSelfTune.TIR.TuneRuleHeat"	Int	"PIDSelfTune.SUT.TuneRuleHeat" = 0..2, "PIDSelfTune.TIR.TuneRuleHeat" = 0..5	Вы можете выбрать алгоритм настройки для нагрева. Возможные варианты: - PID (Temperature) (=по умолчанию) ("PIDSelfTune.SUT.TuneRuleHeat" = 2) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) - PID ("PIDSelfTune.SUT.TuneRuleHeat" = 0) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) - PI ("PIDSelfTune.SUT.TuneRuleHeat" = 1) ("PIDSelfTune.TIR.TuneRuleHeat" = 4) Любая другая комбинация показывает "User defined", но "User defined" не предлагается к выбору. "PID (Temperature)" является новым для PID_Temp со специфическим методом предварительной настройки (SUT) для температурных процессов.
Proportional gain (cooling) ⁴	"Retain.CtrlParams.Cool.Gain"	Real	Gain >= 0.0	Пропорциональное усиление ПИД для охлаждения
Integral action time (cooling) ^{1,4}	"Retain.CtrlParams.Cool.Ti"	Real	100000.0 >=Ti >= 0.0	Интегральное действие ПИД для охлаждения
Derivative action time (cooling) ^{1,4}	"Retain.CtrlParams.Cool.Td"	Real	100000.0 >=Td >= 0.0	Дифференциальное действие ПИД для охлаждения
Derivative delay coefficient (cooling) ⁴	Retain.CtrlParams.Cool.TdFiltRatio"	Real	TdFiltRatio>= 0.0	Коэффициент задержки дифференциального действия ПИД для охлаждения, определяющий время запаздывания дифференциального действия как коэффициент от вре-

				мени дифференцирования ПИД.
Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Proportional action weighting (cooling) ⁴	"Retain.CtrlParams.Cool.PWeighting"	Real	1.0 >=PWeighting >= 0.0	Вес пропорционального усиления ПИД для охлаждения в прямом, либо в замкнутом контуре управления.
Derivative action weighting (cooling) ⁴	Retain.CtrlParams.Cool.DWeighting"	Real	1.0 >=DWeighting >= 0.0	Вес дифференциального усиления ПИД для нагрева в прямом, либо в замкнутом контуре управления.
Sampling time of PID algorithm (cooling) ^{1,4}	"Retain.CtrlParams.Cool.Cycle"	Real	100000.0 >=Cycle > 0.0	Внутренний цикл вызова ПИД-регулятора для охлаждения. Округлен до целого кратного числу времени цикла вызова FB.
Deadband width (cooling) ^{3,4}	"Retain.CtrlParams.Cool.DeadZone"	Real	DeadZone>= 0.0	Ширина мертвой зоны для рассогласования управления охлаждением.
Control Zone (cooling) ^{3,4}	"Retain.CtrlParams.Cool.ControlZone"	Real	ControlZone> 0.0	Ширина зоны рассогласования управления для охлаждения, в которой ПИД-управление активно. Если отклонение управления выходит за этот диапазон, выход переключается на максимальные выходные значения. Значение по умолчанию "MaxReal", таким образом, зона контроля деактивирована, пока автоматическая настройка не выполняется. Значение "0.0" запрещено для зоны контроля; со значением "0.0", PID_Temp ведет себя как двухпозиционный регулятор, который всегда нагревается или охлаждается на полной мощности.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Controller structure (cooling)	"PIDSelfTune.SUT.TuneRuleCool", "PIDSelfTune.TIR.TuneRuleCool"	Int	"PIDSelfTune.SUT.TuneRuleHeat" = 0..2, "PIDSelfTune.TIR.TuneRuleHeat" = 0..5	Вы можете выбрать алгоритм настройки для охлаждения. Возможные варианты: • PID (Temperature) (=по умолчанию) ("PIDSelfTune.SUT.TuneRuleCool" = 2) ("PIDSelfTune.TIR.TuneRuleCool" = 0) • PID ("PIDSelfTune.SUT.TuneRuleCool" = 0) ("PIDSelfTune.TIR.TuneRuleCool" = 0) • PI ("PIDSelfTune.SUT.TuneRuleCool" = 1) ("PIDSelfTune.TIR.TuneRuleCool" = 4) Любая другая комбинация показывает "User defined", но "User defined" не предлагается к выбору. "PID (Temperature)" является новым для PID_Temp со специфическим методом предварительной настройки (SUT) для температурных процессов. Доступно только, если Вы устанавливаете / выбираете следующие элементы: "Activate output (cooling)" в представлении "Basic settings" ("Config.ActivateCooling" = TRUE) и "PID parameter switchover" в представлении "Output settings" (Config.AdvancedCooling = TRUE).

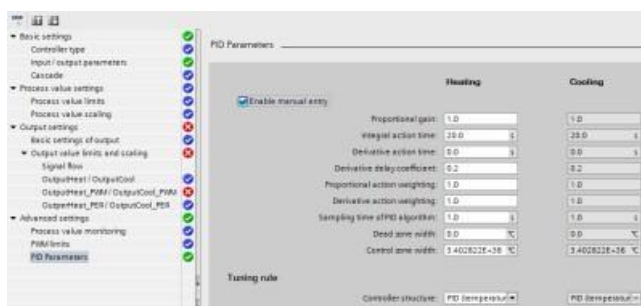
Настройка	ТО-DB параметр	Тип данных	Диапазон значений	Описание
-----------	----------------	------------	-------------------	----------

- 1 Поле выводит на экран "s" (секунды) в качестве единиц измерения времени.
- 2 Доступно только, если Вы выбираете "Enable manual entry" в ПИД-параметрах ("Retain.CtrlParams.SetByUser" = TRUE).
- 3 Единицы измерения, которые выбраны в представлении "Basic settings", отображаются в конце поля.
- 4 Доступно только, если Вы устанавливаете /выбираете следующие элементы: "Enable manual entry" в ПИД-параметрах ("Retain.CtrlParams.SetByUser" = TRUE), "Activate output (cooling)" в представлении "Basic settings" ("Config.ActivateCooling" = TRUE) и "PID parameter switchover" в представлении "Output settings" (Config.AdvancedCooling = TRUE).

Управление стартовыми значениями ПИД

Вы можете отредактировать фактические значения параметров конфигурации ПИД так, чтобы поведение ПИД-регулятора могло быть оптимизировано в режиме онлайн.

Откройте "Technology objects" для своего ПИД-регулятора и его объект "Configuration". Чтобы получить доступ к управлению стартовыми значениями, нажмите "значок очков" в левом верхнем углу диалогового окна:



Вы можете теперь изменить значение любого из Ваших параметров конфигурации ПИД-регулятора, как показано на рисунке ниже.

Вы можете сравнить фактическое значение со стартовым значением в проекте (оффлайн) и стартовым значением ПЛК(онлайн) каждого параметра. Это необходимо, чтобы выявить он-лайн/офф-лайн различия Блока данных технологического объекта (TO-DB) и быть проинформированным о значениях, которые будут использоваться в качестве текущих значений при следующем переходе ПЛК из Stop в Run. Кроме того, значок сравнения предоставляет визуальную индикацию, чтобы облегчить идентификацию он-лайн/офф-лайн различий:

Parameter	Heating	Cooling
Proportional gain:	1.0	1.0
Integral action time:	20.0 s	20.0 s
Derivative action time:	0.0 s	0.0 s
Derivative delay coefficient:	0.2	0.2
Proportional action weighting:	1.0	1.0
Derivative action weighting:	1.0	1.0
Sampling time of PID algorithm:	1.0 s	1.0 s
Dead zone width:	0.0 °C	0.0 °C
Control zone width:	3.402822E+38 °C	3.402822E+38 °C

Tuning rule

Controller structure: PD (temperatur)

Рисунок выше представляет экран параметров ПИД со значками сравнения, показывающими, какие из значений различаются в он-лайн и офф-лайн проектах. Зеленый значок указывает на то, что значения одинаковы; синий / оранжевый значок указывает на то, что значения отличаются.

Дополнительно, нажмите кнопку параметра с направленной вниз стрелкой, чтобы открыть окно, которое отображает стартовое значение проекта (оффлайн) и стартовое значение ПЛК (онлайн) каждого параметра:

Parameter	Heating	Cooling
Proportional gain:	1.0	1.0
Integral action time:	20.0 s	20.0 s
Derivative action time:	0.0 s	0.0 s
Derivative delay coefficient:	0.2	0.2
Proportional action weighting:	1.0	1.0
Derivative action weighting:	1.0	1.0
Sampling time of PID algorithm:	1.0 s	1.0 s
Dead zone width:	0.0 °C	0.0 °C
Control zone width:	3.402822E+38 °C	3.402822E+38 °C

Tuning rule

Controller structure: PD (temperatur)

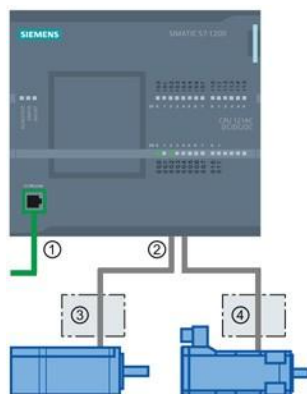
10.3 Управление перемещением

ЦПУ обеспечивает функциональность управления перемещением для работы шаговых двигателей и серводвигателей с импульсным интерфейсом. Функциональность управления перемещением включает в себя управление и контроль приводов.

- Технологический объект "Axis" конфигурирует механические данные привода, интерфейс привода, динамические параметры и другие свойства привода.
- Вы конфигурируете выходы ЦПУ для импульсов и направления для управления приводом.
- Ваша пользовательская программа использует команды управления перемещением, чтобы управлять осью и инициировать задачи перемещения.
- Используйте PROFINET-интерфейс, чтобы установить он-лайн соединение между ЦПУ и программатором. В дополнение к он-лайн функциям ЦПУ для управления перемещением доступны дополнительный ввод в эксплуатацию и диагностические функции.

Примечание

Изменения, которые Вы вносите в конфигурацию управления перемещением и загружаете в режиме RUN, не вступают в силу до перехода ЦПУ из режима STOP в режим RUN.



- ① PROFINET
- ② Импульсный выход и выход направления
- ③ Силовая секция для шагового двигателя
- ④ Силовая секция для серводвигателя

У DC/DC/DC вариантов ЦПУ S7-1200 есть встроенные выходы для прямого управления приводами. Релейные варианты ЦПУ требуют сигнальной платы с DC-выходами для управления приводами.

Сигнальная плата (SB) расширяет встроенный ввод-вывод, чтобы добавить несколько дополнительных каналов ввода-вывода. SB с двумя цифровыми выходами может использоваться в качестве выходов импульса и направления для управления одним двигателем. SB с четырьмя цифровыми выходами может использоваться для выходов импульсов и направления, чтобы управлять двумя двигателями. Встроенные релейные выходы не могут использоваться в качестве импульсных выходов управления двигателями. Используете ли Вы встроенный ввод-вывод или ввод-вывод SB или комбинацию обоих, у Вас может быть максимум четыре импульсных генератора.

Четырех импульсных генератора имеют назначения ввода-вывода по умолчанию; однако, они могут быть сконфигурированы на любом цифровом выходе ЦПУ или SB. Импульсные генераторы на ЦПУ не могут быть назначены SM или распределенному вводу-выводу.

Примечание

Выходы пачки импульсов не могут использоваться другими инструкциями в пользовательской программе

Когда Вы конфигурируете выходы ЦПУ или сигнальной платы, как импульсные генераторы (для использования с ШИМ или командами управления перемещением), соответствующие выходные адреса больше не управляют выходами. Если Ваша пользовательская программа пишет значение на выход, используемый в качестве импульсного генератора, ЦПУ не записывает это значение в физический выход.

Таблица 10- 35 Максимальное количество управляемых приводов

Тип ЦПУ		Встроенный ввод-вывод; SB не установлена		С установленной SB (2 x DC выхода)		С установленной SB (4 x DC выхода)	
		С направлением	Без направления	С направлением	Без направления	С направлением	Без направления
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

Примечание

Максимальное количество импульсных генераторов равняется четырем

Используете ли Вы встроенный ввод-вывод, ввод-вывод SB или комбинацию обоих, у Вас может быть максимум четырех импульсных генератора.

Таблица 10- 36 Выход ЦПУ: максимальная частота

ЦПУ	Выходной канал ЦПУ	Выход импульсов и направления	A/B, квадратура, вверх / вниз и имп./направление
1211C	от Qa.0 до Qa.3	100 кГц	100 кГц
1212C	от Qa.0 до Qa.3	100 кГц	100 кГц
	Qa.4, Qa.5	20 кГц	20 кГц
1214C и 1215C	от Qa.0 до Qa.3	100 кГц	100 кГц
	от Qa.4 до Qb.1	20 кГц	20 кГц
1217C	от DQa.0 до DQa.3 (от .0+, .0- до .3+, .3-)	1 МГц	1 МГц
	от DQa.4 до DQb.1	100 кГц	100 кГц

Таблица 10- 37 Выход сигнальной платы SB : максимальная частота (дополнительная плата)

Сигнальная плата	Выходной канал SB	Выход импульсов и направления	A/B, квадратура, вверх / вниз и имп./направление
SB 1222, 200 kHz	от DQe.0 до DQe.3	200 кГц	200 кГц
SB 1223, 200 kHz	DQe.0, DQe.1	200 кГц	200 кГц
SB 1223	DQe.0, DQe.1	20 кГц	20 кГц

Таблица 10- 38 Предельные частоты импульсных выходов

Импульсный выход	Частота
Встроенный	4 РТО: $2 \text{ Гц} \leq f \leq 1 \text{ МГц}$, 4 РТО: $2 \text{ Гц} \leq f \leq 100 \text{ кГц}$ или любая комбинация этих значений для 4 РТОs. ^{1 2}
Стандартная SB	$2 \text{ Гц} \leq f \leq 20 \text{ кГц}$
Высокоскоростные SB	$2 \text{ Гц} \leq f \leq 200 \text{ кГц}$

¹ Смотри таблицу ниже для четырех возможных комбинаций скорости выхода CPU 1217C.

² Смотри таблицу ниже для четырех возможных комбинаций скорости выхода CPU 1211C, CPU 1212C, CPU 1214C или CPU 1215C.

Пример: Конфигурации скорости импульсного выхода CPU 1217C

Примечание

CPU 1217C может генерировать импульсные выходы до 1 МГц, используя встроенные дифференциальные выходы.

Примеры ниже демонстрируют четыре возможных комбинации скорости выхода:

- Пример 1: 4 - 1 МГц РТО, без выхода направления
- Пример 2: 1 - 1 МГц, 2 - 100 кГц и 1 - 20 кГц РТО все с выходом направления
- Пример 3: 4 - 200 кГц РТО, без выхода направления
- Пример 4: 2 - 100 кГц РТО и 2 - 200 кГц РТО, все с выходом направления

P = Импульс D = Направление		Встроенные выходы ЦПУ										Выходы высоко- скоростной SB				Выходы станд. SB	
		Выходы 1 МГц (Q)				Выходы 100 кГц (Q)						Выходы 200 кГц (Q)				Выходы 20 кГц (Q)	
		0.0+	0.1+	0.2+	0.3+	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0-	0.1-	0.2-	0.3-												
Пр. 1: 4 - 1 МГц (без выхода направ.)	РТО1	P															
	РТО2		P														
	РТО3			P													
	РТО4				P												
Пр. 2: 1 - 1 МГц; 2 - 100 и 1 - 20 кГц (все с выходом направ.)	РТО1	P	D														
	РТО2					P	D										
	РТО3							P	D								
	РТО4															P	D
Пр. 3: 4 - 200 кГц (без выхода направ.)	РТО1											P					
	РТО2												P				
	РТО3													P			
	РТО4														P		
Пр. 4: 2 - 100 кГц; 2 - 200 кГц (все с выхо- дом на- прав.)	РТО1					P	D										
	РТО2							P	D								
	РТО3											P	D				
	РТО4													P	D		

Пример: Конфигурации скорости импульсного выхода CPU 1211C, CPU 1212C, CPU 1214C и CPU 1215C

Примеры ниже демонстрируют четыре возможных комбинации скорости выхода:

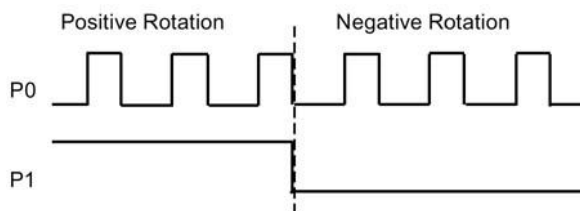
- Пример 1: 4 - 100 кГц РТО, без выхода направления
- Пример 2: 2 - 100 кГц РТО и 2 - 20 кГц РТО, все с выходом направления
- Пример 3: 4 - 200 кГц РТО, без выхода направления
- Пример 4: 2 - 100 кГц РТО и 2 - 200 кГц РТО, все с выходом направления

P = Импульс D = Направление		Встроенные выходы ЦПУ										Выходы высоко- скоростной SB				Выходы станд. SB	
		Выходы 100 кГц (Q)				Выходы 20 кГц (Q)						Выходы 200 кГц (Q)				Выходы 20 кГц (Q)	
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		CPU 1211C															
		CPU 1212C				CPU 1212C											
		CPU 1214C				CPU 1214C		CPU 1214C									
		CPU 1215C				CPU 1215C		CPU 1215C									
Пр. 1: 4 - 100 кГц (без выхода направ.)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Пр. 2: 2 - 100 кГц; 2 - 20 кГц (все с выходом направ.)	PTO1	P	D														
	PTO2			P	D												
	PTO3					P	D										
	PTO4							P	D								
Пр. 3: 4 - 200 кГц (без выхода направ.)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Пр.. 4: 2 - 100 кГц; 2 - 200 кГц (все с выходом направ.)	PTO1	P	D														
	PTO2			P	D												
	PTO3											P	D				
	PTO4													P	D		

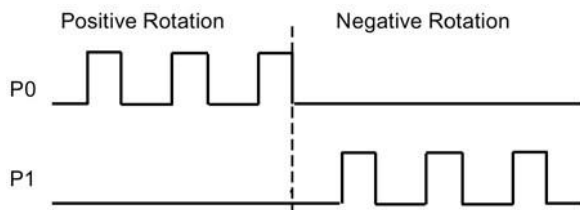
10.3.1. Фазирование

У Вас есть четыре опции для интерфейса "Phasing" для шагового/серво- привода. Эти следующие опции:

- РТО (импульс А и направление В): Если Вы выбираете РТО (импульс А и направление В) опцию, то один выход (P0) управляет подачей импульсов, а другой выход (P1) управляет направлением. P1 находится в состоянии высокого уровня (активен), если импульсы следуют в положительном направлении. P1 находится в состоянии низкого уровня (неактивен), если импульсы следуют в отрицательном направлении:



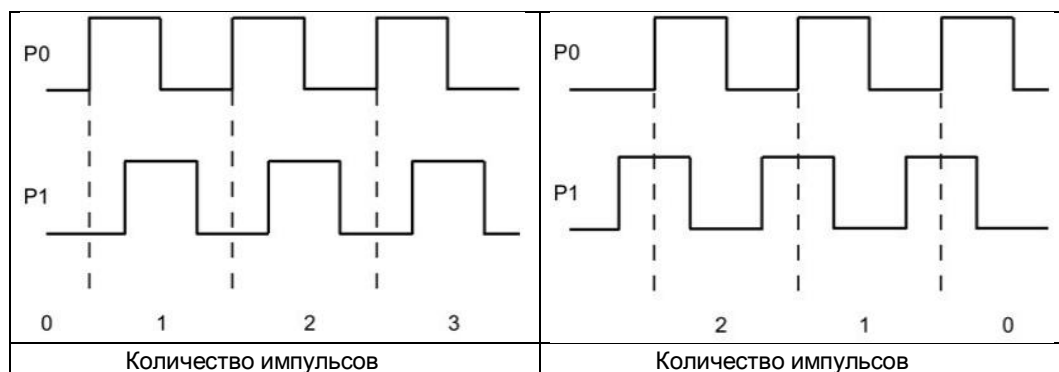
- РТО (счет вверх А и счет вниз В): Если Вы выбираете РТО (счет вверх А и счет вниз В) опцию, то один выход (P0) выдает импульсы для положительных направлений, а дифференциальный выход (P1) выдает импульсы для отрицательных направлений:



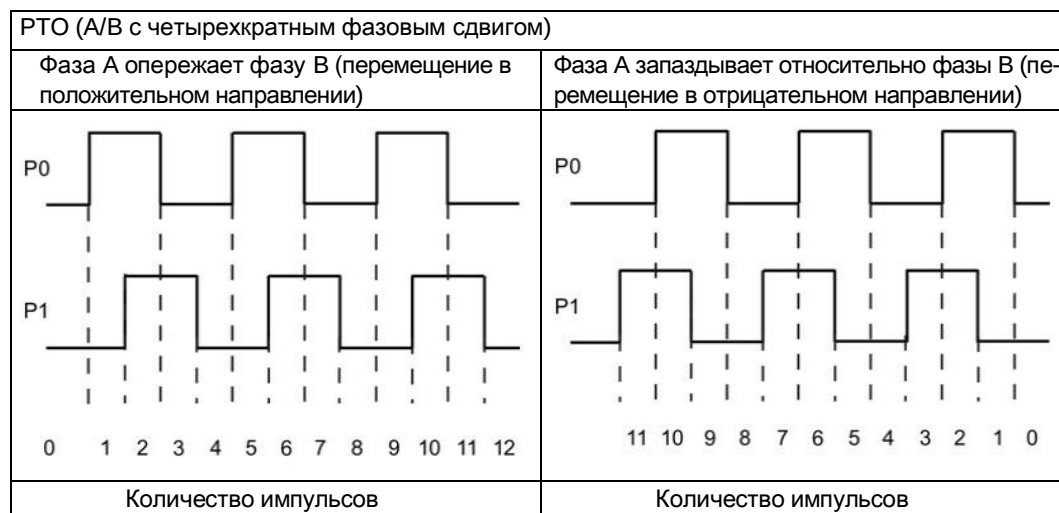
- РТО (А/В с фазовым сдвигом): Если Вы выбираете РТО (А/В с фазовым сдвигом) опцию, то оба выхода выдают импульсы на определенной скорости, но не совпадают по фазе на 90 градусов. Это 1X конфигурация, означающая, что один импульс соответствует времени между положительными фронтами P0. В этом случае направление определяется по тому выходу, фронт которого придет вначале. P0 опережает P1 для положительного направления. P1 опережает P0 для отрицательного направления.

Количество сгенерированных импульсов базируется на числе переходов от 0 к 1 Фазы А. Фазовое соотношение определяет направление перемещения:

РТО (А/В с фазовым сдвигом)	
Фаза А опережает фазу В (перемещение в положительном направлении)	Фаза А запаздывает относительно фазы В (перемещение в отрицательном направлении)



- РТО (А/В с четырехкратным фазовым сдвигом): Если Вы выбираете РТО (А/В с четырехкратным фазовым сдвигом) опцию, то оба выхода выдают импульсы на определенной скорости, но несовпадающие по фазе на 90 градусов. Четырехкратная 4X конфигурация, означает, что один импульс представляет собой переход каждого выхода (как положительный, так и отрицательный). В этом случае направление определяется по выходу, который переходит в состояние высокого уровня первым. Р0 опережает Р1 для положительного направления. Р1 опережает Р0 для отрицательного направления. Четырехкратная конфигурация основана на положительных и отрицательных переходах как фазы А, так и фазы В. Вы конфигурируете число переходов. Фазовое соотношение (А опережает В или В опережает А) определяет направление перемещения.



- РТО (импульс и направление (выбор направления отменен)): Если Вы отменяете выбор выхода направления в РТО (выбор направления отменен), то выход (Р0) управляет выдачей пульсов. Выход Р1 не используется и доступен для другого использования программой. Только команды положительного перемещения принимаются ЦПУ в этом режиме. Управление перемещением ограничивает Вас в создании недопустимых отрицательных конфигураций, когда Вы выбираете этот режим. Вы можете сохранить выход, если Ваше приложение перемещением работает только с одним направлением. Единственная фаза (один выход) показана на рисунке ниже (предполагая положительную полярность):



10.3.2. Конфигурирование импульсного генератора

1. Добавьте технологический объект:

- В дереве Проекта разверните узел "Technology Objects" и выберите "Add new object".
- Выберите значок "Axis" (переименуйте при необходимости), и нажмите "ОК", чтобы открыть редактор конфигурации для объекта оси.
- Отобразите свойства "Select PTO for Axis Control" под "Basic parameters" и выберите желаемый импульс.

Примечание

Если PTO не был ранее сконфигурирован в свойствах ЦПУ, PTO конфигурируется для использования одного из встроенных выходов.

Если Вы используете сигнальную плату выходов, то нажмите кнопку "Device configuration", чтобы перейти к свойствам ЦПУ. Под "Parameter assignment", в "Pulse options" сконфигурируйте как источник выходного сигнала выход сигнальной платы.

- Сконфигурируйте оставшиеся базовые и расширенные параметры.

2. Запрограммируйте Ваше приложение: Вставьте инструкцию MC_Power в кодовый блок.

- Для входа Axis выберите технологический объект оси, который Вы создали и сконфигурировали.
- Установка входа Enable в TRUE позволяет другим инструкциям перемещения функционировать.
- Установка входа Enable в FALSE отменяет другие инструкции перемещения.

Примечание

Используйте только одну инструкцию MC_Power на ось

3. Вставьте другие инструкции перемещения, чтобы выполнить требуемое движение.

Примечание

Конфигурирование импульсного генератора для выходов сигнальной платы: Выберите свойства "Pulse generators (PTO/PWM)" для ЦПУ (в Конфигурации устройства) и включите импульсный генератор. Два импульсных генератора доступны для каждого ЦПУ S7-1200 V1.0, V2.0, V2.1 и V2.2.

ЦПУ S7-1200 V3.0 и V4.0 имеет четыре импульсных генератора в наличии. В той же области конфигурации под "Pulse options", выберите генератор импульсов, используемый как: "PTO".

Примечание

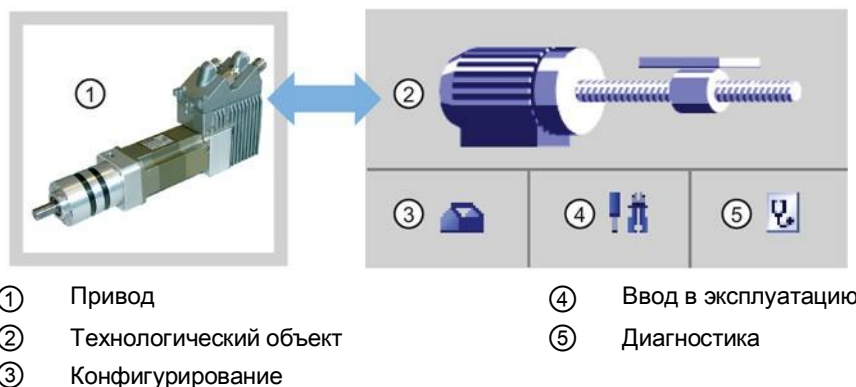
ЦПУ обчисляет задачи перемещения в "секциях" или сегментах по 10 мс. Пока одна секция выполняется, следующая секция ожидает выполнения в очереди. Если Вы прерываете задачу перемещения на оси (выполняя другую новую задачу перемещения для этой оси), новая задача перемещения может не выполняться максимум в течение 20 мс (остаток от текущей секции плюс ожидающая в очереди секция).

10.3.3. Управление перемещением с разомкнутой обратной связью

10.3.3.1. Конфигурирование оси

Вы подключаете ось разомкнутого цикла к ПЛК и управляете ей посредством РТО (Выход пачки импульсов).

STEP 7 обеспечивает инструменты конфигурирования, инструменты ввода в эксплуатацию и инструменты диагностики для технологического объекта "Axis".



Примечание

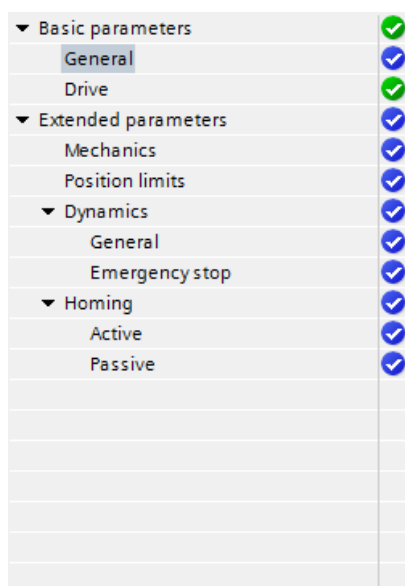
Для ЦПУ со встроенным ПО выпуска V2.2 и более ранних, РТО требует внутренней функциональности высокоскоростного счетчика (HSC). Это означает, что соответствующий HSC не может использоваться в другом месте.

Привязка между РТО и HSC фиксированная. Если РТО1 будет активирован, то он будет соединен с HSC1. Если РТО2 будет активирован, то он будет соединен с HSC2. Вы не можете контролировать текущее значение (например, в ID1000), когда приходят импульсы.

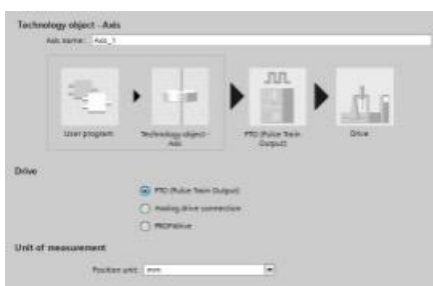
У ЦПУ S7-1200 V3.0 и более поздних этого ограничения нет; когда импульсные выходы сконфигурированы в этих ЦПУ, все HSC остаются доступными для использования в программе.

Таблица 10- 39 STEP 7 инструменты для управления перемещением

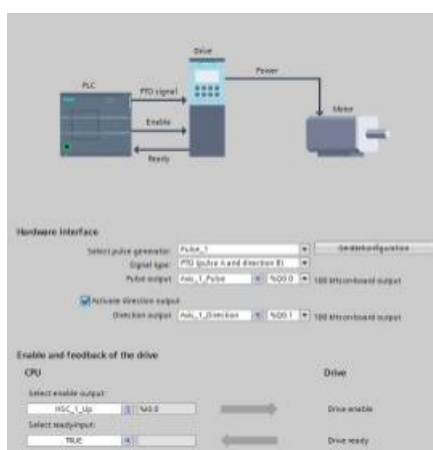
Инструмент	Описание
Configuration	Конфигурирует следующие свойства технологического объекта "Axis": • Выбор РТО, который будет использоваться и конфигурирование интерфейса привода • Свойства механики и передаточное отношение привода (или машины, или системы) • Свойства для ограничителей положения, динамики и возврата в начальное положение Сохраните конфигурацию в блоке данных технологического объекта.
Commissioning	Тестирует функцию Вашей оси без необходимости создания пользовательской программы. Когда инструмент будет запущен, будет отображена панель управления. Следующие команды доступны на панели управления: • Включать и выключать ось • Перемещать ось в старт-стопном режиме • Позиционировать ось в абсолютных и относительных величинах • Переводить ось в начальное положение • Квотировать ошибки Для команд перемещения могут быть определены скорость и ускорение / замедление . Панель управления также показывает текущее состояние оси.
Diagnostics	Контролирует текущее состояние и информацию об ошибках для оси и привода.



Древовидный селектор для PTO оси не включает конфигурационные меню Encoder, Modulo, Position monitoring и Control loop.



После того, как Вы создадите технологический объект для оси, Вы конфигурируете ось, определяя основные параметры, такие как PTO и конфигурация интерфейса привода. Вы также конфигурируете другие свойства оси, такие как ограничители, динамика и начальное положение.

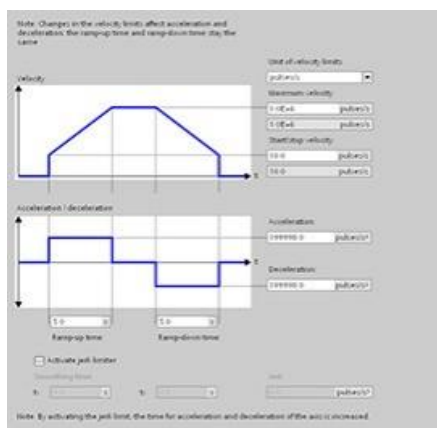


Примечание

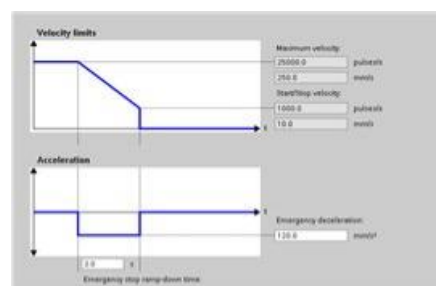
Вам, вероятно, придется адаптировать значения входных параметров инструкций управления перемещением к новым единицам размерности в пользовательской программе.



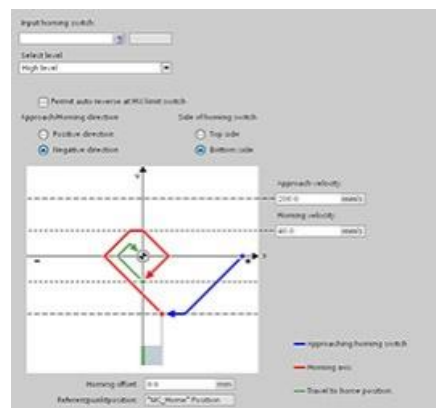
Сконфигурируйте свойства для сигналов привода, управляйте механикой и контролем положения (аппаратные и программные ограничители).



Вы конфигурируете динамику перемещения и поведение при команде аварийной остановки.



Вы также конфигурируете поведение при переходе в начальное положение (пассивное и активное).



Используйте панель управления "Commissioning", чтобы протестировать функциональность независимо от Вашей пользовательской программы.



Щелкните по значку "Startup", чтобы ввести ось в эксплуатацию.

Панель управления показывает текущий статус оси. Кроме того, что Вы можете включить и выключить ось, Вы можете также протестировать позиционирование оси (в абсолютных и относительных значениях), а также можете определить скорость, ускорение и замедление. Вы можете также протестировать задач перехода в начальное положение и старт-стопный режим. Панель управления также позволяет Вам квитировать ошибки.

10.3.3.2. Ввод в эксплуатацию

Диагностическая функция "Status and error bits"

Используйте диагностическую функцию "Status and error bits", чтобы контролировать наиболее важные сообщения о состоянии и ошибках для оси. Диагностический дисплей функции доступен он-лайн в режиме "Manual control" и "Automatic control", когда ось активна.

Таблица 10- 40 Состояние оси

Состояние	Описание
Включена	Ось включена и готова к управлению через задачи управления перемещением. (Тег технологического объекта: <Axis name>.StatusBits.Enable)
Находится в начальном положении	Ось возвращена в начальное положение и способна к выполнению задач инструкции управления перемещением "MC_MoveAbsolute" по абсолютному позиционированию. Ось не должна переводиться в относительное начальное положение. Особые ситуации: - Во время активного перевода в начальное положение, состояние соответствует FALSE. - Если ось подвергается пассивному перемещению в начальное положение, то состояние установлено в TRUE во время пассивного перемещения. (Тег технологического объекта: <Axis name>.StatusBits.HomingDone)
Ошибка	Ошибка произошла в технологическом объекте "Axis". Больше информации об ошибке доступно при автоматическом управлении в параметрах ErrorID и ErrorInfo команд управления перемещением. В ручном режиме поле "Last error" панели управления отображает подробную информацию о причине ошибки. (Тег технологического объекта: <Axis name>.StatusBits.Error)
Панель управления активна	Режим "Manual control" был включен в панели управления. Панель управления имеет приоритет над технологическим объектом "Axis". Осью нельзя управлять из пользовательской программы. (Тег технологического объекта: <Axis name>.StatusBits.ControlPanelActive)

Таблица 10- 41 Состояние привода

Состояние	Описание
Привод готов	Привод готов к работе. (Тег технологического объекта: <Axis name>.StatusBits.DriveReady)
Error	Привод сообщил об ошибке после сбоя его сигнала готовности. (Тег технологического объекта: <Axis name>.ErrorBits.DriveFault)

Таблица 10- 42 Состояние перемещения оси

Состояние	Описание
Состояние покоя	Ось в состоянии покоя. (Тег технологического объекта: <Axis name>.StatusBits.StandStill)
Ускорение	Ось ускоряется. (Тег технологического объекта: <Axis name>.StatusBits.Acceleration)
Постоянная скорость	Ось перемещается с постоянной скоростью. (Тег технологического объекта: <Axis name>.StatusBits.ConstantVelocity)
Замедление	Ось замедляется (уменьшает скорость). (Тег технологического объекта: <Axis name>.StatusBits.Deceleration)

Таблица 10- 43 Состояние режима перемещения

Состояние	Описание
Позиционирование	Ось выполняет задачу позиционирования от инструкций управления перемещением "MC_MoveAbsolute" или "MC_MoveRelative" или от панели управления. (Тег технологического объекта: <Axis name>.StatusBits.PositioningCommand)
Команда скорости	Ось выполняет задачу установки скорости от инструкции управления перемещением "MC_MoveVelocity" или "MC_MoveJog" или от панели управления. (Тег технологического объекта: <Axis name>.StatusBits.SpeedCommand)
Перемещение в начальное положение	Ось выполняет задачу перемещения в начальное положение от инструкции управления перемещением "MC_Home" или от панели управления. (Тег технологического объекта: <Axis name>.StatusBits.Homing)

Таблица 10- 44 Биты ошибок

Ошибка	Описание
Минимальное программное ограничение достигнуто	Нижний программный ограничитель был достигнут. (Тег технологического объекта: <Axis name>.ErrorBits.SwLimitMinReached)
Минимальное программное ограничение превышено	Нижний программный ограничитель был превышен. (Тег технологического объекта: <Axis name>.ErrorBits.SwLimitMinExceeded)
Максимальное программное ограничение достигнуто	Верхний программный ограничитель был достигнут. (Тег технологического объекта: <Axis name>.ErrorBits.SwLimitMaxReached)
Максимальное программное ограничение превышено	Верхний программный ограничитель был превышен. (Тег технологического объекта: <Axis name>.ErrorBits.SwLimitMaxExceeded)
Отрицательное аппаратное ограничение	Нижний аппаратный ограничитель сработал. (Тег технологического объекта: <Axis name>.ErrorBits.HwLimitMin)
Положительное аппаратное ограничение	Верхний аппаратный ограничитель сработал. (Тег технологического объекта: <Axis name>.ErrorBits.HwLimitMax)
РТО уже используется	Вторая ось использует тот же РТО и включена с помощью "MC_Power". (Тег технологического объекта: <Axis name>.ErrorBits.HwUsed)

Ошибка	Описание
Ошибка конфигурации	Технологический объект "Axis" был неправильно сконфигурирован, или доступные для редактирования данные конфигурации были изменены неправильно во время работы пользовательской программы. (Тег технологического объекта: <Axis name>.ErrorBits.ConfigFault)
Общая ошибка	Произошла внутренняя ошибка. (Тег технологического объекта: <Axis name>.ErrorBits.SystemFault)

Диагностическая функция "Motion status"

Используйте диагностическую функцию "Motion status", чтобы контролировать состояние перемещения оси. Диагностический дисплей функции доступен он-лайн в "Manual control" и в "Automatic control" режиме, когда ось активна.

Таблица 10- 45 Состояние перемещения

Ошибка	Описание
Заданное положение	Поле "Target position" указывает текущее заданное положение активной задачи позиционирования от инструкции управления перемещением "MC_MoveAbsolute" или "MC_MoveRelative" или от панели управления. Значение "Target position" допустимо только во время выполнения задачи позиционирования. (Тег технологического объекта: <Axis name>.MotionStatus.TargetPosition)
Текущее положение	Поле "Current position" указывает текущее положение оси. Если ось не в начальном положении, значение указывает положение оси относительно положения включения оси. (Тег технологического объекта: <Axis name>.MotionStatus.Position)
Текущая скорость	Поле "Current velocity" указывает текущую скорость оси. (Тег технологического объекта: <Axis name>.MotionStatus.Velocity)

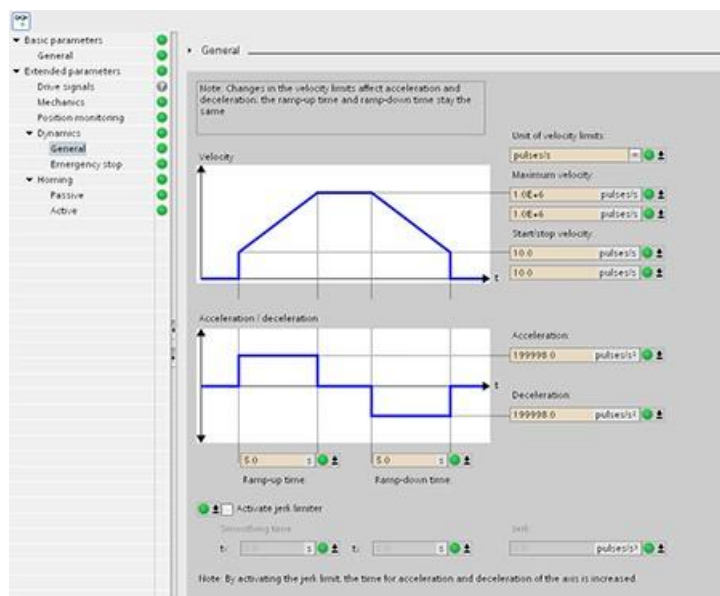
Таблица 10- 46 Динамические ограничения

Динамическое ограничение	Описание
Скорость	Поле "Velocity" указывает сконфигурированную максимальную скорость оси. (Тег технологического объекта: <Axis name>.Config.DynamicLimits.MaxVelocity)
Ускорение	Поле "Acceleration" указывает текущее сконфигурированное ускорение оси. (Тег технологического объекта: <Axis name>.Config.DynamicDefaults.Acceleration)
Замедление	Поле "Deceleration" указывает текущее сконфигурированное замедление оси. (Тег технологического объекта: <Axis name>.Config.DynamicDefaults.Deceleration)

Управление стартовыми значениями перемещения

Вы можете отредактировать фактические значения параметров конфигурации перемещения, то есть поведение процесса может быть оптимизировано в режиме он-лайн.

Откройте "Technology objects" для своего управления перемещением и его объект "Configuration". Чтобы получить доступ к управлению стартовыми значениями, нажмите "символ очков" в левом верхнем углу диалогового окна:



Вы можете теперь изменить значение любого из Ваших параметров конфигурации управления перемещением, как показано на рисунке ниже.

Вы можете сравнить фактическое значение со стартовым значением проекта (офф-лайн) и стартовым значением ПЛК (он-лайн) каждого параметра. Это необходимо, чтобы определить он-лайн / офф-лайн различия Блока данных технологического объекта (ТО-DB) и быть информированным о значениях, которые будут использоваться в качестве текущих значений при следующем переходе ПЛК из Stop в Run. Кроме того, значок сравнения предоставляет визуальную индикацию, чтобы облегчить идентификацию он-лайн/офф-лайн различий.

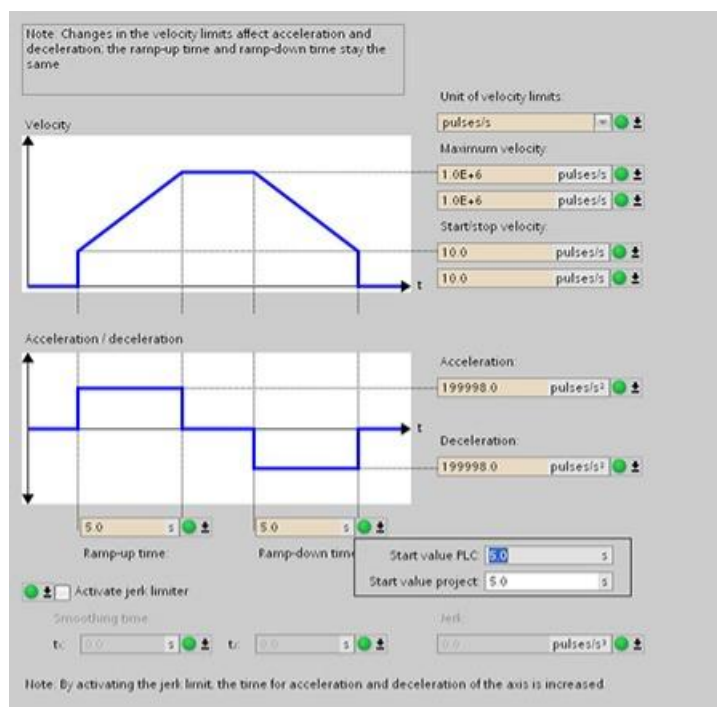
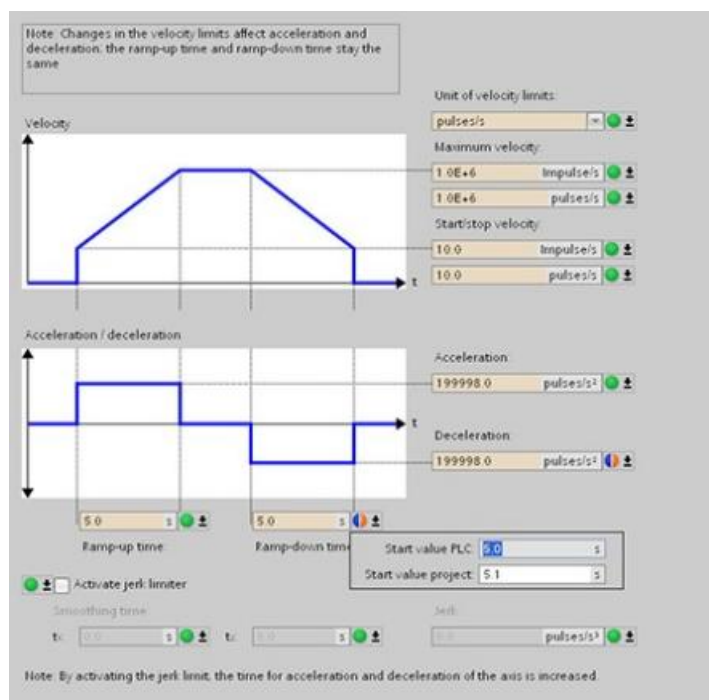


Рисунок выше показывает экран параметра перемещения со значками сравнения, отображающими отличия в значениях между он-лайн и офф-лайн проектами. Зеленый значок указывает на то, что значения совпадают; синий / оранжевый значок указывает на то, что значения отличаются.

Кроме того, нажмите кнопку параметра с направленной вниз стрелкой, чтобы открыть небольшое окно, которое показывает стартовое значение проекта (офф-лайн) и стартовое значение ПЛК (он-лайн) для каждого параметра.



10.3.4. Управление перемещением с замкнутой обратной связью

10.3.4.1. Конфигурирование оси

Вы подключаете ось с замкнутой обратной связью к ПЛК и управляете ей посредством аналогового привода или PROFIdrive. Оси замкнутого контура также требуется энкодер.

STEP 7 обеспечивает инструменты конфигурирования, инструменты ввода в эксплуатацию и инструменты диагностики для технологического объекта "Axis".

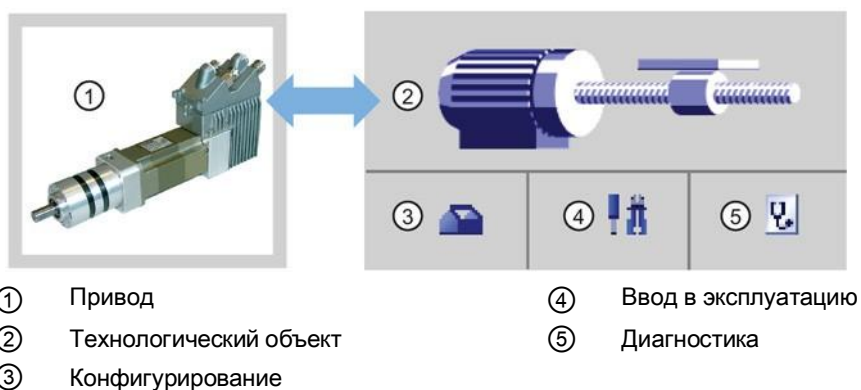


Таблица 10- 47 STEP 7 инструменты для управления перемещением с обратной связью

Инструмент	Описание
Configuration	Конфигурирует следующие свойства технологического объекта "Axis": - Выбор подключения используемого аналогового привода или PROFIdrive, и конфигурирование интерфейса привода и энкодера - Свойства механики и передаточное отношение привода и энкодера (или машины, или системы) - Свойства для ограничителей, динамики и возврата в начальное положение Сохраните конфигурацию в блоке данных технологического объекта.
Commissioning	Тестирует функцию Вашей оси без необходимости создания пользовательской программы. Когда инструмент будет запущен, будет отображена панель управления. Следующие команды доступны на панели управления: - Включить и выключить ось - Перемещать ось в старт-стопном режиме - Позиционировать ось в абсолютных и относительных величинах - Переводить ось в начальное положение - Квитиловать ошибки Для команд перемещения могут быть определены скорость и ускорение / замедление . Панель управления также показывает текущее состояние оси.
Diagnostics	Контролирует текущее состояние и информацию об ошибках для оси и привода.

Примечание

Вам, вероятно, придется адаптировать значения входных параметров инструкций управления перемещением к новым единицам размерности в пользовательской программе.

После того, как Вы создадите технологический объект для оси, Вы конфигурируете ось, определяя основные параметры либо аналогового привода, либо PROFIdrive-подключения и выполняете конфигурацию привода и кодера.

▼ Basic parameters	✓
General	✓
Drive	✓
Encoder	✓
▼ Extended parameters	✓
Mechanics	✓
Modulo	✓
Position limits	✓
▼ Dynamics	✓
General	✓
Emergency stop	✓
▼ Homing	✓
Active	✓
Passive	✓
▼ Position monitoring	✓
Position monitoring	✓
Following error	✓
Standstill signal	✓
Control loop	✓

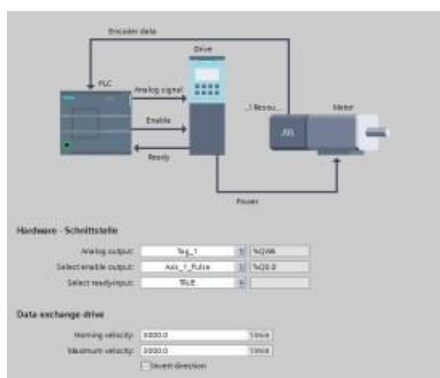
Древовидный селектор для аналогового привода или PROFIdrive соединения включает в себя конфигурационные меню Encoder, Modulo, Position monitoring и Control loop.

Конфигурирование подключения аналогового привода



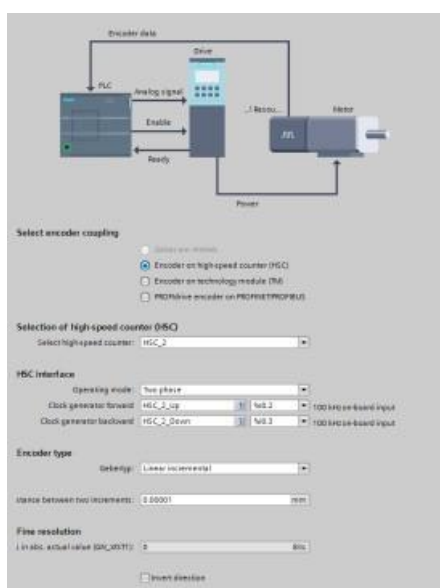
В диалоговом окне общей конфигурации Вы выбираете следующие параметры:

- Радиокнопку "Analog drive connection"
- Единицы измерения



В диалоговом окне конфигурации привода Вы выбираете следующие параметры:

- Аппаратные выходы аналогового привода
- Скорости обмена данными с приводом



В диалоговом окне конфигурации энкодера Вы выбираете следующие параметры:

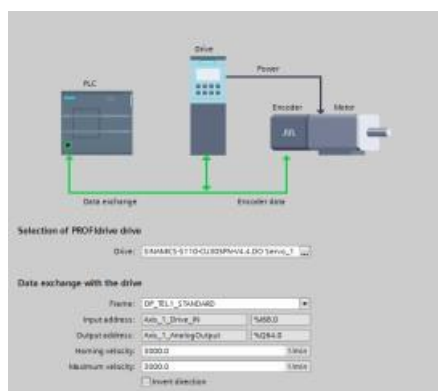
- Соединение с энкодером аналогового привода (например, высокоскоростной счетчик (HSC))
- Интерфейс в/с счетчика
- Тип энкодера
- Высокое разрешение

Конфигурирование PROFIdrive



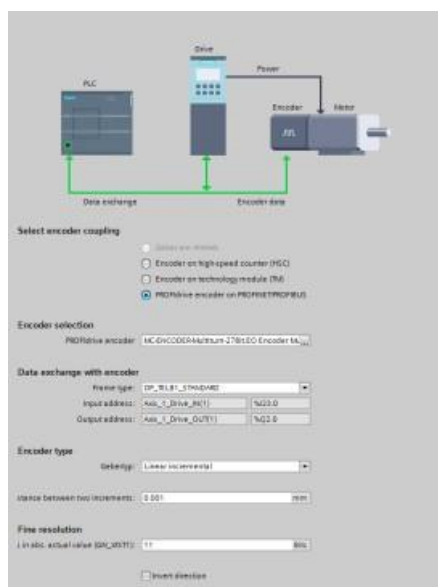
В диалоговом окне общей конфигурации Вы выбираете следующие параметры:

- Радиокнопку "PROFIdrive"
- Единицы измерения



В диалоговом окне конфигурации привода Вы выбираете следующие параметры:

- Привод PROFIdrive
- Обмен данными с приводом



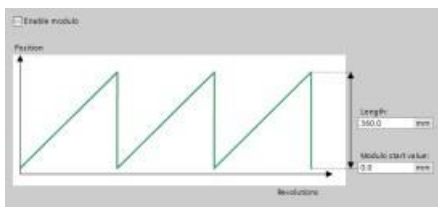
В диалоговом окне конфигурации энкодера Вы выбираете следующие параметры:

- Соединение с энкодером PROFIdrive (например, PROFIdrive энкодер или PROFINET)
- PROFIdrive энкодер
- Обмен данными с энкодером
- Тип энкодера
- Высокое разрешение

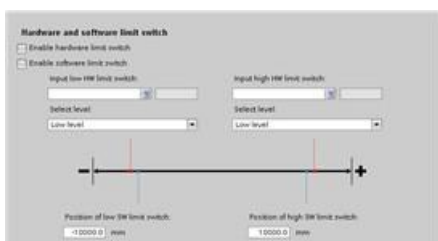
Расширенные параметры

Вы можете также сконфигурировать следующие свойства оси с обратной связью:

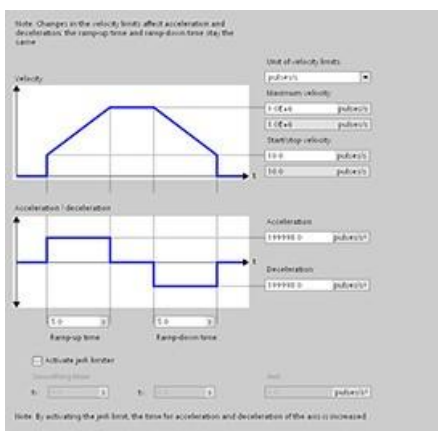
- Modulo
- Position limits
- Dynamics
- Homing
- Position monitoring
- Following error
- Standstill signal
- Control loop



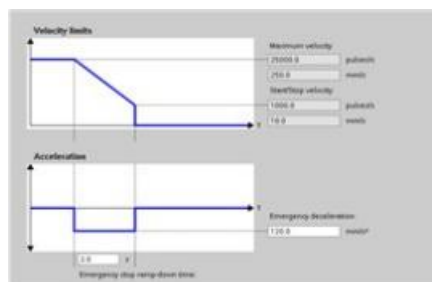
Modulo: Вы можете сконфигурировать "Modulo" ось, чтобы перемещать нагрузку в замкнутой области, которая имеет начальное значение / начальное положение и заданную длину. Если положение нагрузки достигает конца этой области, оно автоматически снова устанавливается на начальное положение. Вы активируете поля "Length" и "Modulo start value", когда устанавливаете флажок "Enable Modulo".



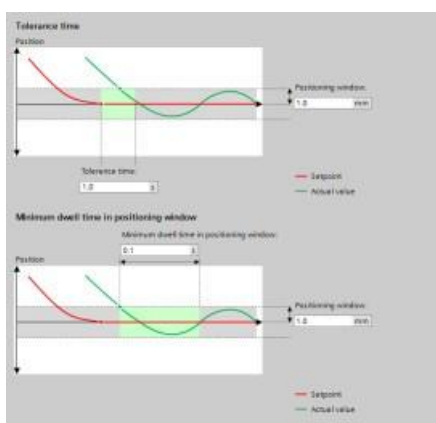
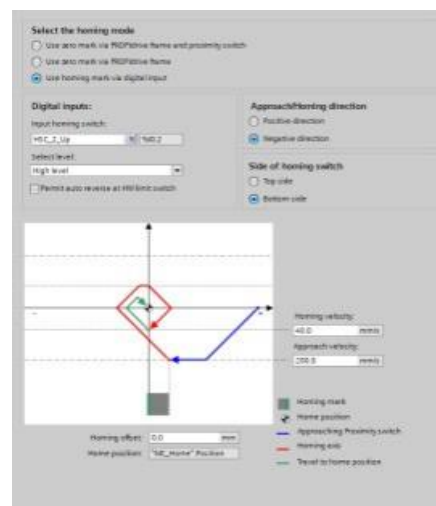
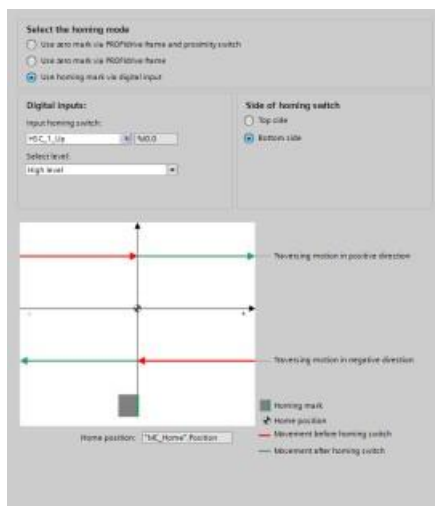
Position limits: Вы можете сконфигурировать свойства для сигналов привода, механики привода и контроля положения (аппаратные и программные ограничители).



Dynamics: Вы можете сконфигурировать динамику перемещения и поведение для команды аварийной остановки.



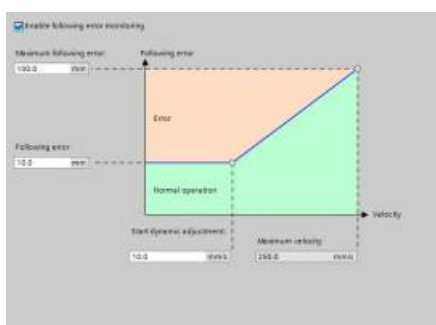
Homing: Вы можете сконфигурировать поведение перемещения в начальное положение (пассивное и активное).



"Positioning monitoring": Вы можете сконфигурировать время допуска, а также минимальное время простоя для окна позиционирования.

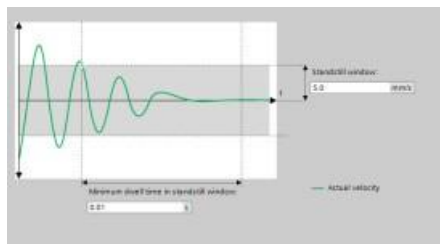
Система подключает следующие три параметра непосредственно с помощью TO-DB оси:

- Окно позиционирования
- Время допуска
- Минимальное время простоя в окне позиционирования



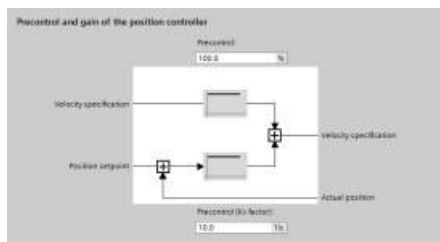
"Following error": Вы можете сконфигурировать различие разрешенного отклонения положения по диапазону скорости. Вы устанавливаете флажок "Enable following error monitoring", чтобы активировать ошибку рассогласования. Вы можете сконфигурировать следующие параметры:

- Максимальная ошибка рассогласования
- Ошибка рассогласования
- Настройка начальной динамики
- Максимальная скорость



"Standstill signal": Вы можете сконфигурировать следующие параметры:

- Минимальное время простоя в окне покоя
- Окно покоя.



"Control loop": Вы можете сконфигурировать прирост скорости, известный как "Предварительное управление (Kv-коэффициент)".

Используйте панель управления "Commissioning", чтобы протестировать функциональность независимо от Вашей пользовательской программы.



Щелкните по значку "Startup", чтобы ввести ось в эксплуатацию.

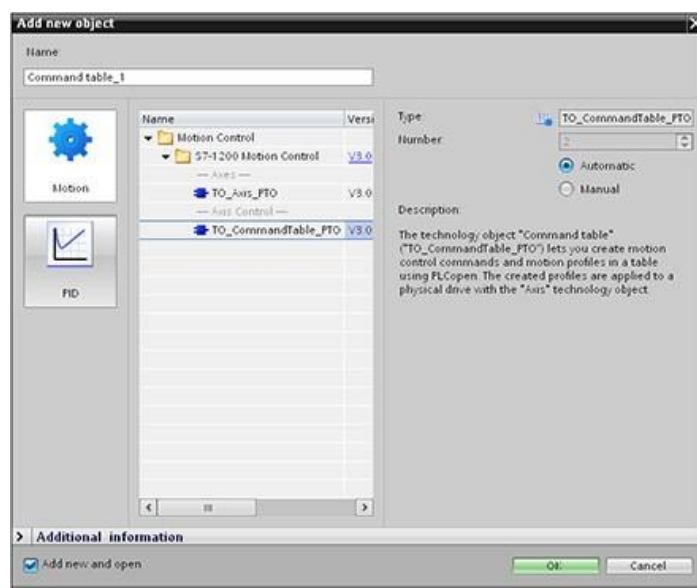
Панель управления показывает текущий статус оси. Кроме того, что Вы можете включить и выключить ось, Вы можете также протестировать позиционирование оси (в абсолютных и относительных значениях), а также можете определить скорость, ускорение и замедление. Вы можете также протестировать задач перехода в начальное положение и старт-стопный режим. Панель управления также позволяет Вам квитировать ошибки.

10.3.5. Конфигурирование TO_CommandTable_PTO

Вы можете сконфигурировать инструкцию MC_CommandTable, используя технологические объекты. Следующий пример демонстрирует, как это делается.

Добавление технологического объекта

1. В дереве проекта разверните узел "Technology Objects" и выберите "Add new object".
2. Выберите значок "CommandTable" (переименуйте при необходимости), и нажмите "OK", чтобы открыть редактор конфигурирования для объекта CommandTable.



Планирование шагов для Вашего приложения

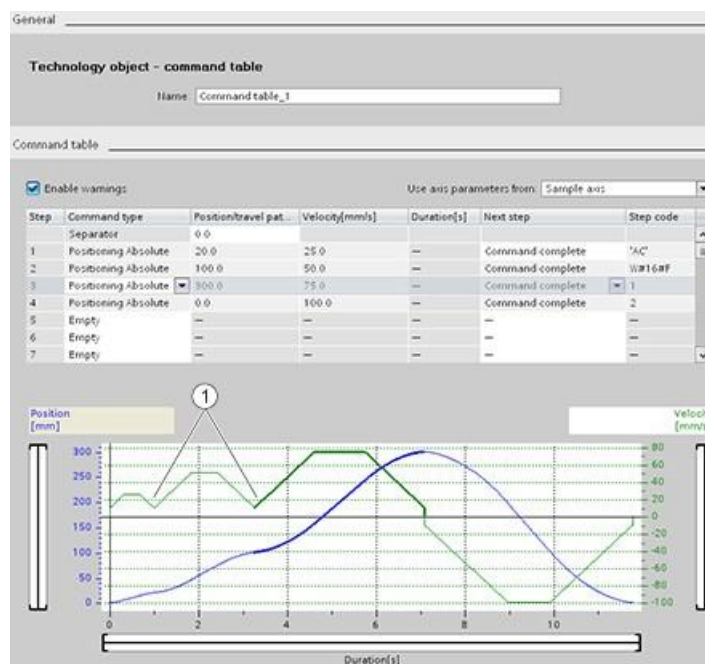
Вы можете создать желаемую последовательность перемещения в конфигурационном окне "Command Table" и проверить результат по графическому представлению на диаграмме тренда.

Вы можете выбрать типы команд, которые должны использоваться для обработки таблицы команд. Могут быть введены до 32 шагов. Команды обрабатываются последовательно, легко формируя сложный профиль перемещения.

Таблица 10- 48 Типы команд MC_CommandTable

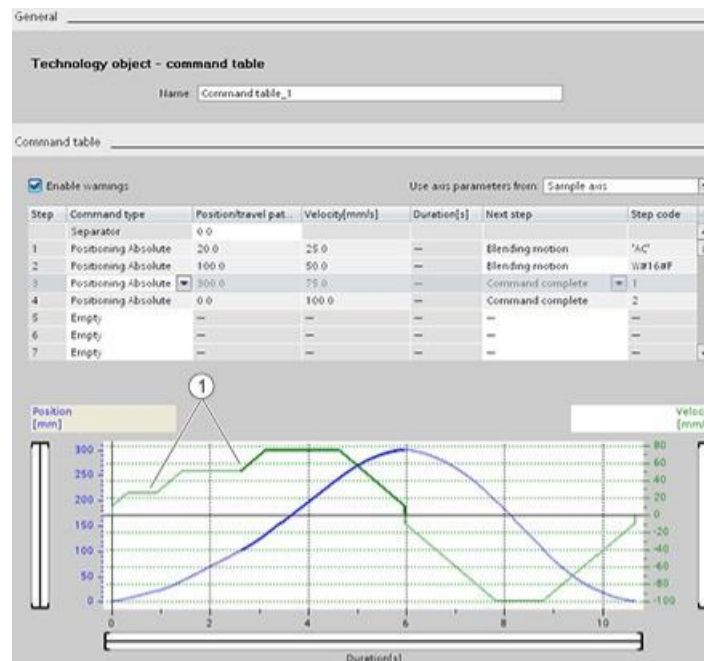
Тип команды	Описание
Empty	Пустая запись служит как заполнитель для любых команд, которые будут добавлены. Пустая запись игнорируется, при обработке таблицы команд
Halt	Приостановить ось. Примечание: команда встречается только после команды "Velocity setpoint".
Positioning Relative	Позиционирует ось на основании расстояния. Команда перемещает ось согласно заданным расстоянию и скорости.
Positioning Absolute	Позиционирует ось на основании положения. Команда перемещает ось в заданное положение, используя определенную скорость.
Velocity setpoint	Перемещает ось с заданной скоростью.
Wait	Ожидает, пока установленный период не истек. "Wait" не останавливает активное вращательное движение.
Separator	Добавляет строку "Разделителя" над выбранной строкой. Строка разделителя позволяет определять более одного профиля в единственной таблице команд.

На рисунке ниже "Command complete" используется в качестве перехода к следующему шагу. Этот тип перехода позволяет Вашему устройству замедляться до скорости запуска/остановки и затем снова ускоряться в начале следующего шага.



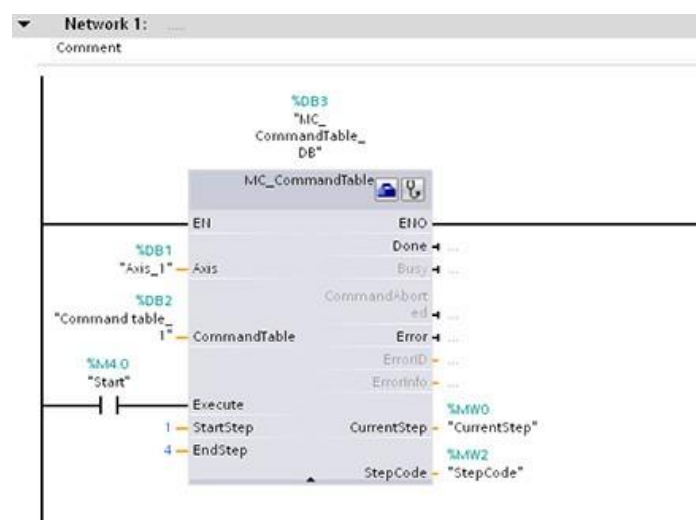
① Ось замедляется до скорости старта/остановки между шагами.

На рисунке ниже "Blending motion" используется в качестве перехода к следующему шагу. Этот тип перехода позволяет Вашему устройству поддерживать свою скорость через запуск следующего шага, приводя к плавному переходу устройства от одного шага к другому. Использование смешивания может сократить общее время, требуемое для полного выполнения профиля. Без смешивания выполнение этого примера занимает семь секунд. Со смешиванием время выполнения уменьшается на одну секунду до общей продолжительности в шесть секунд.



① Ось продолжает перемещаться и ускоряется или замедляется до скорости следующего шага, экономя время и снижая механический износ.

Работой CommandTable управляет инструкция MC_CommandTable, как показано ниже:



10.3.6. Работа управления перемещением для S7-1200

10.3.6.1. Выходы ЦПУ, используемые для управления перемещением

ЦПУ обеспечивает четыре генератора с импульсным выходом. Каждый генератор с импульсным выходом обладает одним импульсным выходом и одним выходом направления для управления приводом шагового двигателя или серводвигателя с импульсным интерфейсом. Импульсный выход предоставляет приводу импульсы, требуемые для движения двигателя. Выход направления управляет направлением перемещения привода.

PTO выход генерирует прямоугольные импульсы переменной частоты. Генерация импульсов управляется конфигурацией, и информация о выполнении, предоставляется через аппаратную конфигурацию и/или SFC/SFB.

На основании выбора пользователя, пока ЦПУ находится в режиме RUN, либо значения, сохраненные в регистре образа, либо выходы генератора импульсов, управляют цифровыми выходами. В режиме STOP PTO генератор не управляет выходами.

Встроенные выходы ЦПУ и выходы сигнальной платы могут использоваться в качестве выходов направления и импульсов. Вы выбираете между встроенными выходами ЦПУ и выходами сигнальной платы во время конфигурирования устройства в разделе "Pulse generators (PTO/PWM)" на вкладке "Properties". Только PTO (Выход пачки импульсов) применяется для управления перемещением.

Таблица ниже показывает назначения ввода-вывода по умолчанию; однако, четыре импульсных генератора могут быть сконфигурированы на любой цифровой выход.

Примечание

Выходы пачки импульсов не могут использоваться другими инструкциями в пользовательской программе

Когда Вы конфигурируете выходы ЦПУ или сигнальной платы как импульсные генераторы (для использования с инструкциями ШИМ или управления перемещением), соответствующие выходные адреса больше не управляют выходами. Если Ваша пользовательская программа пишет значение на выход, используемый в качестве импульсного генератора, ЦПУ не записывает это значение в физический выход.

Примечание

Выходы направления PTO могут быть освобождены для использования в другом месте Вашей программы

Каждый PTO требует назначения двух выходов: одного в качестве выхода импульсов и одного как выхода направления. Вы можете использовать только импульсный выход, и не использовать выход направления. Вы можете в этом случае освободить выход направления для других целей в Вашей пользовательской программе. Выход не может использоваться одновременно и для выхода направления PTO и в пользовательской программе.

Таблица 10- 49 Назначение адреса по умолчанию выходов импульсов и направления

Использование выходов для управления перемещением		
	Импульс	Направление
PTO1		
Встроенный ввод-вывод	Q0.0	Q0.1
Ввод-вывод на SB	Q4.0	Q4.1
PTO2		
Встроенный ввод-вывод	Q0.2	Q0.3
Ввод-вывод на SB	Q4.2 ¹	Q4.3 ¹
PTO3		
Встроенный ввод-вывод	Q0.4 ²	Q0.5 ²
Ввод-вывод на SB	Q4.0	Q4.1
PTO4		
Встроенный ввод-вывод	Q0.6 ³	Q0.7 ³
Ввод-вывод на SB	Q4.2	Q4.3

¹ Выходы Q4.2 и Q4.3 доступны только на SB1222 DQ4.

² У CPU 1211C нет выходов Q0.4, Q0.5, Q0.6 или Q0.7. Поэтому, эти выходы не могут использоваться в CPU 1211C.

³ У CPU 1212C нет выходов Q0.6 или Q0.7. Поэтому, эти выходы не могут использоваться в CPU 1212C.

⁴ Эта таблица применяется к PTO функциям CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C и CPU 1217C.

Интерфейс привода

Для управления перемещением Вы можете дополнительно сконфигурировать интерфейс привода на "Drive enabled" и "Drive ready". При использовании интерфейса привода может быть свободно выбран цифровой выход для включения привода и цифровой вход для "готовности привода".

Примечание

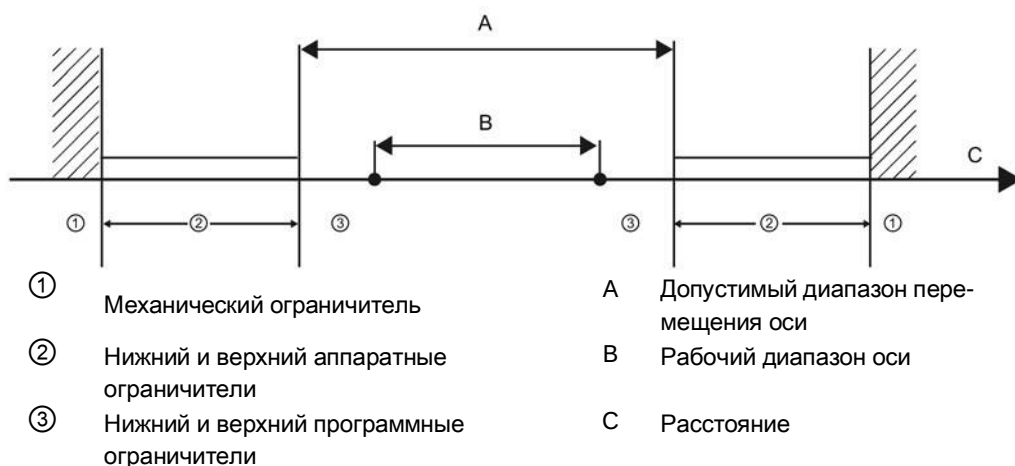
Встроенное ПО возьмет на себя управление через соответствующие выходы импульсов и направления, если PTO (Выход пачки импульсов) был выбран и назначен оси.

Вместе с этим перехватом функции управления, связи между образом процесса и выходом аппаратного ввода-вывода также разрываются. В то время как у пользователя есть возможность записи в образ процесса выходов импульса и направления через пользовательскую программу или таблицу наблюдения, это никогда не передается в аппаратный ввод-вывод. Соответственно, также невозможно контролировать аппаратный ввод-вывод через пользовательскую программу или таблицу наблюдения. Считываемая информация просто отражает значение образа процесса и не соответствует фактическому состоянию аппаратного ввода-вывода в любом отношении.

Для всех других выходов ЦПУ, которые не используются постоянно встроенным ПО ЦПУ, состоянием аппаратного выхода можно управлять или контролировать, как обычно, через образ процесса.

10.3.6.2. Аппаратные и программные ограничители для управления перемещением

Используйте аппаратные и программные ограничители положения, чтобы ограничить "допустимый диапазон перемещения" и "рабочий диапазон" Вашей оси.



Аппаратные и программные ограничители должны быть активированы до использования в конфигурации или в пользовательской программе. Программные ограничители активны только после перемещения оси в начальное положение.

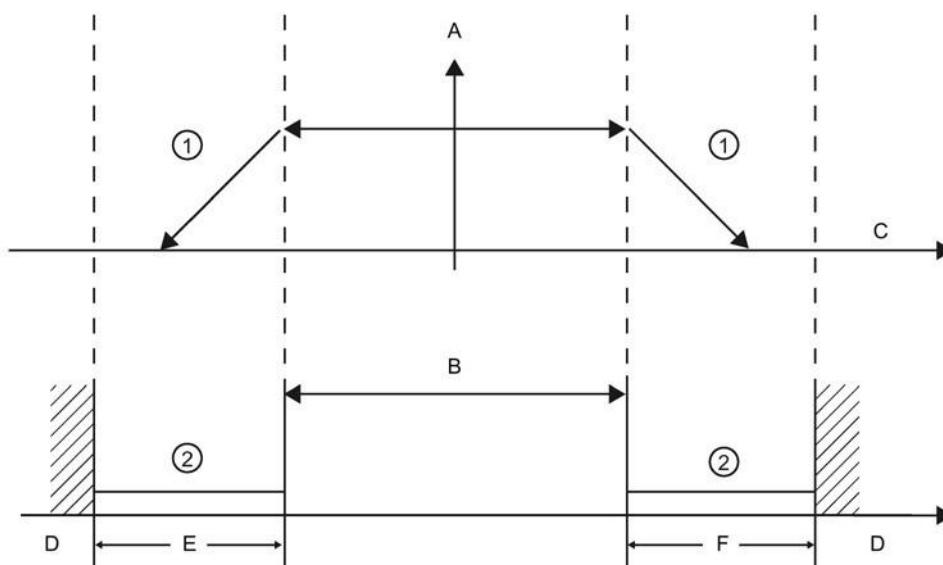
Аппаратные ограничители

Аппаратные ограничители определяют максимальный диапазон перемещения оси. Аппаратные ограничители представляют собой физические коммутационные элементы, которые должны быть подключены к входам ЦПУ с возможностью обработки прерывания. Используйте только аппаратные ограничители положения, которые остаются постоянно переключенными после срабатывания. Это переключенное состояние может быть отменено только после возврата в допустимый диапазон перемещения.

Таблица 10- 50 Входы, подходящие для аппаратных ограничителей

Описание	RPS	LIM-	LIM+
Встроенный ввод-вывод		I0.0 - I1.5	
Ввод-вывод на SB		I4.0 - I4.3	

Когда срабатывает аппаратный ограничитель, ось тормозится до полной остановки со сконфигурированным аварийным замедлением. Заданное аварийное замедление должно быть достаточным, чтобы надежно остановить ось перед механическим ограничителем. Следующая диаграмма показывает поведение оси после того, как она приблизится к аппаратным ограничителям.



- ① Ось тормозится до полной остановки со сконфигурированным аварийным замедлением.
- ② Диапазон, в котором аппаратные ограничители сигнализируют о "срабатывании".
- A [Скорость]
- B Допустимый диапазон перемещения
- C Расстояние
- D Механический упор
- E Нижний аппаратный ограничитель
- F Верхний аппаратный ограничитель



ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с изменением настройки времени фильтра для цифровых входных каналов

Если предыдущая настройка времени фильтра для цифрового входного канала изменена, то, возможно, новое входное значение "0" уровня должно было бы быть представлено в течение накопленного времени до максимум 20.0 мс, прежде чем фильтр станет полностью чувствительным к новым входным значениям. В течение этого времени короткие "0" импульсные события длительностью меньше 20.0 мс не могут обнаруживаться или подсчитываться.

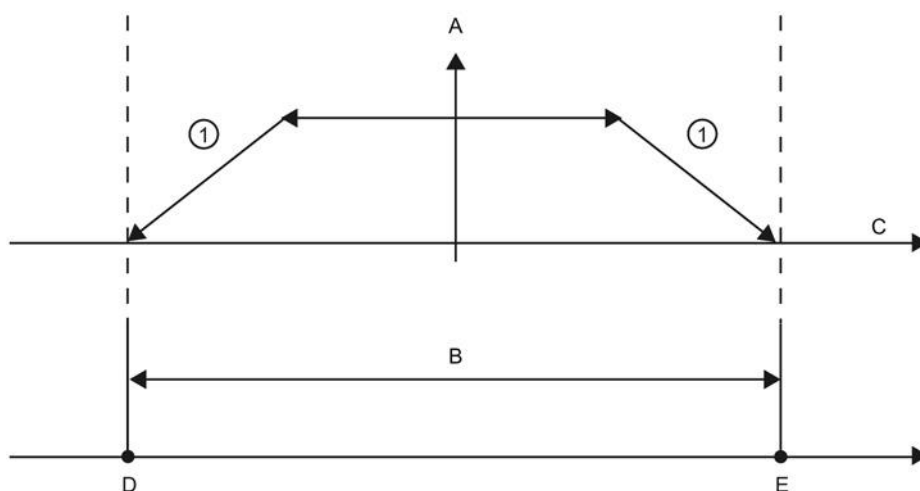
Это изменение времен фильтра может привести к неожиданной работе машины или процесса, что может вызвать смерть или серьезную травму персонала, и/или повредить оборудование.

Чтобы гарантировать, что новое время фильтра сразу вступает в силу, выполните цикл выключения и включения питания ЦПУ.

Программные ограничители

Программные ограничители контролируют "рабочий диапазон" оси. Они должны сработать до аппаратных ограничителей относительно диапазона перемещения. Поскольку позиции программных ограничителей могут быть установлены гибко, рабочий диапазон оси может быть ограничен индивидуально в зависимости от текущего профиля перемещения. В отличие от аппаратных ограничителей, программные ограничители реализованы исключительно посредством программного обеспечения и не требуют своих собственных элементов коммутации.

Если программные ограничители активированы, активное перемещение прекращается в позиции программного ограничителя. Ось тормозится со сконфигурированным замедлением. Следующая диаграмма показывает поведение оси до достижения программных ограничителей.



① Ось тормозится до полной остановки со сконфигурированным замедлением.

A [Скорость]

B Рабочий диапазон

C Расстояние

D Нижний программный ограничитель

E Верхний программный ограничитель

Используйте дополнительные аппаратные ограничители, если механический ограничитель хода расположен после программных ограничителей, и существует риск механического повреждения.

Дополнительная информация

Ваша пользовательская программа может преобладать над действием аппаратных или программных ограничителей, активируя или деактивируя как аппаратную, так и программную функцию ограничения. Выбор выполняется из DB для Axis.

- Чтобы включить или отключить аппаратную функцию ограничения, выполните доступ к тегу "Active" (Bool) согласно следующему пути "<Имя оси>/Config/**PositonLimits_HW**". Состояние этого тега "Active" включает или отключает использование аппаратных ограничителей.
- Чтобы включить или отключить программную функцию ограничения, выполните доступ к тегу "Active" (Bool) согласно следующему пути "<Имя оси>/Config/**Position Limits_SW**". Состояние этого тега "Active" включает или отключает использование программных ограничителей.

Вы можете также изменить позиции программных ограничителей в своей пользовательской программе (к примеру, чтобы сделать гибкой настройку машины или сократить время переналадки машины). Ваша пользовательская программа может записывать новые значения в теги "MinPosition" и "MaxPosition" (инженерные единицы в формате Real) в DB "< Имя оси >/Config/**PositionLimits_SW**".

10.3.6.3. Перемещение в начальное положение

Перемещение в начальное положение имеет отношение к согласованию координат оси с реальным, физическим положением привода. (Если привод в настоящий момент будет находиться в положении x, то ось будет скорректирована до положения x.) Для оси с контролем положения, вводимые значения и индикация для положения точно ссылаются на эти координаты оси.

Примечание

Согласование между координатами оси и действительной ситуацией чрезвычайно важно. Этот шаг необходим, чтобы гарантировать, что абсолютная заданная позиция оси также точно достигается с помощью привода.

Инструкция MC_Home инициирует перемещение оси в начальное положение.

Существует 4 различных функции перемещения в начальное положение. Первые две функции позволяют пользователю устанавливать текущее положение оси, а вторые две позиционируют ось относительно датчика начального положения.

- Режим 0 - Прямая абсолютная ссылка: Во время выполнения этот режим точно сообщает оси, где она находится. Он устанавливает внутреннюю переменную положения в значение входа Position инструкции Homing. Он используется для калибровки и настройки машины.

Положение оси устанавливается независимо от датчика начального положения. Активные перемещения не прерываются. Значение входного параметра Position инструкции MC_Home сразу устанавливается как начальное положение оси. Чтобы привязать начальную точку к точной механической позиции, ось должна быть в состоянии покоя в этой позиции во время операции ввода начального положения.

- Режим 1 - Прямая абсолютная ссылка: При выполнении этот режим добавляет значение входа Position для инструкции Homing к внутренней переменной положения. Он обычно используется, чтобы учесть смещение машины.

Положение оси устанавливается независимо от датчика начального положения. Активные перемещения не прерываются. Следующее присвоение применяется к положению оси после ввода начального положения: Новое положение оси = текущее положение оси + значение параметра Position инструкции MC_Home.

- Режим 2 - Пассивный ввод начального положения: Когда ось перемещается и проходит датчик начального положения, текущее положение устанавливается в качестве исходного. Эта функция поможет учесть нормальный износ машины и боковой износ шестерен в редукторе и устраняет потребность в ручной компенсации износа. Вход Position инструкции Homing, как прежде, добавляется к положению, связанному с датчиком начального положения, позволяя простое смещение начального положения.

Во время пассивного ввода начального положения инструкция MC_Home не инициирует переход в начальное положение. Перемещение, требуемое для этого шага, должно быть реализовано пользователем через другие команды управления перемещением. Когда датчик начального положения обнаружен, ось перенимает начальное положение согласно конфигурации. Активные перемещения не прерываются при запуске пассивного ввода начального положения.

- Режим 3 - Активный ввод начального положения: Этот режим является самым точным методом установки начального положения оси. Начальное направление и скорость перемещения сконфигурированы в Extended Parameters-Homing конфигурации технологического объекта. Это зависит от конфигурации машины. Существует также возможность определить, положительный или отрицательный фронт сигнала датчика соответствует начальному положению. Практически у всех датчиков есть активный диапазон и если бы положение устойчивого срабатывания использовалось в качестве сигнала начального положения, то тогда существовала бы вероятность получения ошибки, так как активный диапазон сигнала срабатывания соответствует диапазону расстояния. При помощи положительного или отрицательного фронта этого сигнала достигается намного более точное начальное положение. Как во всех других режимах значение входа Position инструкции Homing добавляется к определенному аппаратно начальному положению.

В активном режиме ввода начального положения инструкция MC_Home выполняет требуемый подход к начальному положению. Когда датчик начального положения обнаружен, начальное положение оси вводится согласно конфигурации. Активные перемещения прерываются.

Режимы 0 и 1 не требуют каких-либо перемещений оси. Они обычно используются при настройке и калибровке. Режимы 2 и 3 требуют, чтобы ось перемещалась и проходила датчик, который сконфигурирован в технологическом объекте "Axis" как датчик начального положения. Начальное положение может быть определено в рабочей области оси или за пределами нормальной рабочей области, но в диапазоне перемещения.

Конфигурирование параметров начального положения

Сконфигурируйте параметры для активного и пассивного ввода начального положения в конфигурационном окне "Homing". Метод ввода начального положения выбирается, используя входной параметр "Mode" команды управления перемещением. Здесь, Mode = 2 означает пассивное, а Mode = 3 означает активный ввод начального положения.

Примечание

Используйте одну из следующих мер, чтобы гарантировать, что машина не перемещается до механического упора в случае реверсирования направления:

- Поддерживайте скорость подхода на низком уровне
 - Увеличьте сконфигурированное ускорение/замедление
 - Увеличьте расстояние между аппаратным ограничителем и механическим упором
-

Таблица 10- 51 Конфигурационные параметры для ввода начального положения оси

Параметр	Описание
Input reference point switch (Активный и пассивный ввод начального положения)	Выберите цифровой вход для датчика начального положения из выпадающего списка. Вход должен обладать способностью к прерыванию. Встроенные входы ЦПУ и входы вставленной сигнальной платы могут быть выбраны как входы для датчика начального положения. Время фильтра по умолчанию для цифровых входов составляет 6.4 мс. Когда цифровые входы используются в качестве датчика начального положения, это может привести к нежелательным замедлениям и таким образом к неточностям. В зависимости от пониженной скорости и протяженности датчика, начальное положение может быть не обнаружено. Время фильтра может быть установлено под элементом "Input filter" в конфигурации устройства цифровых входов. Указанное время фильтра должно быть меньше, чем длительность входного сигнала от датчика начального положения.
Auto reverse after reaching the hardware limit switches (Только активный ввод начального положения)	Активируйте флажок, чтобы использовать аппаратный ограничитель в качестве активатора реверсирования для подхода к начальному положению. Аппаратные ограничители должны быть сконфигурированы и активированы для реверсирования направления. Если аппаратный ограничитель достигнут во время активного перехода в начальное положение, ось тормозится со сконфигурированным замедлением (не с аварийным) и инвертирует направление. Датчик начального положения обнаруживается затем в обратном направлении. Если реверсирование направления не активно, и ось достигает аппаратного ограничителя во время активного перехода в начальное положение, подход к начальному положению прерывается с ошибкой, и ось тормозится с аварийным замедлением.
Approach direction (Активный и пассивный ввод начального положения)	Выбором направления Вы определяете "направление подхода", используемое во время активного ввода начального положения, чтобы найти датчик начального положения, а также направление начального положения. Направление начального положения определяет направление перемещения, которое использует ось, чтобы приблизиться к сконфигурированной стороне датчика начального положения, чтобы выполнить операцию установки в начальное положение.
Reference point switch (Активный и пассивный ввод начального положения)	- Активная установка в начальное положение: Выберите, нужно ли выполнять установку оси в начальное положение с левой или правой стороны датчика начального положения. В зависимости от стартовой позиции оси и конфигурации параметров начального положения, последовательность подхода к начальному положению может отличаться от диаграммы в окне конфигурации. - Пассивная установка в начальное положение: С пассивной установкой в начальное положение перемещения в целях установки в начальное положение должны быть реализованы пользователем через команды перемещения. Сторона датчика начального положения, на которой выполняется установка начального положения, зависит от следующих факторов: - конфигурация "направления подхода" - конфигурация "датчика начального положения" - текущее направления перемещения во время пассивной установки начального положения

Параметр	Описание
Approach velocity (Только активный ввод начального положения)	Определите скорость, с которой должен выполняться поиск датчик во время подхода к начальному положению. Предельные значения (независимо от выбранных пользовательских единиц): Скорость запуска/остановки $\leq$ скорость подхода $\leq$ максимальная скорость
Reduced velocity (Только активный ввод начального положения)	Определите скорость, с которой ось приближается к датчику для установки в начальное положение. Предельные значения (независимо от выбранных пользовательских единиц): Скорость запуска/остановки $\leq$ скорость подхода $\leq$ максимальная скорость
Home position offset (Только активный ввод начального положения)	Если желаемое начальное положение отличается от позиции датчика начального положения, в этом поле может быть определено смещение начального положения. Если значение не равно 0, ось выполняет следующие действия после установки в начальное положение по датчику: 1. Смещается на пониженной скорости согласно значению смещения начального положения. 2. Когда позиция смещения исходного положения достигнута, положение оси установлено на значение абсолютной ссылки. Позиция абсолютной ссылки определена в параметре "Position" команды управления перемещением "MC_Home". Предельные значения (независимо от выбранных пользовательских единиц): $-1.0e12 \leq$ смещение начального положения $\leq 1.0e12$

Таблица 10- 52 Факторы, влияющие на установку в начальное положение

Факторы воздействия:			Результат:
Конфигурация направления подхода	Конфигурация датчика начального положения	Текущее направление перемещения	Начальное положение относительно датчика
Положительное	"Левая (отрицательная) сторона"	Положительное направление	Слева
		Отрицательное направление	Справа
Положительное	"Правая (положительная) сторона"	Положительное направление	Справа
		Отрицательное направление	Слева
Отрицательное	"Левая (отрицательная) сторона"	Положительное направление	Справа
		Отрицательное направление	Слева
Отрицательное	"Правая (положительная) сторона"	Положительное направление	Слева
		Отрицательное направление	Справа

Последовательность активной установки в начальное положение

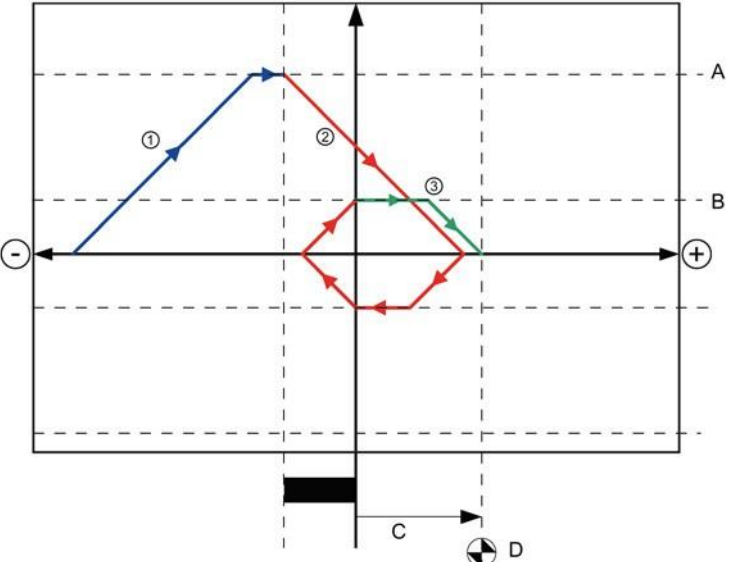
Вы запускаете активную установку в начальное положение с помощью инструкции управления перемещением "MC_Home" (входной параметр

Mode = 3). Входной параметр "Position" определяет в этом случае координаты точки абсолютной ссылки. Альтернативно, Вы можете запустить активную установку в начальное положение в тестовых целях на панели управления.

Следующая диаграмма показывает пример характерной кривой для активной установки в начальное положение со следующими параметрами конфигурации:

- "Направление подхода" = "Положительное направление"
- "Датчик начального положения" = "Правая (положительная) сторона"
- Значение "смещения начального положения" > 0

Таблица 10- 53 Кинематические характеристики MC homing

Операция		Примечания	
		A	Скорость приближения
		B	Пониженная скорость
		C	Координата начального положения
		D	Смещение начального положения
①	Фаза поиска (синий сегмент кривой): Когда активная установка начального положения запускается, ось ускоряется до сконфигурированной "скорости подхода" и ищет на этой скорости датчик начального положения.		
②	Подход к начальному положению (красный раздел кривой): Когда датчик начального положения обнаружен, ось в этом примере тормозится и реверсируется, чтобы быть установленной в начальное положение на сконфигурированной стороне датчика начального положения при сконфигурированной "пониженной скорости".		
③	Перемещение в начальное положение (зеленый сегмент кривой): После перехода в начальное положение по датчику ось перемещается в "позицию начального положения" на "пониженной скорости". При достижении "позицию начального положения", ось останавливается в значении, которое было определено во входном параметре "Position" инструкции "MC_Home".		

Примечание

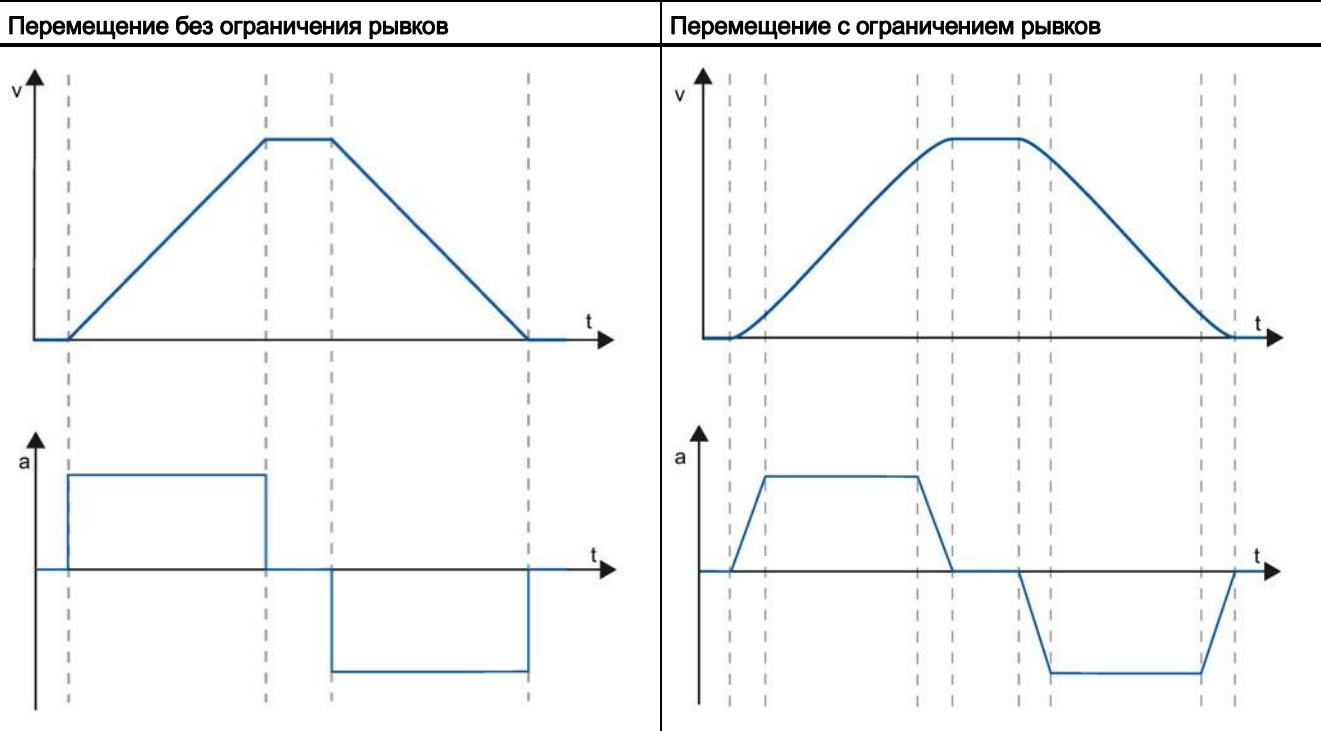
Если поиск начального положения не функционирует, как Вы ожидали, проверьте входы, назначенные аппаратным ограничителям или начальному положению. Для этих входов, возможно, отключены прерывания по фронту в конфигурации устройства.

Исследуйте данные конфигурации для технологического объекта оси беспокойства, чтобы определить какие входы (если таковые имеются) назначены для "HW Low Limit Switch Input", "HW High Limit Switch Input" и "Input reference point switch". Затем откройте конфигурацию устройства для ЦПУ и проверьте каждый из назначенных входов. Проверьте выбраны ли обе опции "Enable rising edge detection" и "Enable falling edge detection". Если эти свойства не выбраны, удалите указанные входы в конфигурации оси и выберите их снова.

10.3.6.4. Ограничение рывков

С ограничением рывков Вы можете уменьшить стрессовые воздействия в своей механике во время линейного замедления и ускорения. Значение для ускорения и замедления не изменяется резко, когда ограничитель рывков активен; оно адаптируется в фазе перехода. Рисунок ниже показывает кривые скорости и ускорения без ограничения рывков и с ним.

Таблица 10- 54 Ограничение рывков



Ограничение рывков дает "сглаженный" профиль скорости перемещения оси. Это гарантирует, например, мягкий запуск и торможение ленточного конвейера.

10.3.7. Инструкции управления перемещением

10.3.7.1. Обзор MC-инструкций

Инструкции управления перемещением используют связанный технологический блок данных и выделенные РТО (выходы пачки импульсов) ЦПУ, чтобы управлять перемещением на оси.

- MC_Power (стр. 572) включает и выключает ось управления перемещением.
- MC_Reset (стр. 575) сбрасывает все ошибки управления перемещением. Все ошибки управления перемещением, которые могут быть квитированы, квитируются.
- MC_Home (стр. 576) устанавливает отношение между программой управления осью и механической системой позиционирования оси.
- MC_Halt (стр. 578) отменяет все процессы перемещения и вызывает остановку оси. Позиция остановки не определяется.
- MC_MoveAbsolute (стр. 580) запускает перемещение в абсолютную позицию. Задание заканчивается, когда заданная позиция достигнута.
- MC_MoveRelative (стр. 582) запускает перемещение относительно стартовой позиции.
- MC_MoveVelocity (стр. 584) заставляет ось перемещаться с указанной скоростью.
- MC_MoveJog (стр. 587) обрабатывает старт-стопный режим в целях тестирования и запуска.
- MC_CommandTable (стр. 589) выполняет команды оси как последовательность перемещений.
- MC_ChangeDynamic (стр. 592) изменяет настройки динамики для оси.
- MC_WriteParam (стр. 594) выполняет запись избранного числа параметров из пользовательской программы, чтобы изменить функциональность оси.
- MC_ReadParam (стр. 596) читает избранное число параметров, которые указывают текущую позицию, скорость, и т.д. для оси, определенной в параметре Axis.

Уровни встроенного ПО ЦПУ

Если у Вас есть ЦПУ S7-1200 со встроенным ПО V4.1, выберите версию V5.0 для каждой из инструкций перемещения.

Если у Вас есть ЦПУ S7-1200 с версией встроенного ПО V4.0 или более ранней, выберите применимую V4.0, V3.0, V2.0 или V1.0 версию каждой из инструкций перемещения.

Примечание

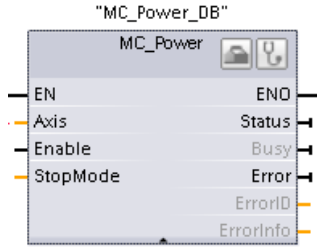
ЦПУ обсчитывает задачи перемещения в "секциях" или сегментах по 10 мс. Пока одна секция выполняется, следующая секция ожидает выполнения в очереди. Если Вы прерываете задачу перемещения на оси (выполняя другую новую задачу перемещения для этой оси), новая задача перемещения может не выполняться максимум в течение 20 мс (остаток от текущей секции плюс ожидающая в очереди секция).

10.3.7.2. Инструкция MC_Power (Освободить/блокировать ось)

Примечание

Если ось выключается из-за ошибки, то она будет снова включена автоматически после того, как ошибка была устранена и квитирована. Это требует, чтобы входной параметр Enable сохранял значение TRUE в течение этого времени.

Таблица 10- 55 Инструкция MC_Power

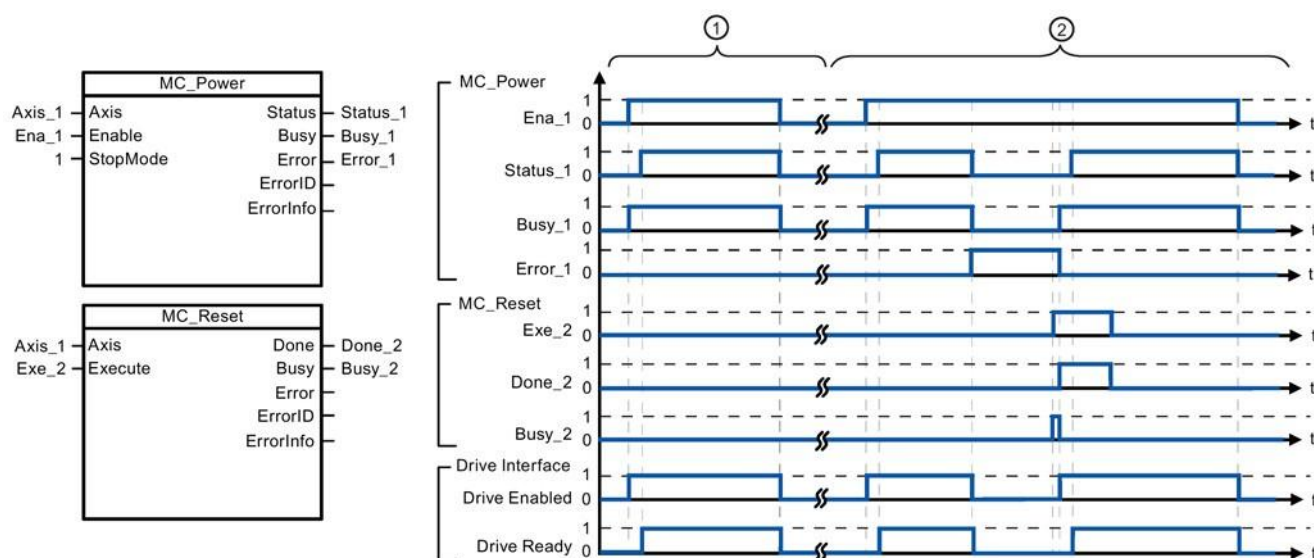
LAD / FBD	SCL	Описание
	<pre>"MC_Power_DB" (Axis:=_multi_fb_in_, Enable:=_bool_in_, StopMode:=_int_in_, Status=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Инструкция управления перемещением MC_Power включает или выключает ось. Прежде чем Вы сможете включить или выключить ось, обеспечьте следующие условия: • Технологический объект был сконфигурирован правильно. • Нет активной препятствующей включению ошибки. Выполнение MC_Power не может быть прервано задачей управления перемещением. Отключение оси (входной параметр Enable = FALSE) прерывает все задачи управления перемещением для связанного технологического объекта.

- ¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- ² В примере на SCL, "MC_Power_DB " является именем экземплярного DB.

Таблица 10- 56 Параметры для инструкция MC_Power

Параметр и тип	Тип данных	Описание
Axis	IN	TO_Axis_1 Технологический объект Axis
Enable	IN	Bool • FALSE (по умолчанию): Все активные задачи прерываются согласно запараметрированному "StopMode", и ось останавливается. • TRUE: Управление перемещением пытается включить ось.
StopMode	IN	Int • 0: Аварийная остановка: Если запрос на выключение оси активен, ось тормозится со сконфигурированным аварийным замедлением. Ось выключается после достижения остановки. • 1: Мгновенная остановка: Если запрос на выключение оси активен, эта ось выключается без замедления. Импульсный выход останавливается незамедлительно. • 2: Чрезвычайная остановка с ограничением рывков: Если запрос на выключение оси активен, ось тормозится со сконфигурированным аварийным замедлением. Если управление ограничением рывков активировано, принимается во внимание сконфигурированная компенсация. Ось выключается после достижения остановки.

Параметр и тип		Тип данных	Описание
Status	OUT	Bool	Состояние включения оси: - FALSE: Ось выключена: - Ось не выполняет задачи управления перемещением и не принимает новых задач (исключение: задача MC_Reset). - Ось не находится в начальном положении. - После выключения состояние не изменяется на FALSE, пока ось не переходит в состояние покоя. - TRUE: Ось включена: - Ось готова выполнить задачи управления перемещением. - После включения оси состояние не изменяется на TRUE, пока сигнал "привод готов" не станет активным. Если связь с приводом "привод готов" не была сформулирована в конфигурации оси, состояние изменяется в TRUE незамедлительно.
Busy	OUT	Bool	FALSE: MC_Power не активна. TRUE: MC_Power активна.
Error	OUT	Bool	FALSE: Ошибок нет. TRUE: Ошибка произошла в команде управления перемещением "MC_Power" или в связанном технологическом объекте. Причина ошибки может быть проанализирована в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUT	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID"



- ① Ось включена и затем вновь выключена. После того, как привод сигнализировал ЦПУ "привод готов", успешное включение может быть считано через "Status_1".
- ② После включения оси произошла ошибка, которая привела к выключению оси. Ошибка устранена и квитируется с помощью "MC_Reset". Ось затем вновь включается.

Чтобы включить ось со сконфигурированным интерфейсом привода, выполните следующие шаги:

1. Проверьте требования, обозначенные выше.
2. Инициализируйте входной параметр "StopMode" требуемым значением. Установите входной параметр "Enable" в TRUE.

Включение выхода "привод включен" изменяется в TRUE, чтобы включить питание привода. ЦПУ ожидает от привода сигнал "привод готов".

Когда сигнал "привод готов" доступен на сконфигурированном входе готовности ЦПУ, ось переходит во включенное состояние. Выходной параметр "Status" и тег технологического объекта <Имя оси>.StatusBits.Enable содержит значение TRUE.

Чтобы включить ось без сконфигурированного интерфейса привода, выполните следующие шаги:

1. Проверьте требования, обозначенные выше.
2. Инициализируйте входной параметр "StopMode" требуемым значением. Установите входной параметр "Enable" в TRUE. Ось включена. Выходной параметр "Status" и тег технологического объекта <Имя оси>.StatusBits.Enable содержит значение TRUE.

Чтобы выключить ось, выполните следующие шаги:

1. Переведите ось в состояние покоя.

Вы можете идентифицировать, когда ось находится в состоянии покоя по значению тега технологического объекта <Имя оси>.StatusBits.StandStill.

2. Установите входной параметр "Enable" в значение FALSE после того, как будет достигнуто состояние покоя.
3. Если выходные параметры "Busy" и "Status" и тег технологического объекта <Axis name>.StatusBits.Enable содержат значения FALSE, выключение оси завершено.

10.3.7.3. Инструкция MC_Reset (Подтвердить ошибку)

Таблица 10- 57 Инструкция MC_Reset

LAD / FBD	SCL	Описание
	<pre>"MC_Reset_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Restart:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_Reset, чтобы квитировать "Ошибка эксплуатации с остановкой оси" и "Ошибка конфигурации". Ошибки, которые требуют подтверждения, могут быть найдены в "Перечне ErrorID и ErrorInfo" под "Устранением неисправностей". Перед использованием инструкции MC_Reset Вы, должны устранить причину активной ошибки конфигурации, требующей подтверждения (например, изменив недопустимую величину ускорения в технологическом объекте "Axis"). С V3.0 и более поздней, команда Restart позволяет конфигурации оси быть загруженной в рабочую память в режиме RUN.

- STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- В примере на SCL, "MC_Reset_DB" является именем экземплярного DB.

Задача MC_Reset не может быть прервана никакой другой задачей управления перемещением. Новая задача MC_Reset не прерывает никакие другие активные задачи управления перемещением.

Таблица 10- 58 Параметры для инструкция MC_Reset

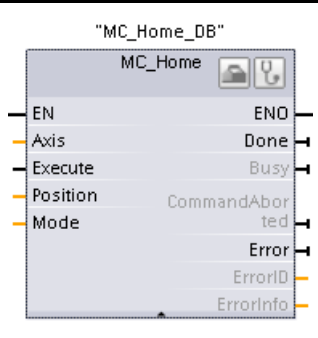
Параметр и тип	Тип данных	Описание
Axis	IN	TO_Axis_1
Execute	IN	Bool
Restart	IN	Bool
		TRUE = Загрузить конфигурацию оси из загрузочной в рабочую память. Команда может быть выполнена только, когда ось включена.
		FALSE = Квитирует активные ошибки
Done	OUT	Bool
		TRUE = Ошибка была квитирована.
Busy	OUT	Bool
		TRUE = Задача выполняется.
Error	OUT	Bool
		TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUTP	Word
		ID ошибки для параметра "Error"
ErrorInfo	OUT	Word
		ID информации об ошибке для параметра "ErrorID"

Чтобы квитировать ошибку с помощью MC_Reset, выполните следующие шаги:

- Проверьте требования, обозначенные выше.
- Запустите подтверждение ошибки по нарастающему фронту на входе Execute.
- Ошибка была квитирована, когда на выходе Done значение TRUE и тег технологического объекта <Имя оси>.StatusBits.Error равен FALSE.

10.3.7.4. Инструкция MC_Home (Перевести ось в начальное положение)

Таблица 10- 59 Инструкция MC_Home

LAD / FBD	SCL	Описание
	<pre>"MC_Home_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Mode:=_int_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_Home, чтобы согласовать координаты оси с реальной, физической позицией привода. Переход в начальное положение требуется для абсолютного позиционирования оси: Чтобы использовать инструкцию MC_Home, ось должна быть вначале включена.

- ¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- ² В примере на SCL, "MC_Home_DB" является именем экземплярного DB.

Доступны следующие типы установки в начальное положение:

- Прямая абсолютная установка в начальное положение (Mode = 0): текущая позиция оси устанавливается в значение параметра "Position".
- Прямая относительная установка в начальное положение (Mode = 1): текущая позиция оси смещается на значением параметра "Position".
- Пассивная установка в начальное положение (Mode = 2): Во время пассивной установки в начальное положение инструкция MC_Home не выполняет перемещений. Перемещение, требуемое для этого шага, должно быть реализовано пользователем через другие команды управления перемещением. Когда датчик обнаружен, ось находится в начальном положении.
- Активная установка в начальное положение (Mode = 3): процедура установки в начальное положение выполняется автоматически.

Таблица 10- 60 Параметры для инструкция MC_Home

Параметр и тип		Тип данных	Описание
Axis	IN	TO_Axis_PTO	Технологический объект оси.
Execute	IN	Bool	Запуск задачи по положительному фронту.
Position	IN	Real	• Mode = 0, 2 и 3 (Абсолютная позиция оси после завершения операции установки в начальное положение) • Mode = 1 (Корректирующее значение для текущего положения оси) Граничные значения: $-1.0e^{12} \leq \text{Положение} \leq 1.0e^{12}$

Параметр и тип		Тип данных	Описание
Mode	IN	Int	Режим установки в начальное положение 0: Прямая абсолютная установка в начальное положение Новым положением оси является значение параметра "Position". 1: Прямая относительная установка в начальное положение Новым положением оси является сумма текущего положения и значения параметра "Position". 2: Пассивная установка в начальное положение Установка в начальное положение согласно конфигурации оси. Вслед за установкой в начальное положение, значение параметра "Position" устанавливается, как новое положение оси. 3: Активная установка в начальное положение Подход к начальному положению согласно конфигурации оси. Вслед за установкой в начальное положение, значение параметра "Position" устанавливается, как новое положение оси.
Done	OUT	Bool	TRUE = Задача завершена
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = Задача была прервана другой задачей во время выполнения.
Error	OUT	Bool	TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUT	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID"

Примечание

Начальное положение оси теряется при следующих условиях

- Отключение оси инструкцией MC_Power
- Переключение между автоматическим и ручным управлением
- После запуска активной установки в начальное положение (После успешного завершения операции установки в начальное положение, начальное положение оси доступно снова.)
- После цикла выключения и включения питания ЦПУ
- После перезапуска ЦПУ (из RUN в STOP или из STOP в RUN)

Чтобы установить ось в начальное положение, выполните следующие шаги:

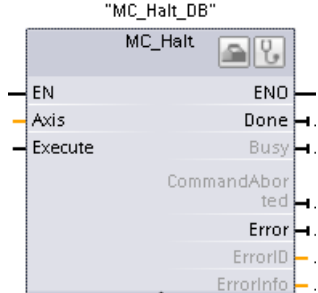
1. Проверьте требования, обозначенные выше.
2. Инициализируйте необходимые входные параметры значениями и запустите операцию установки в начальное положение по нарастающему фронту во входном параметре "Execute".
3. Если выходной параметр "Done" и тег технологического объекта возражает <Имя оси>.StatusBits.HomingDone принимают значение TRUE, установка в начальное положение завершена.

Таблица 10- 61 Реакция на внешнее вмешательство

Mode	Описание	
0 или 1	Задача MC_Home не может быть прервана никакой другой задачей управления перемещением. Новая задача MC_Home не прерывает активных задач управления перемещением. Связанные с позиционированием задачи перемещения возобновляются после установки в начальное положение согласно новому начальному положению (значение во входном параметре Position).	
2	Задача MC_Home может быть прервана следующими задачами управления перемещением: Режим задачи MC_Home = 2, 3: новая задача MC_Home прерывает активную задачу управления перемещением. Режим задачи MC_Home = 2: связанные с позиционированием задачи перемещения возобновляются после установки в начальное положение согласно новому начальному положению (значение во входном параметре Position).	
3	Задача MC_Home может быть прервана следующими задачами управления перемещением: • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog	Новая задача MC_Home прерывает следующие активные задачи управления перемещением: • MC_Home Mode = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.3.7.5. Инструкция MC_Halt (Приостановить ось)

Таблица 10- 62 Инструкция MC_Halt

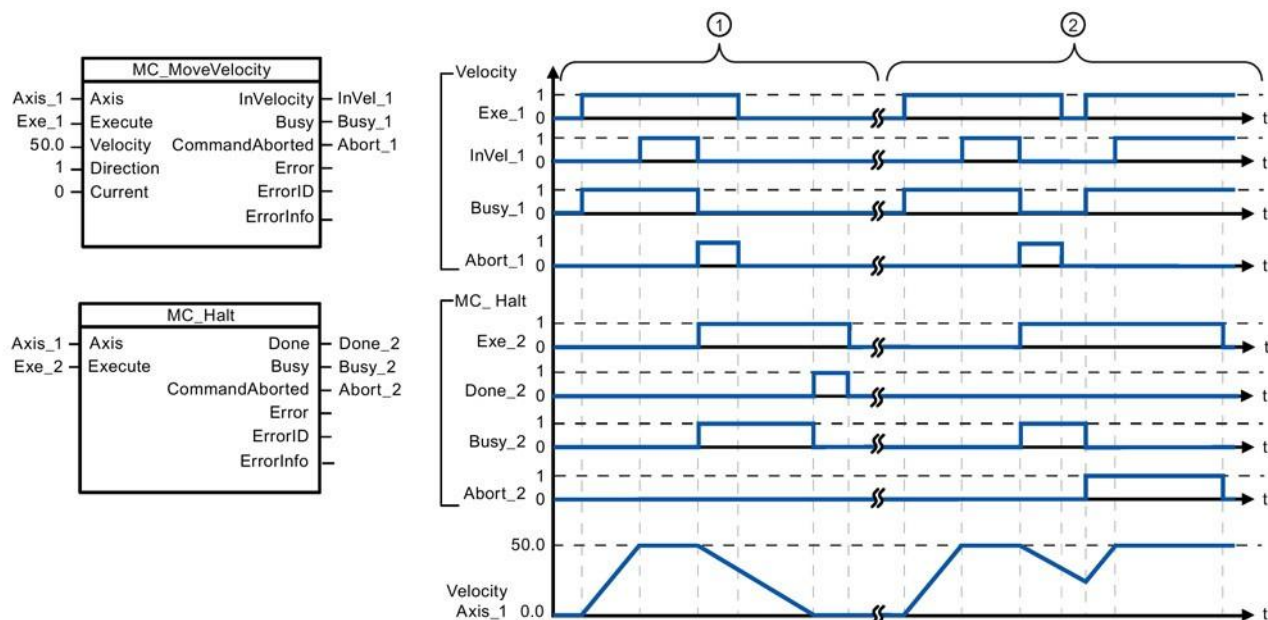
LAD / FBD	SCL	Описание
	<pre>"MC_Halt_DB"(Axis:=_multi_fb_in_, Exe- cute:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_Halt, чтобы остановить любые перемещения и перевести ось в состояние покоя. Позиция покоя не выбирается. Чтобы использовать инструкцию MC_Halt, ось должна вначале быть включена.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

² В примере на SCL, "MC_Halt_DB" является именем экземплярного DB.

Таблица 10- 63 Параметры для инструкция MC_Halt

Параметр и тип		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект оси.
Execute	IN	Bool	Запуск задачи по положительному фронту.
Done	OUT	Bool	TRUE = Нулевая скорость достигнута
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = Задача была прервана другой задачей во время выполнения.
Error	OUT	Bool	TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUT	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID"



Следующие значения были сконфигурированы в конфигурационном окне "Dynamics > General": Acceleration = 10.0 и Deceleration = 5.0

- Ось тормозится задачей MC_Halt, пока она не перейдет в состояние покоя. О состоянии покоя оси сообщается через "Done_2".
- Пока задача MC_Halt тормозит ось, эта задача прерывается другой задачей перемещения. О прерывании операции сообщается через "Abort_2".

Отклик на внешнее воздействие

Задача MC_Halt может быть прервана следующими задачами управления перемещением:

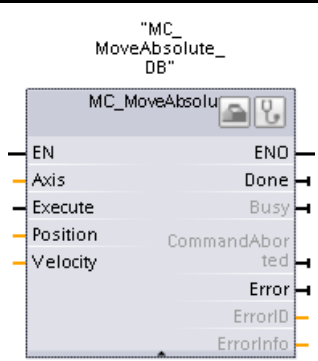
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_Halt прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.6. Инструкция MC_MoveAbsolute (Позиционировать ось абсолютно)

Таблица 10- 64 Инструкция MC_MoveAbsolute

LAD / FBD	SCL	Описание
	<pre>"MC_MoveAbsolute_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_MoveAbsolute, чтобы запустить перемещение оси в абсолютную позицию. Чтобы использовать инструкцию MC_MoveAbsolute, ось должна сначала быть включена, а также должна быть установлена в начальное положение.

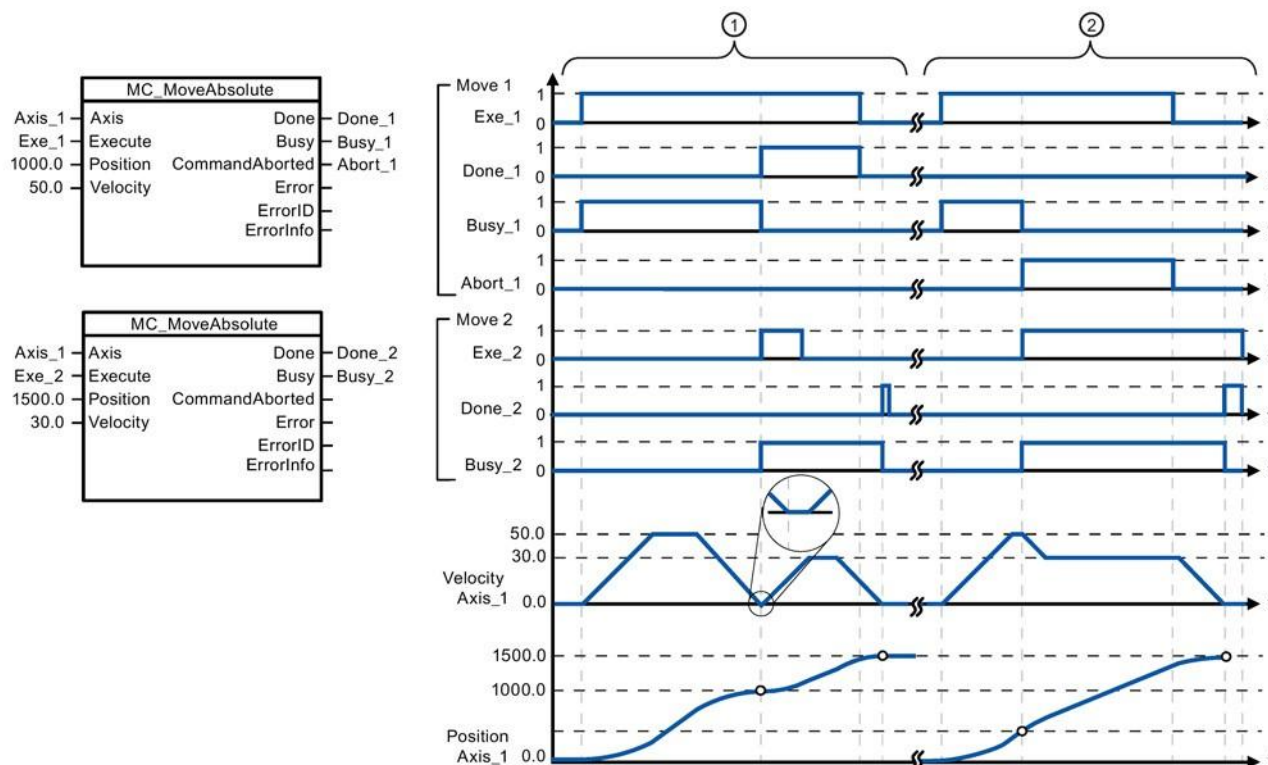
¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

² В примере на SCL, "MC_MoveAbsolute_DB" является именем экземплярного DB.

Таблица 10- 65 Параметры для инструкция MC_MoveAbsolute

Параметр и тип	Тип данных	Описание
Axis	IN	TO_Axis_1
Execute	IN	Bool
Position	IN	Real
Velocity	IN	Real
Done	OUT	Bool
Busy	OUT	Bool
CommandAborted	OUT	Bool

Параметр и тип	Тип данных	Описание
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word



Следующие значения были сконфигурированы в конфигурационном окне "Dynamics > General": Acceleration = 10.0 и Deceleration = 10.0

- ① Ось перемещается в абсолютную позицию 1000.0 с помощью задачи MC_MoveAbsolute. Когда ось достигает целевой позиции, об этом сообщается через "Done_1". Когда "Done_1" = TRUE, запущена другая задача MC_MoveAbsolute с заданной позицией 1500.0. Из-за запаздывания отклика (например, время цикла пользовательской программы, и т.д.), ось кратковременно переходит в состояние покоя (см. увеличенный рисунок). Когда ось достигает новой заданной позиции, об этом сообщается через "Done_2".
- ② Активная задача MC_MoveAbsolute прерывается другой задачей MC_MoveAbsolute. О прерывании операции сообщается через "Abort_1". Ось затем перемещается с новой скоростью в новую заданную позицию 1500.0. Когда новая заданная позиция достигнута, об этом сообщается через "Done_2".

Отклик на внешнее воздействие

Задача MC_MoveAbsolute может быть прервана следующими задачами управления перемещением:

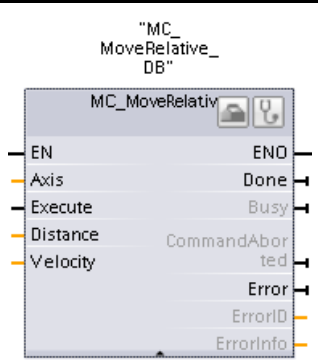
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_MoveAbsolute прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.7. Инструкция MC_MoveRelative (Позиционировать относительно)

Таблица 10- 66 Инструкция MC_MoveRelative

LAD / FBD	SCL	Описание
	<pre>"MC_MoveRelative_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Distance:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_MoveRelative, чтобы запустить позиционирование относительно стартовой позиции. Чтобы использовать инструкцию MC_MoveRelative, ось должна вначале быть включена.

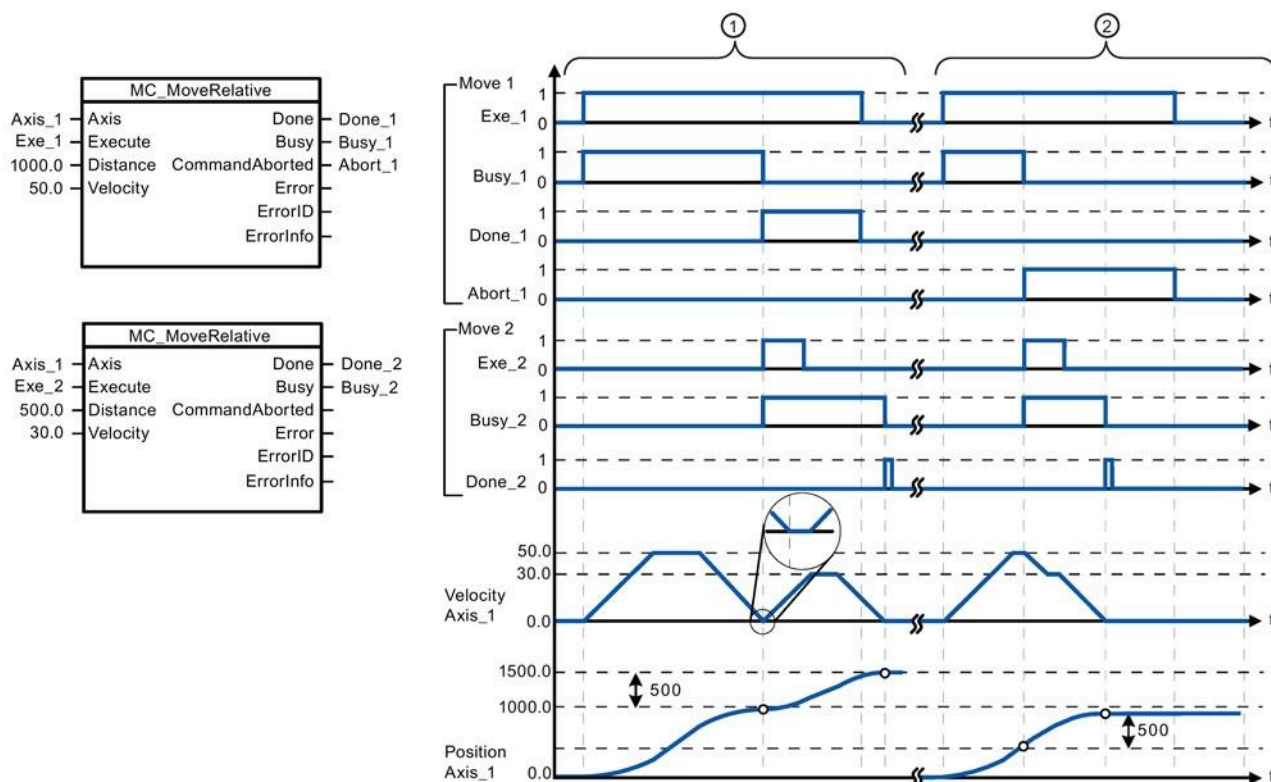
1 STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

2 В примере на SCL, "MC_MoveRelative_DB" является именем экземплярного DB.

Таблица 10- 67 Параметры инструкции MC_MoveRelative

Параметр и тип		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект оси.
Execute	IN	Bool	Запуск задачи по положительному фронту (По умолчанию: False)
Distance	IN	Real	Длина перемещения для операции позиционирования (По умолчанию: 0.0)
Velocity	IN	Real	Скорость оси (По умолчанию: 10.0) Эта скорость не всегда достигается вследствие сконфигурированного ускорения и замедления и расстояния, на которое следует переместиться. Предельные значения: Скорость запуска/остановки ≤ Скорость ≤ Максимальная скорость
Done	OUT	Bool	TRUE = Заданное положение достигнуто.
Busy	OUT	Bool	TRUE = Задача выполняется.

Параметр и тип		Тип данных	Описание
CommandAborted	OUT	Bool	TRUE = Задача была прервана другой задачей во время выполнения.
Error	OUT	Bool	TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUT	Word	ID ошибки для параметра "Error" (По умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (По умолчанию: 0000)



Следующие значения были сконфигурированы в конфигурационном окне "Dynamics > General": Acceleration = 10.0 и Deceleration = 10.0

- ① Ось перемещается задачей MC_MoveRelative на расстояние ("Distance") 1000.0. Когда ось достигает заданной позиции, об этом сообщается через "Done_1". Когда "Done_1" = TRUE, запускается другая задача MC_MoveRelative с перемещением на расстояние 500.0. Из-за запаздывания отклика (например, время цикла пользовательской программы), ось временно переходит в состояние покоя (см. увеличенный рисунок). Когда ось достигает новой заданной позиции, об этом сообщается через "Done_2".
- ② Активная задача MC_MoveRelative прерывается другой задачей MC_MoveRelative. О прерывании операции сообщается через "Abort_1". Ось затем перемещается с новой скоростью в новую заданную позицию 500.0. Когда новая заданная позиция достигнута, об этом сообщается через "Done_2".

Отклик на внешнее воздействие

Задача MC_MoveRelative может быть прервана следующими задачами управления перемещением:

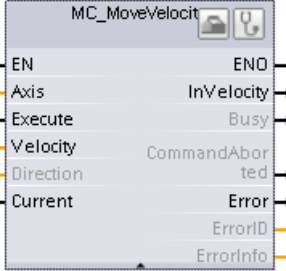
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_MoveRelative прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.8. Инструкция MC_MoveVelocity (Переместить ось с заданной скоростью)у

Таблица 10- 68 Инструкция MC_MoveVelocity

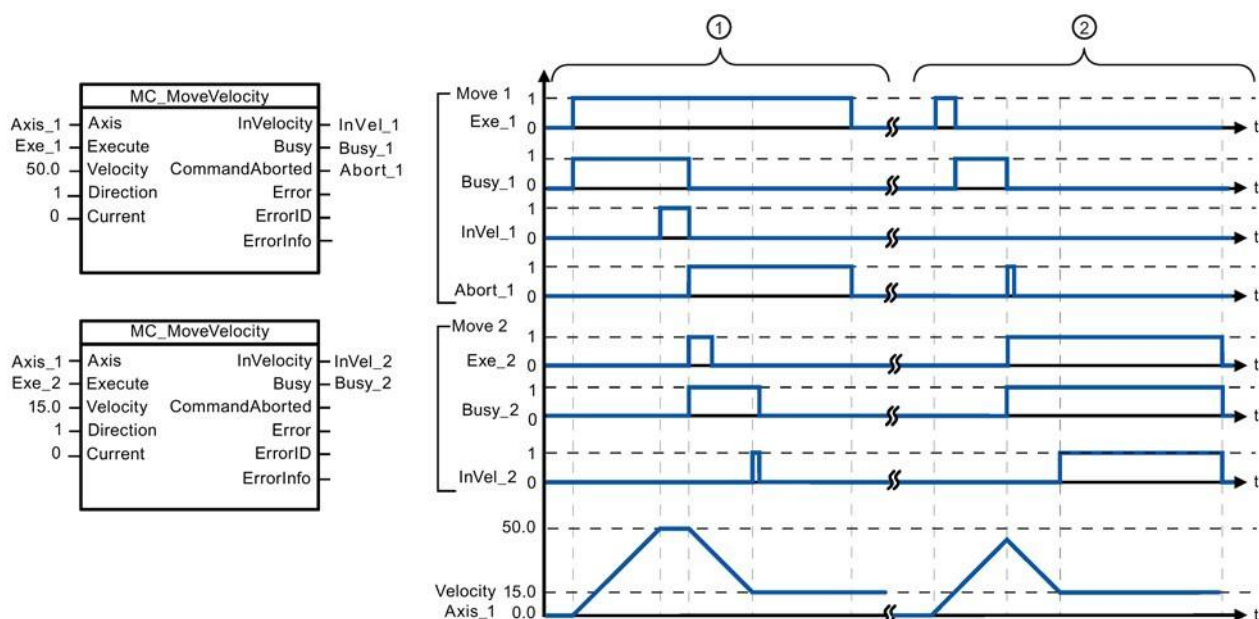
LAD / FBD	SCL	Описание
	<pre>"MC_MoveVelocity_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Velocity:=_real_in_, Direction:=_int_in_, Current:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Используйте инструкцию MC_MoveVelocity, чтобы непрерывно перемещать ось с указанной скоростью. Чтобы использовать инструкцию MC_MoveVelocity, ось должна вначале быть включена.

1 STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

2 В примере на SCL, "MC_MoveVelocity_DB" является именем экземплярного DB.

Таблица 10- 69 Параметры инструкции MC_MoveVelocity

Параметр и тип		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект оси.
Execute	IN	Bool	Запуск задачи по положительному фронту (По умолчанию: False)
Velocity	IN	Real	Определение скорости для перемещения оси (По умолчанию: 10.0) Предельные значения: Скорость запуска/остановки $\leq$ Скорость $\leq$ Максимальная скорость (Velocity = 0.0 допустимо).
Direction	IN	Int	Определение направления: 0: Направление вращения соответствует знаку значения параметра "Velocity" (По умолчанию) 1: Положительное направление вращения (Знак значения параметра "Velocity" игнорируется.) 2: Отрицательное направление вращения (Знак значения параметра "Velocity" игнорируется.)
Current	IN	Bool	Поддерживать текущую скорость: - FALSE: "Поддержка текущей скорости" деактивирована. Используются значения параметров "Velocity" и "Direction". (По умолчанию) - TRUE: "Поддержка текущей скорости", активирована. Значения параметров "Velocity" и "Direction" не принимаются в расчет. Когда ось продолжает перемещение с текущей скоростью, параметр "InVelocity" возвращает значение TRUE.
InVelocity	OUT	Bool	TRUE: - Если "Current" = FALSE: скорость, определенная в параметре "Velocity", была достигнута. - Если "Current" = TRUE: ось перемещается с текущей во время запуска скоростью.
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = Задача была прервана другой задачей во время выполнения.
Error	OUT	Bool	TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo".
ErrorID	OUT	Word	ID ошибки для параметра "Error" (По умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (По умолчанию: 0000)



Следующие значения были сконфигурированы в конфигурационном окне "Dynamics > General": Acceleration = 10.0 и Deceleration = 10.0

- ① Активная задача MC_MoveVelocity сигнализирует через "InVel_1", что его заданная скорость была достигнута. Она затем прерывается другой задачей MC_MoveVelocity. О прерывании работы сообщается через "Abort_1". Когда новая заданная скорость 15.0 достигнута, об этом сообщается через "InVel_2". Ось затем продолжает перемещаться с новой постоянной скоростью.
- ② Активная задача MC_MoveVelocity прерывается другой задачей MC_MoveVelocity до достижения ее заданной скорости. О прерывании работы сообщается через "Abort_1". Когда новая заданная скорость 15.0 достигнута, об этом сообщается через "InVel_2". Ось затем продолжает перемещаться с новой постоянной скоростью.

Отклик на внешнее воздействие

Задача MC_MoveVelocity может быть прервана следующими задачами управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_MoveVelocity прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Примечание

Поведение с выбранной нулевой скоростью (Velocity = 0.0)

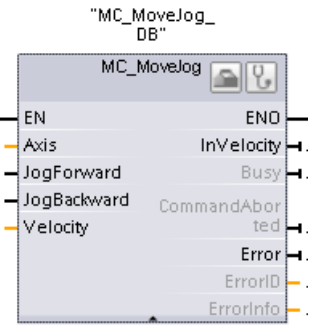
Задача MC_MoveVelocity с "Velocity" = 0.0 (такая же как задача MC_Halt) прерывает активные задачи перемещения и останавливает ось со сконфигурированным замедлением. Когда ось переходит в состояние покоя, выходной параметр "InVelocity" равен TRUE по крайней мере на один цикл программы.

"Busy" равен TRUE во время замедления и изменяется на FALSE вместе с "InVelocity". Если параметр "Execute" установлен в TRUE, "InVelocity" и "Busy" удерживаются.

Когда задача MC_MoveVelocity запущена, бит состояния "SpeedCommand" установлен в технологическом объекте. Бит состояния "ConstantVelocity" установлен при состоянии покоя оси. Оба бита адаптируются к новой ситуации, когда запускается новая задача перемещения.

10.3.7.9. Инструкция MC_MoveJog (Перемещать ось в старт-стопном режиме)

Таблица 10- 70 Инструкция MC_MoveJog

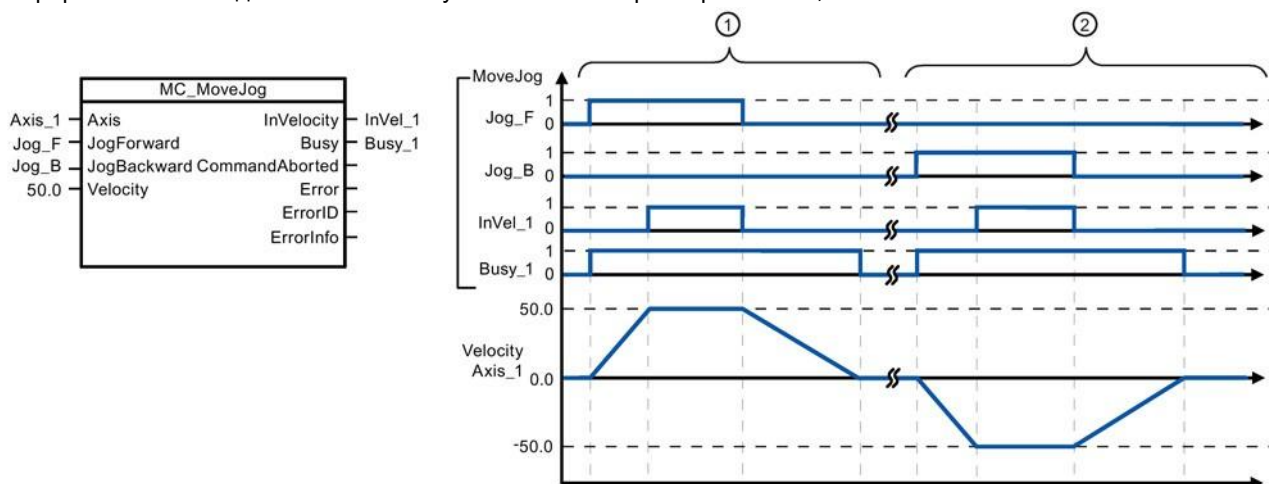
LAD / FBD	SCL	Описание
	<pre>"MC_MoveJog_DB"(Axis:= multi_fb_in_, JogForward:= bool_in_, JogBackward:= bool_in_, Velocity:= real_in_, InVelocity=> bool_out_, Busy=> bool_out_, CommandAborted=> bool_out_, Error=> bool_out_, ErrorID=> word_out_, ErrorInfo=> word_out_);</pre>	Используйте инструкцию MC_MoveJog, чтобы непрерывно перемещать ось с указанной скоростью в старт-стопном режиме. Эта инструкция обычно используется для целей тестирования и ввода в эксплуатацию. Чтобы использовать инструкцию MC_MoveJog, ось должна вначале быть включена.

- ¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- ² В примере на SCL, "MC_MoveJog_DB" является именем экземплярного DB.

Таблица 10- 71 Параметры инструкции MC_MoveJog

Параметр и тип	Тип данных	Описание
Axis	IN	TO_Axis_1
JogForward ¹	IN	Bool
JogBackward ¹	IN	Bool
Velocity	IN	Real
InVelocity	OUT	Bool
Busy	OUT	Bool
CommandAborted	OUT	Bool
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word

¹ Если оба параметра JogForward и JogBackward одновременно равны TRUE, ось останавливается со сконфигурированным замедлением. Ошибка указывается в параметрах "Error", "ErrorID" и "ErrorInfo".



Следующие значения были сконфигурированы в конфигурационном окне "Dynamics > General": Acceleration = 10.0 и Deceleration = 5.0

- ① Ось перемещается в положительном направлении в старт-стопном режиме через "Jog_F". Когда заданная скорость 50.0 достигнута, об этом сообщается через "InVelo_1". Ось вновь тормозится до состояния покоя после сброса "Jog_F".
- ② Ось перемещается в отрицательном направлении в старт-стопном режиме через "Jog_B". Когда заданная скорость 50.0 достигнута, об этом сообщается через "InVelo_1". Ось вновь тормозится до состояния покоя после сброса "Jog_B".

Отклик на внешнее воздействие

Задача MC_MoveJog может быть прервана следующими задачами управления перемещением:

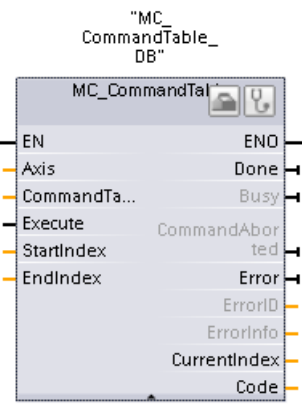
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_MoveJog прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.10. Инструкция MC_CommandTable (Выполнение команд оси как последовательности перемещений)

Таблица 10- 72 Инструкция MC_CommandTable

LAD / FBD	SCL	Описание
	<pre>"MC_CommandTable_DB"(Axis:=_multi_fb_in_, CommandTable:=_multi_fb_in_, Execute:=_bool_in_, StartIndex:=_uint_in_, EndIndex:=_uint_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_, CurrentIndex=>_uint_out_, Code=>_word_out_);</pre>	Обрабатывает серию отдельных перемещений для управления осью, которые могут объединяться в последовательность перемещений. Отдельные перемещения сконфигурированы в таблице команд технологического объекта для выхода пачки импульсов (TO_CommandTable_PTO).

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

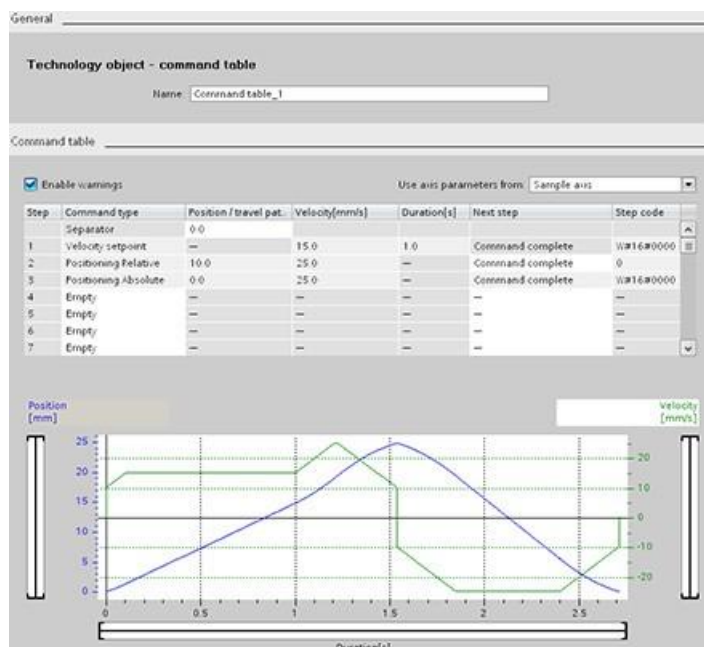
² В примере на SCL, "MC_CommandTable_DB" является именем экземплярного DB.

Таблица 10- 73 Параметры инструкции MC_CommandTable

Параметр и тип		Тип данных	Начальное значение	Описание
Axis	IN	TO_Axis_1	-	Технологический объект оси.
Table	IN	TO_CommandTable_1	-	Технологический объект таблицы команд
Execute	IN	Bool	FALSE	Старт задания по нарастающему фронту.
StartIndex	IN	Int	1	Запуск обработки таблицы команд с этого шага Пределы: $1 \leq \text{StartIndex} \leq \text{EndIndex}$
EndIndex	IN	Int	32	Остановка обработки таблицы команд с этого шага Пределы: $\text{StartIndex} \leq \text{EndIndex} \leq 32$
Done	OUT	Bool	FALSE	Обработка MC_CommandTable завершена успешно

Параметр и тип		Тип данных	Начальное значение	Описание
Busy	OUT	Bool	FALSE	Операция выполняется
CommandAborted	OUT	Bool	FALSE	Задача была прервана во время обработки другой задачей.
Error	OUT	Bool	FALSE	Ошибка произошла во время обработки. Причина указана в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	16#0000	Идентификатор ошибки.
ErrorInfo	OUT	Word	16#0000	Информация об ошибке.
Step	OUT	Int	0	Шаг в настоящее время обрабатывается
Code	OUT	Word	16#0000	Пользовательский идентификатор обрабатываемого в настоящее время шага

Вы можете создать желаемую последовательность перемещения в конфигурационном окне "Command Table" и проверить результат по графическому представлению в диаграмме тренда.



Вы можете выбрать типы команд, которые должны использоваться при обработке таблицы. Могут быть введены до 32 заданий. Команды обрабатываются последовательно.

Таблица 10- 74 Типы команд MC_CommandTable

Тип команды	Описание
Empty	Пустая запись служит заполнителем для любых команд, которые будут добавлены. Пустая запись игнорируется при обработке таблицы команд.
Halt	Приостановить ось. Примечание: Команда выполняется только после команды "Velocity setpoint".
Positioning Relative	Позиционирует ось на основании расстояния. Команда перемещает ось согласно заданным расстоянию и скорости.
Positioning Absolute	Позиционирует ось на основании положения. Команда перемещает ось в заданное положение, используя определенную скорость.

Тип команды	Описание
Velocity setpoint	Перемещает ось с заданной скоростью.
Wait	Ожидает, пока не истекло установленное время. "Wait" не останавливает активное перемещение.
Separator	Добавляет строку "Separator" над выбранной строкой. Строка разделителя позволяет определить более одного профиля в одной таблице команд.

Предпосылки для выполнения MC_CommandTable:

- Технологический объект TO_Axis_PTO V2.0 должен быть правильно сконфигурирован.
- Технологический объект TO_CommandTable_PTO должен быть правильно сконфигурирован.
- Ось должна быть выключена.

Отклик на внешнее воздействие

Задача MC_CommandTable может быть прервана следующими задачами управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

Новая задача MC_CommandTable прерывает следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- Текущее задание управления перемещения с запуском первой команды "Positioning Relative", "Positioning Absolute", "Velocity setpoint" или "Halt".

10.3.7.11. Инструкция MC_ChangeDynamic (Изменить динамические настройки для оси)

Таблица 10- 75 Инструкция MC_ChangeDynamic

LAD / FBD	SCL	Описание
	<pre>"MC_ChangeDynamic_DB"(Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	Изменяет динамические настройки оси управления перемещением: - Изменить значение времени нарастания (ускорение) - Изменить значение времени спада (замедление) - Изменить значение времени спада при аварийной остановке (аварийное замедление) - Изменить значение времени сглаживания (изменение ускорения)

1 STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

2 В примере на SCL, "MC_ChangeDynamic_DB" является именем экземплярного DB.

Таблица 10- 76 Параметры инструкции MC_ChangeDynamic

Параметр и тип		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект оси.
Execute	IN	Bool	Запуск задачи по положительному фронту (По умолчанию: False)
ChangeRampUp	IN	Bool	TRUE = Изменение времени нарастания в соответствии с входным параметром "RampUpTime". По умолчанию: FALSE
RampUpTime	IN	Real	Время (в секундах), чтобы ускориться от состояния покоя до сконфигурированной максимальной скорости без ограничения рывков. По умолчанию: 5.00 Изменение будет влиять на тег <Имя оси>. Config.DynamicDefaults.Acceleration. Эффективность изменения показана в описании этого тега.
ChangeRampDown	IN	Bool	TRUE = Изменение времени спада в соответствии с входным параметром "RampDownTime". По умолчанию: FALSE
RampDownTime	IN	Real	Время (в секундах), чтобы замедлить ось от сконфигурированной максимальной скорости до полной остановки без ограничителя рывков. Значение по умолчанию: 5.00 Изменение будет влиять на тег <Имя оси>. Config.DynamicDefaults.Deceleration. Эффективность изменения показана в описании этого тега.
ChangeEmergency	IN	Bool	TRUE = Изменение времени аварийного замедления в соответствии с входным параметром "EmergencyRampTime". По умолчанию: FALSE.

Параметр и тип		Тип данных	Описание
EmergencyRampTime	IN	Real	Время (в секундах), чтобы замедлить ось от сконфигурированной максимальной скорости до состояния покоя без ограничения рывков в режиме аварийной остановки. По умолчанию: 2.00 Изменение будет влиять на тег <Имя оси>. Config.DynamicDefaults.EmergencyDeceleration. Эффективность изменения показана в описании этого тега.
ChangeJerkTime	IN	Bool	TRUE = Изменение времени сглаживания согласно входному параметру "JerkTime". По умолчанию: FALSE.
JerkTime	IN	Real	Время сглаживания (в секундах), используемое для наклонов ускорения и замедления оси. По умолчанию: 0.25 Изменение будет влиять на тег <Имя оси>. Config.DynamicDefaults.Jerk. Эффективность изменения показана в описании этого тега.
Done	OUT	Bool	TRUE = измененные значения были записаны в технологический блок данных. Описание тегов покажет, когда изменение вступит в силу. По умолчанию: FALSE.
Error	OUT	Bool	TRUE = Ошибка произошла во время выполнения задачи. Причина ошибки может быть определена в параметрах "ErrorID" и "ErrorInfo". По умолчанию: FALSE
ErrorID	OUT	Word	ID ошибки. По умолчанию: 16#0000
ErrorInfo	IN	Word	Информация об ошибке. По умолчанию: 16#0000

Предпосылки для выполнения MC_ChangeDynamic:

- Технологический объект TO_Axis_PTO V2.0 должен быть правильно сконфигурирован.
- Ось должна быть выключена.

Отклик на внешнее воздействие

Команда MC_ChangeDynamic не может быть прервана никакой другой командой управления перемещением.

Новая команда MC_ChangeDynamic не прерывает активных заданий управления перемещением.

Примечание

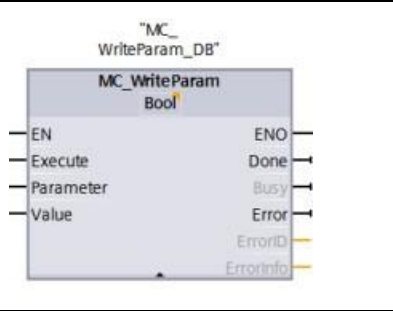
Входные параметры "RampUpTime", "RampDownTime", "EmergencyRampTime" и "RoundingOffTime" могут быть определены со значениями, которые выводят результирующие параметры оси "ускорение", "задержка", "задержка аварийной остановки" и "изменение ускорения" за допустимые пределы.

Убедитесь, что Вы сохраняете параметры MC_ChangeDynamic в рамках динамических конфигурационных настроек для технологического объекта оси.

10.3.7.12. Инструкция MC_WriteParam (Записать параметры технологического объекта

Вы используете инструкцию MC_WriteParam, чтобы записать избранное число параметров из пользовательской программы, чтобы изменить функциональность оси.

Таблица 10- 77 Инструкция MC_WriteParam

LAD / FBD	SCL	Описание
	<pre>"MC_WriteParam_DB"(Parameter:=_variant_in_, Value:=_variant_in_, Execute:=_bool_in_, Done:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	Вы используете инструкцию MC_WriteParam, чтобы выполнить запись в общедоступные параметры (например, ускорение и значения пользовательского DB).

- 1 STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- 2 В примере на SCL, " MC_WriteParam_DB" является именем экземплярного DB.

Вы можете выполнить запись в параметры, которые общедоступны. Вы не можете записать в "MotionStatus" и "StatusBits". Доступные параметры перечислены в таблице ниже:

Имя записываемого параметра	Имя записываемого параметра
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.SideInput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.SideInput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

Таблица 10- 78 Параметры инструкции MC_WriteParam

Параметр и тип		Тип данных	Описание
PARAMNAME	IN	Variant	Имя параметра, в который записывается значение
VALUE	IN	Variant	Значение для записи в назначенный параметр
EXECUTE	IN	Bool	Запуск инструкцию. По умолчанию: FALSE
DONE	OUT	Bool	Значение было записано. По умолчанию: FALSE
BUSY	OUT	Bool	Если TRUE, инструкция работает. По умолчанию: FALSE
ERROR	OUT	Real	Если TRUE, произошла ошибка. По умолчанию: FALSE
ERRORID	OUT	Word	ID ошибки
ERRORINFO	OUT	Word	Соответствующая информация к ERRORID

Таблица 10- 79 Коды состояния для ERRORID и ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Описание
0	0	Успешное изменение параметра Axis TO-DB
8410 _[1]	0028 _[1]	Установка недопустимого параметра (параметр TO-DB оси с неправильной длиной)
8410 _[1]	0029 _[1]	Установка недопустимого параметра (нет параметра TO-DB оси)
8410 _[1]	002B _[1]	Установка недопустимого параметра (параметр только для чтения TO-DB оси)
8410 _[1]	002C _[1]	Установка допустимого параметра, но ось не выключена
Config Error _[2]	Config Error _[2]	Установка допустимого параметра (общедоступный параметр только для чтения TO-DB оси) вне диапазона
Config Error _[3]	Config Error _[3]	Установка допустимого параметра (общедоступный параметр TO-DB оси) вне диапазона

[1] Ошибка в MC_WriteParam

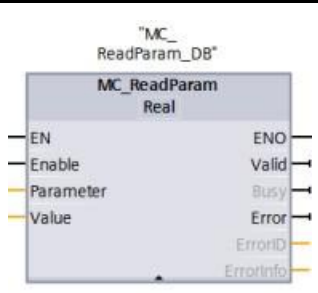
[2] Ошибка в MC_Power

[3] Ошибка в MC_Power и MC_MoveXXX или MC_CommandTable

10.3.7.13. Инструкция MC_ReadParam (считать параметры технологического объекта)

Вы используете инструкцию MC_ReadParam, чтобы считать избранное число параметров, которые указывают текущее положение, скорость, и т.д. для оси, определенной на входе Axis.

Таблица 10- 80 Инструкция MC_ReadParam

LAD / FBD	SCL	Описание
	<pre>"MC_ReadParam_DB"(Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	Вы используете инструкцию MC_ReadParam, чтобы считать одиночные значения состояния, независимо от контрольной точки цикла.

- STEP 7 автоматически создает DB, когда Вы вставляете инструкцию
- В примере на SCL, "MC_ReadParam_DB" является именем экземплярного DB.

Инструкция MC_ReadParam работает согласно разрешению. Пока вход "Enable" находится в состоянии истина, инструкция читает указанный "Parameter" в ячейку хранения "Value".

Значение "MotionStatus" "Position" обновляется в каждой контрольной точке цикла (CCP), на основании текущего значения в/с счетчика.

Значение "MotionStatus" "Velocity" представляет собой скорость команды в конце текущего сегмента (обновление ~10 мс). MC_ReadParam может также считать это значение.

Если происходит ошибка, инструкция переключается в состояние ошибки, которое может быть сброшено только новым нарастающим фронтом на входе "Enable".

Таблица 10- 81 Параметры инструкции MC_ReadParam

Параметр и тип	Тип данных	Описание
ENABLE	IN	Bool
PARAMETER	IN	Variant
VALID	OUT	Bool
BUSY	OUT	Bool
ERROR	OUT	Real
ERRORID	OUT	Word
ERRORINFO	OUT	Word
VALUE	INOUT	Variant

Таблица 10- 82 Коды состояния для ERRORID и ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Описание
0	0	Успешное чтение параметра
8410	0028	Неверный параметр (неправильная длина)
8410	0029	Неверный параметр (нет TO-DB)
8410	0030	Неверный параметр (не читаемый)
8411	0032	Неверный параметр (неправильное значение)

ТО параметры

"MotionStatus" оси состоит из четырех значений. Вы захотите наблюдать изменения этих значений, которые могут быть считаны во время выполнения программы:

Имя переменной	Тип данных	Читаемый через MC_ReadParam
MotionStatus:	Structure	Нет
• Position	REAL	Да
• Velocity	REAL	Да
• Distance	REAL	Да
• TargetPosition	REAL	Да

10.3.8. Контроль активных команд

10.3.8.1. Контроль MC-инструкций с помощью выходного параметра "Done"

Команды управления перемещением с выходным параметром "Done" запускаются входным параметром "Execute" и имеют определенное завершение (например, для инструкции управления перемещением "MC_Home": Установка в начальное положение была успешной). Задача завершена, и ось находится в состоянии покоя.

- Выходной параметр "Done" содержит значение TRUE, если задача была успешно выполнена.
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют о том, что задача все еще обрабатывается, была прервана, или ошибка активна. Команда управления перемещением "MC_Reset" не может быть прервана и таким образом не имеет никакого выходного параметра "CommandAborted".
 - Во время обработки задачи управления перемещением выходной параметр "Busy" содержит значение TRUE. Если задача была выполнена, прервана или остановлена вследствие ошибки, выходной параметр "Busy" изменяет значение на FALSE. Это изменение происходит независимо от сигнала во входном параметре, "Execute".
 - Выходные параметры "Done", "CommandAborted" и "Error" содержат значение TRUE по крайней мере на один цикл. Эти сообщения о состоянии фиксируются, в то время как входной параметр "Execute" устанавливается в TRUE.

Задачи следующих инструкций управления перемещением обладают определенным завершением:

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative

Поведение битов состояния представлено ниже для различных ситуаций в качестве примера.

- Первый пример показывает поведение оси для выполненной задачи. Если задача управления перемещением была полностью выполнена ко времени ее завершения, это обозначено значением TRUE в выходном параметре "Done". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "Done".
- Второй пример показывает поведение оси для прерванной задачи. Если задача управления перемещением прервана во время выполнения, это обозначено значением TRUE в выходном параметре "CommandAborted". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "CommandAborted".
- Третий пример показывает поведение оси, если происходит ошибка. Если ошибка происходит во время выполнения задачи управления перемещением, это обозначено значением TRUE в выходном параметре "Error". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "Error".

Таблица 10- 83 Пример 1 - Полное выполнение задачи

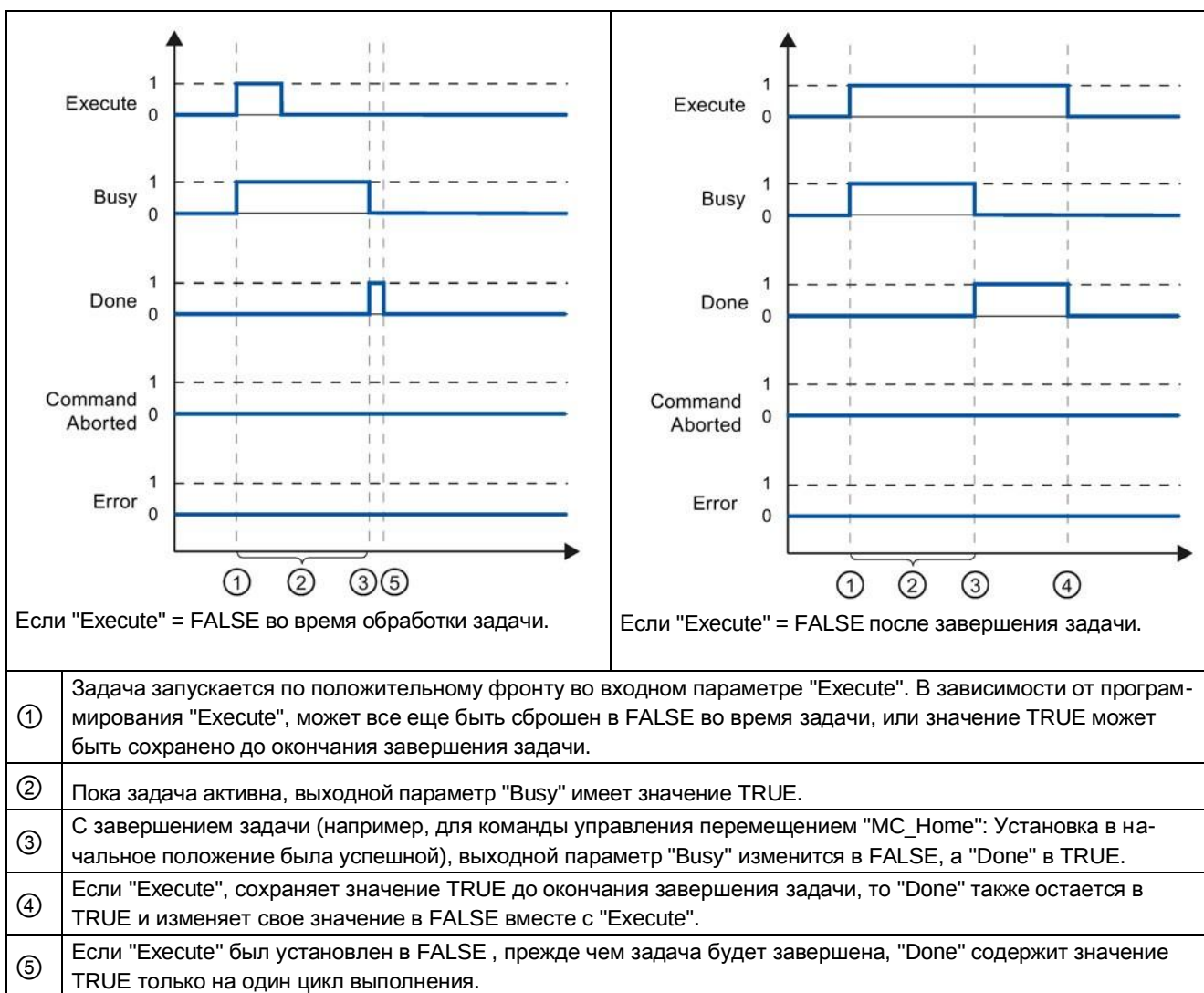


Таблица 10- 84 Пример 2 - Прерывание задачи

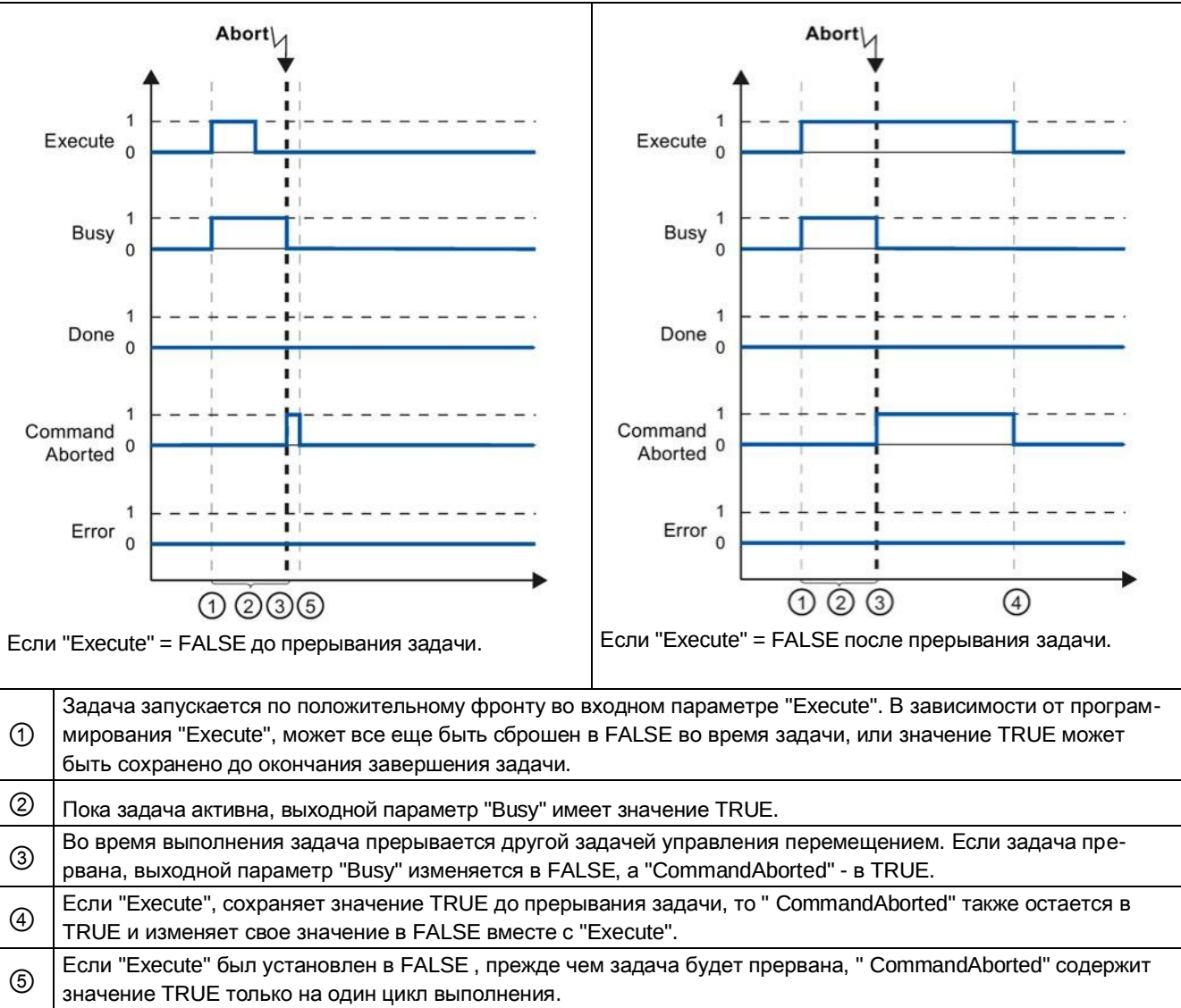
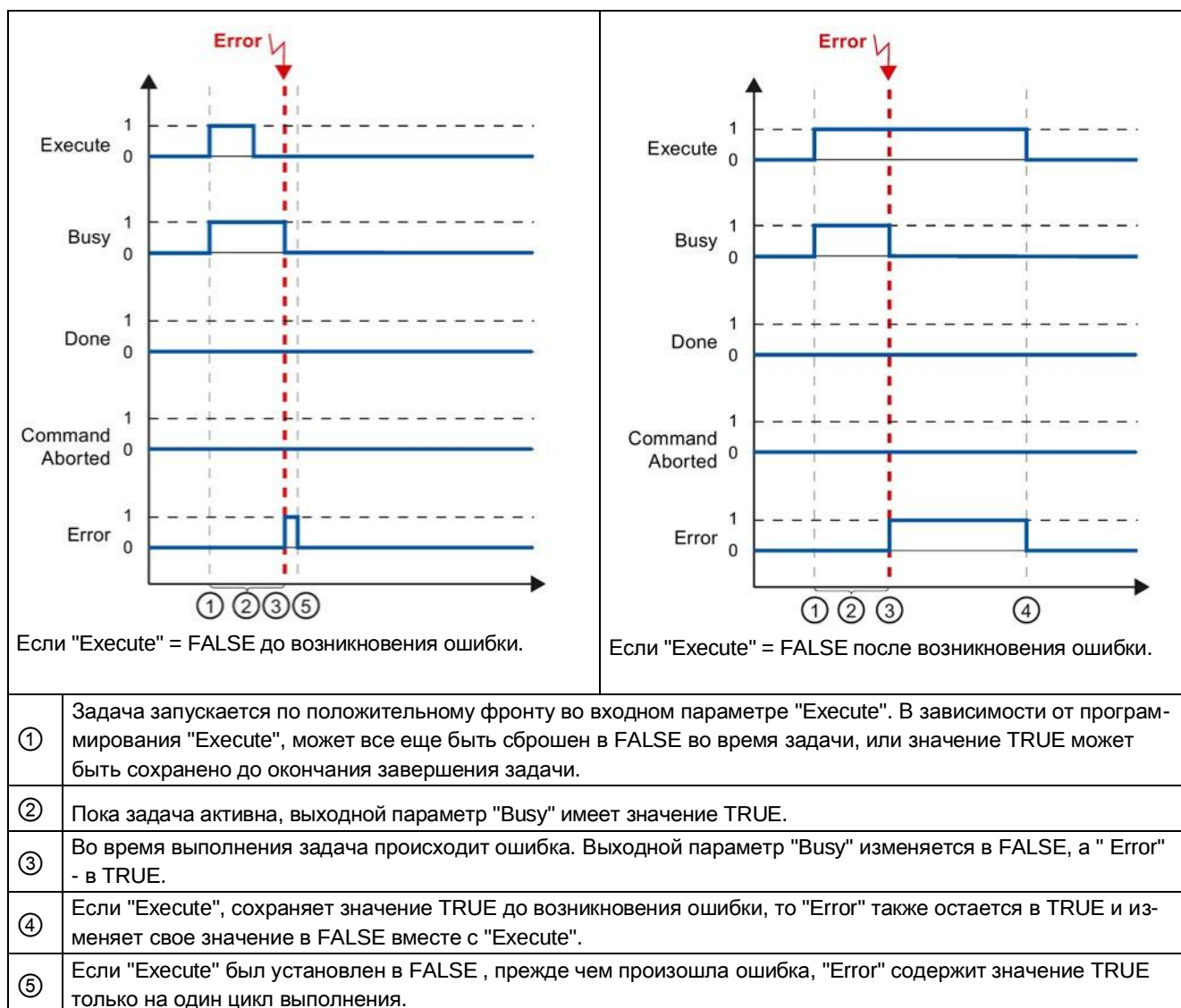


Таблица 10- 85 Пример 3 – Ошибка в процессе выполнения задачи



10.3.8.2. Контроль инструкции MC_Velocity

Задачи команды управления перемещением "MC_MoveVelocity" реализуют перемещение с указанной скоростью:

- У задач команды управления перемещением "MC_MoveVelocity" нет определенного завершения. Цель задачи реализована в первый момент достижения заданной параметром скорости, и ось перемещается с постоянной скоростью. Когда заданная параметром скорость достигнута, это обозначается состоянием TRUE в выходном параметре "InVelocity".
- Задача завершена, заданная параметром скорость достигнута, и входной параметр "Execute", был установлен в значение FALSE. Однако перемещение оси еще не завершено после завершения задачи. Например, перемещение оси может быть остановлено с помощью задачи управления перемещением "MC_Halt".
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют о том, что задача все еще обрабатывается, была прервана, или есть активная ошибка, соответственно.
 - Во время выполнения задачи управления перемещением выходной параметр "Busy" имеет значение TRUE. Если задача была выполнена, прерывалась или вызвала ошибку, выходной параметр "Busy" изменяет свое значение в FALSE. Это изменение происходит независимо от сигнала во входном параметре, "Execute".
 - Выходные параметры "InVelocity", "CommandAborted" и "Error" имеют значение TRUE по крайней мере на один цикл, когда их условия соблюдены. Эти сообщения о состоянии фиксируются, в то время как входной параметр "Execute" устанавливается в TRUE.

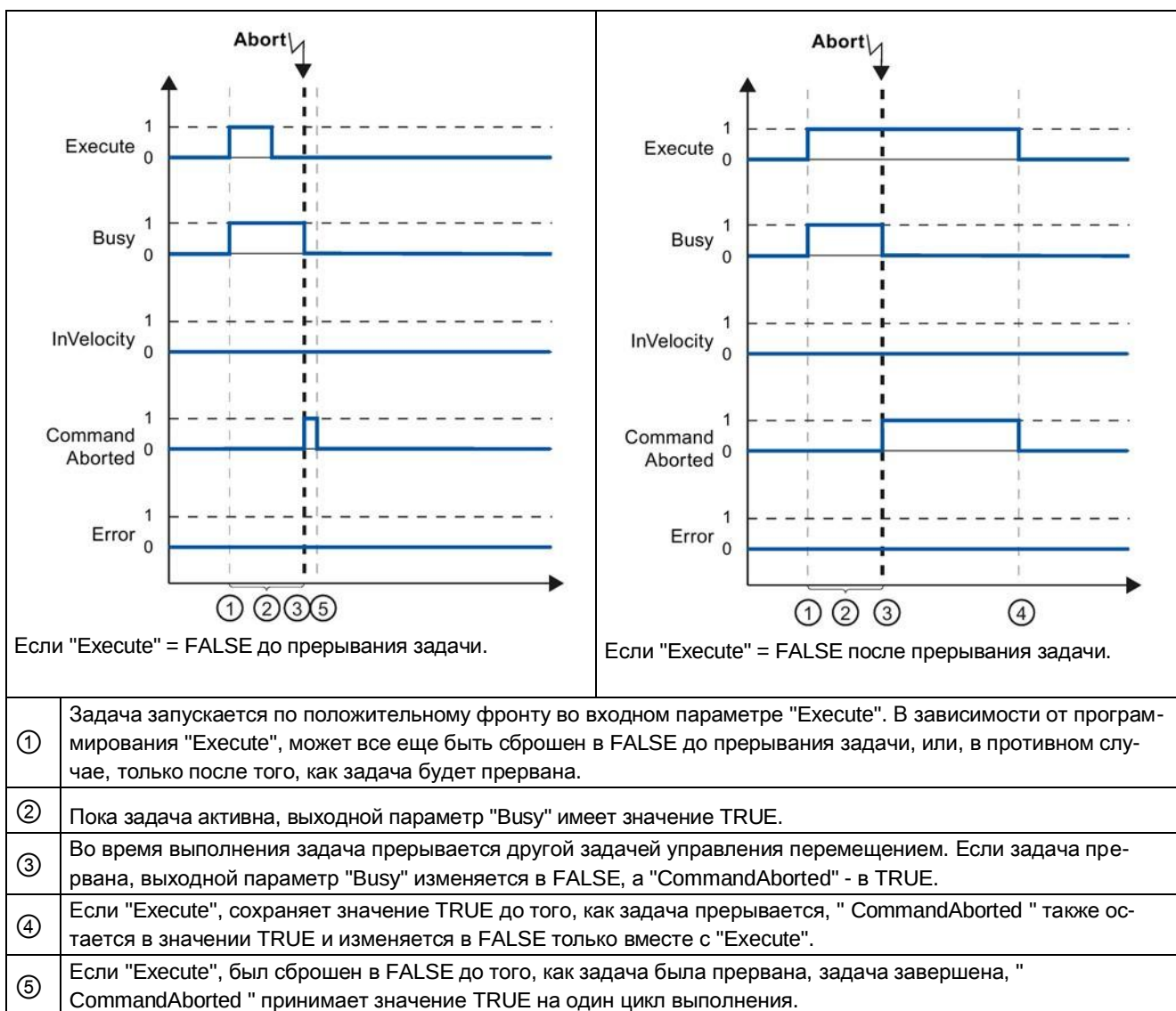
Поведение битов состояния представлено ниже для различных ситуаций в качестве примера:

- Первый пример демонстрирует поведение, когда ось достигает заданной параметром скорости. Если задача управления перемещением была выполнена к тому моменту, когда заданная параметром скорость достигнута, это соответствует значению TRUE в выходном параметре "InVelocity". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "InVelocity".
- Второй пример демонстрирует поведение, если задача прерывается прежде, чем достигнуть заданной параметром скорости. Если задача управления перемещением прервана, прежде чем заданная параметром скорость достигнута, это обозначается состоянием TRUE в выходном параметре "CommandAborted". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "CommandAborted".
- Третий пример демонстрирует поведение оси, если ошибка происходит прежде, чем ось достигает заданной параметром скорости. Если ошибка происходит во время выполнения задачи управления перемещением, прежде чем заданная параметром скорость была достигнута, это обозначается состоянием TRUE в выходном параметре "Error". Состояние сигнала входного параметра "Execute" влияет на продолжительность отображения в выходном параметре "Error".

Таблица 10- 86 Пример 1 – Если заданная параметром скорость достигнута

Если "Execute" = FALSE прежде, чем достигается заданная скорость	Если "Execute" = FALSE после того, как достигается заданная скорость
①	Задача запускается по положительному фронту во входном параметре "Execute". В зависимости от программирования "Execute", может все еще быть сброшен в FALSE до достижения заданной параметром скорости, или, в противном случае, только после того, как она будет достигнута.
②	Пока задача активна, выходной параметр "Busy" имеет значение TRUE.
③	Когда заданная параметром скорость достигнута, выходной параметр "InVelocity" изменяется в TRUE.
④	Если "Execute", сохраняет значение TRUE даже после того, как заданная параметром скорость была достигнута, задача остается активной. "InVelocity" и "Busy" сохраняют значение TRUE и их состояния изменяются в FALSE только вместе с "Execute".
⑤	Если "Execute", был сброшен в FALSE, прежде чем заданная параметром скорость будет достигнута, задача завершена, когда параметризованная скорость достигнута. "InVelocity" имеет значение TRUE на один цикл выполнения и изменяется в FALSE вместе с "Busy".

Таблица 10- 87 Пример 2 – Если задача прерывается до того, как заданная параметром скорость достигнута

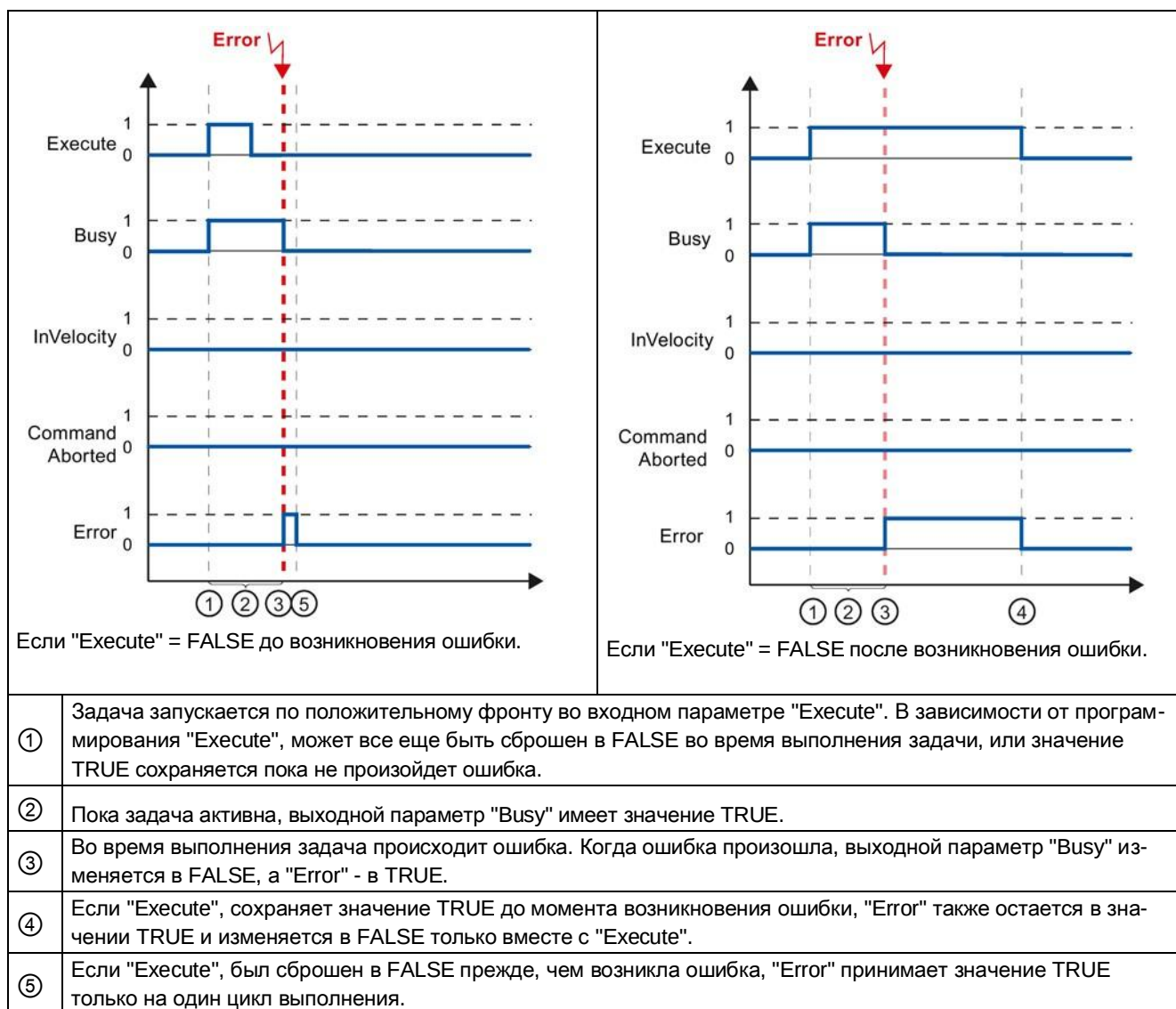


Примечание

При следующих условиях аварийное прекращение работы не обозначено в выходном параметре "CommandAborted":

- Заданная параметром скорость была достигнута, входной параметр "Execute", имеет значение FALSE, и инициируется новая задача управления перемещением.
- Когда заданная параметром скорость была достигнута, и входной параметр "Execute" имеет значение FALSE, задача завершена. Поэтому, запуск новой задачи не обозначен, как аварийное прекращение работы.

Table 10- 88 Пример 3 – Если ошибка происходит до того, как заданная параметром скорость достигнута



Примечание

При следующих условиях ошибка не отображается в выходном параметре "Error":

- Заданная параметром скорость была достигнута, входной параметр "Execute", имеет значение FALSE, и возникает ошибка оси (например, приближение к программному ограничителю).
- Когда заданная параметром скорость была достигнута, и входной параметр "Execute" имеет значение FALSE, задача завершена. После завершения задачи ошибка оси указывается только в команде управления перемещением "MC_Power".

10.3.8.3. Контроль инструкции MC_MoveJog

Задачи команды управления перемещением "MC_MoveJog" реализуют старт-стопный режим.

- Задачи управления перемещением "MC_MoveJog" не имеют определенного завершения. Цель задачи реализована в первый момент достижения заданной параметром скорости, и ось перемещается с постоянной скоростью. Когда заданная параметром скорость достигнута, это обозначено значением TRUE в выходном параметре "InVelocity".
- Задача выполнена, когда введенный параметр "JogForward" или "JogBackward" был установлен в значение FALSE, и ось перешла в состояние покоя.
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют о том, что задача все еще обрабатывается, была прервана, или есть активная ошибка, соответственно.
 - Во время обработки задачи управления перемещением выходной параметр "Busy" имеет значение TRUE. Если задача была выполнена, прервалась или вызвала ошибку, выходной параметр "Busy" изменяется в FALSE.
 - Выходной параметр "InVelocity" имеет состояние TRUE, пока ось перемещается с заданной параметром скоростью. Выходные параметры "CommandAborted" и "Error" указывают состояние по крайней мере на один цикл. Эти сообщения о состоянии фиксируются, пока входной параметр "JogForward" или "JogBackward" установлены в TRUE.

Поведение битов состояния представлено ниже для различных ситуаций в качестве примера.

- Первый пример демонстрирует поведение оси, если заданная параметром скорость достигается и сохраняется. Если задача управления перемещением была выполнена к тому моменту, когда заданная параметром скорость достигнута, это обозначается значением TRUE в выходном параметре "InVelocity".
- Второй пример демонстрирует поведение оси, если задача прервана. Если задача управления движением прервана во время выполнения, это обозначается значением TRUE в выходном параметре "CommandAborted". Поведение не зависит от того, была ли достигнута заданная параметром скорость.
- Третий пример демонстрирует поведение оси, если происходит ошибка. Если ошибка происходит во время выполнения задачи управления перемещением, это обозначается значением TRUE в выходном параметре "Error". Поведение не зависит от того, была ли достигнута заданная параметром скорость.

Таблица 10- 89 Пример 1 - Если заданная параметром скорость достигается и сохраняется

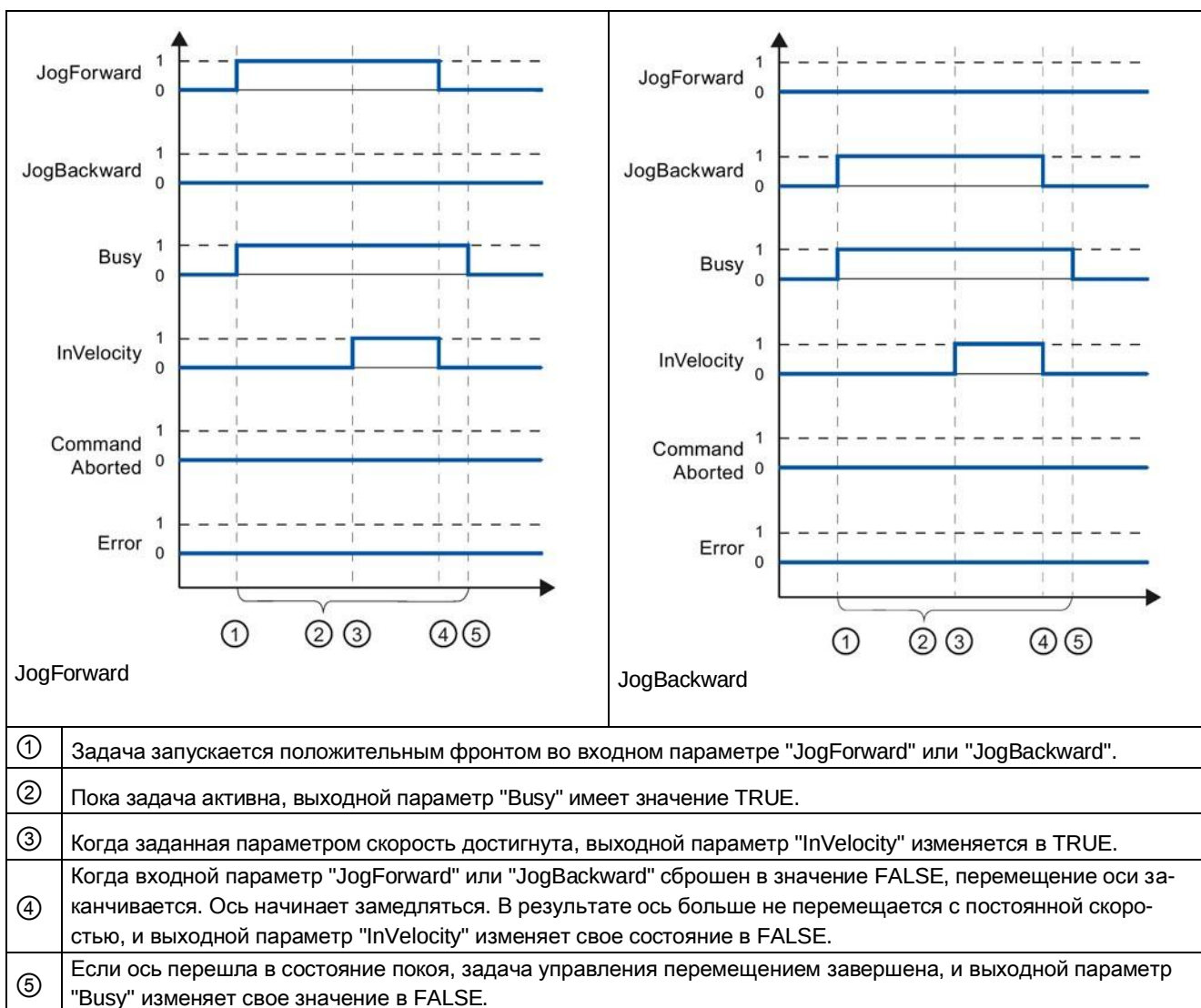
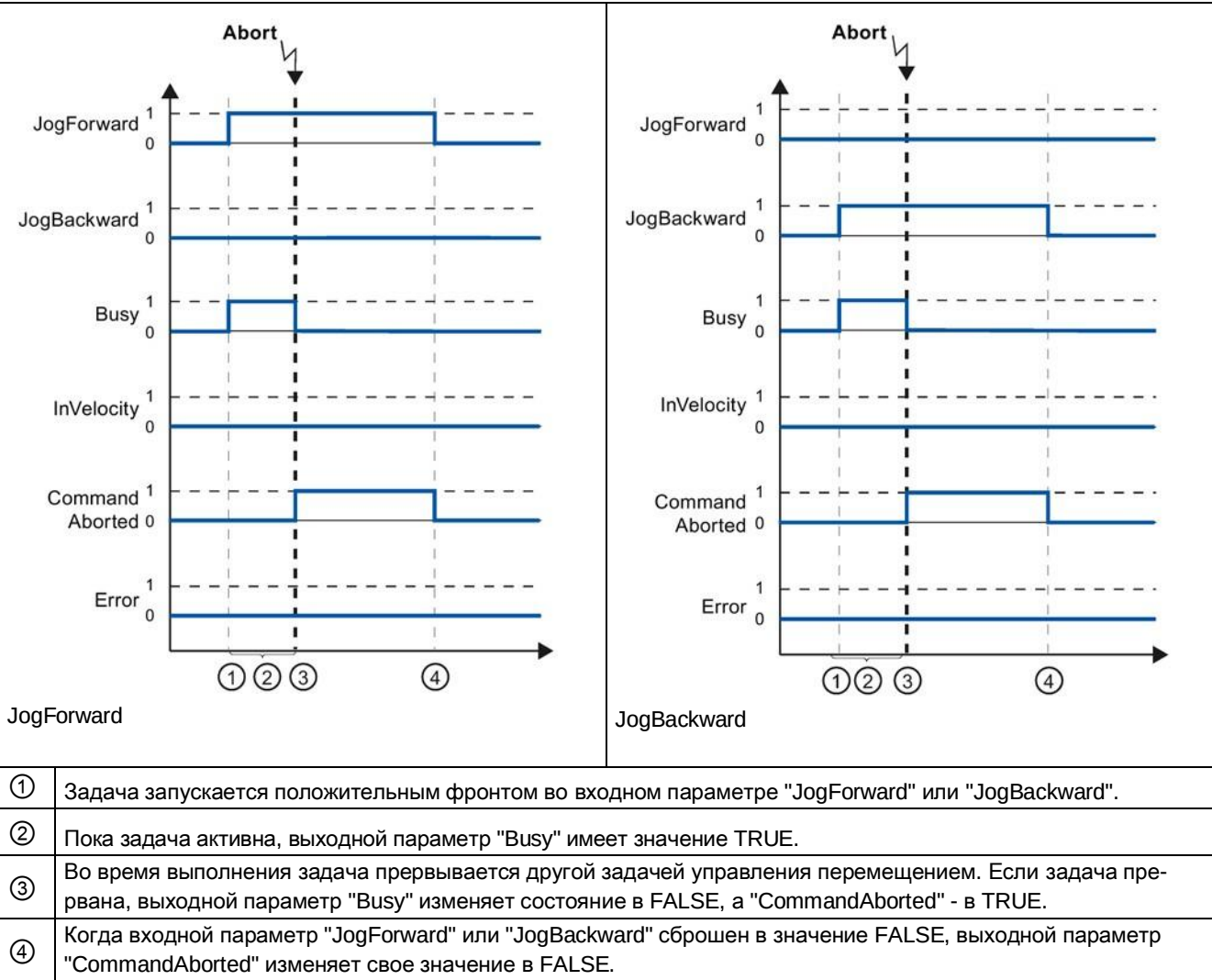


Таблица 10- 90 Пример 2 - Если задача прерывается во время выполнения

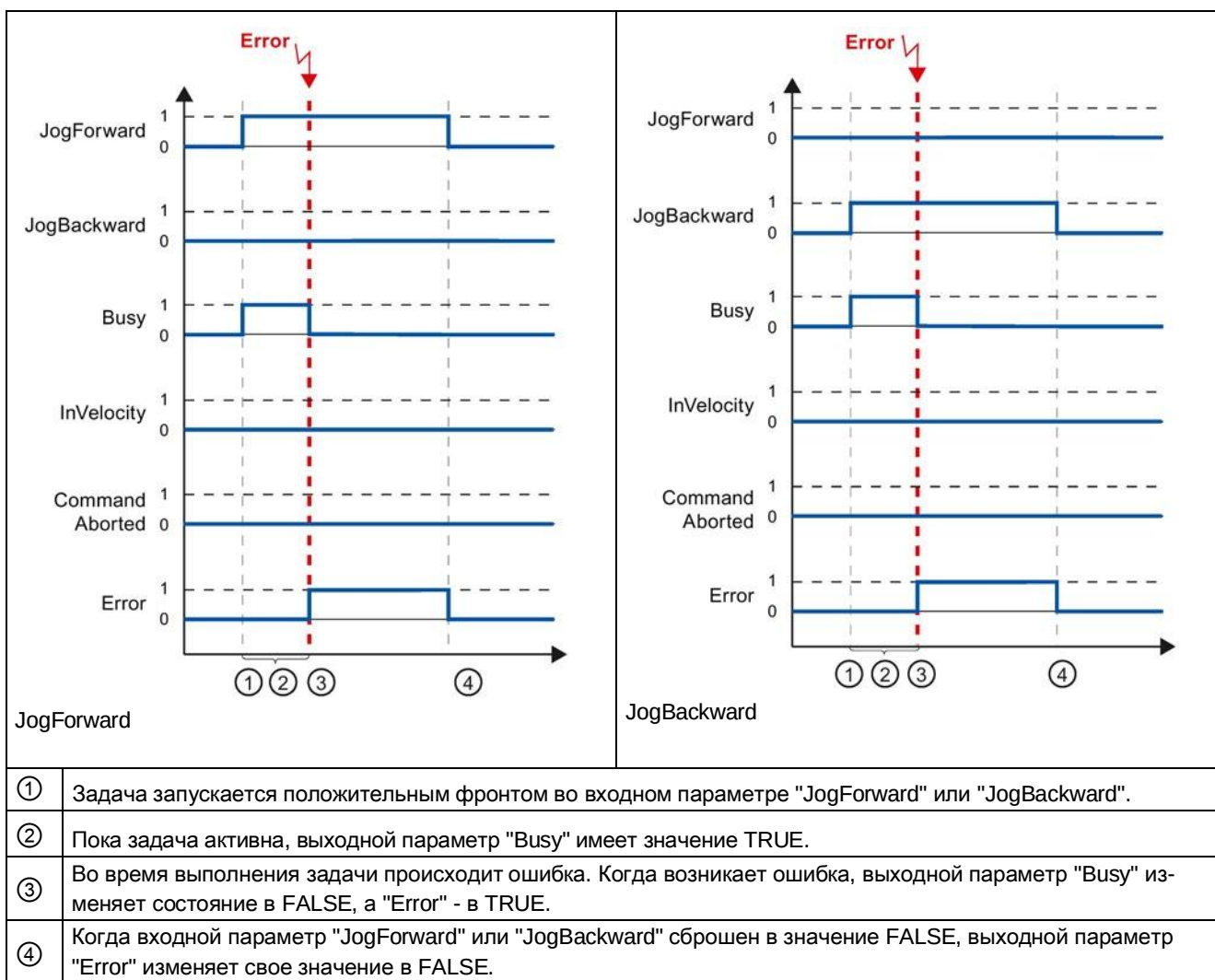


Примечание

Прекращение работы задачи обозначается в выходном параметре "CommandAborted" только на один цикл выполнения, если все условия ниже соблюдены:

Входные параметры "JogForward" и "JogBackward" имеют значение FALSE (но ось все еще замедляется), и инициируется новая задача управления перемещением.

Таблица 10- 91 Пример 3 - Если ошибка происходит во время выполнения задачи



Примечание

Возникновения ошибки обозначается в выходном параметре "Error" только на один цикл выполнения, если все условия ниже соблюдены:

Входные параметры "JogForward" и "JogBackward" имеют значение FALSE (но ось все еще замедляется), и возникает новая ошибка (например, приближение к программному ограничителю).

S7-1200 предлагает несколько типов коммуникаций между ЦПУ и программаторами, устройствами HMI и другими ЦПУ.

ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник может физически получить доступ к Вашим сетям, то он может считать и записать данные.

TIA Portal, ЦПУ и устройства HMI (кроме HMI, использующих GET/PUT), используют безопасный обмен, который защищает от атак повторением и "человек посередине". Как только коммуникация разрешена, обмен заверенными сообщениями осуществляется открытым текстом, который позволяет злоумышленнику считывать данные, но защищает от несанкционированной записи данных. TIA Portal, не коммуникационный процесс, шифрует данные ноу-хау защищенных блоков.

У всех других способов коммуникаций (обмен ввода-вывода через PROFIBUS, PROFINET, ASI или другую шину ввода-вывода, GET/PUT, Т-блок и коммуникационные модули (CM)) нет средств защиты. Вы должны защитить эти формы коммуникаций, ограничив физический доступ. Если злоумышленник может физически получить доступ к Вашим сетям, использующим эти формы коммуникаций, то он сможет считать и записать данные.

Для получения информации о безопасности и рекомендаций, смотри "Рабочие инструкции для промышленной безопасности"

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте обслуживания поддержки от Сименс.

PROFINET

PROFINET используется для обмена данными через пользовательскую программу с другими коммуникационными партнерами по Ethernet:

- В S7-1200 PROFINET поддерживает 16 IO-устройств с максимум 256 submodule, а PROFIBUS позволяет 3 независимых PROFIBUS DP ведущих устройства с поддержкой 32 ведомых на 1 ведущее устройство DP, с максимум 512 модулями на 1 ведущее устройство DP.
- S7-коммуникации
- Протокол пользовательских дейтаграмм (UDP)
- ISO on TCP (RFC 1006)
- Протокол управления передачей (TCP)

PROFINET IO контроллер

- В качестве IO-контроллера, используя PROFINET IO, ЦПУ выполняет обмен максимум с 16 PN устройствами в локальной PN сети или через PN/PN коммутатор (переход). Обратитесь к PROFIBUS and PROFINET International, PI (www.us.profinet.com) для получения дополнительной информации.

PROFIBUS

PROFIBUS используется для обмена данными через пользовательскую программу с другими коммуникационными партнерами по сети PROFIBUS:

- С СМ 1242-5 ЦПУ работает в качестве ведомого устройства PROFIBUS DP.
- С СМ 1243-5 ЦПУ работает в качестве ведущего устройства PROFIBUS DP класса 1.
- Ведомые устройства PROFIBUS DP, ведущие устройства PROFIBUS DP, а также AS-i (3 коммуникационных модуля с левой стороны) и PROFINET являются отдельными коммуникационными сетями, которые не ограничивают друг друга.

AS-i

Коммуникационный модуль ведущего устройства AS-i шины S7-1200 СМ 1243-2 позволяет подключение AS-i сети к ЦПУ S7-1200.

S7 коммуникации: ЦПУ-ЦПУ

Вы можете создать коммуникационное соединение со станцией партнера и использовать инструкции GET и PUT, чтобы обмениваться данными с S7 ЦПУ.

Коммуникации TeleService

В TeleService через GPRS инженерная станция, на которой установлен STEP 7, обменивается по GSM-сети и Интернет со станцией SIMATIC S7-1200 с СР 1242-7.

Соединение работает через сервер телеуправления, который служит посредником и подключен к Интернет.

IO-Link

Ведущее устройство S7-1200 SM 1278 4xIO-Link позволяет подключать IO-Link устройства к ЦПУ S7-1200.

11.1 Соединения асинхронной передачи данных V4.1

Обзор коммуникационных служб

ЦПУ поддерживает следующие коммуникационные службы:

Коммуникационная служба	Выполняемые функции	Используя PROFIBUS		Используя Ethernet
		CM 1243-5 DP master module	CM 1242-5 DP slave module	
PG коммуникации	Ввод в действие, тестирование, диагностика	Да	Нет	Да
HMI коммуникации	Управление и контроль оператором	Да	Нет	Да
S7 коммуникации	Обмен данными, используя сконфигурированные соединения	Да	Нет	Да
Маршрутизация функций PG	Например, тестирование и диагностика вне границ сети	Нет	Нет	Нет
PROFIBUS DP	Обмен данными между ведущим и ведомым устройствами	Да	Да	Нет
PROFINET IO	Обмен данными между I/O контроллерами и I/O устройствами	Нет	Нет	Да
Веб-сервер	Диагностика	Нет	Нет	Да
SNMP (Simple Network Management Protocol)	Стандартный протокол для сетевой диагностики и параметризации	Нет	Нет	Да
Открытые коммуникации через TCP/IP	Обмен данными по Industrial Ethernet с протоколом TCP/IP (с загружаемыми FB)	Нет	Нет	Да
Открытые коммуникации через ISO on TCP	Обмен данными по Industrial Ethernet с протоколом ISO onTCP (с загружаемыми FB)	Нет	Нет	Да
Открытые коммуникации через UDP	Обмен данными по Industrial Ethernet с протоколом UDP (с загружаемыми FB)	Нет	Нет	Да

Доступные соединения

ЦПУ поддерживает следующее максимальное количество одновременных, асинхронных коммуникационных соединений для PROFINET и PROFIBUS. Максимальное количество коммуникационных ресурсов, выделенных каждой категории, фиксировано; Вы не можете изменить эти значения. Однако Вы можете сконфигурировать 6 "свободных доступных соединений", чтобы увеличить количество для любой категории соответственно требованиям Вашего приложения.

Connection resources

Available connection resources reserved for

PG communication: 4

HMI communication: 12

S7 communication: 8

Open user communication: 8

Free available connections: 6

Maximum number available resources: 38

Already configured: 0

Already configured: 0

На основании выделенных коммуникационных ресурсов, для каждого устройства доступно следующее число соединений:

	Терминал программирования (PG)	Человеко-машинный интерфейс (HMI)	GET/PUT клиент/сервер	Открытые пользовательские коммуникации	Веб-браузер
Максимальное число коммуникационных ресурсов	3 (гарантируется поддержка 1 PG)	12 (гарантируется поддержка 4 HMI устройств)	8	8	30 (гарантируется поддержка 3 Веб-браузеров)

- К примеру, у PG есть 3 доступных коммуникационных ресурса. В зависимости от текущих используемых функций PG, он мог бы фактически использовать 1, 2, или 3 из доступных ресурсов. В S7-1200 Вам всегда гарантируют по крайней мере 1 PG; однако, не больше, чем 1 PG допустим.
- Другой пример приведен для HMI, как показано на рисунке ниже. У HMI есть 12 доступных коммуникационных ресурсов. В зависимости от типа HMI или модели, которая у Вас имеется и функций HMI, которые Вы используете, каждый HMI мог бы фактически использовать 1, 2, или 3 из доступных коммуникационных ресурсов. Учитывая число доступных используемых коммуникационных ресурсов, возможно использовать когда-либо более 4 HMI. Однако Вам всегда гарантируют по крайней мере 4 HMI. HMI может использовать свои доступные коммуникационные ресурсы (по 1 на каждую из 3 в общей сложности) для следующих функций:
 - Чтение
 - Запись
 - Вывод тревог плюс диагностика

Пример	HMI 1	HMI 2	HMI 3	HMI 4	HMI 5	Общее число доступных ресурсов
Используемые коммуникационные ресурсы	2	2	2	3	3	12

Примечание

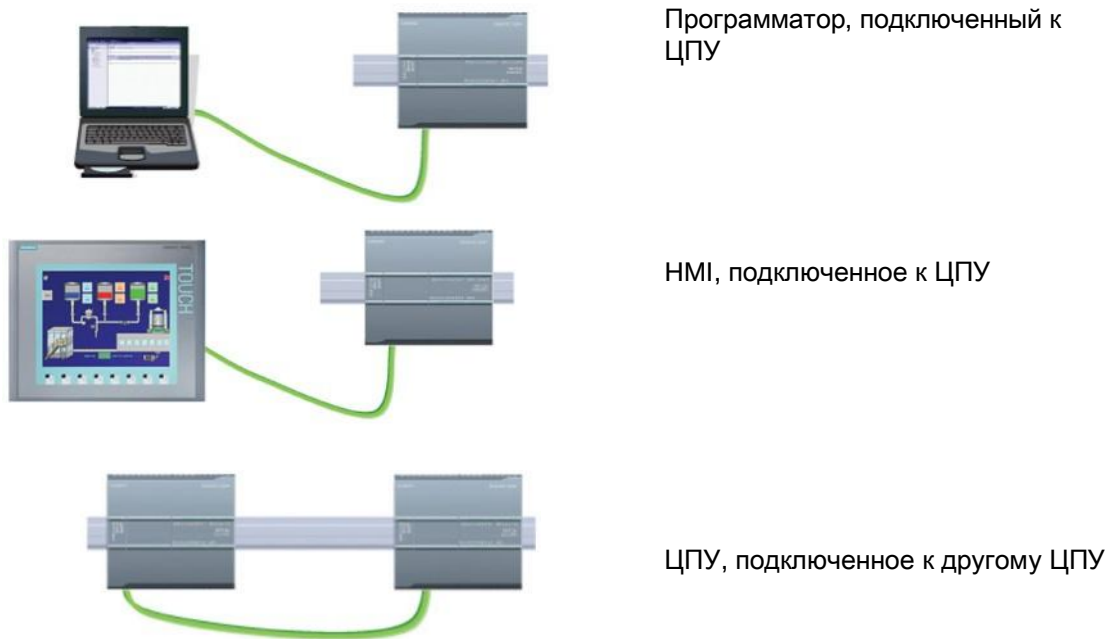
Подключения Веб-сервера (HTTP) : ЦПУ обеспечивает подключение для нескольких веб-браузеров. Количество браузеров, которые может одновременно поддерживать ЦПУ, зависит от того, сколько соединений данный веб-браузер запрашивает/использует.

Примечание

Открытые пользовательские коммуникации, S7-соединение , соединения с HMI, программатором и веб-сервером (HTTP) могут использовать несколько коммуникационных ресурсов в зависимости от используемой функции.

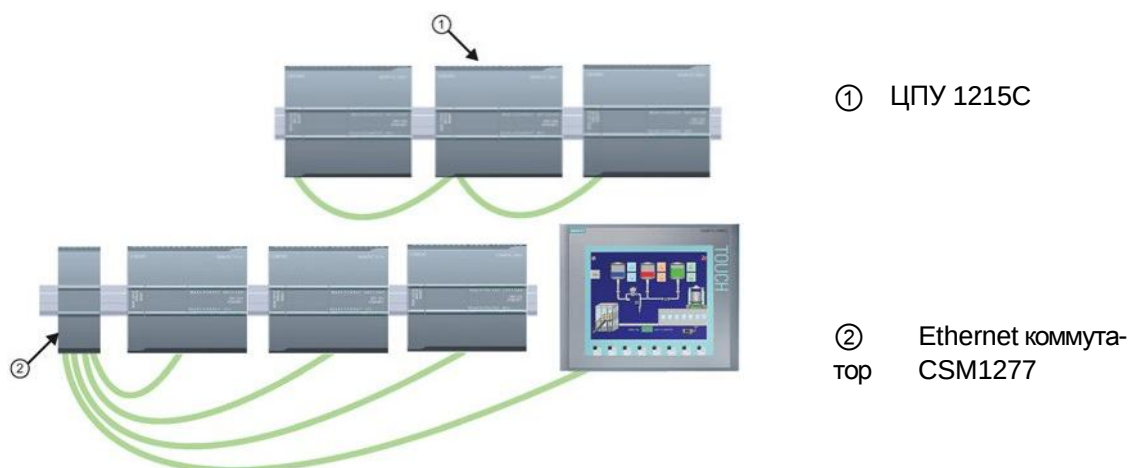
11.2 PROFINET

ЦПУ может выполнять обмен с другими ЦПУ, с программатором, с HMI-устройствами и с устройствами, произведенными не фирмой Сименс, используя стандартные коммуникационные протоколы TCP.



Коммутации в сети Ethernet

Порт PROFINET в ЦПУ 1211C, 1212C и 1214C не содержит устройство коммутации Ethernet. Прямое подключение между программатором или HMI и ЦПУ не требует Ethernet-коммутатора. Однако сети больше чем с двумя ЦПУ или устройствами HMI требуется Ethernet-коммутатор.

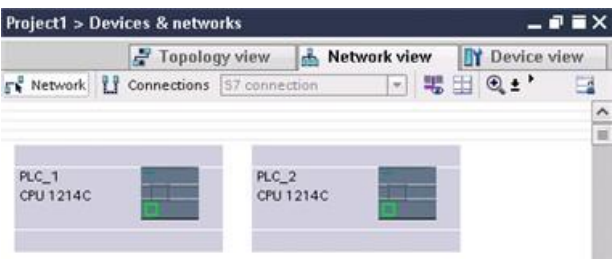
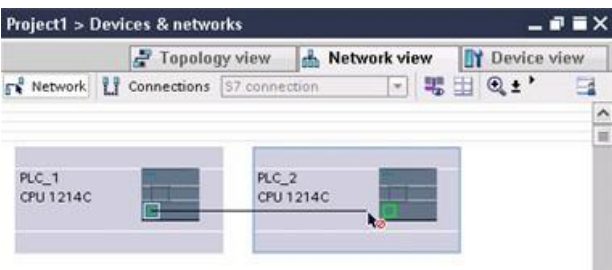
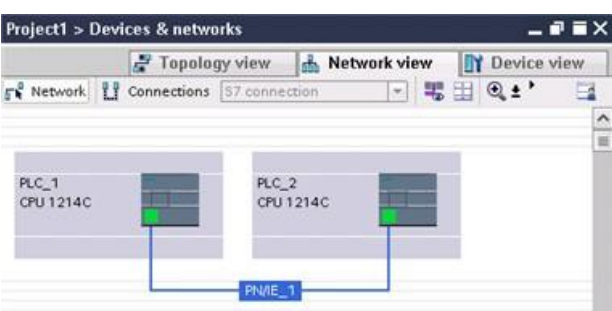


У ЦПУ 1215C и 1217C есть встроенный Ethernet-коммутатор с 2 портами. У Вас может быть сеть с ЦПУ 1215C и двумя другими ЦПУ S7-1200. Вы можете также использовать, монтируемый в стойке Ethernet-коммутатор CSM1277 с 4 портами для соединения нескольких ЦПУ и устройств HMI.

11.2.1. Создание сетевого соединения

Используйте представление "Network view" в конфигурации устройства, чтобы создать сетевые соединения между устройствами в Вашем проекте. После создания сетевого соединения используйте вкладку "Properties" окна инспектора, чтобы сконфигурировать параметры сети.

Таблица 11- 1 Создание сетевого соединения

Действие	Результат
Выберите представление "Network view" чтобы отобразить соединяемые устройства.	
Выберите порт на одном устройстве и выполните перетаскивание соединения на порт второго устройства.	
Отпустите кнопку мыши, чтобы создать сетевое соединение.	

11.2.2. Конфигурирование локального/партнерского пути соединения

Локальное / партнерское (удаленное) соединение определяет логическое назначение двух коммуникационных партнеров для организации коммуникационных служб. Соединение определяет следующее:

- Участвующих в коммуникациях партнеров (один активный, один пассивный)
- Тип соединения (например, ПЛК, HMI или соединение с устройством)
- Путь соединения

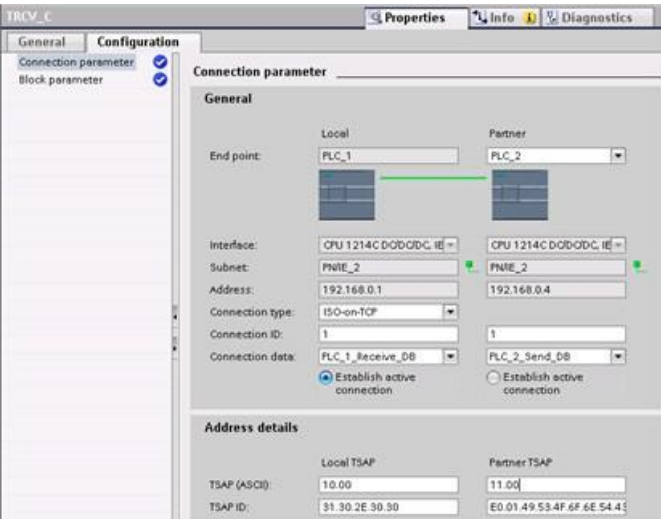
Коммуникационные партнеры выполняют инструкции, чтобы настроить и установить коммуникационное соединение. Вы используете параметры, чтобы определить активного и пассивного коммуникационных партнеров по конечной точке. После того, как соединение настроено и установлено, оно автоматически сохраняется и контролируется ЦПУ.

Если соединение завершено (например, из-за разрыва линии), активный партнер пытается восстановить сконфигурированное соединение. Вы не должны выполнять коммуникационную инструкцию повторно.

Пути соединения

После вставки инструкции TSEND_C, TRCV_C или TCON в пользовательскую программу, окно инспектора выводит на экран свойства соединения каждый раз, когда Вы выбрали любую часть инструкции. Определите коммуникационные параметры на вкладке "Configuration" раздела "Properties" для коммуникационной инструкции.

Таблица 11- 2 Конфигурирование пути соединения (используя свойства инструкции)

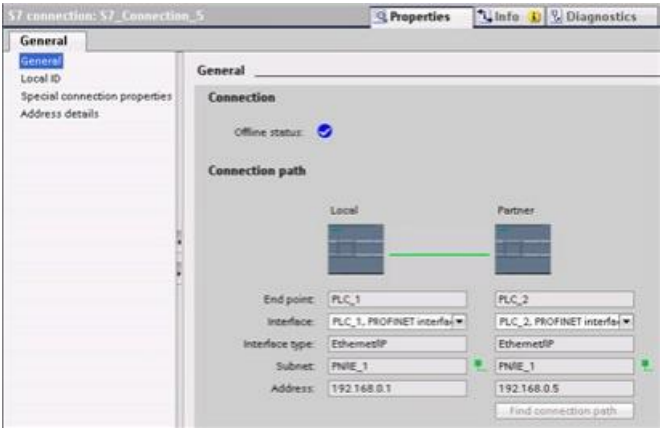
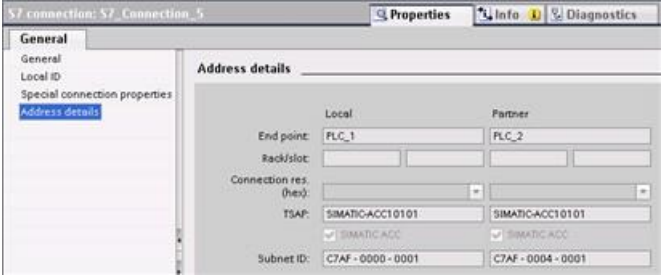
TCP, ISO-on-TCP и UDP	Свойства соединения
Для TCP, ISO-on-TCP и UDP Ethernet-протоколов, используйте "Properties" инструкции (TSEND_C, TRCV_C или TCON), чтобы сконфигурировать "локальные/партнерские" соединения. Иллюстрация показывает "Свойства соединения" со вкладки "Configuration" для ISO-on-TCP соединения.	

Примечание

Когда Вы конфигурируете свойства соединения для одного ЦПУ, STEP 7 разрешает Вам либо выбрать определенный DB соединения в ЦПУ партнера (если такой существует), либо создать DB соединения для ЦПУ партнера. ЦПУ партнера, должен уже быть создан в проекте и не может быть "неопределенным" ЦПУ.

Вы должны вставить инструкцию TSEND_C, TRCV_C или TCON в пользовательскую программу ЦПУ партнера. Когда Вы вставляете инструкцию, выберите DB соединения, который создавался конфигурацией.

Таблица 11- 3 Конфигурирование пути соединения для S7 –коммуникаций (конфигурация устройства)

S7 –коммуникации (GET и PUT)	Свойства соединения
Для S7-коммуникаций используйте редактор сети "Devices & networks" , чтобы сконфигурировать локальное/партнерское соединение . Вы можете нажать кнопку "Highlighted: Connection", чтобы получить доступ к "Properties". Вкладка "General" предлагает несколько разделов свойств: • "General" (показано) • "Local ID" • "Special connection properties" • "Address details" (показано)	 

Обратитесь к теме "Протоколы" (стр. 634) в разделе "PROFINET" или к теме "Создание S7-соединения" (стр. 774) в разделе "S7-коммуникации" для получения дополнительной информации и списка доступных коммуникационных инструкций.

Таблица 11- 4 Параметры для соединения нескольких ЦПУ

Параметр		Определение
Address		Назначенные IP-адреса
General	End point	Имя, назначенное ЦПУ партнера (принимающему)
	Interface	Имя, назначенное интерфейсам
	Subnet	Имя, назначенное подсетям
	Interface type	<i>Только S7-коммуникации.</i> Тип интерфейса
	Connection type	Тип Ethernet-протокола
	Connection ID	Номер ID
	Connection data	Расположение хранилища данных локального и партнерского ЦПУ
	Establish active connection	Радиокнопка для выбора локального или партнерского ЦПУ, в качестве активного участника
Address details	End point	<i>Только S7-коммуникации.</i> Имя, назначенное ЦПУ партнера (принимающему)
	Rack/slot	<i>Только S7-коммуникации.</i> Положение стойки и слота
	Connection resource	<i>Только S7-коммуникации.</i> Компонент TSAP, используемый при конфигурировании S7-соединения с S7-300 или S7-400 ЦПУ
	Port (decimal):	TCP и UDP: Порт ЦПУ партнера в десятичном формате
	TSAP ¹ and Subnet ID:	ISO on TCP (RFC 1006) и S7-коммуникации: TSAP локального и партнерского ЦПУ в ASCII и шестнадцатеричных форматах

¹ При конфигурировании соединения с ЦПУ S7-1200 для ISO-on-TCP используйте только ASCII-символы в расширении TSAP для пассивных коммуникационных партнеров.

Точки доступа к услугам транспортного уровня (TSAP)

Использование TSAP, протокола ISO-on-TCP и S7-коммуникаций позволяет множественные соединения по единственному IP-адресу (до 64K соединений). TSAP однозначно определяют эти коммуникационные соединения конечной точки с IP-адресом.

В разделе "Address Details" диалогового окна "Connection Parameters" Вы определяете TSAP, который будет использоваться. TSAP соединения в ЦПУ вводится в поле "Local TSAP". TSAP, назначенный для соединения в Вашем партнерском ЦПУ, вводится в поле "Partner TSAP".

Номера портов

С TCP и UDP протоколами, конфигурация параметров соединения локального (активного) ЦПУ соединения должна определить удаленный IP-адрес и номер порта партнерского (пассивного) ЦПУ соединения.

В разделе "Address Details" диалогового окна "Connection Parameters" Вы определяете порты, которые будут использоваться. Порт соединения в ЦПУ вводится в поле "Local Port". Порт, назначенный для соединения в Вашем партнерском ЦПУ, вводится в поле "Partner Port".

11.2.3. Назначение адресов Интернет протокола (IP)

11.2.3.1. Назначение IP адресов программатору и сетевым устройствам

Если Ваш программатор использует встроенную карту адаптера, подключенную к LAN Вашего завода (и возможно глобальной сети), то ID сети IP-адреса и маска подсети Вашего ЦПУ и встроенной карты адаптера программатора должны быть одними и теми же. ID сети представляет собой первую часть IP-адреса (три начальных октета) (например, 211.154.184.16), который определяет, к какой IP-сети Вы находитесь. Маска подсети обычно имеет значение 255.255.255.0; однако, так как Ваш компьютер находится в LAN завода, у маски подсети могут быть различные значения (например, 255.255.254.0), чтобы настроить уникальные подсети. Маска подсети при комбинировании с IP-адресом устройства согласно математической операции И, определяет границы IP-подсети.

Примечание

В плане развития глобальной сети, где Ваши программаторы, сетевые устройства и IP-маршрутизаторы связываются с миром, Вы должны присвоить уникальные IP-адреса, чтобы избежать конфликта с другими сетевыми пользователями. Свяжитесь с персоналом ИТ-отдела Вашей компании, который знаком с сетями Вашего завода с целью назначения IP-адресов.

ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к ЦПУ через веб-сервер

Несанкционированный доступ к ЦПУ или изменение переменных ПЛК на недопустимые значения могли бы нарушить работу процесса и привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку активация веб-сервера позволяет авторизованным пользователям выполнять изменения рабочего режима, запись в данные ПЛК и обновления встроенного ПО, Сименс рекомендует, чтобы Вы соблюдали следующие меры безопасности:

- Разрешите доступ к веб-серверу только по протоколу HTTPS.
- Защитите сильными паролями идентификаторы пользователей веб-сервера (стр. 789). Сильные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.
- Не расширяйте минимальные по умолчанию полномочия пользователя веб-сервера "Everybody".
- Выполняйте проверку ошибок и проверку диапазонов Ваших переменных в Вашей программной логике, так как пользователи Веб-страницы могут изменить переменные ПЛК на недопустимые значения.

Если Ваш программатор использует Ethernet-to-USB адаптер, подключенный к изолированной сети, ID сети IP-адреса и маска подсети Вашего ЦПУ и Ethernet-to-USB адаптера программатора должны быть точно такими же. ID сети представляет собой первую часть IP-адреса (три начальных октета) (например, 211.154.184.16), который определяет, к какой IP-сети Вы находитесь. Маска подсети обычно имеет значение 255.255.255.0. Маска подсети при комбинировании с IP-адресом устройства согласно математической операции И, определяет границы IP-подсети.

Примечание

Адаптер Ethernet-USB полезен, когда Вы не хотите подключать программатор к LAN Вашей компании. Во время предпусковых или приемо-сдаточных испытаний, такое решение особенно полезно.

Таблица 11- 5 Назначение Ethernet-адресов

Карта адаптера в программаторе	Тип сети	Адрес Интернет протокола (IP)	Маска подсети
Встроенная карта адаптера	Подключенная к LAN Вашего завода (и, возможно, глобальной сети)	ID сети Вашего ЦПУ и встроенной карты адаптера программатора должны быть точно такими же. ¹	Маска подсети Вашего ЦПУ и встроенной карты адаптера должна быть точно такой же. Маска подсети обычно имеет значение 255.255.255.0 ; однако, так как Ваш компьютер находится в LAN завода, у маски подсети могут быть отличные значения (например, 255.255.254.0), чтобы организовать уникальные подсети. ²
Карта адаптера Ethernet-USB	Подключенная к изолированной сети	ID сети Вашего ЦПУ и карты адаптера Ethernet- USB для программатора должны быть точно такими же. ¹	Маска подсети Вашего ЦПУ и карты адаптера Ethernet- USB должна быть точно такой же. Маска подсети обычно имеет значение 255.255.255.0 . ²

¹ ID сети представляет собой первую часть IP-адреса (три начальных октета) (например, **211.154.184.16**), который определяет, к какой IP-сети Вы находитесь.

² Маска подсети при комбинировании с IP-адресом устройства согласно математической операции И, определяет границы IP-подсети.

Назначение или проверка IP-адреса программатора, используя "My Network Places" (на рабочем столе)

Вы можете назначить или проверить IP-адрес своего программатора, выбирая следующие пункты меню:

- (Щелчок правой клавишей) "My Network Places"
- "Properties"
- (Щелчок правой клавишей) "Local Area Connection"
- "Properties"

В диалоговом окне "Local Area Connection Properties", в поле "This connection uses the following items:" перейдите по списку вниз к "Internet Protocol (TCP/IP)". Щелкните по "Internet Protocol (TCP/IP)" и затем по кнопке "Properties". Выберите "Obtain an IP address automatically (DHCP)" или "Use the following IP address" (чтобы ввести статический IP-адрес).

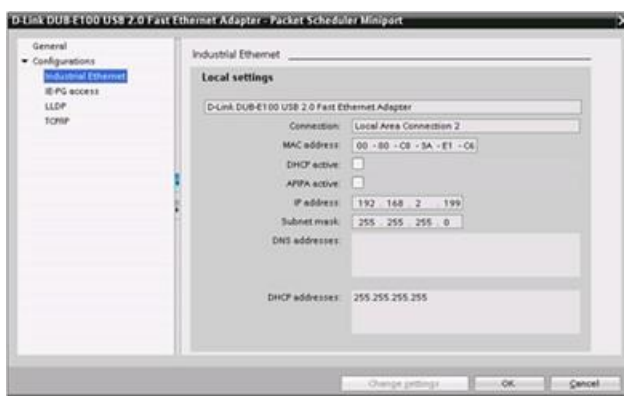
Примечание

Протокол динамического конфигурирования хоста (DHCP) автоматически назначает IP-адрес Вашему программатору при включении питания сервера DHCP.

11.2.3.2. Проверка IP-адреса Вашего программатора

Вы можете проверить MAC-адрес и IP-адрес Вашего программатора, выбирая следующие пункты меню:

1. В "дереве проекта", разверните "Online access".
2. Щелкните правой кнопкой по требуемой сети и выберите "Properties".
3. В сетевом диалоговом окне разверните "Configurations" и выберите "Industrial Ethernet". MAC-адрес и IP-адрес программатора отображаются на экране.

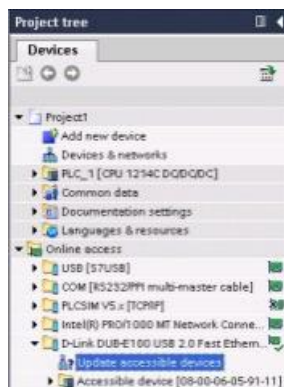


11.2.3.3. Назначение IP-адреса ЦПУ он-лайн

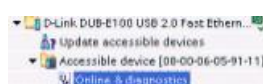
Вы можете назначить IP-адрес сетевому устройству он-лайн. Это особенно полезно при начальном конфигурировании устройства.

1. В "Дереве проекта", проверьте, что никакой IP-адрес не назначен ЦПУ, выбирая следующие пункты меню:

- "Online access"
 - < Карта адаптера для сети, в которой расположено устройство >
 - "Update accessible devices"
- ПРИМЕЧАНИЕ: Если STEP 7 показывает MAC-адрес вместо IP-адреса, то IP-адрес не был назначен.

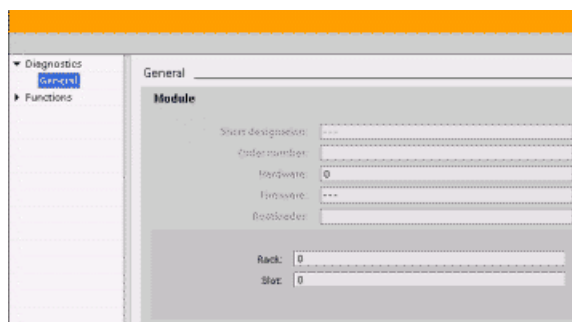


2. Под требуемым доступным устройством дважды щелкните по "Online & diagnostics".

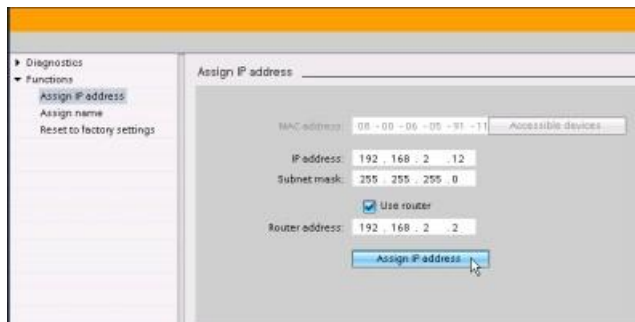


3. В диалоговом окне "Online & diagnostics", выберите следующие пункты меню:

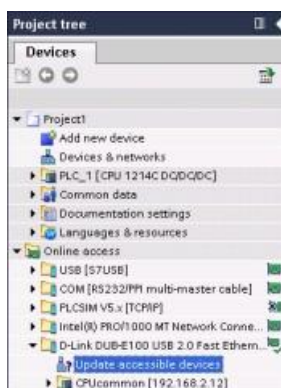
- "Functions"
- "Assign IP address"



4. В поле "IP address" введите новый IP-адрес и щелкните по кнопке "Assign IP address".



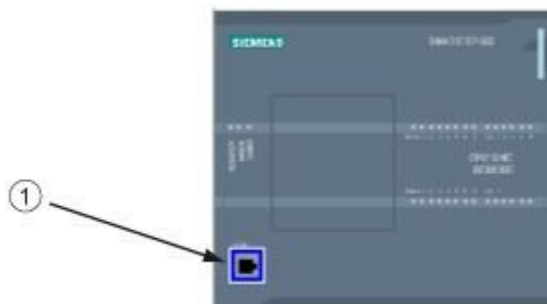
5. В "дереве проекта" проверьте, что Ваш новый IP-адрес был присвоен ЦПУ, выбирая следующие пункты меню:
- "Online access"
 - < Карта адаптера для сети, в которой расположено устройство >
 - "Update accessible devices"



11.2.3.4. Конфигурирование IP-адреса для ЦПУ в проекте

Конфигурирование PROFINET-интерфейса

Чтобы сконфигурировать параметры для PROFINET-интерфейса, выберите зеленое поле PROFINET на ЦПУ. Вкладка "Properties" в окне инспектора отображает PROFINET-порт.



- ① PROFINET-порт

Конфигурирование IP-адреса

Ethernet (MAC) адрес: В сети PROFINET каждому устройству производителем назначен для идентификации адрес на уровне управления доступом к среде (MAC-адрес). MAC-адрес состоит из шести групп по две шестнадцатеричных цифры в каждой, разделенных дефисами (-) или двоеточиями (:), в порядке передачи, (например, 01-23-45-67-89-AB или 01:23:45:67:89:AB).

IP адрес: У каждого устройства должен также быть адрес Интернет протокола (IP). Этот адрес позволяет устройству доставлять данные по более сложной, маршрутизированной сети.

Каждый IP-адрес разделен на четыре 8-разрядных сегмента и выражается в десятичном формате с разделением точкой (например, 211.154.184.16). Первая часть IP-адреса используется для ID сети (В какой сети Вы находитесь?), а вторая часть адреса - для идентификации хоста (уникальна для каждого устройства в сети). IP-адрес 192.168.x.y является стандартным обозначением, распознаваемым как часть частной сети, которая не имеет маршрута в Интернете.

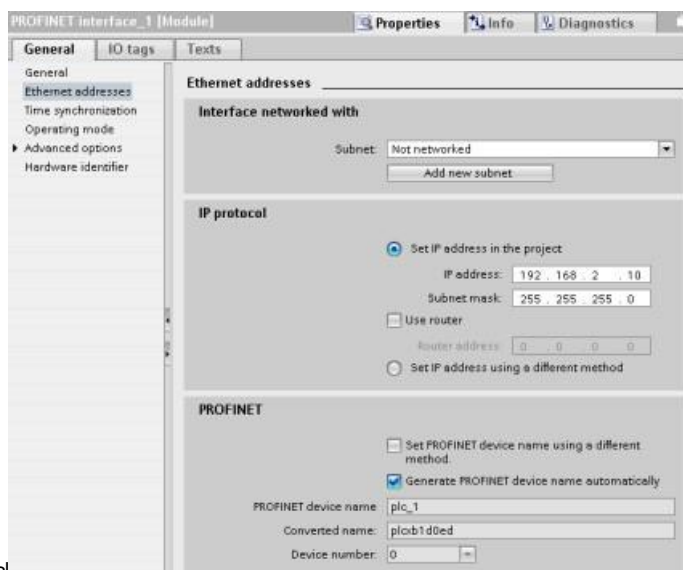
Маска подсети: Подсеть является логическим группированием подключенных сетевых устройств. Узлы подсети обычно расположены в непосредственной близости друг от друга в Локальной сети (LAN). Маска (известная как маска подсети или сетевая маска) определяет границы IP подсети.

Маска подсети 255.255.255.0 обычно подходит для небольшой локальной сети. Это означает, что у всех IP-адресов в этой сети должны быть те же первые 3 октета, а различные устройства в этой сети идентифицированы последним октетом (8-разрядное поле). Пример является назначением маски подсети 255.255.255.0 и IP-адреса от 192.168.2.0 до 192.168.2.255 для устройств небольшой локальной сети.

Соединение между различными подсетями осуществляется только через маршрутизатор. Если используются подсети, то должен использоваться IP-маршрутизатор.

IP-маршрутизатор: Маршрутизаторы представляют собой канал связи между LAN. Используя маршрутизатор, компьютер в LAN может отправить сообщения в любые другие сети, за которыми могли бы быть другие LAN. Если место назначения данных не в LAN, то маршрутизатор передает данные другой сети или группе сетей, где они могут быть доставлены в место назначения.

Маршрутизаторы опираются на IP-адреса, чтобы доставлять и получать пакеты данных.



Свойства IP-адресов: В окне свойств выберите пункт конфигурации "Ethernet addresses". STEP 7 отображает диалоговое окно конфигурации Ethernet-адреса, который связывает проект с IP-адресом ЦПУ, который получает этот проект.

Таблица 11- 6 Параметры IP-адреса

Параметр		Описание
Подсеть	Имя подсети, к которой подключено устройство. Нажмите кнопку "Add new subnet", чтобы создать новую подсеть. "Not connected" является значением по умолчанию. Два типа подключения возможны:	
	- Значение по умолчанию "Not connected" обеспечивает локальное подключение. - Требуется подсеть, когда в Вашей сети есть два или более устройств.	
IP -протокол	IP address	Назначенный для ЦПУ IP-адрес
	Subnet mask	Назначенная маска подсети
	Use IP router	Щелкните по флажку, чтобы выбрать использование IP-маршрутизатора
	Router address	Назначенный для маршрутизатора IP-адрес, если используется

Примечание

Все IP-адреса конфигурируются, когда Вы загружаете проект. Если у ЦПУ нет предварительно сконфигурированного IP-адреса, Вы должны связать проект с MAC-адресом целевого устройства. Если Ваш ЦПУ соединен с маршрутизатором в сети, Вы должны также ввести IP-адрес маршрутизатора.

Радиокнопка "Set IP address using a different method" позволяет Вам изменять IP-адрес онлайн или при помощи инструкции "T_CONFIG (стр. 704)" после того, как программа будет загружена. Этот метод назначения действует только для IP-адреса ЦПУ.

**ПРЕДУПРЕЖДЕНИЕ****Загрузка аппаратной конфигурации с помощью "Set IP address using different method"**

После загрузки аппаратной конфигурации с включенной опцией "Set IP address using a different method", невозможно перевести рабочий режим ЦПУ из RUN в STOP или из STOP в RUN.

Пользовательское оборудование продолжает работать при этих условиях, что может привести к неожиданной работе машины или процесса, которая могла бы привести к смерти, тяжелым телесным повреждениям или материальному ущербу, если не приняты надлежащие меры предосторожности.

Гарантируйте, что IP-адрес(-а) Вашего(-их) ЦПУ установлены перед использованием ЦПУ в фактической среде автоматизации. Это может быть сделано при помощи Вашего пакета программирования STEP 7, инструмента S7-1200 или подключенного устройства HMI в сочетании с инструкцией T_CONFIG.

ПРЕДУПРЕЖДЕНИЕ

Состояние, в котором могла бы остановиться сеть PROFINET

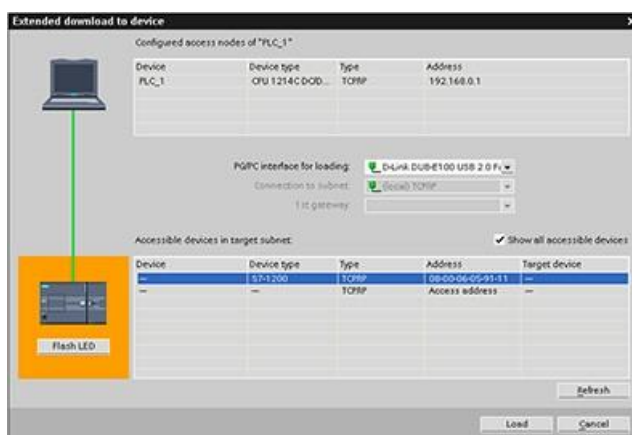
При изменении IP-адреса ЦПУ он-лайн или из пользовательской программы, возможно создать состояние, в котором могла бы остановиться сеть PROFINET.

Если IP-адрес ЦПУ будет изменен на IP-адрес, находящийся вне подсети, то сеть PROFINET потеряет возможность выполнять коммуникации, и весь обмен данными остановится. Пользовательское оборудование может быть сконфигурировано таким образом, чтобы продолжить работать при этих условиях. Потеря PROFINET-коммуникаций может привести к неожиданной работе машины или процесса, которая могла бы привести к смерти, тяжелым телесным повреждениям или материальному ущербу, если не приняты надлежащие меры предосторожности.

Если IP-адрес должен быть изменен вручную, убедитесь, что новый IP-адрес находится в подсети.

11.2.4. Тестирование сети PROFINET

После завершения конфигурирования загрузите проект (стр. 203) в ЦПУ. Все IP-адреса конфигурируются, когда Вы загружаете проект.



Назначение IP-адреса устройству он-лайн

У S7-1200 ЦПУ нет предварительно сконфигурированного IP-адреса. Вы должны вручную назначить IP-адрес для ЦПУ:

- Чтобы назначить IP-адрес устройству он-лайн, обратитесь к теме "Конфигурация устройства: назначение IP-адреса ЦПУ он-лайн" (стр. 625) для этой пошаговой процедуры.
- Чтобы назначить IP-адрес в Вашем проекте, Вы должны сконфигурировать IP-адрес в конфигурации устройства, сохранить конфигурацию и загрузить ее в ПЛК. Обратитесь к теме "Конфигурация устройства: конфигурирование IP-адреса ЦПУ в Вашем проекте" (стр. 626) для получения более подробной информации.

Использование диалога "Extended download to device" для тестирования подключенных сетевых устройств

Функция S7-1200 ЦПУ "Download to device" и ее диалоговое окно "Extended download to device" может показать все доступные сетевые устройства и были ли назначены уникальные IP-адреса всем устройствам. Чтобы отобразить все допустимые и доступные устройства с их назначенными MAC- или IP-адресами, проверьте флажок "Show all accessible devices".



Если требуемое сетевое устройство не находится в этом списке, то обмен с этим устройством был прерван по некоторым причинам. Устройство и сеть должны быть проверены на предмет конфигурационных и/или аппаратных ошибок.

11.2.5. Определение Ethernet (MAC) адреса в ЦПУ

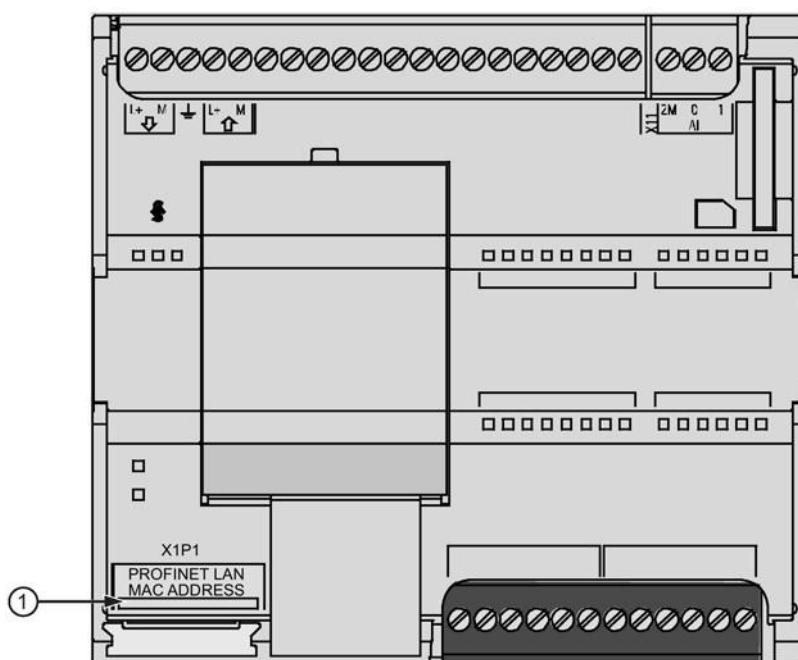
В сетях PROFINET адрес на уровне управления доступом к среде (MAC-адрес) является идентификатором, назначенным сетевому интерфейсу производителем для идентификации. MAC-адрес обычно кодирует зарегистрированный идентификационный номер производителя.

Стандартный (IEEE 802.3) формат для вывода MAC-адресов в удобной для восприятия форме представляет собой шесть групп из двух шестнадцатеричных цифр, разделенных дефисами (-) или двоеточиями (:), в порядке передачи, (например, 01-23-45-67-89-ab или 01:23:45:67:89:ab).

Примечание

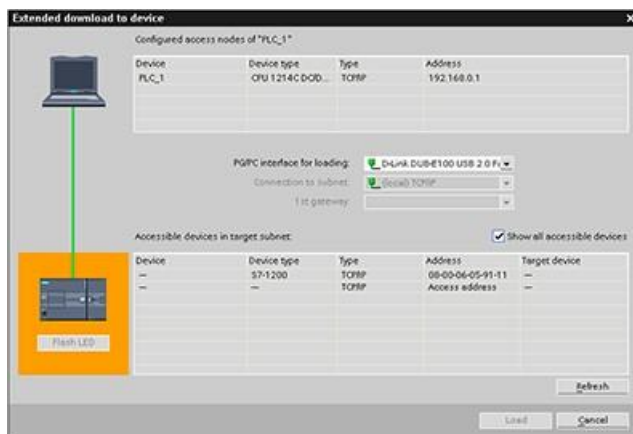
В каждый ЦПУ на заводе загружается постоянный, уникальный MAC-адрес. Вы не можете изменить MAC-адрес ЦПУ.

MAC-адрес отпечатан на переднем, нижнем левом углу ЦПУ. Обратите внимание на то, что Вы должны поднять нижнюю крышку, чтобы увидеть информацию о MAC-адресе.



① MAC-адрес

Первоначально, у ЦПУ нет IP-адреса, только установленный на заводе MAC-адрес. PROFINET коммуникации требуют, чтобы все устройства были назначены уникальным IP-адресом.



Используйте функцию ЦПУ "Download to device" и диалоговое окно "Extended download to device", чтобы отобразить все доступные сетевые устройства и гарантировать, что уникальные IP-адреса были назначены всем устройствам. Это диалоговое окно отображает все допустимые и доступные устройства с их назначенными MAC- или IP-адресами. MAC-адреса первоначально важны в идентификации устройств, у которых отсутствует требуемый уникальный IP-адрес.

11.2.6. Конфигурирование синхронизации протокола сетевого времени

ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник может физически получить доступ к Вашим сетям посредством синхронизации протокола сетевого времени (NTP), то он может получить частичный контроль над Вашим процессом, сдвигая системное время ЦПУ.

Опция NTP-клиента в ЦПУ S7-1200 по умолчанию отключена, и, когда активируется, позволяет только сконфигурированным IP-адресам действовать в качестве NTP-сервера. ЦПУ отключает эту опцию по умолчанию, и Вы должны сконфигурировать ее, чтобы позволить дистанционно управляемые корректировки системного времени ЦПУ.

ЦПУ S7-1200 поддерживает прерывания "по времени суток" и инструкции часов, которые зависят от точного системного времени ЦПУ. Если Вы конфигурируете NTP и принимаете синхронизацию времени от сервера, Вы должны гарантировать, что сервер является доверенным источником. Несоблюдение этого может вызвать нарушение защиты, которое позволяет неизвестному пользователю получить частичный контроль над Вашим процессом, сдвигая системное время ЦПУ.

Для получения информации о безопасности и рекомендаций, смотри "Рабочие инструкции для промышленной безопасности"

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте обслуживания поддержки от Сименс.

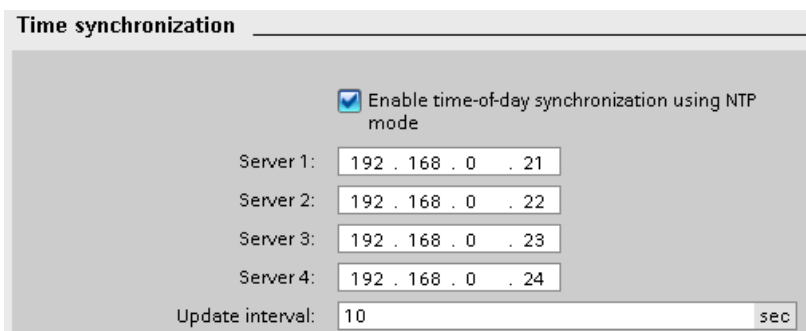
Протокол сетевого времени (NTP) широко используется, чтобы синхронизировать часы компьютерных систем с интернет-серверами времени. В режиме NTP ЦПУ отправляет запросы времени суток с регулярным интервалом (в клиентском режиме) NTP-серверу в подсети (LAN). На основе ответов от сервера вычисляется самое надежное и самое точное время, и время суток на станции синхронизируется.

Преимущество этого режима состоит в том, что он позволяет синхронизацию времени по всей подсети.

IP-адреса до четырех NTP-серверов должны быть сконфигурированы. Интервал обновления определяет интервал между запросами времени (в секундах). Значение интервала составляет от 10 секунд до суток.

В NTP-режиме, обычно передается UTC (Всемирное скоординированное время); это соответствует GMT (Среднее время по Гринвичу).

В окне свойств выберите запись конфигурации "Time synchronization". STEP 7 отображает диалоговое окно конфигурирования синхронизации времени:



Примечание

Все IP-адреса конфигурируются, когда Вы загружаете проект.

Таблица 11- 7 Параметры синхронизации времени

Параметр	Определение
Enable time-of-day synchronization using Network Time Protocol (NTP) servers	Щелкните по флажку, чтобы включить синхронизацию времени суток, используя серверы NTP.
Server 1	Назначенный IP-адрес для сетевого сервера времени 1
Server 2	Назначенный IP-адрес для сетевого сервера времени 2
Server 3	Назначенный IP-адрес для сетевого сервера времени 3
Server 4	Назначенный IP-адрес для сетевого сервера времени 4
Time synchronization interval	Значение интервала (в секундах)

11.2.7. Время запуска, присвоение имени и назначение адреса PROFINET-устройству

PROFINET IO может увеличить время запуска для Вашей системы (конфигурируемый тайм-аут). Большее количество устройств и медленные устройства влияют на время, которое необходимо, чтобы переключиться в RUN.

В V4.0 и более поздних у Вас может быть максимум 16 PROFINET IO устройств в Вашей S7-1200 PROFINET-сети.

Каждая станция (или IO-устройство) запускается независимо, и это влияет на общее время запуска ЦПУ. Если Вы настраиваете конфигурируемый тайм-аут слишком малым, то общее время запуска ЦПУ может быть недостаточным для всех станций, чтобы завершить процесс запуска. Если происходит подобная ситуация, это приводит к ошибкам ложных станций.

В свойствах ЦПУ в разделе "Startup" Вы можете найти "Parameter assignment time for distributed I/O" (тайм-аут). Сконфигурированный тайм-аут по умолчанию составляет 60,000 мс (1 минута); пользователь может настроить это время.

Назначение имени и адреса PROFINET-устройству в STEP 7

У всех PROFINET-устройств должны быть имя устройства и IP-адрес. Используйте STEP 7, чтобы определить имена устройств и сконфигурировать IP-адреса. Имена устройств загружаются в IO-устройства, используя PROFINET DCP (Протокол обнаружения и конфигурирования).

Назначение PROFINET-адреса при запуске системы

Контроллер рассылает имена устройств по сети, и устройства отвечают своими MAC-адресами. Контроллер затем назначает IP-адрес устройству, используя протокол PROFINET DCP:

- Если у MAC-адреса есть сконфигурированный IP-адрес, то станция выполняет запуск.
- Если у MAC-адреса нет сконфигурированного IP-адреса, STEP 7 назначает адрес, который сконфигурирован в проекте, и станция затем выполняет запуск.
- Если существует проблема с этим процессом, имеет место ошибка станции, и запуск не выполняется. Эта ситуация приводит к превышению сконфигурированного значения тайм-аута.

11.2.8. Открытые пользовательские коммуникации

11.2.8.1. Протоколы

Интегрированный PROFINET-порт ЦПУ поддерживает несколько коммуникационных стандартов по Ethernet-сети:

- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- User Datagram Protocol (UDP)

Таблица 11- 8 Протоколы и коммуникационные инструкции для всех стандартов

Протокол	Примеры использования	Ввод данных в область приема	Коммуникационные инструкции	Тип адресации
TCP	Коммуникации ЦПУ-ЦПУ Транспортировка кадров	Режим Ad hoc	Только TRCV_C и TRCV (V4.1 и унаследованные инструкции)	Назначает номера портов Локальному (активное) и Партнерскому (пассивное) устройствам
		Прием данных с указанной длиной	TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV(V4.1 и унаследованные инструкции)	
ISO on TCP	Коммуникации ЦПУ-ЦПУ Фрагментация и восстановление сообщения	Режим Ad hoc	Только TRCV_C и TRCV (V4.1 и унаследованные инструкции)	Назначает TSAP Локальному (активное) и Партнерскому (пассивное) устройствам
		Управляемый протоколом	TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV(V4.1 и унаследованные инструкции)	

Протокол	Примеры использования	Ввод данных в область приема	Коммуникационные инструкции	Тип адресации
UDP	Коммуникации ЦПУ-ЦПУ Коммуникации пользовательской программы	Протокол дейтаграмм пользователя	TUSEND и TURCV	Назначает номера портов Локальному (активное) и Партнерскому (пассивное) устройствам, но не является выделенным соединением
S7 коммуникации	Коммуникации ЦПУ-ЦПУ Чтение/запись данных из/в ЦПУ	Передача и прием данных с указанной длиной	GET и PUT	Назначает TSAP Локальному (активное) и Партнерскому (пассивное) устройствам
PROFINET IO	Коммуникации ЦПУ-PROFINET IO устройство	Передача и прием данных с указанной длиной	Встроенные	Встроенный

11.2.8.2. TCP и ISO on TCP

Протокол управления передачей (TCP) является стандартным протоколом, описанным RFC 793: Transmission Control Protocol. Основная цель TCP состоит в том, чтобы предоставить услугу надежного, безопасного соединения между партнерскими процессами. Этот протокол обладает следующими функциями:

- Эффективный коммуникационный протокол, так как он близко связан с аппаратными средствами
- Подходящий для данных среднего и большого объема (до 8192 байтов)
- Предоставляет значительно больше средств для приложений, в особенности, восстановление после ошибки, управления потоком и надежность
- Протокол с установлением соединения
- Может использоваться очень гибко со сторонними системами, которые поддерживают исключительно TCP
- Способен к маршрутизации
- Только статические длины данных применимы.
- Сообщения подтверждаются.
- Приложения адресуются, используя номера портов.
- Большинство пользовательских прикладных протоколов, таких как TELNET и FTP, использует TCP.
- Работа по программированию требуется для управления данными из-за интерфейса программирования SEND / RECEIVE.

Международная организация по стандартизации (ISO) на Протоколе управления передачей (TCP) (RFC 1006) (ISO on TCP) является механизмом, который позволяет ISO приложениям быть перенаправленным в TCP/IP сеть. Этот протокол обладает следующими функциями:

- Эффективный коммуникационный протокол, так как он близко связан с аппаратными средствами
- Подходящий для данных среднего и большого объема (до 8192 байтов)

- В отличие от TCP, сообщения обладают идентификацией завершения данных и ориентированы на сообщение.
- Способен к маршрутизации; может использоваться в WAN
- Возможны динамические длины данных.
- Работа по программированию требуется для управления данными из-за интерфейса программирования SEND / RECEIVE.

Используя точки доступа к транспортной службе (TSAP), протокол TCP позволяет множественные соединения с единственным IP-адресом (до 64K соединений). С RFC 1006 TSAP однозначно определяют принадлежность этих коммуникационных соединений конечной точки к IP-адресу.

11.2.8.3. Коммуникационные услуги и используемые номера портов

S7-1200 ЦПУ поддерживает протоколы, перечисленные в таблице ниже. Для каждого протокола ЦПУ назначает параметры адреса, соответствующий коммуникационный уровень, а также коммуникационную роль и направление коммуникаций.

Эта информация позволяет подобрать меры безопасности для защиты системы автоматизации к используемым протоколам (например, брандмауэр). Только Ethernet-или PROFINET-сети обладают мерами безопасности. Так как у PROFIBUS отсутствуют меры безопасности, таблица не содержит PROFIBUS-протоколы.

Таблица ниже показывает различные уровни и протоколы, которые использует ЦПУ:

Протокол	Номер порта	(2) Уровень канала (4) Транспортный уровень	Функция	Описание
PROFINET-протоколы				
DCP (протокол обнаружения и конфигурирования)	Не имеет значения	(2) Ethernet II и IEEE 802.1Q и Ethertype 0x8892 (PROFINET)	Доступные устройства PROFINET Обнаружение и конфигурирование	PROFINET использует DCP, чтобы обнаружить устройства и обеспечить основные настройки. DCP использует специальный групповой MAC-адрес: xx-xx-xx-01-0E-CF, xx-xx-xx = уникальный идентификатор организации
LLDP (протокол обнаружения канального уровня)	Не имеет значения	(2) Ethernet II и IEEE 802.1Q и Ethertype 0x88CC (PROFINET)	PROFINET протокол обнаружения канального уровня	PROFINET использует LLDP для обнаружения и управления локальными взаимодействиями между PROFINET-устройствами. LLDP использует специальный групповой MAC-адрес: 01-80-C2-00-00-0E

11.2.8.4. Ситуативный режим

Как правило, TCP и ISO-on-TCP принимают пакеты данных указанной длины, в пределах от 1 до 8192 байтов. Однако коммуникационные инструкции TRCV_C и TRCV также обеспечивают "ситуативный" коммуникационный режим, который позволяет принимать пакеты данных переменной длины от 1 до 1472 байтов.

Примечание

Если Вы храните данные в "оптимизированном" DB (только символьная адресация), Вы можете принимать данные только в массивах данных типа Byte, Char, USInt и SInt.

Чтобы сконфигурировать инструкцию TRCV_C или TRCV для ситуативного режима, установите входной параметр ADHOC инструкции.

Если Вы не вызываете инструкцию TRCV_C или TRCV в ситуативном режиме часто, то Вы могли бы получить больше чем один пакет в одном вызове. Например: Если бы Вы должны были получить пять 100-байтовых пакетов с одним вызовом, TCP поставил бы эти пять пакетов как один 500-байтовый пакет, в то время как ISO-on-TCP реструктурирует пакеты в пять 100-байтовых пакетов.

11.2.8.5. ID соединений для инструкций открытых пользовательских коммуникаций

Когда Вы вставляете PROFINET-инструкции TSEND_C, TRCV_C или TCON в Вашу пользовательскую программу, STEP 7 создает экземплярный DB, чтобы сконфигурировать коммуникационный канал (или соединение) между устройствами. Используйте "свойства" (стр. 619) инструкции, чтобы сконфигурировать параметры для соединения. Среди параметров есть ID для этого соединения.

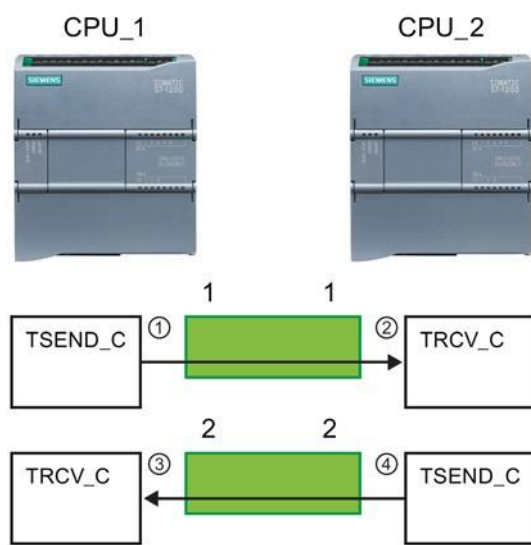
- ID соединения должен быть уникальным для ЦПУ. У каждого соединения, которое Вы создаете, должны быть отличающиеся от других DB и ID соединения.
- Как локальный, так и партнерский ЦПУ могут использовать тот же ID номер для того же соединения, но ID номера соединения не должны обязательно совпадать. ID номер соединения важен только для PROFINET-инструкций в пользовательской программе отдельного ЦПУ.
- Вы можете использовать любой номер для ID соединения ЦПУ. Однако конфигурирование ID соединений последовательно от "1" обеспечивает простой метод для отслеживания числа используемых соединений для определенного ЦПУ.

Примечание

Каждая инструкция TSEND_C, TRCV_C или TCON в Вашей пользовательской программе создает новое соединение. Важно использовать корректный ID для каждого соединения.

Следующий пример показывает коммуникации между двумя ЦПУ, которые используют два отдельных соединения для отправки и получения данных.

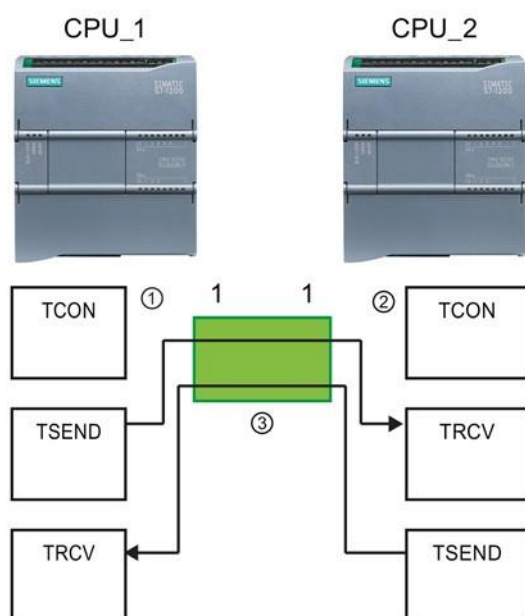
- Инструкция TSEND_C в CPU_1 соединяется с TRCV_C в CPU_2 по первому соединению ("ID соединения = 1" как в CPU_1, так и в CPU_2).
- Инструкция TRCV_C в CPU_1 соединяется с TSEND_C в CPU_2 по второму соединению ("ID соединения = 2" как в CPU_1, так и в CPU_2).



- ① TSEND_C в CPU_1 создает соединение и назначает ID этому соединению (ID соединения 1 для CPU_1).
- ② TRCV_C в CPU_2 создает соединение для CPU_2 и назначает ID соединения (ID соединения 1 для CPU_2).
- ③ TRCV_C в CPU_1 создает второе соединение для CPU_1 и назначает другой ID для этого соединения (ID соединения 2 для CPU_1).
- ④ TSEND_C в CPU_2 создает второе соединение и назначает другой ID для этого соединения (ID соединения 2 для CPU_2).

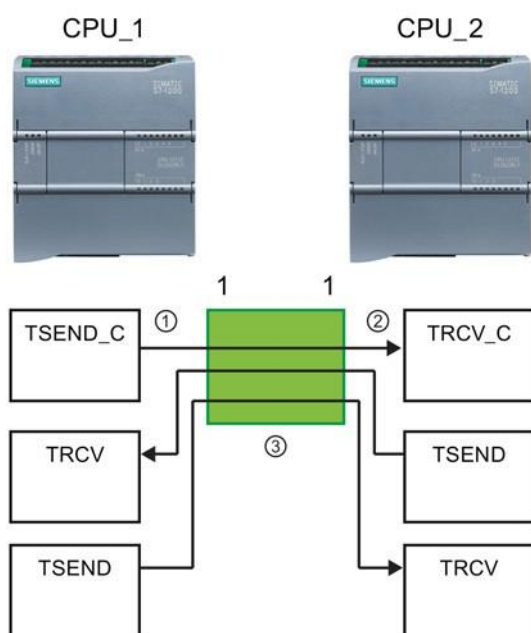
Следующий пример показывает коммуникации между двумя ЦПУ, которые используют 1 соединение как для отправки, так и для получения данных.

- Каждый ЦПУ использует инструкцию TCON, чтобы сконфигурировать соединение между этими двумя ЦПУ.
- Инструкция TSEND в CPU_1 соединяется с инструкцией TRCV в CPU_2 при помощи ID соединения ("ID соединения = 1"), которое было сконфигурировано инструкцией TCON в CPU_1. Инструкция TRCV в CPU_2 соединяется с инструкцией TSEND в CPU_1 при помощи ID соединения ("ID соединения = 1"), которое было сконфигурировано инструкцией TCON в CPU_2.
- Инструкция TSEND в CPU_2 соединяется с инструкцией TRCV в CPU_1 при помощи ID соединения ("ID соединения = 1"), которое было сконфигурировано инструкцией TCON в CPU_2. Инструкция TRCV в CPU_1 соединяется с инструкцией TSEND в CPU_2 при помощи ID соединения ("ID соединения = 1"), которое было сконфигурировано инструкцией TCON в CPU_1.



- ① TCON в CPU_1 создает соединение и назначает ID для этого соединения в CPU_1 (ID=1).
- ② TCON в CPU_2 создает соединение и назначает ID для этого соединения в CPU_2 (ID=1).
- ③ TSEND и TRCV на CPU_1 используют ID соединения, созданного TCON в CPU_1 (ID=1).
TSEND и TRCV в CPU_2 используют ID соединения, созданного TCON в CPU_2 (ID=1).

Как показано в следующем примере, Вы можете также использовать отдельную инструкцию TSEND и TRCV для коммуникаций через соединение, созданное инструкцией TSEND_C или TRCV_C. Инструкции TSEND и TRCV самостоятельно не создают новое соединение, так что должны использовать DB и ID соединения, которые создавались инструкцией TSEND_C, TRCV_C или TCON.



- ① TSEND_C в CPU_1 создает соединение и назначает ID этому соединению (ID=1).
- ② TRCV_C в CPU_2 создает соединение и назначает ID этому соединению в CPU_2 (ID=1).
- ③ TSEND и TRCV в CPU_1 используют ID соединения, созданного TSEND_C в CPU_1 (ID=1).
TSEND и TRCV в CPU_2 используют ID соединения, созданного TRCV_C в CPU_2 (ID=1).

11.2.8.6. Параметры для PROFINET-соединения

Инструкции TSEND_C, TRCV_C и TCON требуют, чтобы связанные с соединением параметры были определены, чтобы соединиться с устройством партнера. Эти параметры назначаются структурой TCON_Param для TCP, ISO-on-TCP и UDP протоколов. Как правило, Вы используете вкладку "Configuration" (стр. 619) "свойств" инструкции, чтобы определить эти параметры. Если вкладка "Configuration" недоступна, то Вы должны определить структуру TCON_Param программно.

С V4.1 структура TCON_IP_V4 назначает параметры для TCP-протокола, а структура TCON_IP_RFC - для ISO-on-TCP протокола.

TCON_Param

Таблица 11- 9 Структура с описанием соединения (TCON_Param)

Байт	Параметр и тип данных		Описание
0 ... 1	block_length	UInt	Длина: 64 байта (фиксировано)
2 ... 3	id	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Укажите значение этого параметра для инструкции TSEND_C, TRCV_C или TCON под ID.
4	connection_type	USInt	Тип соединения: 17: TCP (по умолчанию) 18: ISO-on-TCP 19: UDP
5	active_est	Bool	ID для типа соединения: - TCP и ISO-on-TCP: - FALSE: пассивное соединение - TRUE: активное соединение (по умолчанию) - UDP: FALSE
6	local_device_id	USInt	ID для локального интерфейса PROFINET или Industrial Ethernet: 1 (по умолчанию)
7	local_tsap_id_len	USInt	Длина используемого параметра local_tsap_id в байтах; возможные значения: - TCP: 0 (активный, по умолчанию) или 2 (пассивный) - ISO-on-TCP: от 2 до 16 - UDP: 2
8	rem_subnet_id_len	USInt	Этот параметр не используется.
9	rem_staddr_len	USInt	Длина адреса конечной точки партнера, в байтах: 0: не определено (параметр rem_staddr не важен) 4 (по умолчанию): Допустимый IP-адрес в параметре rem_staddr (только для TCP и ISO-on-TCP)
10	rem_tsap_id_len	USInt	Длина используемого параметра rem_tsap_id в байтах; возможные значения: - TCP: 0 (пассивный) или 2 (активный, по умолчанию) - ISO-on-TCP: от 2 до 16 - UDP: 0

Байт	Параметр и тип данных		Описание
11	next_staddr_len	USInt	Этот параметр не используется.
12 ... 27	local_tsap_id	Array [1..16] of Byte	Компонент локального адреса соединения: - TCP и ISO-on-TCP: номер локального порта (возможные значения: от 1 до 49151; рекомендуемые значения: 2000...5000): - local_tsap_id[1] = старший байт номера порта в шестнадцатеричном представлении; - local_tsap_id[2] = младший байт номера порта в шестнадцатеричном представлении; - local_tsap_id[3-16] = не важно - ISO-on-TCP: local TSAP-ID: - local_tsap_id[1] = B#16#E0; - local_tsap_id[2] = стойка и слот локальных конечных точек (биты с 0 по 4: номер слота, биты с 5 по 7: номер стойки); - local_tsap_id[3-16] = расширение TSAP, дополнительно - UDP: Этот параметр не используется. Примечание: Убедитесь, что каждое значение local_tsap_id уникально в ЦПУ.
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Этот параметр не используется.
34 ... 39	rem_staddr	Array [1..6] of USInt	Только TCP и ISO-on-TCP: IP-адрес конечной точки партнера. (Не важно для пассивных соединений). Например, IP-адрес 192.168.002.003 сохраняется в следующих элементах массива: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = не важно
40 ... 55	rem_tsap_id	Array [1..16] of Byte	Компонент партнерского адреса соединения - TCP: номер порта партнера. Диапазон: от 1 до 49151; Рекомендуемые значения: от 2000 до 5000): - rem_tsap_id[1] = старший байт номера порта в шестнадцатеричном представлении - rem_tsap_id[2] = младший байт номера порта в шестнадцатеричном представлении; - rem_tsap_id[3-16] = не важно - ISO-on-TCP: ID TSAP партнера: - rem_tsap_id[1] = B#16#E0 - rem_tsap_id[2] = стойка и слот локальных конечных точек (биты с 0 по 4: номер слота, биты с 5 по 7: номер стойки) - rem_tsap_id[3-16] = расширение TSAP, дополнительно - UDP: Этот параметр не используется.
56 ... 61	next_staddr	Array [1..6] of Byte	Этот параметр не используется.
62 ... 63	spare	Word	Зарезервировано: W#16#0000

TCON_IP_V4

Таблица 11- 10 Структура с описанием соединения (TCON_IP_V4): для использования с TCP

Байт	Параметр и тип данных		Описание
0 ... 1	Interfaceld	HW_ANY	Аппаратный идентификатор субмодуля IE-интерфейса
2 ... 3	ID	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Укажите значение этого параметра для инструкции TSEND_C, TRCV_C или TCON под ID.
4	ConnectionType	Byte	Тип соединения: • 11: TCP/IP (по умолчанию) • 17: TCP/IP (Этот тип соединения включен по причине использования унаследованных инструкций. Рекомендуется, чтобы Вы использовали "11: TCP/IP (по умолчанию)".) • 19: UDP
5	ActiveEstablished	Bool	Активное/пассивное установление соединения: • TRUE: активное соединение (по умолчанию) • FALSE: пассивное соединение
	V4 IP-адрес		
6	ADDR[1]	Byte	Октет 1
7	ADDR[1]	Byte	Октет 2
8	ADDR[1]	Byte	Октет 3
9	ADDR[1]	Byte	Октет 4
10 ... 11	RemotePort	UInt	Номер удаленного UDP/TCP порта
12 ... 13	LocalPort	UInt	Номер локального UDP/TCP порта

TCON_IP_RFC

Таблица 11- 11 Структура с описанием соединения (TCON_IP_RFC): для использования с ISO on TCP

Байт	Параметр и тип данных		Описание
0 ... 1	Interfaceld	HW_ANY	Аппаратный идентификатор субмодуля IE-интерфейса
2 ... 3	ID	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Укажите значение этого параметра для инструкции TSEND_C, TRCV_C или TCON под ID.
4	ConnectionType	Byte	Тип соединения: • 12: ISO-on-TCP (по умолчанию) • 17: ISO-on-TCP (Этот тип соединения включен по причине использования унаследованных инструкций. Рекомендуется, чтобы Вы использовали "12: ISO-on-TCP (по умолчанию)".)
5	ActiveEstablished	Bool	Активное/пассивное установление соединения: • TRUE: активное соединение (по умолчанию) • FALSE: пассивное соединение

Байт	Параметр и тип данных		Описание
6 ... 7	Spare		Не используется
	V4 IP-адрес		
8	ADDR[1]	Byte	Октет 1
9	ADDR[1]	Byte	Октет 2
10	ADDR[1]	Byte	Октет 3
11	ADDR[1]	Byte	Октет 4
	Удаленный транспортный селектор		
12 ... 13	TSelLength	UInt	Длина TSelector
14 ... 45	TSel	array [1..32] of Byte	Массив символов для имени TSAP
	Локальный транспортный селектор		
46 ... 47	TSelLength	UInt	Длина TSelector
48 ... 79	TSel	array [1..32] of Byte	Массив символов для имени TSAP

11.2.8.7. Инструкции TSEND_C и TRCV_C

Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможности инструкций TSEND_C и TRCV_C для использовать параметров соединения со структурами "TCON_IP_v4" и "TCON_IP_RFC".

По этой причине S7-1200 поддерживает два набора инструкций TSEND_C и TRCV_C:

- Унаследованные инструкции TSEND_C и TRCV_C (стр. 655): Эти инструкции TSEND_C и TRCV_C существовали до версии V4.0 S7-1200 и работали только с параметрами соединения со структурами "TCON_Param".
- Инструкции TSEND_C и TRCV_C (стр. 645): Эти инструкции TSEND_C и TRCV_C обеспечивают всю функциональность унаследованных инструкций плюс возможность использовать параметры соединения со структурами "TCON_IP_v4" и "TCON_IP_RFC".

Выбор версии инструкций TSEND_C и TRCV_C

Существует две версии инструкций TSEND_C и TRCV_C, доступных в STEP 7:

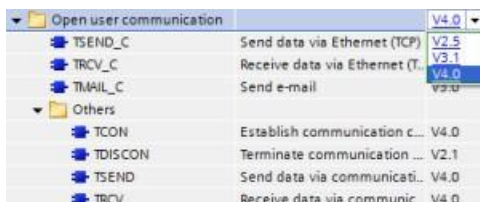
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних.
- Версия 4.0 доступна в STEP 7 Basic/Professional V13 SP1.

Для совместимости и простоты миграции, Вы можете выбрать какую из версий инструкции вставить в Вашу пользовательскую программу.

Не используйте различные версии инструкций в одной программе ЦПУ.



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.



Чтобы изменить версию TSEND_C и TRCV_C инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить TSEND_C или TRCV_C инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной TSEND_C или TRCV_C инструкции создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию TSEND_C или TRCV_C инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите TSEND_C или TRCV_C экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии TSEND_C или TRCV_C инструкции.

Инструкции TSEND_C и TRCV_C (послать и получить данные, используя Ethernet)

Инструкция TSEND_C комбинирует функции инструкций TCON, TDISCON и TSEND. Инструкция TRCV_C комбинирует функции инструкций TCON, TDISCON и TRCV. (Обратитесь к теме "TCON, TDISCON, TSEND и TRCV (стр. 664)" для получения дополнительной информации об этих инструкциях.)

Минимальный размер данных, которые Вы можете передать (TSEND_C) или получить (TRCV_C), составляет один байт; максимальный размер составляет 8192 байта. TSEND_C не поддерживает передачу данных из булевых ячеек, и TRCV_C не принимает данные в булевые ячейки. Для получения информации о передаче данных с помощью этих инструкций смотрите раздел по согласованности данных (стр. 185).

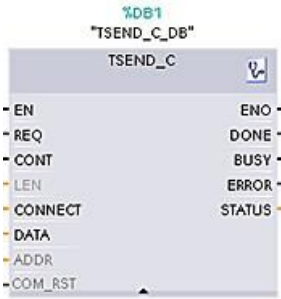
Примечание

Инициализация коммуникационных параметров

После того, как Вы вставите инструкцию TSEND_C или TRCV_C, используйте "свойства" инструкции (стр. 619), чтобы сконфигурировать коммуникационные параметры (стр. 641). Поскольку Вы вводите параметры для коммуникационных партнеров в окне инспектора, STEP 7 вводит соответствующие данные в DB для инструкции.

Если Вы хотите использовать мультиэкземплярный DB, Вы должны вручную сконфигурировать DB в обоих ЦПУ.

Таблица 11- 12 Инструкции TSEND_C и TRCV_C

LAD / FBD	SCL	Описание
	<pre> TSEND_C_DB"(req:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, con- nect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_); </pre>	TSEND_C устанавливает TCP или ISO on TCP коммуникационное соединение со станцией партнера, отправляет данные и может завершить соединение. После того, как соединение настроено и установлено, оно автоматически обслуживается и контролируется ЦПУ.
	<pre> TRCV_C_DB"(en_r:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, adhoc:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, rcvd_len=>_uint_out_, connect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_); </pre>	TRCV_C устанавливает TCP или ISO on TCP коммуникационное соединение со станцией партнера, принимает данные и может завершить соединение. После того, как соединение настроено и установлено, оно автоматически обслуживается и контролируется ЦПУ.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 13 Типы данных для параметров TSEND_C и TRCV_C

Параметр и тип		Тип данных	Описание
REQ (TSEND_C)	IN	Bool	Запускает задание передачи данных по нарастающему фронту
EN_R (TRCV_C)	IN	Bool	Разрешить прием
CONT	IN	Bool	Управляет коммуникационным соединением: 0: Установить соединение автоматически. Прервать коммуникационное соединение после отправки данных. 1: Прервать коммуникационное соединение после приема данных. Параметр CONT оценивается только при положительном фронте на REQ или, когда COM_RST = "1".
LEN	IN	UDInt	Дополнительный (скрытый) параметр Максимальное количество байтов, которые будут отправлены (TSEND_C) или получены (TRCV_C) согласно заданию. Если Вы используете чисто символьные значения в параметре DATA, у параметра LEN должно быть значение "0".

Параметр и тип		Тип данных	Описание
ADHOC (TRCV_C)	IN	Bool	Дополнительный (скрытый) параметр Запрос ситуативного режима для соединения типа TCP.
CONNECT	IN_OUT	TCON_Param	Указатель на описание соединения, соответствующее структуре соединения, которое будет описано. - Для TCP или UDP, используйте структуру TCON_IP_v4 - За описанием обратитесь к теме: "Параметры соединения со структурой TCON_IP_v4". - Для ISO-on-TCP используйте структуру TCON_IP_RFC - За описанием обратитесь к теме: "Параметры соединения со структурой TCON_IP_RFC". Параметр CONNECT оценивается только при положительном фронте на REQ (TSEND_C), когда запускается установление соединения (TRCV_C), или когда COM_RST = 1.
DATA	IN_OUT	Variant	Указатель на область отправки, содержащую: - Адрес и длину данных, подлежащих отправке (TSEND_C) - Адрес и максимальную длину принятых данных (TRCV_C)
ADDR	IN_OUT	Variant	Дополнительный (скрытый) параметр Указатель на адрес получателя с соединением UDP типа. Адресная информация отображена в структуре TADDR_Param ###.
COM_RST	IN_OUT	Bool	Дополнительный (скрытый) параметр Перезапуск инструкции: 0: не важно 1: Полностью перезапускает инструкцию; существующее соединение либо завершается, либо сбрасывается и устанавливается повторно в соответствии с CONT. Параметр COM_RST сбрасывается после оценки инструкцией TSEND_C или TRCV_C и не должен, таким образом, переключаться статически.
DONE	OUT	Bool	Параметр состояния со следующими значениями: 0: Задание по отправке еще не запускалось или все еще выполняется. 1: Задание по отправке выполнено без ошибок. Это состояние отображается только на один цикл.
BUSY	OUT	Bool	Параметр состояния со следующими значениями: 0: Задание по отправке еще не запускалось или уже завершено. 1: Задание по отправке еще не завершено. Новое задание не может быть запущено.
ERROR	OUT	Bool	Параметр состояния со следующими значениями: 0: Без ошибок 1: Ошибка произошла во время установления соединения, передачи данных или завершения соединения.

Параметр и тип		Тип данных	Описание
STATUS	OUT	Word	Состояние инструкции (см. описание параметров ERROR и STATUS).
RCVD_LEN (TRCV_C)	OUT	Int	Объем фактически принятых данных (в байтах).

Примечание

Инструкции TSEND_C требуется переход от низкого к высокому уровню на входе REQ, чтобы запустить задание по отправке. Параметр BUSY в этом случае устанавливается в 1 на время обработки. Завершение задания по отправке обозначается установкой в 1 на 1 скан либо параметра DONE, либо параметра ERROR. В это время любой переход от низкого к высокому уровню на входе REQ игнорируется.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) использует параметр DATA, чтобы определить длину передаваемых данных. Гарантируйте, что DATA, переданные инструкцией TSEND_C имеют тот же размер, что и параметр DATA инструкции TRCV_C.

Операции TSEND_C

Инструкция TSEND_C выполняется асинхронно и реализует последовательно следующие функции:

1. Настройка и установление коммуникационного соединения:

TSEND_C настраивает коммуникационное соединение и устанавливает это соединение, если нарастающий фронт обнаружен в параметре REQ, и никакое коммуникационное соединение еще не установлено. Как только соединение было настроено и установлено, оно автоматически поддерживается и контролируется ЦПУ. Описание соединения, определенное в параметре CONNECT, используется, чтобы настроить коммуникационное соединение. Следующие типы соединения могут использоваться:

- TCON_Param структура для TCP, ISO-on-TCP и UDP протоколов
- C V4.1, TCP/UDP: Описание соединения, использующее структуру TCON_IP_v4 в параметре CONNECT
- C V4.1, ISO-on-TCP: Описание соединения, использующее структуру TCON_IP_RFC в параметре CONNECT

Существующее соединение завершается и соединение, которое было настроено, удаляется, когда ЦПУ переходит в режим STOP. Чтобы настроить и установить соединение снова, Вы должны повторно выполнить TSEND_C. Для получения информации о количестве возможных коммуникационных соединений, см. технические спецификации на Ваш ЦПУ.

2. Отправка данных через существующее коммуникационное соединение:

Задание по отправке выполняется, когда обнаруживается нарастающий фронт в параметре REQ. Как описано выше, коммуникационное соединение сначала устанавливается. Вы определяете область отправки в параметре DATA. Он включает адрес и длину данных, которые будут отправлены. Не используйте область данных с типом данных BOOL или Array of BOOL в параметре DATA. В параметре LEN Вы определяете максимальное количество байтов, отправленных с заданием. При использовании символического имени в параметре DATA у параметра LEN должно быть значение "0".

Данные, которые будут отправлены, не должны редактироваться, пока задание по отправке не завершено.

3. Завершение коммуникационного соединения:

Коммуникационное соединение завершается после того, как данные были отправлены, если параметр CONT имел значение "0" во время нарастающего фронта в параметре REQ.

Иначе, коммуникационное соединение будет поддерживаться.

Если задание по отправке выполняется успешно, то параметр DONE установлен в "1". Коммуникационное соединение может быть завершено перед этим (см. вышеупомянутое описание зависимости от параметра CONT). Состояние "1" в параметре DONE не является подтверждением, что отправленные данные уже были считаны коммуникационным партнером.

TSEND_C сбрасывается, когда параметр COM_RST установлен в "1". Потеря данных может произойти, если данные передаются в этой точке.

Следующие сценарии возможны в зависимости от параметра CONT:

- CONT = "0":
Существующее коммуникационное соединение устанавливается.
- CONT = "1" и было установлено коммуникационное соединение:
Существующее коммуникационное соединение сбрасывается и устанавливается повторно.
- CONT = "1" и не было установлено коммуникационного соединения.
Никакое коммуникационное соединение не устанавливается.

Параметр COM_RST сбрасывается после оценки инструкцией T_SEND. Чтобы активировать TSEND_C снова после выполнения (DONE = 1), вызовите инструкцию один раз с REQ = 0.

Операции TRCV_C

Инструкция TRCV_C выполняется асинхронно и реализует последовательно следующие функции:

1. Настройка и установление коммуникационного соединения:

TRCV_C настраивает коммуникационное соединение и устанавливает это соединение, если параметр EN_R = "1", и никакое коммуникационное соединение еще не установлено. Как только соединение было настроено и установлено, оно автоматически поддерживается и контролируется ЦПУ.

Описание соединения, определенное в параметре CONNECT, используется, чтобы настроить коммуникационное соединение. Следующие типы соединения могут использоваться:

- TCON_Param структура для TCP, ISO-on-TCP и UDP протоколов
- C V4.1, TCP/UDP: Описание соединения, использующее структуру TCON_IP_v4 в параметре CONNECT
- C V4.1, ISO-on-TCP: Описание соединения, использующее структуру TCON_IP_RFC в параметре CONNECT

Существующее соединение завершается и соединение, которое было настроено, удаляется, когда ЦПУ переходит в режим STOP. Чтобы настроить и установить соединение снова, Вы должны повторно выполнить TRCV_C с EN_R = "1".

Если EN_R будет переведен в "0", прежде чем коммуникационное соединение было установлено, то соединение будет установлено и сохранится даже если CONT = "0". Однако никакие данные не будут приняты (DONE останется в состоянии "0").

Для получения информации о количестве возможных коммуникационных соединений, см. технические спецификации на Ваш ЦПУ.

2. Получение данных через существующее коммуникационное соединение:

Получение данных разрешено, когда параметр EN_R установлен в значение "1". Как описано выше, вначале устанавливается коммуникационное соединение. Принятые данные вводятся в область приема. Вы определяете длину области приема либо с помощью параметра LEN (если LEN <> 0) или с помощью информации о длине в параметре DATA (если LEN = 0), в зависимости от используемого варианта протокола. Если Вы используете чисто символьные значения в параметре DATA, то параметр LEN должен иметь значение "0".

Если EN_R будет переведен в "0", прежде чем данные будут получены в первый раз, то коммуникационное соединение сохранится даже если CONT = "0". Однако никакие данные не будут приняты (DONE останется в состоянии "0").

3. Завершение коммуникационного соединения:

Коммуникационное соединение завершается после того, как данные были приняты, если параметр CONT имел значение "0", когда было начало установление соединения. Иначе, коммуникационное соединение будет поддерживаться.

Если задание на получение выполняется успешно, то параметр DONE устанавливается в "1". Коммуникационное соединение может быть завершено перед этим (см. вышеупомянутое описание зависимости от параметра CONT).

TRCV_C сбрасывается, когда параметр COM_RST установлен. Если принимаются данные, когда она выполняются повторно, то это может привести к потере данных. Следующие сценарии возможны в зависимости от состояния параметра CONT:

- CONT = "0":
Существующее коммуникационное соединение устанавливается.
- CONT = "1" и было установлено коммуникационное соединение:
Существующее коммуникационное соединение сбрасывается и устанавливается повторно.
- CONT = "1" и не было установлено коммуникационного соединения.
Никакое коммуникационное соединение не устанавливается.

Параметр COM_RST сбрасывается после оценки инструкцией TRCV_".

TRCV_C работает в тех же режимах, что и инструкция TRCV. Следующая таблица показывает, как данные вводятся в область приема:

Вариант протокола	Доступность данных в области приема	Параметр Connection_type в описании соединения	Параметр LEN	Параметр RCVD_LEN
TCP (ситуативный режим)	Данные доступны сразу.	B#16#11	Выбирается входом ADHOC инструкции TRCV_C	от 1 до 1472
TCP (получение данных с указанной длиной)	Данные доступны, как только получены данные с длиной, определенной в параметре LEN.	B#16#11	от 1 до 8192	Идентично значению параметра LEN
ISO on TCP (управляемая протоколом передача данных)	Данные доступны, как только получены данные с длиной, определенной в параметре LEN.	B#16#12	от 1 до 8192	Идентично значению параметра LEN

Примечание

Ситуативный (Ad hoc) режим

"Ситуативный режим" доступен только с вариантом протокола TCP. Чтобы сконфигурировать инструкцию TRCV_C для ситуативного режима, установите входной параметр ADHOC для инструкции. Длина области приема определяется указателем в параметре DATA. Длина фактически принятых данных, выводится в параметре RCVD_LEN. Может быть получено максимум 1460 байтов.

Примечание

Импорт S7-300/400 STEP 7 проектов, содержащих "ситуативный режим" в S7-1200

В S7-300/400 STEP 7 проектах "ситуативный режим" выбирается назначением "0" параметру LEN. В S7-1200 Вы конфигурируете инструкцию TRCV_C для ситуативного режима, устанавливая входной ADHOC параметр инструкции.

Если Вы импортируете S7-300/400 STEP 7 проект, содержащий "ситуативный режим" в S7-1200, Вы должны изменить параметр LEN на "65535".

Примечание

TCP (получение данных с указанной длиной)

Вы используете значение параметра LEN, чтобы определить длину для получения данных. Данные, определенные в параметре DATA, доступны в области приема, как только была полностью получена информация с длиной, определенной в параметре LEN.

Примечание

ISO on TCP (управляемая протоколом передача данных)

С вариантом протокола ISO on TCP данные передаются с управлением протоколом. Область приема определяется параметрами LEN и DATA.

Параметры BUSY, DONE и ERROR

Примечание

Вследствие асинхронной обработки TSEND_C, Вы должны сохранять данные в области передачи согласованными пока параметры DONE или ERROR не примет значение TRUE.

Для TSEND_C состояние TRUE в параметре DONE означает, что данные были отправлены успешно. Это не означает, что ЦПУ партнера по соединению фактически прочитал приемный буфер.

Из-за асинхронной обработки TRCV_C, данные в области приема являются согласованными, только когда параметр DONE = 1.

Таблица 11- 14 Параметры BUSY, DONE и ERROR инструкций TSEND_C и TRCV_C

BUSY	DONE	ERROR	Описание
1	0	0	Задание на отправку обрабатывается.
0	1	0	Задание на отправку было успешно завершено.
0	0	1	Установление соединения или задание на отправку были завершены с ошибкой. Причина ошибки определена в параметре STATUS.
0	0	0	Не было назначено никаких новых заданий на отправку.

Вы можете проверить состояние выполнения с помощью параметров BUSY, DONE, ERROR и STATUS. Параметр BUSY указывает на состояние обработки. С параметром DONE Вы можете проверить, выполнено ли задание на отправку успешно. Параметр ERROR установлен, когда во время выполнения TSEND_C или TRCV_C произошли ошибки. Информация об ошибке выводится в параметре STATUS.

Параметры Error и Status

Таблица 11- 15 Коды состояния ERROR и STATUS для инструкций TSEND_C и TRCV_C

ERROR	STATUS (W#16#...)	Описание
0	0000	Задание на отправку (TSEND_C) или получение (TRCV_C) выполнено без ошибок.
0	7000	Нет выполнения активного задания на отправку; нет установленных коммуникационных соединений.
0	7001	- Запустить выполнение задания на отправку (TSEND_C) или получение (TRCV_C). - Установить соединение. - Ожидать коммуникационного партнера.
0	7002	Данные отправляются (TSEND_C) или принимаются (TRCV_C).
0	7003	Коммуникационное соединение завершается.
0	7004	Коммуникационное соединение, установлено и контролируется; нет активного выполнения задания на отправку (TSEND_C) или получение (TRCV_C).
0	7005	Коммуникационное соединение сбрасывается.
1	80A0	Ошибка группы для кодов ошибки W#16#80A1 и W#16#80A2.
1	80A1	- Соединение или порт, уже используются пользователем. - Коммуникационная ошибка: - Указанное соединение еще не было установлено. - Указанное соединение завершается. Передача через это соединение невозможна. - Интерфейс повторно инициализируется.
1	80A2	Локальный или удаленный порт используется системой.
1	80A3	- Предпринимается попытка восстановить существующее соединение. - Предпринимается попытка завершить несуществующее соединение.
1	80A4	IP-адрес удаленной конечной точки соединения - неправильный, то есть соответствует IP-адресу локального партнера.
1	80A7	Ошибка связи: Вы вызвали инструкцию с COM_RST = 1, прежде чем задание на отправку было завершено.
1	80B2	Параметр CONNECT указывает на блок данных, который был сгенерирован с атрибутом "Only store in load memory".
1	80B3	Несогласованное назначение параметра: ошибка группы для кодов ошибки от W#16#80A0 до W#16#80A2, W#16#80A4, от W#16#80B4 до W#16#80B9.
1	80B4	Вы нарушили одно или оба из следующих условий для пассивного установления соединения (active_est = FALSE) при использовании ISO on TCP варианта протокола (connection_type = B#16#12): - local_tsap_id_len >= B#16#02 - local_tsap_id[1] = B#16#E0
1	80B5	Только пассивное установление соединения разрешено для типа соединения 13 = UDP.
1	80B6	Ошибка назначения параметра в параметре connection_type блока данных для описания соединения.
1	80B7	Ошибка в одном из следующих параметров блока данных для описания соединения: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len.

ERROR	STATUS (W#16#...)	Описание
1	8085	- Параметр LEN больше, чем наибольшее разрешенное значение. - Значение в LEN или DATA параметре было изменено после первого вызова.
1	8086	Параметр ID внутри параметра CONNECT вне разрешенного диапазона.
1	8087	Максимальное количество соединений достигнуто; никакое дополнительное соединение невозможно.
1	8088	Значение в параметре LEN не соответствует заданию области приема в параметре DATA.
1	8089	Параметр CONNECT не указывает на блок данных.
1	8091	Максимальная глубина вложения превышена.
1	809A	Параметр CONNECT указывает на поле, которое не соответствует длине описания соединения.
1	809B	ID локального устройства в описании соединения не соответствует ЦПУ.
1	80C3	- Все ресурсы для соединений (стр. 637) используются. - Блок с этим ID уже обрабатывается в другой группе приоритета.
1	80C4	Временная коммуникационная ошибка: - В этот момент времени не может быть установлено соединение. - Интерфейс получает новые параметры, или соединение устанавливается. - Сконфигурированное соединение удаляется инструкцией "TDISCON (стр. 664)". - Используемое соединение завершается вызовом с COM_RST = 1.
1	8722	Ошибка в параметре CONNECT: Недопустимая область источника (область, не объявленная в блоке данных).
1	873A	Ошибка в параметре CONNECT: Доступ к описанию соединения невозможен (нет доступа к блоку данных).
1	877F	Ошибка в параметре CONNECT: Внутренняя ошибка
1	8822	TSEND_C: параметр DATA: Недопустимая область источника, область не существует в DB.
1	8824	TSEND_C: параметр DATA: ошибка области в указателе VARIANT.
1	8832	TSEND_C: параметр DATA: номер DB слишком большой.
1	883A	TSEND_C: параметр ПОДКЛЮЧЕНИЯ: Доступ к указанным данным соединения, не возможным (для экс-вполне достаточный, потому что DB не существует).
1	887F	TSEND_C: параметр DATA: Внутренняя ошибка (например, недопустимая ссылка VARIANT)
1	893A	TSEND_C: параметр DATA: Доступ к области отправки невозможен (например, так как DB не существует).
1	8922	TRCV_C: параметр DATA: Недопустимая область цели; область не существует в DB.
1	8924	TRCV_C: параметр DATA: ошибка области в указателе VARIANT.
1	8932	TRCV_C: параметр DATA: номер DB слишком большой.
1	893A	TRCV_C: параметр CONNECT: Доступ к указанным данным соединения невозможен (например, так как DB не существует).
1	897F	TRCV_C: параметр DATA: Внутренняя ошибка (например, недопустимая ссылка VARIANT).
1	8A3A	TRCV_C: параметр DATA: Нет доступа к области данных (например, так как блок данных не существует).

Примечание

Сообщения об ошибках инструкций TCON, TSEND, TRCV и TDISCON

Внутренне, инструкция TSEND_C использует инструкции TCON, TSEND и TDISCON; а инструкция TRCV_C использует инструкции TCON, TRCV и TDISCON. Обратитесь к теме "TCON, TDISCON, TSEND и TRCV (стр. 664)" для получения дополнительной информации о сообщениях об ошибках для этих инструкций.

Протоколы Ethernet соединения

Каждый ЦПУ обладает интегрированным портом PROFINET, который поддерживает стандартные PROFINET-коммуникации. Инструкции TSEND_C и TRCV_C, а также TSEND и TRCV все поддерживают TCP и ISO on TCP Ethernet-протоколы.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

11.2.8.8. Унаследованные инструкции TSEND_C и TRCV_C

До выпуска STEP 7 V13 SP1 и S7-1200 ЦПУ V4.1 инструкции TSEND_C и TRCV_C могли работать только с параметрами соединения со структурами "TCON_Param". Общие понятия применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным инструкциям TSEND_C и TRCV_C для получения информации о программировании.

Выбор версии инструкций TSEND_C и TRCV_C

Существует две версии инструкций TSEND_C и TRCV_C, доступных в STEP 7:

- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних.
- Версия 4.0 доступна в STEP 7 Basic/Professional V137 V13 SP1.

Для совместимости и простоты миграции, Вы можете выбрать какую из версий инструкции вставить в Вашу пользовательскую программу.

Не используйте различные версии инструкций в одной программе ЦПУ.



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.

Open user communication		V3.1
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V2.1
TCON	Establish communication c...	V3.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V3.0
TRCV	Receive data via communic...	V3.0

Чтобы изменить версию TSEND_C и TRCV_C инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить TSEND_C или TRCV_C инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной TSEND_C или TRCV_C инструкции создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_х > Program blocks > System blocks > Program resources.

Чтобы проверить версию TSEND_C или TRCV_C инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите TSEND_C или TRCV_C экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии TSEND_C или TRCV_C инструкции.

Унаследованные инструкции TSEND_C и TRCV_C (послать и получить данные, используя Ethernet)

Унаследованная инструкция TSEND_C комбинирует функции унаследованных инструкций TCON, TDISCON и TSEND. Инструкция TRCV_C комбинирует функции инструкций TCON, TDISCON и TRCV. (Обратитесь к теме "Унаследованные TCON, TDISCON, TSEND и TRCV (стр. 673)" для получения дополнительной информации об этих инструкциях.)

Минимальный размер данных, которые Вы можете передать (TSEND_C) или получить (TRCV_C), составляет один байт; максимальный размер составляет 8192 байта. TSEND_C не поддерживает передачу данных из булевых ячеек, и TRCV_C не принимает данные в булевые ячейки. Для получения информации о передаче данных с помощью этих инструкций смотрите раздел по согласованности данных (стр. 185).

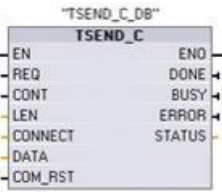
Примечание

Инициализация коммуникационных параметров

После того, как Вы вставите инструкцию TSEND_C или TRCV_C, используйте "свойства" инструкции (стр. 619), чтобы сконфигурировать коммуникационные параметры (стр. 641). Поскольку Вы вводите параметры для коммуникационных партнеров в окне инспектора, STEP 7 вводит соответствующие данные в DB для инструкции.

Если Вы хотите использовать мультиэкземплярный DB, Вы должны вручную сконфигурировать DB в обоих ЦПУ.

Таблица 11- 16 Инструкции TSEND_C и TRCV_C

LAD / FBD	SCL	Описание
	<pre>"TSEND_C_DB" (req:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, con- nect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_);</pre>	TSEND_C устанавливает TCP или ISO on TCP коммуникационное соединение со станцией партнера, отправляет данные и может завершить соединение. После того, как соединение настроено и установлено, оно автоматически обслуживается и контролируется ЦПУ.
	<pre>"TRCV_C_DB" (en_r:=_bool_in_, cont:=_bool_in_, len:=_uint_in_, adhoc:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, rcvd_len=>_uint_out_, connect:=_struct_inout_, data:=_variant_inout_, com_rst:=_bool_inout_);</pre>	TRCV_C устанавливает TCP или ISO on TCP коммуникационное соединение со станцией партнера, принимает данные и может завершить соединение. После того, как соединение настроено и установлено, оно автоматически обслуживается и контролируется ЦПУ.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 17 Типы данных для параметров TSEND_C и TRCV_C

Параметр и тип		Тип данных	Описание
REQ (TSEND_C)	IN	Bool	REQ = 1 запускает задание на отправку TSEND_C по нарастающему фронту с соединением, описанным в параметре CONNECT. (CONT = 1 также требуется, чтобы устанавливать и поддерживать коммуникационное соединение.
EN_R (TRCV_C)	IN	Bool	Когда EN_R = 1, TRCV_C готов к получению. Задание на получение обрабатывается. (CONT = 1 также требуется, чтобы установить и поддерживать коммуникационное соединение.)
CONT	IN	Bool	Управляет коммуникационным соединением: 0: Разорвать коммуникационное соединение. 1: Установить и поддерживать коммуникационное соединение. При отправке данных (TSEND_C) (нарастающий фронт в параметре REQ) параметр CONT должен иметь значение TRUE, чтобы установить или поддерживать соединение. При получении данных (TRCV_C) (нарастающий фронт в параметре EN_R) параметр CONT должен иметь значение TRUE, чтобы установить или поддерживать соединение.
Параметр и тип		Тип данных	Описание

LEN	IN	UInt	Максимальное количество байтов, которые необходимо отправить (TSEND_C) или получить (TRCV_C): - По умолчанию = 0: Параметр DATA определяет длину данных, которые необходимо отправить (TSEND_C) или получить (TRCV_C). - Ситуативный режим = 65535: Переменная длина данных устанавливается для приема (TRCV_C).
CONNECT	IN_OUT	TCON_Param	Указатель на описание соединения (стр. 641)
DATA	IN_OUT	Variant	- Содержит адрес и длину данных, которые необходимо отправить (TSEND_C) - Содержит начальный адрес и максимальную длину принятых данных (TRCV_C).
COM_RST	IN_OUT	Bool	Позволяет перезапускать инструкцию: 0: не важно 1: Полный перезапуск функционального блока, существующее соединение будет завершено.
DONE	OUT	Bool	0: Задание еще не запущено или все еще выполняется. 1: Задание завершено без ошибок.
BUSY	OUT	Bool	0: Задание завершено. 1: Задание еще не завершено. Новое задание не может быть запущено.
ERROR	OUT	Bool	Параметры состояния со следующими значениями: 0: Ошибок нет 1: Во время обработки произошла ошибка. STATUS содержит детальную информацию о типе ошибки.
STATUS	OUT	Word	Информация о состоянии, включая информацию об ошибке. (Обратитесь к таблице "Параметры Error и Status" ниже.)
RCVD_LEN (TRCV_C)	OUT	Int	Объем фактически принятых данных, в байтах

Примечание

Инструкции TSEND_C требуется переход от низкого к высокому уровню на входе REQ, чтобы запустить задание по отправке. Параметр BUSY в этом случае устанавливается в 1 на время обработки. Завершение задания по отправке обозначается установкой в 1 на 1 скан либо параметра DONE или ERROR. В это время любой переход от низкого к высокому уровню на входе REQ игнорируется.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) использует параметр DATA, чтобы определить длину передаваемых данных. Гарантируйте, что DATA, переданные инструкцией TSEND_C имеют тот же размер, что и параметр DATA инструкции TRCV_C.

Операции TSEND_C

Следующие функции описывают работу инструкции TSEND_C:

- Чтобы установить соединение, выполните TSEND_C с CONT = 1.
- После успешного установления соединения TSEND_C устанавливает параметр DONE на один цикл.
- Чтобы завершить коммуникационное соединение, выполните TSEND_C с CONT = 0. Соединение будет прервано незамедлительно. Это также влияет на приемную станцию. Соединение в ней будет закрыто, и данные в приемном буфере могут быть потеряны.
- Чтобы отправить данные по установленному соединению, выполните TSEND_C с нарастающим фронтом на REQ. После успешной операции по отправке, TSEND_C устанавливает параметр DONE на один цикл.
- Чтобы установить соединение и отправить данные, выполните TSEND_C с CONT =1 и REQ = 1. После успешной операции по отправке, TSEND_C устанавливает параметр DONE на один цикл.

Операции TRCV_C

Следующие функции описывают работу инструкции TRCV_C:

- Чтобы установить соединение, выполните TRCV_C с параметром CONT = 1.
- Чтобы принять данные, выполните TRCV_C с параметром EN_R = 1. TRCV_C принимает данные непрерывно, когда параметры EN_R = 1 и CONT = 1.
- Чтобы завершить соединение, выполните TRCV_C с параметром CONT = 0. Соединение будет прервано незамедлительно, и данные могут быть потеряны.

TRCV_C работает в тех же режимах, что и инструкция TRCV. Следующая таблица показывает, как данные вводятся в область приема:

Таблица 11- 18 Ввод данных в область приема

Вариант протокола	Ввод данных в область приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ситуативный режим	B#16#11	65535	от 1 до 1472
TCP	Прием данных с указанной длиной	B#16#11	0 (рекомендуется) или от1 до 8192, исключая 65535	от 1 до 8192
ISO on TCP	Ситуативный режим	B#16#12	65535	от 1 до 1472
ISO on TCP	С управлением по протоколу	B#16#12	0 (рекомендуется) или от1 до 8192, исключая 65535	от 1 до 8192

у

Примечание

Ситуативный (Ad hoc) режим

"Ситуативный режим" существует с TCP и ISO on TCP вариантами протокола. Вы устанавливаете "ситуативный режим", назначая "65535" для параметра LEN. Область приема идентична области, сформированной посредством DATA. Длина полученных данных будет выведена в параметре RCVD_LEN.

Если Вы храните данные в "оптимизированном" DB (только символьный доступ), Вы можете получить данные только в массивах с типом Byte, Char, USInt и SInt.

Примечание

Импорт S7-300/400 STEP 7 проектов, содержащих "ситуативный режим" в S7-1200

В S7-300/400 STEP 7 проектах "ситуативный режим" выбирается назначением "0" параметру LEN. В S7-1200 Вы конфигурируете инструкцию TRCV_C для ситуативного режима, назначая "65535" для параметра LEN.

Если Вы импортируете S7-300/400 STEP 7 проект, содержащий "ситуативный режим" в S7-1200, Вы должны изменить параметр LEN на "65535".

Примечание

Необходимо сохранять данные в области отправки согласованными, пока параметр DONE или параметр ERROR не примет значение TRUE

Вследствие асинхронной обработки TSEND_C, Вы должны сохранять данные в области отправки согласованными, пока параметр DONE или параметр ERROR не примет значение TRUE.

Для TSEND_C состояние TRUE в параметре DONE означает, что данные были отправлены успешно. Это не означает, что ЦПУ партнера по соединению фактически прочитал приемный буфер.

Вследствие асинхронной обработки TRCV_C данные в области приема являются согласованными только тогда, когда параметр DONE = 1.

Таблица 11- 19 Параметры BUSY, DONE и ERROR для инструкций TSEND_C и TRCV_C

BUSY	DONE	ERROR	Описание
TRUE	не важно	не важно	Задание обрабатывается.
FALSE	TRUE	FALSE	Задание успешно завершено.
FALSE	FALSE	TRUE	Задание было завершено с ошибкой. Причина ошибки можно узнать из параметра STATUS.
FALSE	FALSE	FALSE	Новое задание не было назначено.

Коды состояния Error и Status инструкций TSEND_C и TRCV_C

ERROR	STATUS	Описание
0	0000	Задание выполняется без ошибок
0	7000	Нет выполнения активного задания
0	7001	Запустить обработку задания, установление соединения, ожидание коммуникационного партнера
0	7002	Данные отправляются или принимаются
0	7003	Коммуникационное соединение завершается
0	7004	Коммуникационное соединение, установлено и контролируется; нет активного выполнения задания
1	8085	Параметр LEN больше, чем наибольшее разрешенное значение.
1	8086	Параметр CONNECT вне разрешенного диапазона.
1	8087	Максимальное количество соединений достигнуто; дополнительное соединение невозможно.
1	8088	Параметр LEN недопустим для области памяти, определенной в DATA.
1	8089	Параметр CONNECT не указывает на блок данных.
1	8091	Максимальная глубина вложения превышена.
1	809A	Параметр CONNECT указывает на поле, которое не соответствует длине описания соединения.
1	809B	local_device_id в описании соединения не соответствует ЦПУ.
1	80A1	Коммуникационная ошибка: - Указанное соединение еще не было установлено - Указанное соединение в настоящее время завершается; передача по этому соединению невозможна - Интерфейс повторно инициализируется
1	80A3	Предпринята попытка завершить несуществующее соединение
1	80A4	IP-адрес соединения с удаленным партнером неправильный. Например, IP-адрес удаленного партнера совпадает с IP-адресом локального партнера.
1	80A5	ID соединения (стр. 637) уже используется.
1	80A7	Коммуникационная ошибка: Вы вызвали TDISCON до завершения TSEND_C.
1	80B2	Параметр CONNECT указывает на блок данных, который был сгенерирован с ключевым словом UNLINKED.
1	80B3	Несогласованные параметры: - Ошибка в описании соединения - Локальный порт (параметр local_tsap_id) уже присутствует в другом описании соединения. - ID в описании соединения, отличается от ID, определенного как параметр

ERROR	STATUS	Описание
1	80B4	При использовании ISO on TCP (connection_type = B#16#12), чтобы установить пассивное соединение, код состояния 80B4 предупреждает Вас, что введенный TSAP не соответствовал одному из следующих, требований к адресу: - Для длины локального TSAP равной 2 и значения идентификатора TSAP равного E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01. - Для длины локального TSAP равной 3 или больше и значения идентификатора TSAP равного E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01, а все остальные байты должны быть допустимыми ASCII-символами. - Для длины локального TSAP равной 3 или больше и первого байта TSAP ID не имеющего значения E0 или E1 (в шестнадцатеричном формате) все байты TSAP ID должны быть допустимыми ASCII-символами. Действительные ASCII-символы представляют собой байтовые значения от 20 до 7E (в шестнадцатеричном формате).
1	80B7	Тип данных и/или длина переданных данных не согласуются с областью в ЦПУ партнера, в которую они должны быть записаны.
1	80C3	Все коммуникационные ресурсы используются.
1	80C4	Временная коммуникационная ошибка: - В данный момент времени не может быть установлено соединение - Интерфейс принимает новые параметры - Сконфигурированное соединение в настоящее время удаляется TDISCON.
1	8722	Параметр CONNECT: Область источника недействительна: область не находится в DB.
1	873A	Параметр CONNECT: Обращение к описанию соединения невозможно (например, DB недоступен)
1	877F	Параметр CONNECT: Внутренняя ошибка, такая как неверная ссылка ANY
1	893A	Параметр содержит номер DB, который не загружен.

Протоколы Ethernet соединения

Каждый ЦПУ обладает интегрированным PROFINET-портом, который поддерживает стандартные PROFINET-коммуникации. Инструкции TSEND_C и TRCV_C, а также TSEND и TRCV поддерживают Ethernet-протоколы TCP и ISO on TCP.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

11.2.8.9. Инструкции TCON, TDISCON, TSEND и TRCV

Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможности инструкций TCON, TDISCON, TSEND и TRCV для использовать параметров соединения со структурами "TCON_IP_v4" и "TCON_IP_RFC".

По этой причине S7-1200 поддерживает два набора инструкций TCON, TDISCON, TSEND и TRCV:

- Унаследованные инструкции TCON, TDISCON, TSEND и TRCV (стр. 673): Эти инструкции TCON, TDISCON, TSEND и TRCV существовали до версии V4.0 S7-1200 и работали только с параметрами соединения со структурами "TCON_Param".
- Инструкции TCON, TDISCON, TSEND и TRCV (стр. 664): Эти инструкции TCON, TDISCON, TSEND и TRCV обеспечивают всю функциональность унаследованных инструкций плюс возможность использовать параметры соединения со структурами "TCON_IP_v4" и "TCON_IP_RFC".

Выбор версии инструкций TCON, TDISCON, TSEND и TRCV

Существует две версии инструкций TCON, TDISCON, TSEND и TRCV, доступных в STEP 7:

- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних.
- Версия 4.0 доступна в STEP 7 Basic/Professional V137 V13 SP1.

Для совместимости и простоты миграции, Вы можете выбрать какую из версий инструкции вставить в Вашу пользовательскую программу.

Не используйте различные версии инструкции в одной программе ЦПУ.



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.

Open user communication		V4.0
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V3.0
TCON	Establish communication c...	V4.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V4.0
TRCV	Receive data via communic...	V4.0

Чтобы изменить версию TCON, TDISCON, TSEND и TRCV инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить TCON, TDISCON, TSEND или TRCV инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной TCON, TDISCON, TSEND или TRCV инструкции создается в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию TCON, TDISCON, TSEND или TRCV инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите TCON, TDISCON, TSEND или TRCV экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии TCON, TDISCON, TSEND или TRCV инструкции.

Инструкции TCON, TDISCON, TSEND и TRCV (TCP коммуникации)

Ethernet-коммуникации с использованием протоколов TCP и ISO on TCP

Примечание

Инструкции TSEND_C и TRCV_C

Чтобы облегчить программирование PROFINET/Ethernet коммуникаций, инструкция TSEND_C и инструкция TRCV_C сочетают функциональность инструкций TCON, TDISCON, TSEND и TRCV:

- TSEND_C комбинирует инструкции TCON, TDISCON и TSEND;
 - TRCV_C комбинирует инструкции TCON, TDISCON и TRCV.
-

Следующие инструкции управляют коммуникационным процессом:

- TCON устанавливает TCP/IP соединение между клиентом и ПК-сервером (ЦПУ).
- TSEND и TRCV отправляют и принимают данные.
- TDISCON разрывает соединение.

Минимальный размер данных, которые Вы можете передать (TSEND) или принять (TRCV), составляет один байт; максимальный размер составляет 8192 байта. TSEND не поддерживает передачу данных из булевых ячеек, а TRCV не принимает данные в булевые ячейки. Для получения информации о передаче данных с помощью этих инструкций, смотри раздел по согласованности данных (стр. 185).

TCON, TDISCON, TSEND и TRCV работают асинхронно, то есть обработка задания продолжается за несколько вызовов команды. Например, Вы запускаете задание по настройке и установлению соединения, выполняя инструкцию TCON с параметром REQ = 1. Затем Вы используете дополнительные вызовы TCON, чтобы контролировать прогресс задания и проверять завершение задания с помощью параметра DONE.

Следующая таблица показывает отношения между BUSY, DONE и ERROR. Используйте таблицу, чтобы определить текущее состояние задания:

Таблица 11- 20 Взаимодействие между параметрами BUSY, DONE и ERROR

BUSY	DONE	ERROR	Описание
1	0	0	Задание обрабатывается.
0	1	0	Задание успешно завершено.
0	0	1	Задание завершилось с ошибкой. Причина ошибки выведена в параметре STATUS.
0	0	0	Нет новых назначенных заданий.

TCON и TDISCON

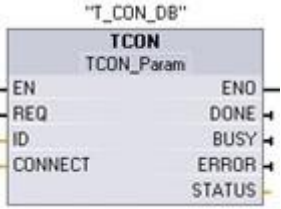
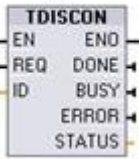
Примечание

Инициализация коммуникационных параметров

После того, как Вы вставите инструкцию TCON, используйте "свойства" инструкции (стр. 619), чтобы сконфигурировать коммуникационные параметры (стр. 641). Поскольку Вы вводите параметры для коммуникационных партнеров в окне инспектора, STEP 7 вводит соответствующие данные в экземплярный DB для инструкции.

Если Вы хотите использовать мультиэкземплярный DB, Вы должны вручную сконфигурировать DB в обоих ЦПУ.

Таблица 11- 21 Инструкции TCON и TDISCON

LAD / FBD		Описание
	<pre>"TCON_DB" (req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, connect:=_struct_inout_);</pre>	TCP и ISO on TCP: TCON иницирует коммуникационное соединение ЦПУ с коммуникационным партнером.
	<pre>"TDISCON_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	TCP и ISO on TCP: TDISCON прерывает коммуникационное соединение ЦПУ с коммуникационным партнером.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 22 Типы данных для параметров TCON и TDISCON

Параметр	Объявление	Тип данных	Описание
REQ	IN	Bool	Запускает задание по установлению определенного соединения по нарастающему фронту на ID.
ID	IN	CONN_OUC (Word)	Ссылка на назначенное соединение. Диапазон значений: от W#16#0001 до W#16#0FFF

Параметр	Объявление	Тип данных	Описание
CONNECT (TCON)	IN_OUT	VARIANT	Указатель на описание соединения - Для TCP или UDP, используйте структуру TCON_IP_v4 За описанием обратитесь к теме: "Параметры соединения со структурой TCON_IP_v4" в TIA Portal. - Для ISO-on-TCP, используйте структуру TCON_IP_RFC За описанием обратитесь к теме: "Параметры соединения со структурой TCON_IP_RFC" в TIA Portal.
DONE	OUT	Bool	Параметр состояния со следующими значениями: 0: Задание еще не запущено или все еще обрабатывается 1: Задание выполнено без ошибок
BUSY	OUT	Bool	Параметр состояния со следующими значениями: 0: Задание еще не запущено или уже завершено 1: Задание еще не завершено. Новое задание не может быть запущено
ERROR	OUT	Bool	Параметр для отображения состояния ошибки: 0: Нет ошибок 1: Произошла ошибка
STATUS	OUT	Word	Состояние инструкции

Оба коммуникационных партнера выполняют инструкцию TCON, чтобы настроить и установить коммуникационное соединение. Вы используете параметры, чтобы определить активного и пассивного коммуникационных партнеров конечной точки. После того, как соединение настроено и установлено, оно автоматически сохраняется и контролируется ЦПУ.

Если соединение завершено из-за разрыва линии или из-за удаленного коммуникационного партнера, например, активный партнер пытается восстановить сконфигурированное соединение. Вы не должны выполнять TCON снова.

Существующее соединение завершено, и настройка соединения удалена, когда выполняется инструкция TDISCON или когда ЦПУ перешел в режим STOP. Чтобы настроить и восстановить соединение, Вы должны выполнить TCON снова.

Таблица 11- 23 Коды состояния Error и Status инструкций TCON и TDISCON

ERROR	STATUS (W#16#...)	Объяснение
0	0000	Соединение успешно установлено.
0	7000	Нет активной обработки задания
0	7001	Запуск выполнения задания; установить соединение (TCON) или завершить соединение
0	7002	Соединение устанавливается (REQ не важен); установить соединение (TCON) или завершить соединение (TDISCON).
1	8085	TCON: ID соединения уже используется.
1	8086	TCON: параметр ID вне допустимого диапазона.
1	8087	TCON: Максимальное количество соединений достигнуто; дополнительное соединение
1	8089	TCON: параметр CONNECT не указывает на описание соединения, или описание соединения создавалось вручную.

ERROR	STATUS (W#16#...)	Объяснение
1	809A	TCON: Структура в параметре CONNECT не поддерживается или длина неправильная.
1	809B	TCON: ID локального устройства в описании соединения не соответствует ЦПУ или КП или он равен "0".
1	80A0	Ошибка группы для кодов ошибки W#16#80A1 и W#16#80A2.
1	80A1	TCON: Для TCP/UDP (TCON_IP_v4): Соединение ил порт уже используются.
1	80A2	TCON: Локальный или удаленный порт используется системой.
1	80A3	TCON: Значение в параметре ID уже используется соединением (TCON), которое создавалось с использованием пользовательской программы. Соединение использует идентичный ID, но различные настройки соединения в параметре CONNECT.
1	80A4	TCON: IP-адрес удаленной конечной точки соединения неправильный или он соответствует IP-адресу локального партнера.
1	80A5	TCON: ID соединения уже используется.
1	80A7	TCON: Коммуникационная ошибка: Вы выполнили "TDISCON" до завершения "TCON".
1	80B2	TCON: Параметр CONNECT указывает на блок данных, который был сгенерирован с атрибутом, "Only store in load memory".
1	80B3	Несогласованное назначение параметра: ошибка группы для кодов ошибки от W#16#80A0 до W#16#80A2, W#16#80A4, от W#16#80B4 до W#16#80B9.
1	80B4	TCON: Только с TCON_IP_RFC. Локальный T селектор не был определен или первый байт не содержит значение 0x0E или локальный T селектор начинается с "SIMATIC-".
1	80B5	TCON: Разрешено установление только пассивного соединения для соединения типа 13 = UDP (Параметр active_est структуры TCON_IP_v4 имеет значение TRUE).
1	80B6	TCON: Ошибка назначения параметра connection_type блока данных для описания соединения. - Действительно только с TCON_IP_v4: 0x11, 0x0B и 0x13. - Действительно только с TCON_IP_RFC: 0x0C и 0x12
1	80B7	TCON: С TCON_IP_v4: - TCP (установление активного соединения): Удаленный порт "0". - TCP (установление пассивного соединения): Локальный порт "0". - UDP: Локальный порт "0". TCON: С TCON_IP_RFC: - Локальный (local_tselector) или удаленный (remote_tselector) T селектор был определен с длиной больше чем 32 байта. - Для TselLength T селектора (локальный или удаленный) была введена длина, больше чем 32. - Ошибка длины IP-адреса определенного партнера по соединению.
1	80B8	TCON: ID параметра в описании локального соединения (структура в параметре CONNECT) и параметр ID инструкции различны.
1	80C3	TCON: Все коммуникационные (стр. 637) ресурсы используются.
1	80C4	Временная коммуникационная ошибка: - Соединение не может быть установлено в данный момент времени (TCON). - Интерфейс в настоящее время принимает новые параметры (TCON и TDISCON). - Сконфигурированное соединение в настоящее время удаляется инструкцией "TDISCON" (TCON).
1	80C5	TCON: Удаленный партнер отказывается устанавливать соединение, прервал соединение или активно завершил его.

ERROR	STATUS (W#16#...)	Объяснение
1	80C6	TCON: Невозможно установить соединение с удаленным партнером (ошибка сети).
1	80C7	TCON: Тайм-аут выполнения.
1	80C8	TCON: ID используется соединением, созданным пользовательской программой, которое использует то же описание соединения в параметре CONNECT.
1	80C9	TCON: Проверка достоверности удаленного партнера перестала работать. Удаленный партнер, который хочет установить соединение, не соответствует определенному партнеру в структуре параметра CONNECT.
1	80CE	TCON: IP-адрес локального интерфейса 0.0.0.0.

TSEND и TRCV

Примечание

Используя открытые пользовательские коммуникации по PROFINET, если Вы выполняете инструкцию TSEND без соответствующей инструкции TRCV, выполняющейся на удаленном устройстве, инструкция TSEND может находиться неопределенно в "состоянии Занято", ожидая инструкции TRCV, чтобы принять данные. В этом состоянии установлен выход инструкции TSEND "Busy", а выход "Status" имеет значение "0x7002". Это может иметь место, если Вы передаете больше чем 4096 байтов данных. Проблема решается при следующем выполнении инструкции TRCV.

Таблица 11- 24 Инструкции TSEND и TRCV

LAD / FBD	SCL	Описание
	<pre>"TSEND_DB" (req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, data:=_variant_inout_);</pre>	TCP и ISO on TCP: TSEND отправляет данные через коммуникационное соединение из ЦПУ в станцию партнера.
	<pre>"TRCV_DB" (en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, adhoc:=_bool_in_, ndr=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	TCP и ISO on TCP: TRCV принимает данные через коммуникационное соединение от станции партнера в ЦПУ.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 25 Типы данных для параметров TSEND и TRCV

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	TSEND: Запускает задание на отправку по нарастающему фронту. Данные передаются из области, определенной DATA и LEN.
EN_R	IN	Bool	TRCV: Позволяет ЦПУ принимать; с EN_R = 1, TRCV готов к приему. Задание на получение обрабатывается.
ID	IN	CONN_OUC (Word)	Ссылка на связанное соединение. ID должен быть идентичен связанному параметру ID в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
LEN	IN	UDInt	Максимальное количество байтов, подлежащие отправке (TSEND) или получению (TRCV): - По умолчанию = 0: параметр DATA определяет длину данных, подлежащих отправке (TSEND) или получению (TRCV). - Ситуативный режим = 65535: переменная длина данных устанавливается для приема (TRCV).
ADHOC	IN	Bool	TRCV: Дополнительный (скрытый) параметр Запрос ситуативного режима для подключения типа TCP.
DATA	IN_OUT	Variant	Указатель на область данных передачи (TSEND) или приема (TRCV); область данных содержит адрес и длину. Адрес ссылается на I, Q или M области памяти или DB.
DONE	OUT	Bool	TSEND: 0: Задание еще не запущено или все еще выполняется. 1: Задание выполнено без ошибок.
NDR	OUT	Bool	TRCV: - NDR = 0: Задание еще не запущено или все еще выполняется. - NDR = 1: Задание успешно завершено.
BUSY	OUT	Bool	- BUSY = 1: задание еще не завершено. Новое задание не может быть активировано. - BUSY = 0: Задание завершено.
ERROR	OUT	Bool	ERROR = 1: Во время обработки произошла ошибка. STATUS предоставляет подробную информацию о типе ошибки
STATUS	OUT	Word	Информация о статусе, включая информацию об ошибке. (Обратитесь к кодам состояния Error и Status в таблице ниже.)
RCVD_LEN	OUT	UDInt	TRCV: Объем фактически принятых данных в байтах

Примечание

Инструкции TSEND требуется переход от низкого к высокому уровню во входном параметре REQ, чтобы запустить задание на отправку. Параметр BUSY затем устанавливается в 1 на время обработки. Завершение задания на отправку отображается установкой в 1 параметров DONE или ERROR на один скан.

В это время любой переход от низкого к высокому уровню во входном параметре REQ игнорируется.

Операции TRCV

Инструкция TRCV записывает принятые данные в область приема, которая определяется следующими двумя переменными:

- Указателем на начало области;
- Длиной области или значением, представленным на входе LEN если не равно 0.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) использует параметр DATA, чтобы определить длину передаваемых данных. Гарантируйте, что DATA, переданные инструкцией TSEND имеют тот же размер, что и параметр DATA инструкции TRCV.

Как только все данные задания были получены, TRCV передает их в область приема и устанавливает NDR в 1.

Таблица 11- 26 Ввод данных в область приема

Вариант протокола	Ввод данных в область приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ситуативный режим	B#16#11	Выбирается по входу ADHOC инструкции TRCV	от 1 до 1472
TCP	Прием данных с указанной длиной	B#16#11	0 (рекомендуется) или от 1 до 8192, исключая 65535	от 1 до 8192
ISO on TCP	Ситуативный режим	B#16#12	65535	от 1 до 1472
ISO on TCP	С управлением по протоколу	B#16#12	0 (рекомендуется) или от 1 до 8192, исключая 65535	от 1 до 8192

Примечание

Ситуативный (Ad hoc) режим

"Ситуативный режим" существует с TCP и ISO on TCP вариантами протокола. Чтобы сконфигурировать инструкцию TRCV для ситуативного режима, установите входной ADHOC параметр инструкции. Область приема идентична области, сформированной посредством DATA. Длина полученных данных будет выведена в параметре RCVD_LEN. Сразу после получения блока данных, TRCV вводит данные в область приема и устанавливает NDR в 1.

Если Вы храните данные в "оптимизированном" DB (только символьный доступ), Вы можете получить данные только в массивах с типом Byte, Char, USInt и SInt.

Примечание

Импорт S7-300/400 STEP 7 проектов, содержащих "ситуативный режим" в S7-1200

В S7-300/400 STEP 7 проектах "ситуативный режим" выбирается назначением "0" параметру LEN. В S7-1200 Вы конфигурируете инструкцию TRCV для ситуативного режима установкой параметра ADHOC.

Если Вы импортируете S7-300/400 STEP 7 проект, содержащий "ситуативный режим" в S7-1200, Вы должны изменить параметр LEN на "65535".

Таблица 11- 27 Коды состояния Error и Status инструкций TSEND и TRCV

ERROR	STATUS	Описание
0	0000	- Задание на отправку завершено без ошибок (TSEND) - Новые данные получены: текущая длина принятых данных представлена в RCVD_LEN (TRCV).
0	7000	- Нет активного задания в обработке (TSEND) - Блок не готов к приему (TRCV)
0	7001	- Запуск обработки задания, данные отправляются: Во время этой обработки операционная система получает доступ к данным в DATA области передачи (TSEND). - Блок готов к приему, задание на получение было активировано (TRCV).
0	7002	- Последующее выполнение инструкции (REQ не важен), задание обрабатывается: Во время этой обработки операционная система получает доступ к данным в DATA области передачи (TSEND). - Последующее выполнение инструкции, задание на получение обрабатывается : Во время этой обработки данные записываются в область приема. По этой причине ошибка могла бы привести к несогласованным данным в области приема (TRCV).
1	8085	- Параметр LEN больше, чем наибольшее разрешенное значение (TSEND) и (TRCV). - Параметр LEN или DATA изменились с момента первого выполнения инструкции (TRCV).
1	8086	Параметр ID не находится в разрешенном диапазоне адресов.
1	8088	Параметр LEN больше, чем область памяти, определенная в DATA.
1	80A1	Коммуникационная ошибка: - Указанное соединение еще не установлено (TSEND и TRCV). - Указанное соединение в настоящее время завершается. Задания на отправку или прием по этому соединению невозможны (TSEND и TRCV). - Интерфейс повторно инициализируется (TSEND). - Интерфейс принимает новые параметры (TRCV).
1	80C3	Внутреннее отсутствие коммуникационных (стр. 637) ресурсов: блок с этим ID уже обрабатывается в другом классе приоритета.
1	80C4	Временная коммуникационная ошибка: - Соединение с коммуникационным партнером не может быть установлено в данный момент времени. - Интерфейс принимает новые настройки параметров или соединение в настоящий момент устанавливается.

Протоколы Ethernet соединения

Каждый ЦПУ обладает интегрированным PROFINET-портом, который поддерживает стандартные PROFINET-коммуникации. Инструкции TSEND_C и TRCV_C, а также TSEND и TRCV поддерживают Ethernet-протоколы TCP и ISO on TCP.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

11.2.8.10. Унаследованные инструкции TCON, TDISCON, TSEND и TRCV

До выпуска STEP 7 V13 SP1 и S7-1200 ЦПУ V4.1 инструкции TCON, TDISCON, TSEND и TRCV могли работать только с параметрами соединения со структурами "TCON_Param". Общие понятия применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным инструкциям TCON, TDISCON, TSEND и TRCV для получения информации о программировании.

Выбор версии инструкций TCON, TDISCON, TSEND и TRCV

Существует две версии инструкций TCON, TDISCON, TSEND или TRCV, доступных в STEP 7:

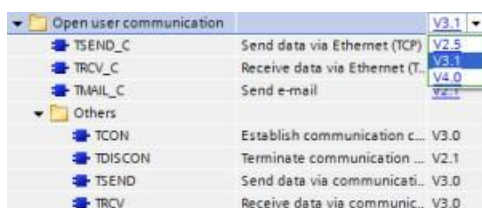
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних.
- Версия 4.0 доступна в STEP 7 Basic/Professional V137 V13 SP1.

Для совместимости и простоты миграции, Вы можете выбрать какую из версий инструкции вставить в Вашу пользовательскую программу.

Не используйте различные версии инструкции в одной программе ЦПУ.



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций



Чтобы изменить версию TCON, TDISCON, TSEND или TRCV инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить TCON, TDISCON, TSEND или TRCV инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной TCON, TDISCON, TSEND или TRCV инструкции создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию TCON, TDISCON, TSEND или TRCV инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите TSEND_C или TRCV_C экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии TCON, TDISCON, TSEND или TRCV инструкции.

Унаследованные инструкции TCON, TDISCON, TSEND и TRCV (TCP коммуникации)

Ethernet-коммуникации с использованием протоколов TCP и ISO on TCP

Примечание

Инструкции TSEND_C и TRCV_C

Чтобы облегчить программирование PROFINET/Ethernet коммуникаций, инструкция TSEND_C и инструкция TRCV_C сочетают функциональность инструкций TCON, TDISCON, TSEND и TRCV:

- TSEND_C комбинирует инструкции TCON, TDISCON и TSEND;
- TRCV_C комбинирует инструкции TCON, TDISCON и TRCV.

Следующие инструкции управляют коммуникационным процессом:

- TCON устанавливает TCP/IP соединение между клиентом и ПК-сервером (ЦПУ).
- TSEND и TRCV отправляют и принимают данные.
- TDISCON разрывает соединение.

Минимальный размер данных, которые Вы можете передать (TSEND) или принять (TRCV), составляет один байт; максимальный размер составляет 8192 байта. TSEND не поддерживает передачу данных из булевых ячеек, а TRCV не принимает данные в булевые ячейки. Для получения информации о передаче данных с помощью этих инструкций, смотри раздел по согласованности данных (стр. 185).

TCON, TDISCON, TSEND и TRCV работают асинхронно, то есть обработка задания продолжается за несколько вызовов команды. Например, Вы запускаете задание по настройке и установлению соединения, выполняя инструкцию TCON с параметром REQ = 1. Затем Вы используете дополнительные вызовы TCON, чтобы контролировать прогресс задания и проверять завершение задания с помощью параметра DONE.

Следующая таблица показывает отношения между BUSY, DONE и ERROR. Используйте таблицу, чтобы определить текущее состояние задания:

Таблица 11- 28 Взаимодействие между параметрами BUSY, DONE и ERROR

BUSY	DONE	ERROR	Описание
TRUE	не важно	не важно	Задание обрабатывается.
FALSE	TRUE	FALSE	Задание успешно завершено.
FALSE	FALSE	TRUE	Задание завершилось с ошибкой. Причина ошибки выведена в параметре STATUS.
FALSE	FALSE	FALSE	Нет новых назначенных заданий.

TCON и TDISCON

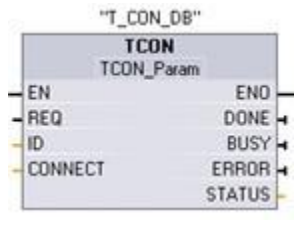
Примечание

Инициализация коммуникационных параметров

После того, как Вы вставите инструкцию TCON, используйте "свойства" инструкции (стр. 619), чтобы сконфигурировать коммуникационные параметры (стр. 641). Поскольку Вы вводите параметры для коммуникационных партнеров в окне инспектора, STEP 7 вводит соответствующие данные в экземплярный DB для инструкции.

Если Вы хотите использовать мультиэкземплярный DB, Вы должны вручную сконфигурировать DB в обоих ЦПУ.

Таблица 11- 29 Инструкции TCON и TDISCON

LAD / FBD		Описание
	<pre>"TCON_DB"(req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, connect:=_struct_inout_);</pre>	TCP и ISO on TCP: TCON иницирует коммуникационное соединение ЦПУ с коммуникационным партнером.
	<pre>"TDISCON_DB"(req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	TCP и ISO on TCP: TDISCON прерывает коммуникационное соединение ЦПУ с коммуникационным партнером.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 30 Типы данных для параметров TCON и TDISCON

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQ запускает задание, устанавливая соединение, определенное в ID. Задание запускается по нарастающему фронту.
ID	IN	CONN_OUC (Word)	Ссылка на соединение, которое будет установлено (TCON) или завершено (TDISCON) с удаленным партнером, или между пользовательской программой и коммуникационным уровнем операционной системы. ID должен быть идентичен связанному параметру ID в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
CONNECT (TCON)	IN_OUT	TCON_Param	Указатель на описание соединения (стр. 641)

Параметр и тип		Тип данных	Описание
DONE	OUT	Bool	0: Задание еще не запущено или все еще обрабатывается 1: Задание выполнено без ошибок
BUSY	OUT	Bool	0: Задание завершено. 1: Задание еще не завершено. Новое задание не может быть запущено.
ERROR	OUT	Bool	Параметр для отображения состояния ошибки: 0: Нет ошибок 1: Произошла ошибка в процессе обработки. STATUS предоставляет подробную информацию о типе ошибки.
STATUS	OUT	Word	Информация о статусе, включая информацию об ошибке. (Обратитесь к кодам состояния Error и Status в таблице ниже.)

Оба коммуникационных партнера выполняют инструкцию TCON, чтобы настроить и установить коммуникационное соединение. Вы используете параметры, чтобы определить активного и пассивного коммуникационных партнеров конечной точки. После того, как соединение настроено и установлено, оно автоматически сохраняется и контролируется ЦПУ.

Если соединение завершено из-за разрыва линии или из-за удаленного коммуникационного партнера, например, активный партнер пытается восстановить сконфигурированное соединение. Вы не должны выполнять TCON снова.

Существующее соединение завершено, и настройка соединения удалена, когда выполняется инструкция TDISCON или когда ЦПУ перешел в режим STOP. Чтобы настроить и восстановить соединение, Вы должны выполнить TCON снова.

Таблица 11- 31 Коды состояния ERROR и STATUS инструкций TCON и TDISCON

ERROR	STATUS (W#16#...)	Описание
0	0000	Соединение было успешно установлено.
0	7000	Нет активной обработки задания
0	7001	Запуск выполнения задания; установить соединение (TCON) или завершить соединение (TDISCON).
0	7002	Повторный вызов (REQ не важен); установить соединение (TCON) или завершить соединение (TDISCON).
1	8086	Параметр ID вне допустимого диапазона адреса.
1	8087	TCON: Максимальное количество соединений достигнуто; дополнительное соединение невозможно.
1	809B	TCON: local_device_id в описании соединения не соответствует ЦПУ.
1	80A1	TCON: Соединение или порт уже задействованы пользователем.
1	80A2	TCON: Локальный или удаленный порт используется системой.
1	80A3	Предпринята попытка восстановить существующее соединение (TCON) или завершиться несуществующее соединение (TDISCON).
1	80A4	TCON: IP-адрес удаленной конечной точки соединения неправильный; он соответствует IP-адресу локального партнера.
1	80A5	TCON: ID соединения (стр. 637) уже используется.
1	80A7	TCON: Коммуникационная ошибка: Вы выполнили "TDISCON" до завершения "TCON". TDISCON должен сначала полностью завершить соединение, на которое ссылается ID.

ERROR	STATUS	Описание
1	80B2	TCON: Параметр CONNECT указывает на блок данных, который был сгенерирован с атрибутом, "Only store in load memory".
1	80B4	TCON: При использовании ISO on TCP (connection_type = B#16#12), чтобы установить пассивное соединение, код состояния 80B4 предупреждает Вас, что введенный TSAP не соответствовал одному из следующих, требований к адресу: - Для длины локального TSAP равной 2 и значения идентификатора TSAP равного E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01. - Для длины локального TSAP равной 3 или больше и значения идентификатора TSAP равного E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01, а все остальные байты должны быть допустимыми ASCII-символами. - Для длины локального TSAP равной 3 или больше и первого байта TSAP ID не имеющего значения E0 или E1 (в шестнадцатеричном формате) все байты TSAP ID должны быть допустимыми ASCII-символами. Действительные ASCII-символы представляют собой байтовые значения от 20 до 7E (в шестнадцатеричном формате).
1	80B5	TCON: Тип соединения "13 = UDP" позволяет установление только пассивного соединения.
1	80B6	TCON: Ошибка назначения параметра CONNECTION_TYPE в SDT TCON_Param.
1	80B7	TCON: Ошибка в одном из следующих параметров блока данных для описания соединения: - block_length - local_tsap_id_len - rem_subnet_id_len - rem_staddr_len - rem_tsap_id_len - next_staddr_len Примечание: При работе TCON в TCP пассивном режиме LOCAL_TSAP_ID_LEN должен быть равен "2", а REM_TSAP_ID_LEN - "0".
1	80B8	TCON: Параметр в описании локального соединения и параметр ID различны.
1	80C3	TCON: Все коммуникационные ресурсы используются.
1	80C4	Временная коммуникационная ошибка: - Соединение не может быть установлено в данный момент времени (TCON). - Сконфигурированное соединение в настоящее время удаляется инструкцией TDISCON (TCON). - Соединение в настоящее время устанавливается (TDISCON). - Интерфейс в настоящее время принимает новые параметры (TCON и TDISCON).

TSEND и TRCV

Примечание

Используя открытые пользовательские коммуникации по PROFINET, если Вы выполняете инструкцию TSEND без соответствующей инструкции TRCV, выполняющейся на удаленном устройстве, инструкция TSEND может находиться неопределенно в "состоянии Занято", ожидая инструкции TRCV, чтобы принять данные. В этом состоянии установлен выход инструкции TSEND "Busy", а выход "Status" имеет значение "0x7002". Это может иметь место, если Вы передаете больше чем 4096 байтов данных. Проблема решается при следующем выполнении инструкции TRCV.

Таблица 11- 32 Инструкции TSEND и TRCV

LAD / FBD	SCL	Описание
"T_SEND_DB_1" 	<pre>"TSEND_DB"(req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, data:=_variant_inout_);</pre>	TCP и ISO on TCP: TSEND отправляет данные через коммуникационное соединение из ЦПУ в станцию партнера.
"T_RCV_DB" 	<pre>"TRCV_DB"(en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, ndr=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	TCP и ISO on TCP: TRCV принимает данные через коммуникационное соединение от станции партнера в ЦПУ.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 33 Типы данных для параметров TSEND и TRCV

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	TSEND: Запускает задание на отправку по нарастающему фронту. Данные передаются из области, определенной DATA и LEN.
EN_R	IN	Bool	TRCV: Позволяет ЦПУ принимать; с EN_R = 1, TRCV готов к приему. Задание на получение обрабатывается.
ID	IN	CONN_OUC (Word)	Ссылка на связанное соединение. ID должен быть идентичен связанному параметру ID в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF

Параметр и тип		Тип данных	Описание
LEN	IN	UDInt	Максимальное количество байтов, подлежащие отправке (TSEND) или получению (TRCV): - По умолчанию = 0: параметр DATA определяет длину данных, подлежащих отправке (TSEND) или получению (TRCV). - Ситуативный режим = 65535: переменная длина данных устанавливается для приема (TRCV).
DATA	IN_OUT	Variant	Указатель на область данных передачи (TSEND) или приема (TRCV); область данных содержит адрес и длину. Адрес ссылается на I, Q или M области памяти или DB.
DONE	OUT	Bool	TSEND: 0: Задание еще не запущено или все еще выполняется. 1: Задание выполнено без ошибок.
NDR	OUT	Bool	TRCV: - NDR = 0: Задание еще не запущено или все еще выполняется. - NDR = 1: Задание успешно завершено.
BUSY	OUT	Bool	- BUSY = 1: задание еще не завершено. Новое задание не может быть активировано. - BUSY = 0: Задание завершено.
ERROR	OUT	Bool	ERROR = 1: Во время обработки произошла ошибка. STATUS предоставляет подробную информацию о типе ошибки
STATUS	OUT	Word	Информация о статусе, включая информацию об ошибке. (Обратитесь к кодам состояния Error и Status в таблице ниже.)
RCVD_LEN	OUT	UDInt	TRCV: Объем фактически принятых данных в байтах

Примечание

Инструкции TSEND требуется переход от низкого к высокому уровню во входном параметре REQ, чтобы запустить задание на отправку. Параметр BUSY затем устанавливается в 1 на время обработки. Завершение задания на отправку отображается установкой в 1 параметров DONE или ERROR на один скан.

В это время любой переход от низкого к высокому уровню во входном параметре REQ игнорируется.

Операции TRCV

Инструкция TRCV записывает принятые данные в область приема, которая определяется следующими двумя переменными:

- Указателем на начало области;
- Длиной области или значением, представленным на входе LEN если не равно 0.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) использует параметр DATA, чтобы определить длину передаваемых данных. Гарантируйте, что DATA, переданные инструкцией TSEND имеют тот же размер, что и параметр DATA инструкции TRCV.

Как только все данные задания были получены, TRCV передает их в область приема и устанавливает NDR в 1.

Таблица 11- 34 Ввод данных в область приема

Вариант протокола	Ввод данных в область приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ситуативный режим	B#16#11	65535	от 1 до 1472
TCP	Прием данных с указанной длиной	B#16#11	0 (рекомендуется) или от 1 до 8192, исключая 65535	от 1 до 8192
ISO on TCP	Ситуативный режим	B#16#12	65535	от 1 до 1472
ISO on TCP	С управлением по протоколу	B#16#12	0 (рекомендуется) или от 1 до 8192, исключая 65535	от 1 до 8192

Примечание

Ситуативный (Ad hoc) режим

"Ситуативный режим" существует с TCP и ISO on TCP вариантами протокола. Вы устанавливаете "ситуативный режим", назначая "65535" параметру LEN. Область приема идентична области, сформированной посредством DATA. Длина полученных данных будет выведена в параметре RCVD_LEN. Сразу после получения блока данных, TRCV вводит данные в область приема и устанавливает NDR в 1.

Если Вы храните данные в "оптимизированном" DB (только символьный доступ), Вы можете получить данные только в массивах с типом Byte, Char, UInt и SInt.

Примечание

Импорт S7-300/400 STEP 7 проектов, содержащих "ситуативный режим" в S7-1200

В S7-300/400 STEP 7 проектах "ситуативный режим" выбирается назначением "0" параметру LEN. В S7-1200 Вы выбираете ситуативный режим, назначая "65535" параметру LEN.

Если Вы импортируете S7-300/400 STEP 7 проект, содержащий "ситуативный режим" в S7-1200, Вы должны изменить параметр LEN на "65535".

Коды состояния Error и Status инструкций TSEND и TRCV

ERROR	STATUS	Описание
0	0000	- Задание на отправку завершено без ошибок (TSEND) - Новые данные приняты: текущая длина принятых данных выведена в RCVD_LEN (TRCV).
0	7000	- Нет активных заданий в обработке (TSEND) - Блок не готов к приему (TRCV)
0	7001	- Запуск обработки задания, данные отправляются: Во время этой обработки операционная система получает доступ к данным в DATA области передачи (TSEND). - Блок готов к приему, задание на получение было активировано (TRCV).
0	7002	- Последующее выполнение инструкции (REQ не важен), задание обрабатывается: Во время этой обработки операционная система получает доступ к данным в DATA области передачи (TSEND). - Последующее выполнение инструкции, задание на получение обрабатывается: Во время этой обработки данные записываются в область приема. По этой причине ошибка могла бы привести к несогласованности данных в области приема (TRCV).
1	8085	- Параметр LEN больше чем наибольшее разрешенное значение (TSEND) и (TRCV). - Параметр LEN или DATA изменились с момента первого выполнения инструкции (TRCV).
1	8086	Параметр ID не находится в разрешенном диапазоне адресов.
1	8088	Параметр LEN больше, чем область памяти, определенная в DATA.
1	80A1	Коммуникационная ошибка: - Указанное соединение еще не установлено (TSEND и TRCV). - Указанное соединение в настоящее время завершается. Задания на передачу или прием по этому соединению невозможны (TSEND и TRCV). - Интерфейс повторно инициализируется (TSEND). - Интерфейс принимает новые параметры (TRCV).
1	80C3	Внутреннее отсутствие коммуникационных ресурсов: блок с этим ID уже обрабатывается в другом классе приоритета.
1	80C4	Временная коммуникационная ошибка: - Соединение с коммуникационным партнером не может быть установлено в данный момент времени. - Интерфейс принимает новые настройки параметров или соединение в настоящий момент устанавливается.

Протоколы Ethernet соединения

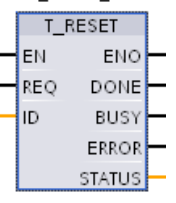
Каждый ЦПУ обладает интегрированным PROFINET-портом, который поддерживает стандартные PROFINET-коммуникации. Инструкции TSEND_C и TRCV_C, а также TSEND и TRCV поддерживают Ethernet-протоколы TCP и ISO on TCP.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

11.2.8.11. Инструкция T_RESET (Разорвать и восстановить существующее соединение)

Инструкция "T_RESET" разрывает и затем восстанавливает существующее соединение:

Таблица 11- 35 Инструкция T_RESET

LAD / FBD	SCL	Описание
	<pre>"T_RESET_DB"(req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, status:=_word_out_);</pre>	Используйте инструкцию T_RESET, чтобы разорвать и затем восстановить существующее соединение.

Локальные конечные точки соединения сохраняются. Они генерируются автоматически:

- Если соединение было сконфигурировано и загружено в ЦПУ.
- Если соединение было сгенерировано пользовательской программой, например, вызовом инструкции "TCON (стр. 664)".

Инструкция "T_RESET" может быть выполнена для всех типов соединения независимо от того, использовались ли для соединения локальный интерфейс ЦПУ или интерфейс CM/CP. Исключение составляют соединения для передачи данных в ситуативном режиме с TCP, так как на подобные соединения нельзя сослаться с помощью ID соединения.

Как только инструкция "T_RESET" была вызвана, используя параметр REQ, соединение, определенное с помощью параметра ID, разрывается и, при необходимости, данные буферов передачи и приема удаляются. Отмена соединения также отменяет любую происходящую передачу данных. Поэтому существует риск потери данных, если происходит их передача. ЦПУ, определенный как активный партнер по соединению, в этом случае автоматически попытается восстановить прерванное коммуникационное соединение. Поэтому Вы не должны вызывать инструкцию "TCON (стр. 664)", чтобы восстановить коммуникационное соединение.

Выходные параметры DONE, BUSY и STATUS указывают на состояние задания.

Типы данных для параметров

В следующей таблице представлены параметры инструкции "T_RESET":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L, T, C или константа	Управляющий параметр REQUEST запускает задание на разрыв соединения, определенного в ID. Задание запускается по нарастающему фронту.
ID	Input	CONN_OUC (WORD)	L, D или константа	Ссылка на соединение с пассивным партнером, которое необходимо разорвать. ID должен быть идентичен соответствующему параметру ID в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
DONE	Output	BOOL	I, Q, M, D, L	Параметр состояния DONE 0: Задание еще не запускалось или все еще выполняется. 1: Задание выполнено без ошибок.
BUSY	Output	BOOL	I, Q, M, D, L	Параметр состояния BUSY 0: Задание завершено. 1: Задание еще не завершено.
ERROR	Output	BOOL	I, Q, M, D, L	Параметр состояния ERROR 0: Ошибок не было. 1: Во время обработки произошла ошибка. Параметр STATUS предоставляет подробную информацию о типе ошибки.
STATUS	Output	WORD	I, Q, M, D, L	Параметр состояния STATUS Информация об ошибке (смотри таблицу "Параметр STATUS").

Параметр STATUS

Бит Error	STATUS* (W#16#...)	Описание
0	0000	Ошибок нет.
0	0001	Соединение не было установлено.
0	7001	Разрыв соединения запущен.
0	7002	Соединение разрывается.
1	8081	Неизвестное соединение определено в параметре ID.

11.2.8.12. Инструкция T_DIAG (Проверяет состояние соединения и считывает информацию)

Инструкция "T_DIAG" проверяет состояние соединения и считывает дополнительную информацию о локальной конечной точке этого соединения:

Таблица 11- 36 Инструкция T_DIAG

LAD / FBD	SCL	Описание
	<pre>"T_DIAG_DB"(req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, status=>_dword_out_);</pre>	Используйте инструкцию T_DIAG, чтобы проверить состояние соединения и считать дополнительную информацию о локальной конечной точке этого соединения.

Инструкция "T_DIAG" работает следующим образом:

На соединение ссылается параметр ID. Вы можете считать обе конечных точки соединения, сконфигурированные в редакторе соединения, и запрограммированные конечные точки соединения (например, с помощью инструкции "TCON").

Конечные точки временного соединения (например, конечные точки, созданные, когда Вы соединяетесь с инженерной станцией), не могут быть диагностированы, поскольку никакой ID соединения не генерируется в этом случае.

- Прочитанная информации о соединении сохраняется в структуре, на которую ссылается параметр RESULT.
- Выходной параметр STATUS указывает на то, было ли возможно считать информацию о соединении. Информация о соединении из структуры в параметре RESULT действительная только, если инструкция "T_DIAG" была завершена со STATUS = W#16#0000 и ERROR = FALSE.

Информация о соединении не может быть оценена, если происходит ошибка.

Возможная информация о соединении

Структура "Tdiag_Status" может использоваться, чтобы считать информацию о соединении в параметре RESULT. Структура Tdiag_Status содержит только наиболее важную информацию о конечной точке соединения (например, используемый протокол, состояние соединения и число байтов данных, отправленных или принятых).

Строение и параметры структуры Tdiag_Status описаны ниже (смотри таблицу "Структура TDIAG_Status").

Типы данных для параметров

В следующей таблице представлены параметры инструкции "T_DIAG":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L, T, C или константа	Запускает инструкцию, чтобы проверить соединение, определенное в параметре ID, когда есть положительный фронт.
ID	Input	CONN_OUC (WORD)	L, D или константа	Ссылка на назначенное соединение. Диапазон значений: от W#16#0001 до W#16#0FFF
RESULT	InOut	VARIANT	D	Указатель на структуру, в которой сохранена информация о соединении. Структура TDiag_Status может использоваться в параметре RESULT (для описания, смотри таблицу "Структура TDIAG_Status").
DONE	Output	BOOL	I, Q, M, D, L	Параметр состояния: 0: Инструкция еще не запускалась или все еще выполняется. 1: Инструкция выполнена без ошибок.
BUSY	Output	BOOL	I, Q, M, D, L	Параметр состояния: 0: Инструкция еще не запущена или уже завершена. 1: Инструкция еще не завершена. Новое задание не может быть запущено.
ERROR	Output	BOOL	I, Q, M, D, L	Параметр состояния: 0: Ошибок нет. 1: Произошла ошибка.
STATUS	Output	WORD	I, Q, M, D, L	Состояние инструкции

Параметры BUSY, DONE и ERROR

Вы можете проверить состояние выполнения инструкции "T_DIAG" с помощью параметров BUSY, DONE, ERROR и STATUS. Параметр BUSY указывает на состояние обработки. Вы используете параметр DONE, чтобы проверить, была ли инструкция выполнена успешно. Параметр ERROR установлен, если ошибки произошли во время выполнения "T_DIAG".

Следующая таблица показывает отношение между параметрами BUSY, DONE и ERROR:

BUSY	DONE	ERROR	Описание
1	-	-	Инструкция обрабатывается.
0	1	0	Инструкция была выполнена успешно. Данные в структуре, на которую ссылается RESULT, действительны только, если это так.
0	0	1	Инструкция завершена с ошибкой. Причина ошибки выведена в параметре STATUS.
0	0	0	Не было назначено новой инструкции.

Параметр STATUS

Следующая таблица показывает толкование значений параметра STATUS:

Бит Error	STATUS* (W#16#...)	Описание
0	0000	Инструкция "T_DIAG" была выполнена успешно. Данные в структуре, на которую ссылаются в параметре RESULT, могут быть оценены.
0	7000	Нет активной инструкции в обработке.
0	7001	Обработка инструкции запущена.
0	7002	Информации о соединении считывается (параметр REQ не важен).
1	8086	Значение параметра ID вне допустимого диапазона (W#16#0001... W#16#0FFF).
1	8089	Параметр RESULT указывает на недопустимый тип данных (только структуры TDIAG_Status и TDIAG_StatusExt).
1	80A3	Параметр ID ссылается на конечную точку соединения, которая не существует. Эта ошибка может также произойти с запрограммированными соединениями после того, как вызывается инструкция "TDISCON".
1	80C4	Внутренняя ошибка. Доступ к конечной точке соединения временно невозможен.

Структура TDIAG_Status

Таблица ниже детализирует внутреннее устройство структуры TDIAG_Status. Значение каждого элемента действительно только, если инструкция была выполнена без ошибок. Если произойдет ошибка, то содержимое параметров не изменится:

Имя	Тип данных	Описание
Структура TDIAG_Status содержит следующие параметры:		
InterfacelD	HW_ANY	Идентификатор интерфейса (LADDR) ЦПУ или СМ/СР.
ID	CONN_OUC	Идентификатор диагностируемого соединения. После успешного вызова значение этого элемента идентично параметру ID инструкции "T_DIAG".
ConnectionType	BYTE	Тип протокола, используемый для соединения: • 0x01: Не используется. □ ... • 0x0B: TCP протокол (IP_v4) • 0x0C: ISO-оп-TCP протокол (RFC1006) • 0x0D: TCP протокол (DNS) • 0x0E: Dial-in протокол • 0x0F: WDC протокол • 0x10: SMTP протокол • 0x11: TCP протокол • 0x12: TCP и ISO-оп-TCP протокол (RFC1006) • 0x13: UDP протокол • 0x14: Зарезервировано • 0x15: Протокол доступа к шине PROFIBUS (FDL) • 0x16: ISO 8073 транспортный протокол (чистый ISO) □ ... • 0x20: SMTP или SMTPS протокол – на основе IPv4 • 0x21: SMTP или SMTPS протокол - на основе IPv6 • 0x22: SMTP или SMTPS протокол - на основе FQDN (Fully Qualified Domain Name) □ ... • 0x70: S7 соединения • Другое: Зарезервировано
ActiveEstablished	BOOL	• FALSE: Локально, пассивная конечная точка соединения • TRUE: Локально, активная конечная точка соединения

Имя	Тип данных	Описание
State	BYTE	Текущий статус конечной точки соединения • 0x00: Не используется. • 0x01: Соединение разорвано. Временное состояние, например, после вызова инструкции "T_RESET". Затем система автоматически пытается восстановить соединение. • 0x02: Активная конечная точка пытается установить соединение с удаленным коммуникационным партнером. • 0x03: Пассивная конечная точка соединения ожидает установления соединения с удаленным коммуникационным партнером. • 0x04: Соединение установлено. • 0x05: Соединение прерывается. Это может происходить вследствие вызова инструкции "T_RESET" или "T_DISCON". Другие возможные причины - ошибки протокола и разрывы линии. • 0x06..0xFF: Не используется.
Kind	BYTE	Режим конечной точки соединения: • 0x00: Не используется. • 0x01: Сконфигурировано, статическое соединение, которое было сконфигурировано и загружено в ЦПУ. • 0x02: Сконфигурировано, динамическое соединение, которое было сконфигурировано и загружено в ЦПУ (не поддерживается в настоящее время). • 0x03: Запрограммированное соединение, сгенерированное в пользовательской программе с помощью инструкции "TCON". Вызов инструкции "TDISCON" или переход ЦПУ в состояние STOP нарушил конечную точку соединения. • 0x04: Временное, динамическое соединение, установленное, например, инженерной станцией (ES) или станцией оператора (OS). (Этот тип подключения не может в настоящее время быть диагностирован, так как не имеет ID).0x05. • 0xFF: Не используется.
SentBytes	UDINT	Число переданных байтов данных.
ReceivedBytes	UDINT	Число принятых байтов данных.

11.2.8.13. Инструкция TMAIL_C (Послать электронное письмо, используя Ethernet-интерфейс ЦПУ)

Обзор

Вы используете инструкцию "TMAIL_C", чтобы послать электронное письмо, используя Ethernet-интерфейс ЦПУ S7-1200.

Инструкция TMAIL_C обладает двойным функционалом:

- Электронная почта через интерфейс ЦПУ (только SMTP без SSL)
- Электронная почта через интерфейс CP (либо SMTP без SSL, либо SMTP с SSL), Если Вы хотите использовать функциональность SSL, Вы должны установить входной параметр CERTINDEX = 1 для TMAIL_C и использовать интерфейс CP. Кроме того, корректный сертификат должен быть помещен в хранилище сертификатов CP.

Инструкция может использоваться только после того, как аппаратные средства были сконфигурированы, и, если сетевая инфраструктура допускает коммуникационное соединение с почтовым сервером.

Таблица 11- 37 Инструкция TMAIL_C

LAD / FBD	SCL	Описание
	<pre>"TMAIL_C_DB"(req:=_bool_in_, to_s:=_string_in_, cc:=_string_in_, sub- ject:=_string_in_, text:=_string_in_, attachment:=_variant_in_, attach- ment_name:=_string_in_, mail_addr_param:=_string_i n_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	Инструкция TMAIL_C посылает электронное письмо, используя Ethernet-интерфейс ЦПУ S7-1200.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Вы определяете содержание электронной почты и данные соединения, используя следующие параметры:

- Вы определяют адреса получателей с помощью параметров TO_S и CC.
- Вы определяют содержание электронной почты с помощью параметров SUBJECT и TEXT.
- Вы можете определить вложение, используя указатели VARIANT в параметрах ATTACHMENT и ATTACHMENT_NAME .
- Данные соединения определяются, а также адресация и аутентификация для почтового сервера выполняются, используя системный тип данных Tmail_v4 или Tmail_FQDN в параметре MAIL_ADDR_PARAM. Если Вы используете интерфейс ЦПУ S7-1200, то должен использоваться системный тип данных Tmail_v4. В этом случае электронное письмо может быть послано только с использованием SMTP.

- Вы запускаете отправку электронной почты по изменению сигнала из "0" в "1" в параметре REQ.
- Состояние задания обозначается выходными параметрами "BUSY", "DONE", "ERROR" и "STATUS".

Вы не можете отправить SMS непосредственно инструкцией "TMAIL_C". Электронная почта может быть в любом случае переслана почтовым сервером, однако посылка SMS зависит от Вашего телекоммуникационного провайдера.

Работа инструкции

Инструкция "TMAIL_C" работает асинхронно, то есть ее выполнение распространяется на несколько вызовов. Вы должны определить экземпляр, когда Вы вызываете инструкцию "TMAIL_C".

В следующих случаях будет потеряно соединение с почтовым сервером:

- Если ЦПУ переключается в STOP, в то время как "TMAIL_C" активна.
- Если коммуникационные проблемы возникают на шине Industrial Ethernet. В этом случае передача электронной почты будет прервана, и она не дойдет до получателя.

Соединение также отменяется, как только инструкция была успешно выполнена, и электронное письмо было отправлено.

ЗАМЕТКА

Изменение пользовательских программ

Вы можете изменить части своей пользовательской программы, которые непосредственно влияют на вызовы "TMAIL_C" только в том случае, если:

- ЦПУ находится в режиме "STOP".
- Никакое электронное письмо не посылается (REQ = 0 и BUSY = 0).

Это имеет отношение, в частности, к удалению и замене блоков программы, которые содержат вызовы "TMAIL_C", или вызовы экземпляров "TMAIL_C".

Игнорирование этого ограничения может связать коммуникационные ресурсы. Система автоматизации может перейти в неопределенное состояние с коммуникационными функциями TCP/IP через Industrial Ethernet.

"Теплый" или "холодный" перезапуск ЦПУ требуется после того, как внесены изменения.

Согласованность данных

Параметры TO_S, CC, SUBJECT, TEXT, ATTACHMENT и MAIL_ADDR_PARAM применяются инструкцией "TMAIL_C", во время ее работы, то есть они могут быть изменены только после того, как задание было завершено (BUSY = 0).

SMTP аутентификация

Под аутентификацией здесь понимается процедура проверки подлинности, например, с помощью запроса пароля.

Если Вы используете интерфейс ЦПУ S7-1200, инструкция "TMAIL_C" поддерживает процедуру SMTP аутентификации AUTH-LOGIN, которая требуется большинством почтовых серверов. Для получения информации о процедуре аутентификации для Вашего почтового сервера, смотри руководство по Вашему почтовому серверу или веб-сайт Вашего Интернет-провайдера.

- Прежде чем Вы сможете использовать процедуру аутентификации AUTH-LOGIN, инструкция "TMAIL_C" требует имя пользователя, с которым она должна войти в систему почтового сервера. Это имя соответствует пользователю, с которым Вы устанавливаете почтовую учетную запись на своем почтовом сервере. Оно передается через параметр UserName в параметр структуры MAIL_ADDR_PARAM. Если никакое имя пользователя не определено в параметре MAIL_ADDR_PARAM, процедура аутентификации AUTH-LOGIN не используется. Электронное письмо в этом случае отправляется без аутентификации.
- Чтобы войти в систему, инструкция "TMAIL_C" требует также связанного пароля. Он соответствует паролю, который Вы определили, когда создавали свою почтовую учетную запись. Он передается через параметр PassWord в параметр структуры MAIL_ADDR_PARAM.

Типы данных для параметров

В следующей таблице представлены параметры инструкции "TMAIL_C":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L, T, C или константа	Управляющий параметр REQUEST: Активирует отправку электронной почты по нарастающему фронту.
TO_S (стр. 694)	Input	STRING	D	Адрес получателя STRING с максимальной длиной 180 символов (байтов). Для формата адреса электронной почты смотрите пример в описании параметра.
CC (стр.694)	Input	STRING	D	Адреса получателей копии (дополнительно) STRING с максимальной длиной 180 символов (байтов). Тот же формат адреса электронной почты что и в параметре TO_S. Если здесь назначена пустая строка, копии электронного письма не посылаются.
SUBJECT	Input	STRING	D	Тема электронного письма STRING с максимальной длиной 180 символов (байтов).

Параметр	Объявление	Тип данных	Область памяти	Описание
TEXT	Input	STRING	D	Текст электронной почты (дополнительно) STRING с максимальной длиной 180 символов (байтов). Если пустая строка назначена этому параметру, электронное письмо посылается без текста.
ATTACHMENT	Input	VARIANT	D	Почтовое вложение (дополнительно) Ссылка на ячейку байт/слово/двойное слово (ArrayOfByte, ArrayOfWord или ArrayOfDWord) с максимальной длиной 64 байта. Если никакое значение не назначено, электронное письмо посылается без вложения.
ATTACHMENT_NAME	Input	VARIANT	D	Наименование почтового вложения (дополнительно) Ссылка на символьную строку с максимальной длиной 50 символов (байтов), чтобы определить имя файла вложения. Если пустая последовательность будет назначена этому параметру, то почтовое вложение будет отправлено с именем файла "attachment.bin".
MAIL_ADDR_PARAM (стр. 692)	Input	VARIANT	D	Параметр соединения и адрес почтового сервера Чтобы определить параметры соединения, используйте структуру Tmail_v4 или Tmail_FQDN (см. описание параметра).
DONE (стр. 694)	Output	BOOL	I, Q, M, D, L	Параметр состояния - DONE = 0: Задание еще не запущено или все еще выполняется. - DONE = 1: Задание было выполнено без ошибок.
BUSY (стр. 694)	Output	BOOL	I, Q, M, D, L	Параметр состояния - BUSY=0: Обработка "TMAIL_C" была остановлена. - BUSY = 1: Передача электронной почты еще не завершена.
ERROR (стр. 694)	Output	BOOL	I, Q, M, D, L	Параметр состояния - ERROR = 0: Ошибка не произошла. - ERROR = 1: Во время обработки произошла ошибка. STATUS предоставляет подробную информацию о типе ошибки.
STATUS (стр. 694)	Output	WORD	I, Q, M, D, L	Параметр состояния Возвращаемое значение или информация об ошибке инструкции "TMAIL_C" (см. описание параметра).

Вы найдете более подробную информацию о допустимых типах данных в разделе "Обзор допустимых типов данных".

Примечание

Необязательные параметры

Необязательные данные из CC, TEXT и ATTACHMENT отправляются с электронной почтой, только если соответствующие параметры содержат строку с длиной > 0.

Параметр MAIL_ADDR_PARAM

В параметре MAIL_ADDR_PARAM Вы определяете соединение для отправки электронной почты в структуре Tmail_v4 или Tmail_FQDN, и сохраняете адрес почтового сервера и данные для входа в систему.

Структура, которую Вы используете в параметре MAIL_ADDR_PARAM, зависит от формата, в котором должен адресоваться почтовый сервер:

- Tmail_v4: Адресация по IP-адресу (IPv4).
- Tmail_FQDN: Адресация по полностью определенному имени домена (FQDN).

Какая структура Вы можете использовать, зависит от интерфейса, указанного в параметре Interfaceld. Если Вы хотите использовать инструкцию "TMAIL_C" с внутренним интерфейсом, то в параметре MAIL_ADDR_PARAM должна использоваться структура Tmail_v4.

Таблица 11- 38 Tmail_v4: Адресация почтового сервера по IP-адресу (IPv4)

Параметр	Тип данных	Описание
Tmail_v4	Struct	
Interfaceld	LADDR	Аппаратный идентификатор интерфейса
ID	CONN_OUC	ID соединения
ConnectionType	BYTE	Тип соединения. Выберите 16#20 для IPv4.
ActiveEstablished	BOOL	Бит состояния. Принимает значение "1" после установления соединения.
CertIndex	BYTE	=0: используется SMTP (Simple Mail Transfer Protocol). SMTP должен использоваться, если электронное письмо посылается через интерфейс ЦПУ S7-1200.
WatchDogTime	TIME	Сторожевой таймер выполнения. Используйте этот параметр, чтобы определить максимальное время выполнения для операции отправки. Примечание: Установление соединения может занять больше времени (приблизительно одна минута), если соединение медленное. Когда Вы определяете параметр WATCH_DOG_TIME, не забудьте учесть время, требуемое на установление соединения. Соединение разрывается, как только установленное время истекло.
MailServerAddress	IP_v4	IP-адрес почтового сервера. IPv4 в следующем формате: XXX.XXX.XXX.XXX (десятичный формат). Пример: 192.142.131.237.
UserName	STRING[254]	Имя для входа на почтовый сервер
PassWord	STRING[254]	Пароль для входа на почтовый сервер

Параметр	Тип данных	Описание
From	EMAIL_ADDR	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Например: "myname@mymailserver.com".
Local-PartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая знак @. Пример: "myname@".
FullQualified-DomainName	STRING[254]	Полностью определенное доменное имя (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com".

Таблица 11- 39 Tmail_FQDN: Адресация почтового сервера по FQDN

Параметр	Тип данных	Описание
Tmail_v6	Struct	
Tmail_FQDN	LADDR	Аппаратный идентификатор интерфейса
ID	CONN_OUC	ID соединения
ConnectionType	BYTE	Тип соединения. Выберите 16#22 FQDN.
ActiveEstablished	BOOL	Бит состояния. Принимает значение "1" после установления соединения.
CertIndex	BYTE	=0: используется SMTP (Simple Mail Transfer Protocol). SMTP должен использоваться, если электронное письмо посылается через интерфейс ЦПУ S7-1200.
WatchDogTime	TIME	Сторожевой таймер выполнения. Используйте этот параметр, чтобы определить максимальное время выполнения для операции отправки. Примечание: Установление соединения может занять больше времени (приблизительно одна минута), если соединение медленное. Когда Вы определяете параметр WATCH_DOG_TIME, не забудьте учесть время, требуемое на установление соединения. Соединение разрывается, как только установленное время истекло.
MailServerAddress	STRING[254]	FQDN (Fully Qualified Domain Name) почтового сервера. Почтовый сервер адресуется, используя полностью определенное доменное имя. Пример: "www.mymailserver.com".
UserName	STRING[254]	Имя для входа на почтовый сервер
PassWord	STRING[254]	Пароль для входа на почтовый сервер
From	Struct	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Например: "myname@mymailserver.com".
Local-PartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая знак @. Пример: "myname@".
FullQualified-DomainName	STRING[254]	Полностью определенное доменное имя (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com".

Параметры TO_S и CC

Параметры TO_S и CC являются строками, например, со следующим содержанием:

- <wenna@mydomain.com>, <ruby@mydomain.com>
- <admin@mydomain.com>, judy@mydomain.com>

Обратите внимание на следующие правила при вводе параметров:

- Пробел и открытая угловая скобка "<" должна быть введены перед каждым адресом.
- Закрытая угловая скобка ">" должна быть введена после каждого адреса.
- Запятая должна быть введена между адресами в TO_S и CC.

По причинам снижения времени обработки и требований к памяти инструкция "TMAIL_C" не выполняет проверку синтаксиса параметра TO_S или CC

Параметры DONE, BUSY и ERROR

Выходные параметры DONE, BUSY и ERROR каждый устанавливаются только на один цикл, если состояние параметра выхода BUSY изменяется из "1" в "0".

Следующая таблица показывает отношение между DONE, BUSY и ERROR. Используя эту таблицу, Вы можете определить текущий статус инструкции "TMAIL_C и момент, когда отправка электронной почты завершена.

DONE	BUSY	ERROR	Описание
0	1	0	Задание обрабатывается.
1	0	0	Задание успешно завершено.
0	0	1	Задание завершено с ошибкой. Причина ошибки может быть считана в параметре STATUS (стр. 694).
0	0	0	Инструкция "TMAIL_C" не было назначено (новое) задание.

Параметр STATUS

Следующая таблица показывает возвращаемые значения TMAIL_C в параметре STATUS:

Возвращаемое значение STATUS* (W#16#...):	Пояснение	Примечания
0000	Обработка TMAIL_C была завершена без ошибок.	Безошибочное завершение TMAIL_C не означает, что посланное электронное письмо обязательно будет получено. Неправильный ввод адресов получателей не генерирует ошибку состояния инструкции TMAIL_C. В этом случае нет никакой гарантии, что электронное письмо будет послано другим получателям, даже если они были введены правильно.
7001	TMAIL_C активна (BUSY = 1).	Первый вызов: Задание активировано.
Возвращаемое значение STATUS* (W#16#...):	Пояснение	Примечания

щаемое значение STATUS* (W#16#...):		
7002	TMAIL_C активна (BUSY = 1).	Промежуточный вызов: Задание уже активно.
8xxx	Обработка TMAIL_C была завершена с кодом ошибки коммуникационных инструкций, вызванных внутри.	Для получения дальнейшей информации обратитесь к описаниям параметра STATUS для коммуникационных инструкций TCON, TDISCON, TSEND и TRCV (стр. 664).
8010	Ошибка во время установления соединения	Вы найдете дополнительную информацию об оценке в параметре SFB_STATUS экземплярного блока данных. Код ошибки, выводимый в параметре SFB_STATUS, разъяснен в описании параметра STATUS для инструкции TCON (стр. 664).
8011	Ошибка передачи данных	Вы найдете дополнительную информацию об оценке в параметре SFB_STATUS экземплярного блока данных. Код ошибки, выводимый в параметре SFB_STATUS, разъяснен в описании параметра STATUS для инструкции TSEND (стр. 664).
8012	Ошибка приема данных	Вы найдете дополнительную информацию об оценке в параметре SFB_STATUS экземплярного блока данных. Код ошибки, выводимый в параметре SFB_STATUS, разъяснен в описании параметра STATUS для инструкции TRCV (стр. 664).
8013	Ошибка во время установления соединения	Вы найдете дополнительную информацию об оценке в параметре SFB_STATUS экземплярного блока данных. Код ошибки, выводимый в параметре SFB_STATUS, разъяснен в описании параметра STATUS для инструкции TCON (стр. 664) и TDISCON (стр. 664).
8014	Установление соединения невозможно.	Вы, возможно, ввели неправильный IP-адрес почтового сервера (MailServerAddress (стр. 692)) или слишком малое время (WatchDogTime (стр. 692)) для установления соединения. Также возможно, что у ЦПУ нет соединения с сетью или конфигурация ЦПУ некорректна.
8015	Неправильный тип данных для MAIL_ADDR_PARAM	Единственными допустимыми типами данных являются системные типы данных (структуры) Tmail_v4 и TMail_FQDN.
8016	Неправильный тип данных для параметра ATTACHMENT	Единственными допустимыми типами данных являются ArrayOfByte, ArrayOfWord and ArrayOfDWord.
8017	Неправильный тип данных для параметра ATTACHMENT	Длина данных должна быть <= 65534 байтов.

Возвращаемое значение STATUS* (W#16#...):	Пояснение	Примечания
82xx, 84xx, или 85xx	Сообщение об ошибке происходит от почтового сервера и соответствует, за исключением "8", коду ошибки SMTP-протокола. В следующих строках приведены несколько кодов ошибки, которые могут иметь место.	Вы найдете более подробную информацию о коде ошибки SMTP и других кодах ошибок SMTP-протокола в Интернете или в документации по ошибкам почтового сервера. Вы можете также просмотреть новое сообщение об ошибке от почтового сервера в Вашем экземплярном DB в параметре BUFFER1. Вы найдете последние данные отправленными инструкцией TMAIL_C под DATEN в экземплярном DB.
8450	Действие не выполнено: Почтовый ящик недоступен / не может быть достигнут	Попытайтесь еще раз позднее.
8451	Действие прервано: Ошибка локальной обработки	Попытайтесь еще раз позднее.
8500	Синтаксическая ошибка: Ошибка нераспознана. Сюда также относят ошибку, когда командная строка слишком длинная. Это может также произойти, когда почтовый сервер не поддерживает процедуру аутентификации LOGIN.	Проверьте параметры TMAIL_C. Попытайтесь послать электронное письмо без аутентификации. Чтобы сделать это, замените содержание параметра UserName пустой строкой. Если имя пользователя не определено, процедура аутентификации LOGIN не используется.
8501	Синтаксическая ошибка: неправильное значение параметра	Возможная причина: Неправильный адрес в параметре TO_S или CC (см. также: параметры TO_S и CC (стр. 694)).
8502	Команда неизвестна или не реализуема	Проверьте введенные значения, в частности параметр FROM. Он может быть неполным, и Вы, возможно, забыли "@" или "." (см. также: Параметры TO_S и CC (стр. 694)).
8535	SMTP-аутентификация неполная	Вы возможно ввели неправильное имя пользователя или неправильный пароль.
8550	Почтовый сервер не может быть достигнут. У Вас нет прав доступа.	Вы, возможно, ввели неправильное имя пользователя или пароль, или почтовый сервер может не поддерживать Ваш вход в систему. Другой причиной могла бы быть ошибка в доменном имени после "@" в параметре TO_S или CC (см. также: Параметры TO_S и CC (стр. 694)).
8552	Действие прервано: Назначенный объем памяти был превышен	Попытайтесь еще раз позднее.
8554	Передача не удалась	Попытайтесь еще раз позднее.
* Вы можете отобразить коды ошибки в программном редакторе как целые или как шестнадцатеричные значения.		

11.2.8.14. UDP

UDP - стандартный протокол, описанный RFC 768: Протокол пользовательских дейтаграмм. UDP обеспечивает механизм для одного приложения, чтобы отправить дейтаграмму другому приложению; однако, доставка данных не гарантируется. Этот протокол обладает следующими функциями:

- Быстрый коммуникационный протокол, так как очень близок к аппаратным средствам
- Подходит для данных от небольшого до среднего объема (до 1472 байтов)
- UDP - более простой протокол управления передачей, чем TCP, тонкослойный, что приводит к малым издержкам
- Может использоваться очень гибко со многими сторонними системами
- Способен к маршрутизации
- Использует номера портов, чтобы направить дейтаграммы
- Сообщения не квитируются: требуется приложение, чтобы отвечать за восстановление после ошибки и безопасность
- Работа по программированию требуется для управления данными из-за интерфейса программирования SEND / RECEIVE

UDP поддерживает широковещательную передачу. Чтобы использовать широковещательную передачу, Вы должны сконфигурировать часть адреса IP конфигурации ADDR. Например: ЦПУ с IP-адресом 192.168.2.10 и маской подсети 255.255.255.0 использовал бы широковещательный адрес 192.168.2.255.

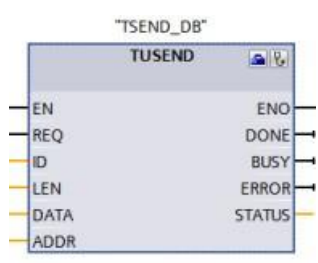
11.2.8.15. TUSEND и TURCV

Следующие инструкции управляют коммуникационным процессом UDP:

- TCON устанавливает обмен между клиентом и ПК-сервером (ЦПУ).
- TUSEND и TURCV отправляют и получают данные.
- TDISCON разрывает обмен между клиентом и сервером.

Обратитесь к теме TCON, TDISCON, TSEND и TRCV (стр. 664) в разделе "TCP и ISO-on-TCP" для получения дополнительной информации о коммуникационных инструкциях TCON и TDISCON.

Таблица 11- 40 Инструкции TUSEND и TURCV

LAD / FBD	SCL	Описание
	<pre>"TUSEND_DB"(req:=_bool_in_, ID:=_word_in_, len:=_udint_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, data:=_variant_inout_);</pre>	Инструкция TUSEND отправляет данные через UDP удаленному партнеру, определенному параметром ADDR . Чтобы запустить задание на отправку данных, вызовите инструкцию TUSEND с REQ = 1.
	<pre>"TURCV_DB"(en_r:=_bool_in_, ID:=_word_in_, len:=_udint_in_, ndr=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, rcvd_len=>_udint_out_, data:=_variant_inout_);</pre>	Инструкция TURCV получает данные через UDP. Параметр ADDR показывает адрес отправителя. После успешного завершения TURCV параметр ADDR содержит адрес удаленного партнера по (отправителя). TURCV не поддерживает ситуативный режим. Чтобы запустить задание на получение данных, вызовите инструкцию TURCV с EN_R = 1.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

TCON, TDISCON, TUSEND и TURCV работают асинхронно, то есть обработка задания распространяется на несколько вызовов инструкции.

Таблица 11- 41 Типы данных для параметров TUSEND и TURCV

Параметр и тип		Тип данных	Описание
REQ (TUSEND)	IN	Bool	Запускает задание по нарастающему фронту. Данные передаются из области, определенной DATA и LEN.
EN_R (TURCV)	IN	Bool	0: ЦПУ не может принимать. 1: Позволяет ЦПУ принимать. Инструкция TURCV готова к приему, и задание на получение обрабатывается.
ID	IN	Word	Ссылка на связанное соединение между пользовательской программой и коммуникационным уровнем операционной системы. ID должен быть идентичен связанному параметру ID в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
LEN	IN	UDInt	Количество байтов, подлежащих отправке (TUSEND) или получению (TURCV): - По умолчанию = 0: параметр DATA определяет длину данных, подлежащих отправке или получению. - Иначе, диапазон значений: от 1 до 1472
DONE (TUSEND)	OUT	Bool	Параметр состояния DONE (TUSEND): 0: Задание еще не запущено или все еще выполняется. 1: Задание завершено без ошибок.
NDR (TURCV)	OUT	Bool	Параметр состояния NDR (TURCV): - NDR = 0: Задание еще не запущено или все еще выполняется. - NDR = 1: Задание успешно завершено.
BUSY	OUT	Bool	1: Задание еще не завершено. Новое задание не может быть активировано. 0: Задание завершено.
ERROR	OUT	Bool	Параметр состояния со следующими значениями: 0: Нет ошибок. 1: Во время обработки произошла ошибка. STATUS предоставляет подробную информацию о типе ошибки
STATUS	OUT	Word	Информация о статусе, включая информацию об ошибке. (Обратитесь к кодам состояния Error и Status в таблице ниже.)
RCVD_LEN	OUT	UDInt	Количество принятых байтов (TURCV)

Параметр и тип		Тип данных	Описание
DATA	IN_OUT	Variant	Адрес области передачи (TUSEND) или области приема (TURCV): • Таблица образа процесса по входам • Таблица образа процесса по выходам • Битовая память • Блок данных
ADDR	IN_OUT	Variant	Указатель на адрес получателя (для TUSEND) или отправителя (для TURCV) (например, P#DB100. byte DBX0.0 8). Указатель может сослаться на любую область памяти. Требуется следующая структура из 8 байтов: • Первые 4 байта содержат удаленный IP-адрес. • Следующие 2 байта определяют номер удаленного порта. • Последние 2 байта зарезервированы.

Состояние задания отображается в выходных параметрах BUSY и STATUS. STATUS соответствует выходному параметру RET_VAL асинхронно работающих инструкций.

Следующая таблица показывает отношения между BUSY, DONE (TUSEND), NDR (TURCV) и ERROR. Используя эту таблицу, Вы можете определить текущий статус инструкции (TUSEND или TURCV) или момент завершения процесса отправки (передачи) / приема.

Таблица 11- 42 Состояния параметров BUSY, DONE (TUSEND) / NDR (TURCV) и ERROR

BUSY	DONE / NDR	ERROR	Описание
TRUE	Не важно	Не важно	Задание обрабатывается.
FALSE	TRUE	FALSE	Задание было завершено успешно.
FALSE	FALSE	TRUE	Задание было завершено с ошибкой. Причина ошибки может быть найдена в параметре STATUS.
FALSE	FALSE	FALSE	Инструкции не было назначено (новое) задание.

¹ Вследствие асинхронной работы инструкций: Для TUSEND Вы должны сохранять данные в области передачи согласованными пока параметр DONE или ERROR не примет значение TRUE. Для TURCV данные в области приема являются согласованными только тогда, когда параметр NDR принимает значение TRUE.

Таблица 11- 27 Коды состояния Error и Status инструкций TUSEND и TURCV

ERROR	STATUS	Описание
0	0000	- Задание на отправку завершено без ошибок (TUSEND) - Новые данные получены: текущая длина принятых данных представлена в RCVD_LEN (TURCV).
0	7000	- Нет активного задания в обработке (TUSEND) - Блок не готов к приему (TURCV)
0	7001	- Запуск обработки задания, данные отправляются (TUSEND): Во время этой обработки операционная система получает доступ к данным в DATA области передачи). - Блок готов к приему, задание на получение было активировано (TURCV).
0	7002	- Последующее выполнение инструкции (REQ не важен), задание обрабатывается: Во время этой обработки операционная система получает доступ к данным в DATA области передачи. - Последующее выполнение инструкции, задание на получение обрабатывается : Во время этой обработки данные записываются в область приема. По этой причине ошибка могла бы привести к несогласованным данным в области приема.
1	8085	Параметр LEN больше, чем наибольшее разрешенное значение, имеет значение 0 (TUSEND) или Вы изменили значение параметра LEN или DATA с момента первого выполнения инструкции (TURCV).
1	8086	Параметр ID не находится в разрешенном диапазоне адресов.
1	8088	- Параметр LEN больше, чем область памяти (TUSEND) или область приема (TURCV), определенная в DATA. - Область приема слишком мала (TURCV).
1	8089	Параметр ADDR не указывает на блок данных.
1	80A1	Коммуникационная ошибка: - Указанное соединение между пользовательской программой и коммуникационным уровнем операционной системы еще не было установлено. - Указанное соединение между пользовательской программой и коммуникационным уровнем операционной системы в настоящее время завершается. Задание на отправку (TUSEND) или на получение (TURCV) по этому соединению невозможно. - Интерфейс повторно инициализируется.
1	80A4	IP-адрес удаленной конечной точки соединения неправильный; возможно, что он совпадает с локальным IP-адресом (TUSEND).
1	80B3	- Выбранный вариант протокола (параметр connection_type в описании соединения) не является UDP. Используйте инструкцию TSEND или TRCV. - Параметр ADDR: Недопустимые настройки для номера порта (TUSEND)
1	80C3	- Блок с этим ID уже обрабатывается в другом классе приоритета. - Внутреннее отсутствие ресурсов.
1	80C4	Временная коммуникационная ошибка: - Соединение между пользовательской программой и коммуникационным уровнем операционной системы не может быть установлено в данный момент времени (TUSEND). - Интерфейс принимает новые параметры (TUSEND). - Соединение в настоящее время повторно инициализируется (TURCV).

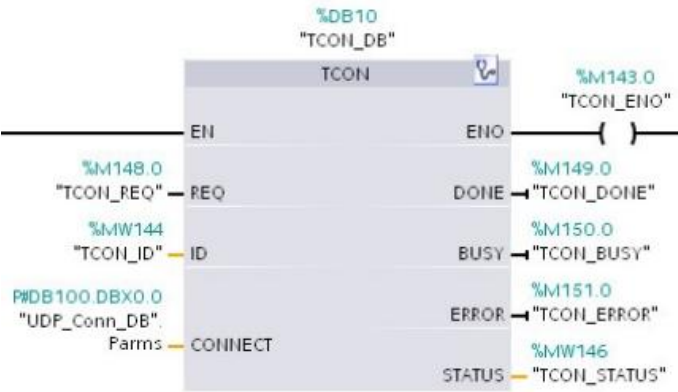
Протоколы Ethernet соединения

Каждый ЦПУ обладает интегрированным PROFINET-портом, который поддерживает стандартные PROFINET-коммуникации. Инструкции TUSEND и TURCV поддерживают UDP Ethernet-протокол.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

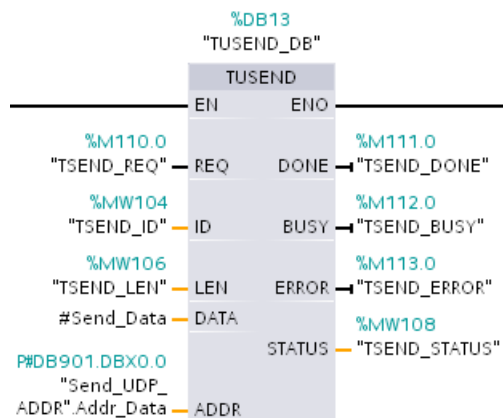
Операции

Оба партнера в UDP коммуникациях являются пассивными. Типовые стартовые значения параметров для типа данных "TCON_Param" представлены на последующих рисунках. Номера портов (LOCAL_TSAP_ID) записаны в 2-байтовом формате. Все порты за исключением 161, 34962, 34963, и 34964 разрешены.



UDP_Conn_DB					
	Name	Data type	Offset	Initial value	Comment
1	Static				
2	Params	TCON_Param	0.0		
3	BLOCK_LENGTH	UInt	0.0	64	byte length of SDT
4	ID	CONN_OUC	2.0	1	reference to the connection
5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP, 18: ISO on TCP
6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment
7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local IE interface
8	LOCAL_TSAP_ID_L...	USInt	7.0	2	byte length of local TSAP id/port number
9	REM_SUBNET_ID_...	USInt	8.0	0	byte length of remote subnet id
10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address
11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id
12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number
14	LOCAL_TSAP_I...	Byte		B#16#07	
15	LOCAL_TSAP_I...	Byte		B#16#D0	

Инструкция TUSEND отправляет данные через UDP удаленному партнеру, определенному в типе данных "TADDR_Param". Инструкция TURCV получает данные через UDP. После успешного выполнения инструкции TURCV тип данных "TADDR_Param" показывает адрес удаленного партнера (отправитель), как представлено на рисунках ниже.



Send_UDP_ADDR					
	Name	Data type	Offset	Initial value	Comment
1	Static				
2	Addr_Data	TADDR_Param	0.0		
3	REM_IP_ADDR	Array[1..4] of USInt	0.0		remote station address
4	REM_IP_ADDR[1]	USInt		192	
5	REM_IP_ADDR[2]	USInt		168	
6	REM_IP_ADDR[3]	USInt		2	
7	REM_IP_ADDR[4]	USInt		10	
8	REM_PORT_NR	UInt	4.0	2000	remote port number
9	RESERVED	Word	6.0	0	unused; has to be 0

11.2.8.16. T_CONFIG

Инструкция T_CONFIG изменяет параметры IP конфигурации PROFINET-порта из пользовательской программы, позволяя постоянное изменение или настройку следующих характеристик:

- Имя станции
- IP-адрес
- Маска подсети
- Адрес маршрутизатора

Примечание

Находящаяся в "свойствах" ЦПУ, страница "Ethernet address", радиокнопка "Set IP address using a different method" (стр. 710) позволяет Вам изменять IP-адрес он-лайн или при помощи инструкции "T_CONFIG" после того, как программа будет загружена. Этот метод назначения IP-адреса предназначен только для ЦПУ.

Находящаяся в "свойствах" ЦПУ, страница "Ethernet address" радиокнопка "Set PROFINET device name using a different method" (стр. 711) позволяет Вам изменять имя PROFINET-устройства он-лайн или при помощи инструкции "T_CONFIG" после того, как программа будет загружена. Этот метод присвоения имени PROFINET-устройства предназначен только для ЦПУ.



ПРЕДУПРЕЖДЕНИЕ

Изменение параметра IP-конфигурации с помощью T_CONFIG вызывает перезапуск ЦПУ.

После того, как Вы используете T_CONFIG, чтобы изменить параметр IP-конфигурации, ЦПУ перезапускается. ЦПУ переходит в режим STOP, выполняет "теплый" перезапуск и возвращается в режим RUN.

Не используйте инструкцию T_CONFIG в производственной среде. Управляющие устройства могут перестать работать в небезопасном состоянии, приводящем к неожиданной работе управляемого оборудования.

Такие неожиданные операции могли бы привести к смерти или серьезной травме персонала и/или повреждению оборудования.

Гарантируйте, что Ваш процесс перейдет в безопасное состояние, когда ЦПУ выполнит "теплый" перезапуск в результате выполнения инструкции T_CONFIG. "Теплый" перезапуск не очищает память. "Теплый" перезапуск инициализирует всю несохраняемые системные и пользовательские данные и поддерживает значения всех сохраняемых пользовательских данных.

Примечание

Вы не должны пытаться выполнить больше чем одну инструкцию T_CONFIG за один раз.

Таблица 11- 44 Инструкция T_CONFIG

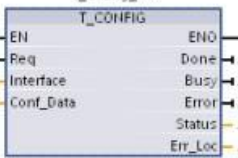
LAD / FBD	SCL	Описание
	<pre>"T_CONFIG_DB"(req:=_bool_in_, inter- face:=_word_in_, conf_Data:=_variant_in_, done=>_bool_out_, busy=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_dword_out_, err_loc=>_word_out_);</pre>	Используйте инструкцию T_CONFIG, чтобы изменить параметры IP-конфигурации из Вашей пользовательской программы. T_CONFIG работает асинхронно. Выполнение распространяется на несколько вызовов.

Таблица 11- 45 Типы данных для параметров инструкции T_CONFIG

Параметр и тип		Тип данных	Описание
REQ	Input	Bool	Запускает инструкцию по нарастающему фронту.
INTERFACE	Input	HW_Interface	ID сетевого интерфейса
CONF_DATA	Input	Variant	Ссылка на структуру данных конфигурации; CONF_DATA определяется System Data Type (SDT).
DONE	Output	Bool	0: Задание еще не запущено или все еще работает. 1: Задание было выполнено без ошибок.
BUSY	Output	Bool	0: Задание завершено. 1: Задание еще не завершено. Новое задание не может быть запущено.
ERROR	Output	Bool	Параметры состояния со следующими значениями: 0: Нет ошибок 1: Ошибка произошла во время обработки. STATUS предоставляет подробную информацию о типе ошибки.
STATUS	Output	DWord	Информация о состоянии, включая информацию об ошибке. (Обратитесь к кодам состояния Error и Status в таблице ниже.)
ERR_LOC	Output	DWord	Локализация отказа (ID поля и ID подполя параметра ошибки)

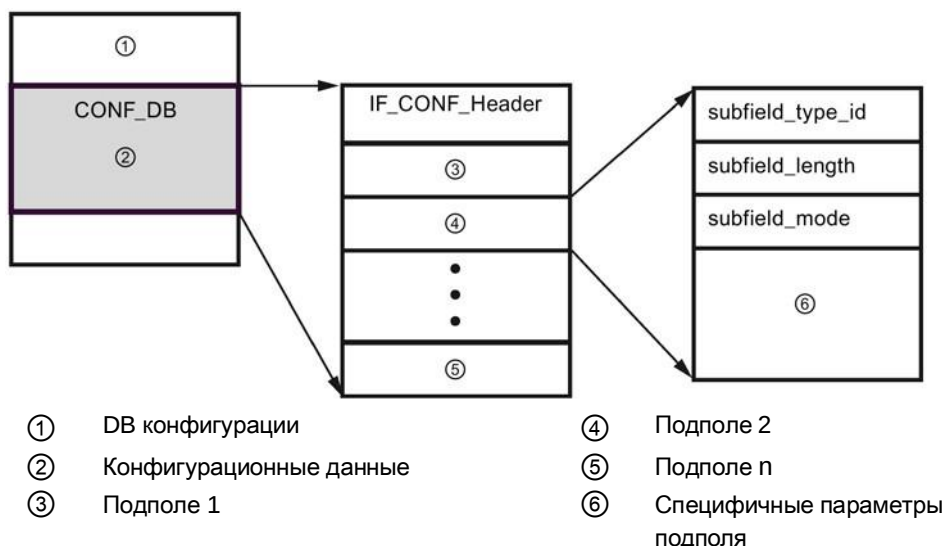
Информация об IP-конфигурации помещается в блок данных CONF_DATA, вместе с указателем Variant на параметр CONF_DATA, упомянутый выше. Успешное выполнение инструкции T_CONFIG заканчивается передачей данных IP-конфигурации сетевому интерфейсу. Ошибки назначаются выходному параметру STATUS.

Таблица 11- 46 Коды состояния для ERROR и STATUS

ERROR	STATUS (DW#16#...)	Описание
0	00000000	Без ошибок Примечание: Если инструкция выполняется успешно, состояние "без ошибок" не может быть возвращено.
0	00700000	Задание не закончено (BUSY = 1).
0	00700100	Запуск выполнения задания
0	00700200	Промежуточный вызов (REQ не важен)
1	C08хуу00	Общий отказ
1	C0808000	Параметры LADDR для идентификации интерфейса ошибочные.
1	C0808100	Параметры LADDR для идентификации интерфейса были назначены не поддерживаемому аппаратному интерфейсу.
1	C0808200	Ошибка параметра CONF_DATA: Тип данных указателя Variant не соответствует типу данных Byte.
1	C0808300	Ошибка параметра CONF_DATA: указатель области не находится в DB указателя Variant.
1	C0808400	Ошибка параметра CONF_DATA: Указатель Variant имеет неправильную длину.
1	C0808600	Зарезервировано
1	C0808700	Несоответствие длины блока данных CONF_DATA IP-конфигурации
1	C0808800	Параметры блока данных CONF_DATA field_type_id являются недопустимыми. (Допустим только field_type_id = 0.)
1	C0808900	Параметры блока данных CONF_DATA field_type_id являются недопустимыми или использовались несколько раз.
1	C0808A00	Ошибки длины LEN параметров IP-конфигурации или subfield_cnt
1	C0808B00	Параметр ID IP-конфигурации неправильный или неподдерживаемый.
1	C0808C00	Подблок IP-конфигурации неправильно размещен (Подблок неверный, неправильный порядок или используется многократно).
1	C0808D00	Длина подблоков оператора LEN неправильная.
1	C0808E00	Значение параметра в режиме подблоков неправильное.
1	C0808F00	Конфликт подблока между IP-конфигурацией и предыдущим подблоком.
1	C0809000	Параметры подполя защищены от записи (например: параметры определены конфигурацией, или активирован режим PNIO).
1	C0809100	Зарезервировано
1	C0809400	Параметр в подблоке IP-конфигурации не был определен или не может использоваться.
1	C0809500	Существует несоответствие между подблоком параметра IP-конфигурации и другими параметрами.
1	C080C200	Инструкция не может быть выполнена. Эта ошибка может произойти, если, например, связь с интерфейсом была потеряна.
1	C080C300	Недостаточно ресурсов. Эта ошибка может произойти, если, например, инструкцию вызывают многократно с различными параметрами
1	C080C400	Коммуникационный сбой. Ошибка может произойти временно и потребует повторения пользовательской программы.
1	C080D200	Выполнение инструкции не поддерживается интерфейсом PROFINET.

Блок данных CONF_DATA

Следующая схема показывает, как данные конфигурации, которые будут переданы, сохраняются в DB конфигурации.



Конфигурационные данные CONF_DB состоят из поля, которое содержит заголовок (IF_CONF_Header) и несколько подполей. IF_CONF_Header предоставляет следующие элементы:

- field_type_id (тип данных UInt): Ноль
- field_id (тип данных UInt): Ноль
- subfield_cnt (тип данных UInt): Количество подполей

Каждое подполе состоит из заголовка (subfield_type_id, subfield_length, subfield_mode) и специфичных для подполя параметров. Каждое подполе должно состоять из четного числа байтов. subfield_mode поддерживает значение 1.

Примечание

Только одно поле (IF_CONF_Header) разрешено на настоящий момент. Его параметры field_type_id и field_id должен иметь нулевое значение. Другие поля с отличными значениями для field_type_id и field_id зарезервированы под будущие расширения.

В поле IF_CONF_Header в настоящий момент разрешены только два подполя, "addr" (IP-адрес) и "nos" (Имя станции).

Таблица 11- 47 Поддерживаемые подполя

subfield_type_id	Тип данных	Описание
30	IF_CONF_V4	IP- параметры: IP-адрес, маска подсети, адрес маршрутизатора
40	IF_CONF_NOS	Имя PROFINET IO устройства (имя станции)

Таблица 11- 48 Элементы типа данных IF_CONF_V4

Имя	Тип данных	Стартовое значение	Описание
Id	UInt	30	subfield_type_id
len	UInt	18	subfield_length
mode	UInt	1	subfield_mode (1: постоянно)
InterfaceAddress	IP_V4	-	Адрес интерфейса
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte		
ADDR[2]	Byte		
ADDR[3]	Byte		
ADDR[4]	Byte		
SubnetMask	IP_V4	-	Маска подсети
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte		
ADDR[2]	Byte		
ADDR[3]	Byte		
ADDR[4]	Byte		
DefaultRouter	IP_V4	-	Маршрутизатор по умолчанию
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte		
ADDR[2]	Byte		
ADDR[3]	Byte		
ADDR[4]	Byte		

Таблица 11- 49 Элементы типа данных IF_CONF_NOS

Имя	Тип данных	Стартовое значение	Описание
id	UInt	40	subfield_type_id
len	UInt	246	subfield_length
mode	UInt	1	subfield_mode (1: постоянно)
Nos (имя станции)	Array[1..240] of Byte	0	Имя станции: Вы должны заполнять массив с первого байта. Если массив большей длины, чем имя станции, которое будет назначено, Вы должны ввести нулевой байт после фактического имени станции (согласно МЭК 61158-6-10). Иначе, nos отклоняется, и инструкция "T_CONFIG (стр. 704)" вводит код ошибки DW#16#C0809400 в STATUS. Если Вы занимаете нулем первый байт, то имя станции удаляется.

Имя станции подвержено следующим ограничениям:

- Компонент в имени станции, т.е., строка символов между двумя точками, не должен быть больше 63 символов.
- Никакие специальные символы, такие как умляуты, скобки, подчеркивание, наклонная черта, пробел, и т.д. Единственным разрешенным специальным символом является тире.
- Имя станции не должно начинаться или заканчиваться символом "-".
- Имя станции не должно начинаться с цифры.
- Формат имени станции n.n.n.n (n = 0... 999) не разрешен.
- Имя станции не должно начинаться с последовательности "port-xyz" или "port-xyz-abcde" (a, b, c, d, e, x, y, z = 0... 9).

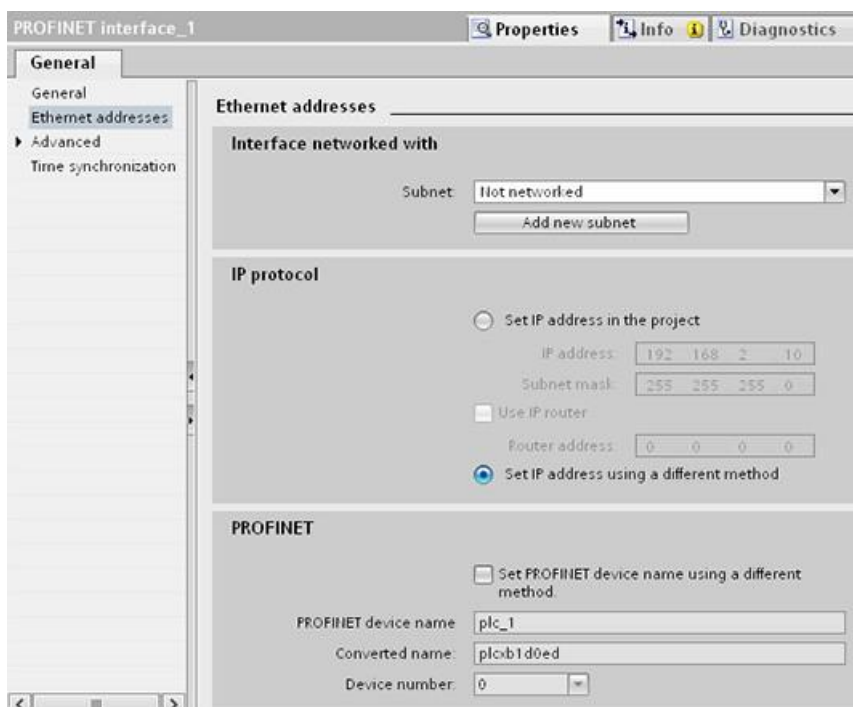
Примечание

Вы можете также создать массив "nos", который короче 240, но не меньше 2 байтов. В этом случае Вы должны соответственно скорректировать тег "len" (длина подполя).

Пример: Использование инструкции T_CONFIG для изменения IP-параметров

В следующем примере в подполе "addr" изменены "InterfaceAddress" (IP-адрес), "SubnetMask" и "DefaultRouter" (IP-маршрутизатор). В "свойствах" ЦПУ, страница "Ethernet address", нужно щелкнуть по радиокнопке "Set IP address using a different method", чтобы позволить изменение IP-адреса, используя инструкцию "T_CONFIG" после того, как программа будет загружена.

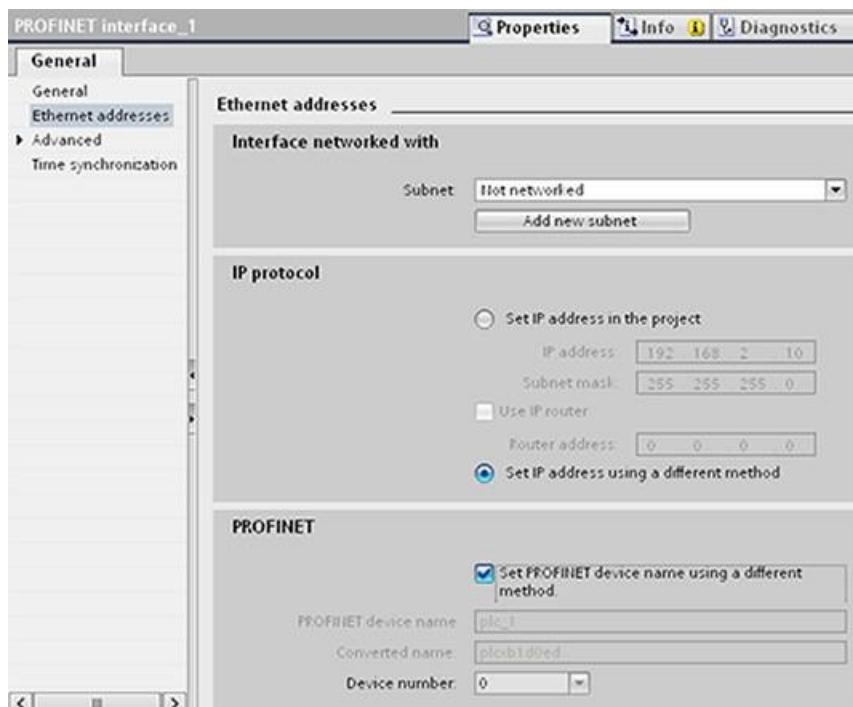
CONF_DATA_1			
	Name	Data type	Start value
1	▼ Static		
2	▼ Conf_data	Struct	
3	▼ header	IF_CONFIG_Header	
4	Fieldtype	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	▼ addr	IF_CONFIG_V4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	▼ InterfaceAddress	IP_V4	
12	▼ ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	Byte	168
15	ADDR[3]	Byte	2
16	ADDR[4]	Byte	30
17	▼ SubnetMask	IP_V4	
18	▼ ADDR	array [1..4] of Byte	
19	ADDR[1]	Byte	255
20	ADDR[2]	Byte	255
21	ADDR[3]	Byte	255
22	ADDR[4]	Byte	0
23	▼ DefaultRouter	IP_V4	
24	▼ ADDR	array [1..4] of Byte	
25	ADDR[1]	Byte	192
26	ADDR[2]	Byte	168
27	ADDR[3]	Byte	2
28	ADDR[4]	Byte	1



Пример: Использование инструкции T_CONFIG, чтобы изменить IP-параметры и имена PROFINET IO устройств

В следующем примере изменяются оба подполя "addr" и "nos" (имя станции). В "свойствах" ЦПУ, страница "Ethernet address", нужно щелкнуть по радиокнопке "Set PROFINET device name using a different method", чтобы позволить изменение имени PROFINET устройства, используя инструкцию "T_CONFIG" после того, как программа будет загружена.

CONF_DATA_2			
	Name	Data type	Start value
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	Fieldid	UInt	0
6	SubfieldCount	UInt	2
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	SubnetMask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_IOS	
18	Id	UInt	40
19	Length	UInt	246
20	Mode	UInt	1
21	IOS	array [1..240] of Byte	



11.2.8.17. Общие параметры для инструкций

Входной параметр REQ

Многие из инструкций Открытых пользовательских коммуникаций используют вход REQ, чтобы инициировать работу по переходу от низкого к высокому уровню. Вход REQ должен иметь высокий уровень (TRUE) на одно выполнение инструкции, но вход REQ может остаться в состоянии TRUE столько, сколько необходимо. Инструкция не инициирует другую операцию, пока она не будет исполнена со входом REQ в состоянии FALSE так, чтобы инструкция могла сбросить состояние истории входа REQ. Это требуется, чтобы инструкция смогла обнаружить переход от низкого уровня к высокому, чтобы инициировать следующую операцию.

Когда Вы помещаете одну из этих инструкций в Вашу программу, STEP 7 предлагает Вам идентифицировать экземплярный DB. Используйте уникальный DB для каждого вызова инструкции. Это гарантирует, что каждая инструкция должным образом обрабатывает входы, такие как REQ.

Входной параметр ID

Это является ссылкой на "Local ID (hex)" в представлении "Network view" для "Devices and networks" в STEP 7 и является ID сети, которую Вы хотите использовать для этого коммуникационного блока. ID должен совпадать со связанным параметром ID в локальном описании соединения.

Выходные параметры DONE, NDR, ERROR и STATUS

Эти инструкции обладают выходами, описывающими состояние завершения:

Таблица 11- 50 Выходные параметры инструкций открытых пользовательских коммуникаций

Параметр	Тип данных	По умолчанию	Описание
DONE	Bool	FALSE	Устанавливается в TRUE на одно выполнение, чтобы указать на то, что последний запрос завершился без ошибок
NDR	Bool	FALSE	Устанавливается в TRUE на одно выполнение, чтобы указать на то, что требуемое действие завершено без ошибок, и новые данные были получены
BUSY	Bool	FALSE	Устанавливается в TRUE, когда активен, чтобы указать на то, что: - Задание еще не завершено. - Новое задание не может быть активировано. Устанавливается в FALSE, когда задание завершено.
ERROR	Bool	FALSE	Устанавливается в TRUE на одно выполнение, чтобы указать на то, что последний запрос завершен с ошибками и действительным кодом ошибки в STATUS; иначе, FALSE..
STATUS	Word	0	Статус результата: - Если бит DONE или NDR установлен, то в STATUS записывается 0 или информационный код. - Если установлен бит ERROR, то в STATUS записывается код ошибки. - Если ни один из вышеупомянутых битов не установлен, то инструкция возвращает значения, которые описывают текущее состояние функции. STATUS сохраняет свое значение на время выполнения функции.

Примечание

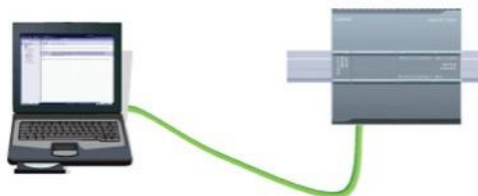
Обратите внимание на то, что DONE, NDR и ERROR устанавливаются только на од выполнение.

Запрещенные номера TSAP и портов для пассивных ISO и TCP коммуникаций

Если Вы используете инструкцию "TCON", чтобы настроить и установить пассивное коммуникационное соединение, следующие адреса портов запрещены и не должны использоваться:

- ISO TSAP (пассивное):
 - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
 - 10.00, 10.01, 11.00, 11.01, ... BF.00, BF.01
- TCP порт (пассивное): 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- UDP порт (пассивное): 161, 34962, 34963, 34964

11.2.9. Коммуникации с программатором



ЦПУ может обмениваться данными с находящимся в сети программатором со STEP 7.

При настройке коммуникаций между ЦПУ и программатором примите во внимание следующее:

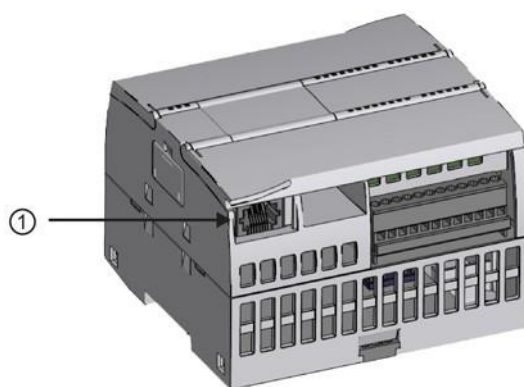
- Конфигурация/Установка: Требуется аппаратное конфигурирование.
- Ethernet-коммутатор не требуется для прямых коммуникаций; Ethernet-коммутатор требуется для более чем двух устройств в сети.

11.2.9.1. Установление аппаратного коммуникационного соединения

PROFINET-интерфейсы устанавливают физические соединения между программатором и ЦПУ. Так как автоперекрестная функциональность встроена в ЦПУ, и для интерфейса может использоваться стандартный или перекрестный Ethernet-кабель, то для того чтобы подключить программатор непосредственно к ЦПУ коммутатор не требуется.

Выполните шаги, представленные ниже, чтобы создать аппаратное соединение между программатором и ЦПУ:

1. Инсталлируйте ЦПУ (стр. 58).
2. Вставьте Ethernet-кабель в PROFINET-порт, показанный ниже.
3. Подключите Ethernet-кабель к программатору.



① PROFINET-порт

Дополнительный компенсатор натяжения доступен, чтобы повысить надежность PROFINET-соединения. Для получения информации о заказе смотрите раздел "Запасные элементы и другие аппаратные средства" (стр. 58).

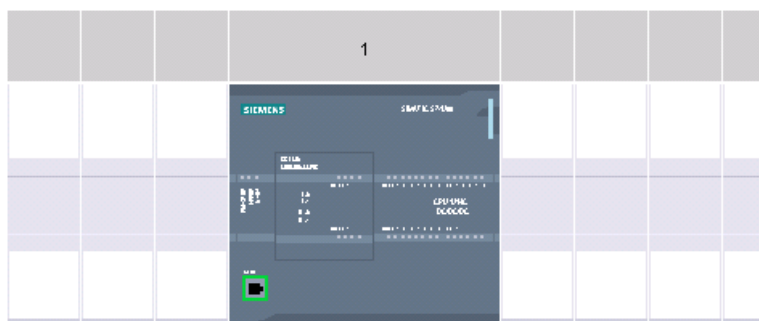
Смотри также

Запасные элементы и другие аппаратные средства (стр. 1282)

11.2.9.2. Конфигурирование устройств

Если Вы уже создали проект с ЦПУ, откройте свой проект в STEP 7.

В противном случае создайте проект и вставьте ЦПУ (стр.146) в стойку. В проекте ниже, ЦПУ показан в представлении "Device View".



11.2.9.3. Назначение адресов интернет протокола (IP)

Назначение IP-адресов

В сети PROFINET у каждого устройства должен также быть адрес Интернета Протокола (IP). Этот адрес позволяет устройству доставлять данные по более сложной, маршрутизируемой сети:

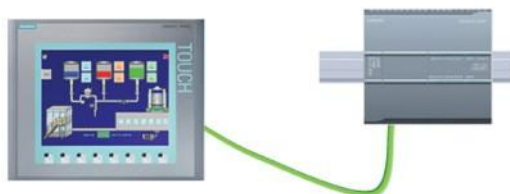
- Если у Вас есть программатор или другие сетевые устройства, которые используют встроенную карту адаптера, соединенную с LAN Вашего завода или Ethernet-USB адаптер, соединенной с изолированной сетью, Вы должны назначить им IP-адреса. Обратитесь к теме "Назначение IP-адресов программаторам и сетевым устройствам" (стр. 622) для получения дополнительной информации.
- Вы можете также назначить IP-адрес ЦПУ или сетевому устройству он-лайн. Это особенно полезно при начальном конфигурировании устройства. Обратитесь к теме "Назначение IP-адреса ЦПУ он-лайн" (стр. 625) для получения дополнительной информации.
- После того, как Вы сконфигурировали свой ЦПУ или сетевое устройство в Вашем проекте, Вы можете сконфигурировать параметры для PROFINET-интерфейса, чтобы включать его IP-адрес. Обратитесь к теме "Конфигурирование IP-адреса ЦПУ в Вашем проекте" (стр. 626) для получения дополнительной информации.

11.2.9.4. Тестирование Вашей PROFINET-сети

После завершения конфигурирования Вы должны загрузить свой проект в ЦПУ. Все IP-адреса конфигурируются, когда Вы загружаете проект.

Функция ЦПУ "Download to device" и ее диалоговое окно "Extended download to device" может показать все доступные сетевые устройства и то, были ли назначены уникальные IP-адреса всем устройствам. Обратитесь к теме "Тестирование PROFINET-сети" (стр. 629) для получения более подробной информации.

11.2.10. Коммуникации HMI-ПЛК



ЦПУ поддерживает коммуникационные соединения PROFINET с устройствами HMI (стр. 30). Следующие требования нужно принять во внимание при настройке коммуникаций между ЦПУ и HMI:

Конфигурация/Установка:

- PROFINET-порт ЦПУ должен быть сконфигурирован для обмена с HMI.
- HMI должен быть настроен и сконфигурирован.
- Информация о конфигурации HMI является частью проекта ЦПУ и может быть изменена и загружена из проекта.
- Ethernet-коммутатор не требуется для прямых коммуникаций; Ethernet-коммутатор требуется для более чем двух устройств в сети.

Примечание

Ethernet-коммутатор CSM1277 для монтажа в стойку с 4 портами может использоваться для соединения Ваших ЦПУ и HMI-устройства. PROFINET-порт ЦПУ не содержит Ethernet-коммутатора.

Поддерживаемые функции:

- HMI может читать /записывать данные в ЦПУ.
- Сообщения могут быть активированы на основании информации, полученной от ЦПУ.
- Системная диагностика

Таблица 11- 51 Требуемые шаги по конфигурированию соединения между HMI и ЦПУ

Шаг	Задача
1	Установление аппаратного коммуникационного соединения PROFINET-интерфейс устанавливает физическое соединение между HMI и ЦПУ. Так как автоперекрестная функциональность встроена в ЦПУ, Вы можете использовать как стандартный, так и перекрестный Ethernet-кабель для интерфейса. Ethernet-коммутатор не требуется, чтобы соединять HMI и ЦПУ. Обратитесь к теме "Коммуникации с программатором: Установление аппаратного коммуникационного соединения" (стр. 714) для получения дополнительной информации.
2	Конфигурирование устройств Обратитесь к теме "Коммуникации с программатором: Конфигурирование устройств" (стр. 715) для получения более подробной информации.
3	Конфигурирование логических сетевых соединений между HMI и ЦПУ Обратитесь к теме "Коммуникации HMI-ПЛК: Конфигурирование логических сетевых соединений между двумя устройствами" (стр. 717) для получения дополнительной информации.

Шаг	Задача
4	Конфигурирование IP-адреса в проекте Используйте ту же процедуру конфигурации; однако, Вы должны сконфигурировать IP-адреса для HMI и ЦПУ. Обратитесь к теме "Конфигурация устройства: Конфигурирование IP-адреса для ЦПУ в Вашем проекте" (стр. 626) для получения дополнительной информации.
5	Тестирование сети PROFINET Вы должны загрузить конфигурацию для каждого ЦПУ и HMI-устройства. Обратитесь к теме "Конфигурация устройства: Тестирование сети PROFINET" (стр. 629) для получения дополнительной информации.

11.2.10.1. Конфигурирование логических сетевых соединений между двумя устройствами

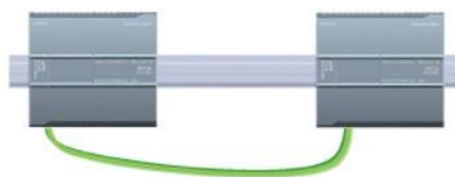
После конфигурирования стойки с ЦПУ, Вы теперь готовы сконфигурировать свои сетевые соединения.

В портале Devices and Networks используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Во-первых, щелкните по вкладке "Connections", а затем выберите тип подключения с помощью выпадающего списка, просто вправо (например, ISO on TCP соединение).

Чтобы создать PROFINET-соединение, щелкните по зеленому полю (PROFINET) на первом устройстве и перетащите линию на поле PROFINET на втором устройстве. Отпустите кнопку мыши, и Ваше PROFINET-соединение будет выполнено.

Обратитесь к теме "Конфигурация устройства: Создание сетевого соединения" (стр. 618) для получения более подробной информации.

11.2.11. Коммуникации ПЛК-ПЛК



ЦПУ может обмениваться данными с другим ЦПУ в сети при помощи инструкций TSEND_C и TRCV_C.

При настройке коммуникаций между двумя ЦПУ примите во внимание следующее:

- Конфигурация/Установка: Требуется аппаратное конфигурирование.
- Поддерживаемые функции: Чтение/запись данных в одноранговый ЦПУ
- Ethernet-коммутатор не требуется для прямых коммуникаций; Ethernet-коммутатор требуется для более чем двух устройств в сети.

у

Таблица 11- 52 Требуемые шаги по конфигурированию соединения между двумя ЦПУ

Шаг	Задача
1	Установление аппаратного коммуникационного соединения PROFINET-интерфейс устанавливает физическое соединение между двумя ЦПУ. Так как автоперекрестная функциональность встроена в ЦПУ, Вы можете использовать как стандартный, так и перекрестный Ethernet-кабель для интерфейса. Ethernet-коммутатор не требуется для соединения двух ЦПУ. Обратитесь к теме "Коммуникации с программатором: Установление аппаратного коммуникационного соединения" (стр. 714) для получения дополнительной информации.
2	Конфигурирование устройств Вы должны сконфигурировать два ЦПУ в своем проекте. Обратитесь к теме "Коммуникации с программатором: Конфигурирование устройств" (стр. 715) для получения дополнительной информации.
3	Конфигурирование логических сетевых соединений между двумя ЦПУ Обратитесь к теме "Коммуникации ПЛК-ПЛК: Конфигурирование логических сетевых соединений между двумя устройствами" (стр. 718) для получения дополнительной информации.
4	Конфигурирование IP-адреса в проекте Используйте ту же процедуру конфигурации; однако, Вы должны сконфигурировать IP-адреса для двух ЦПУ (например, PLC_1 и PLC_2). Обратитесь к теме "Конфигурация устройства: Конфигурирование IP-адреса для ЦПУ в Вашем проекте" (стр. 626) для получения дополнительной информации.
5	Конфигурирование параметров передачи (отправки) и приема Вы должны сконфигурировать инструкции TSEND_C и TRCV_C в обоих ЦПУ, чтобы позволить обмен данными между ними. Обратитесь к теме "Конфигурирование коммуникаций между двумя ЦПУ: Конфигурирование параметров передачи (отправки) и приема" (стр. 719) для получения дополнительной информации.
6	Тестирование сети PROFINET Вы должны загрузить конфигурацию для каждого ЦПУ. Обратитесь к теме "Конфигурация устройства: Тестирование сети PROFINET" (стр. 629) для получения дополнительной информации.

11.2.11.1. Конфигурирование логических сетевых соединений между двумя устройствами

После конфигурирования стойки с ЦПУ, Вы теперь готовы сконфигурировать свои сетевые соединения.

В портале Devices and Networks используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Во-первых, щелкните по вкладке "Connections", а затем выберите тип подключения с помощью выпадающего списка, просто вправо (например, ISO on TCP соединение).

Чтобы создать PROFINET-соединение, щелкните по зеленому полю (PROFINET) на первом устройстве и перетащите линию на поле PROFINET на втором устройстве. Отпустите кнопку мыши, и Ваше PROFINET-соединение будет выполнено.

Обратитесь к теме "Конфигурация устройства: Создание сетевого соединения" (стр. 618) для получения более подробной информации.

11.2.11.2. Конфигурирование локального/партнерского пути соединения между двумя устройствами

Конфигурирование основных параметров

Вы определяете параметры в диалоговом окне конфигурации свойств коммуникационной инструкции. Это диалоговое окно появляется рядом с нижней частью страницы каждый раз, когда Вы выбрали любую часть инструкции.

Обратитесь к теме "Конфигурация устройства: Конфигурирование локального / партнерского пути соединения (стр. 619)" для получения дополнительной информации.

В разделе "Address Details" диалогового окна параметров соединения Вы определяете используемые TSAP или порты. TSAP или порт соединения в ЦПУ вводится в поле "Local TSAP". TSAP или порт, назначенный соединению в Вашем партнерском ЦПУ вводится в поле "Partner TSAP".

11.2.11.3. Конфигурирование параметров передачи (отправки) и приема

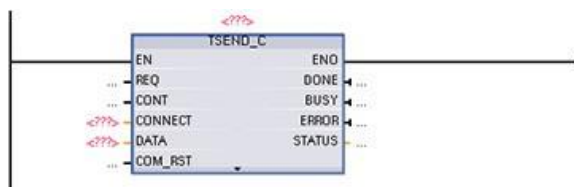
Коммуникационные блоки (например, TSEND_C и TRCV_C) используются, чтобы установить соединения между двумя ЦПУ. Прежде чем ЦПУ смогут участвовать в PROFINET-коммуникациях, Вы должны сконфигурировать параметры для передачи (или отправки) сообщения и приема сообщений. Эти параметры определяют то, как работают коммуникации, когда сообщения передаются в / принимаются от целевого устройства.

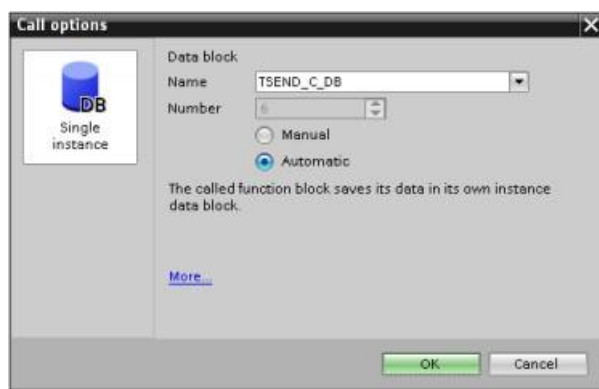
Конфигурирование параметров инструкции передачи (отправки) TSEND_C

Инструкция TSEND_C

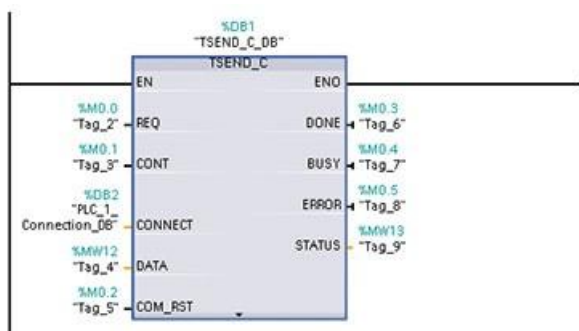
Инструкция TSEND_C (стр. 645) создает коммуникационное соединение со станцией партнера. Соединение настраивается, устанавливается и автоматически контролируется, пока не поступит команда на разъединение от инструкции. Инструкция TSEND_C комбинирует функции инструкций TCON, TDISCON и TSEND.

Из Device configuration в STEP 7 Вы можете сконфигурировать то, как инструкция TSEND_C передает данные. Чтобы начать, Вы вставляете инструкцию в программу из папки "Communications" в карте задач "Instructions". Инструкция TSEND_C отображается вместе с диалоговым окном Call options, где Вы назначаете DB для хранения параметров инструкции.





Вы можете назначить теги ячеек памяти входам и выходам, как это показано на следующем рисунке:



Конфигурирование основных параметров

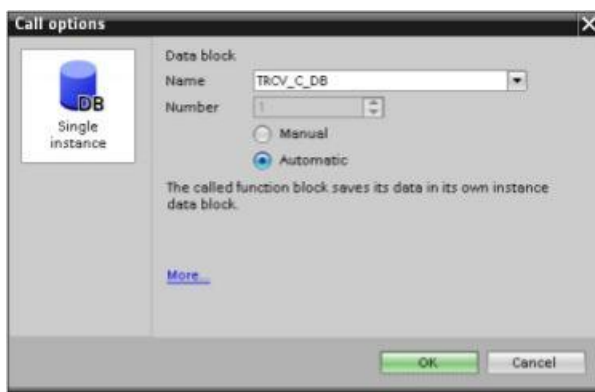
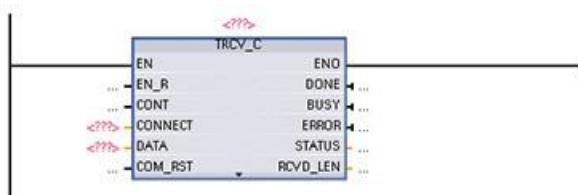
Вы определяете параметры в диалоговом окне конфигурации свойств инструкции TSEND_C. Это диалоговое окно появляется рядом с нижней частью страницы каждый раз, когда Вы выбрали любую часть инструкции TSEND_C.

Конфигурирование параметров приема инструкции TRCV_C

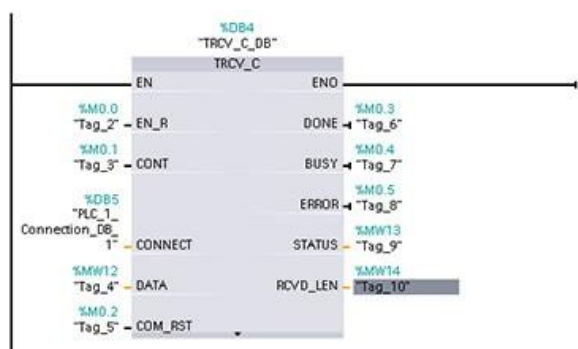
Инструкция TRCV_C

Инструкция TRCV_C (стр. 645) создает коммуникационное соединение со станцией партнера. Соединение настраивается, устанавливается и автоматически контролируется пока не поступит команда на разъединение от инструкции. Инструкция TSEND_C комбинирует функции инструкций TCON, TDISCON и TRCV.

Из Device configuration в STEP 7 Вы можете сконфигурировать то, как инструкция TRCV_C принимает данные. Чтобы начать, Вы вставляете инструкцию в программу из папки "Communications" в карте задач "Instructions". Инструкция TRCV_C отображается вместе с диалоговым окном Call options, где Вы назначаете DB для хранения параметров инструкции.



Вы можете назначить теги ячеек памяти входам и выходам, как это показано на следующем рисунке:



Конфигурирование основных параметров

Вы определяете параметры в диалоговом окне конфигурации свойств инструкции TRCV_C. Это диалоговое окно появляется рядом с нижней частью страницы каждый раз, когда Вы выбрали любую часть инструкции TRCV_C.

11.2.12. Конфигурирование ЦПУ и PROFINET IO устройства

11.2.12.1. Добавление PROFINET IO устройства

Добавление PROFINET IO устройства

В портале "Devices and networks", используйте аппаратный каталог, чтобы добавить PROFINET IO устройства.

Примечание

Чтобы добавить PROFINET IO устройство, Вы можете использовать STEP 7, Professional или Basic, V11 или выше.

Например, разверните следующие контейнеры в аппаратном каталоге, чтобы добавить IO устройство ET200S: Distributed I/O, ET200S, Interface modules и PROFINET. Вы можете затем выбрать интерфейсный модуль из списка ET200S устройств (отсортированный по номеру детали) и добавить IO устройство ET200S.

Таблица 11- 53 Добавление IO устройства ET200S в конфигурацию устройств

Вставить IO устройство	Результат
	

Вы можете теперь подключить PROFINET IO устройство к ЦПУ:

- Щелкните правой кнопкой по ссылке "Not assigned" на устройстве и выберите "Assign new IO controller" из контекстного меню, чтобы отобразить диалоговое окно "Select IO controller".
- Выберите Ваш ЦПУ S7-1200 (в настоящем примере, "PLC_1") из списка IO-контроллеров в проекте.
- Нажмите "ОК", чтобы создать сетевое соединение.

11.2.12.2. Конфигурирование логического сетевого соединения между ЦПУ и PROFINET IO устройством

Конфигурирование логического сетевого соединения

После конфигурирования стойки с ЦПУ, Вы теперь готовы сконфигурировать свои сетевые соединения.

В портале Devices and Networks используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Чтобы создать PROFINET-соединение, щелкните по зеленому полю (PROFINET) на первом устройстве и перетащите линию на поле PROFINET на втором устройстве. Отпустите кнопку мыши, и Ваше PROFINET-соединение будет выполнено.

Обратитесь к теме "Создание сетевого соединения" (стр. 618) для получения более подробной информации.

11.2.12.3. Назначение имен ЦПУ и устройству

Назначение имен ЦПУ и устройству

Сетевые соединения между устройствами также назначают PROFINET IO устройство ЦПУ, что необходимо для того, чтобы ЦПУ управлял устройством. Чтобы изменить это назначение, щелкните по PLC Name, отображаемому на PROFINET IO устройстве. Открывается диалоговое окно, которое позволяет отключить PROFINET IO устройство от текущего ЦПУ и переназначить его или оставить при необходимости неназначенным.

У устройств в Вашей PROFINET-сети должно быть назначенное имя, прежде чем Вы сможете соединиться с ЦПУ. Используйте представление "Network view", чтобы назначить имена Вашим PROFINET-устройствам, если устройствам уже не были назначены имена или если имя устройства должно быть изменено. Щелкните правой кнопкой по PROFINET IO устройству и выберите "Assign device name", чтобы сделать это.

Для каждого PROFINET IO устройства Вы должны назначить то же самое имя как в проекте STEP 7, так и, используя инструмент "Online & diagnostics", в памяти конфигурации PROFINET IO устройства (например, памяти конфигурации интерфейсного модуля ET200 S). Если имя будет отсутствовать или не будет соответствовать, то PROFINET IO режим обмена данными не будет работать. Обратитесь к теме "Он-лайн и диагностические инструменты: Назначение имени устройству PROFINET он-лайн (стр. 1070)" для получения дополнительной информации.

11.2.12.4. Назначение адресов интернет протокола (IP)

Назначение IP-адресов

В сети PROFINET у каждого устройства должен также быть адрес Интернета Протокола (IP). Этот адрес позволяет устройству доставлять данные по более сложной, маршрутизируемой сети:

- Если у Вас есть программатор или другие сетевые устройства, которые используют встроенную карту адаптера, соединенную с LAN Вашего завода или Ethernet-USB адаптер, соединенной с изолированной сетью, Вы должны назначить им IP-адреса. Обратитесь к теме "Назначение IP-адресов программаторам и сетевым устройствам" (стр. 622) для получения дополнительной информации.
- Вы можете также назначить IP-адрес ЦПУ или сетевому устройству он-лайн. Это особенно полезно при начальном конфигурировании устройства. Обратитесь к теме "Назначение IP-адреса ЦПУ он-лайн" (стр. 625) для получения дополнительной информации.
- После того, как Вы сконфигурировали свой ЦПУ или сетевое устройство в Вашем проекте, Вы можете сконфигурировать параметры для PROFINET-интерфейса, чтобы включать его IP-адрес. Обратитесь к теме "Конфигурирование IP-адреса ЦПУ в Вашем проекте" (стр. 626) для получения дополнительной информации.

11.2.12.5. Конфигурирование длительности IO цикла

Конфигурирование длительности IO цикла

PROFINET IO устройству предоставляются новые данные от ЦПУ в течение "IO цикла". Время обновления может быть отдельно сконфигурировано для каждого устройства и определяет временной интервал с которым данные передаются между ЦПУ и устройством.

STEP 7 рассчитывает время обновления "IO цикла" автоматически в настройке по умолчанию для каждого устройства сети PROFINET, принимая во внимание объем данных, по которым выполняется обмен, и число устройств, назначенных этому контроллеру. Если Вы не хотите вычислять время обновления автоматически, Вы можете изменить эти настройки.

Вы определяете параметры "IO цикла" в конфигурационном диалоге "Properties" PROFINET IO устройства. Это диалоговое окно появляется рядом с нижним краем страницы каждый раз, когда Вы выбрали любую часть инструкции.

В представлении "Device view" для PROFINET IO устройства щелкните по порту PROFINET. В "диалоговом окне" "PROFINET Interface" обратитесь к параметрам "IO цикла", выбирая следующие пункты меню:

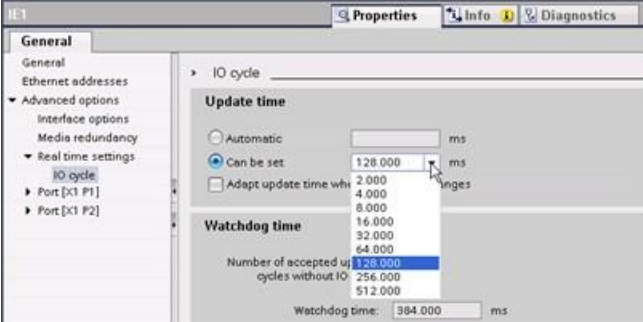
- "Advanced options"
- "Realtime settings"
- "IO cycle"

Определите "время обновления" IO цикла с помощью следующих настроек:

- Чтобы получить надлежащее время обновления, вычисленное автоматически, выберите "Automatic".
- Чтобы установить обновление самостоятельно, выберите "Can be set" и введите требуемое время обновления в мс.
- Чтобы гарантировать согласованность между синхрои импульсами передачи и вре-

менем обновления, активируйте опцию "Adapt update time when send clock changes". Эта опция гарантирует, что время обновления не меньше, чем период синхроимпульсов передачи.

Таблица 11- 54 Конфигурирование времени цикла PROFINET IO для ET200S

ET200 S PROFINET IO устройство	Диалоговое окно PROFINET IO цикла ET200S
	

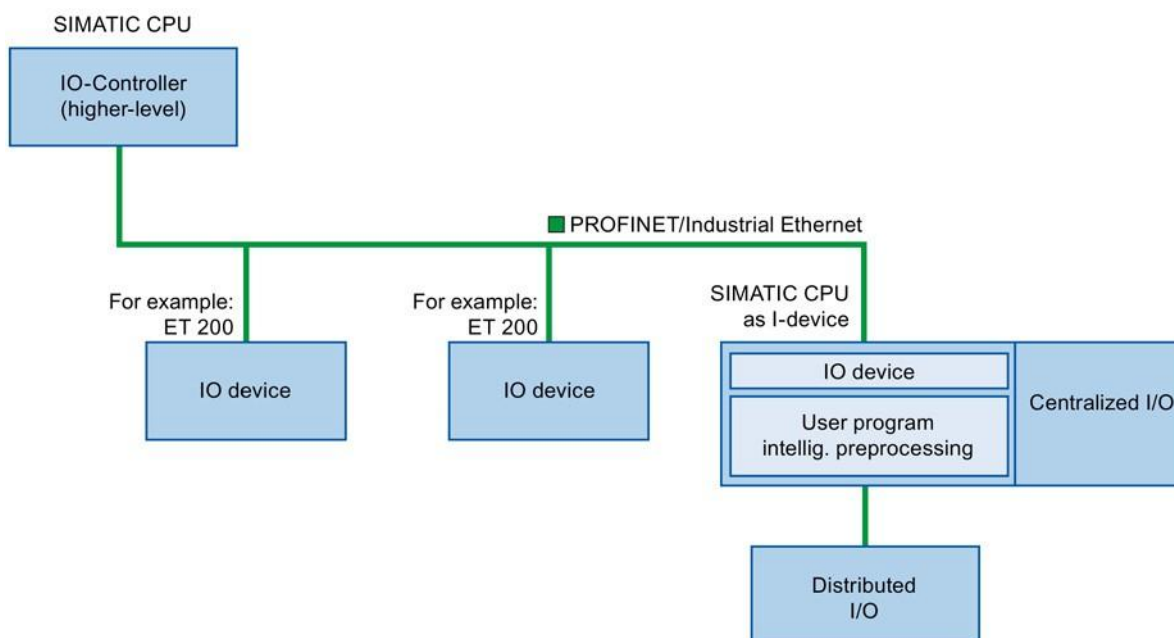
① PROFINET порт

11.2.13. Конфигурирование ЦПУ и PROFINET I-устройства

11.2.13.1. Функциональность I-устройства

Функциональные возможности ЦПУ, как "I-устройства" (интеллектуальное IO устройство) упрощает обмен данными с IO контроллером и работу ЦПУ в качестве, например, интеллектуального модуля предварительной обработки подпроцессов. I-устройство подключается как IO устройство к IO контроллеру "более высокого уровня".

Предварительная обработка выполняется пользовательской программой в ЦПУ. Значения процесса, полученные в централизованном или распределенном (PROFINET IO или PROFIBUS DP) вводе-выводе, предварительно обрабатываются пользовательской программой и становятся доступными посредством PROFINET IO интерфейса для ЦПУ станции более высокого уровня.



Соглашения об именовании "I-устройств"

В оставшейся части описания, ЦПУ или КП с функциональностью I-устройства называется просто "I-устройство".

11.2.13.2. Свойства и преимущества I-устройств

Области применения

Области применения I- устройств:

- **Распределенная обработка:**
Сложная задача автоматизации может быть разделена на меньшие модули / под-процессы. Это сводится к управляемым процессам, что приводит к упрощенным подзадачам.
- **Разделение подпроцессов:**
Сложные, широко распределенные и масштабные процессы могут быть поделены на несколько подпроцессов с управляемыми интерфейсами с помощью I- устройств. Эти подпроцессы могут быть сохранены, при необходимости, в отдельных STEP 7 проектах, которые могут позже быть объединены, чтобы образовать один основной проект.
- **Защита ноу-хау:**
Компоненты могут быть поставлены только с GSD-файлом для описания интерфейса I-устройства вместо проекта STEP 7. Пользователь может защитить свою программу, так как она больше не должна предоставляться.

Свойства

Свойства I-устройств:

- **Независимость от STEP 7 проектов:**
Создатели и пользователи I-устройств могут иметь полностью независимые проекты автоматизации STEP 7. GSD-файл формирует интерфейс между проектами STEP 7. Это позволяет организовать канал связи со стандартными IO-контроллерами через стандартизированный интерфейс.
- **Коммуникации в реальном времени:**
I-устройство оснащено детерминированной PROFINET IO системой посредством PROFINET IO-интерфейса и поэтому поддерживает RT (коммуникации в режиме реального времени) и IRT (изохронное реальное время).

Преимущества

I-устройство обладает следующими преимуществами:

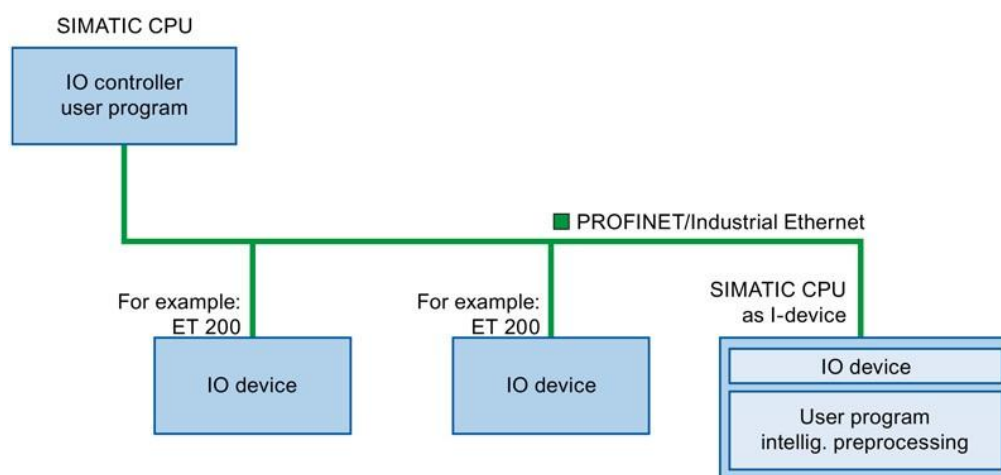
- Простое подключение к IO-контроллерам
- Коммуникации в режиме реального времени между IO-контроллерами
- Разгрузка IO-контроллера, путем распределения вычислительной способности по I-устройствам.
- Сниженные коммуникации загрузки, посредством локальной обработки данных процесса.
- Управляемость, из-за обработки подзадач в отдельных проектах STEP 7

11.2.13.3. Характеристики I-устройства

I-устройство встраивается в IO систему как стандартное IO устройство.

I-устройство без PROFINET IO системы более низкого уровня

У I-устройства нет своего собственного распределенного ввода-вывода. Конфигурирование и назначение параметров I-устройств в роли IO-устройств являются теми же, что и для распределенной системы ввода-вывода (например, ET 200).



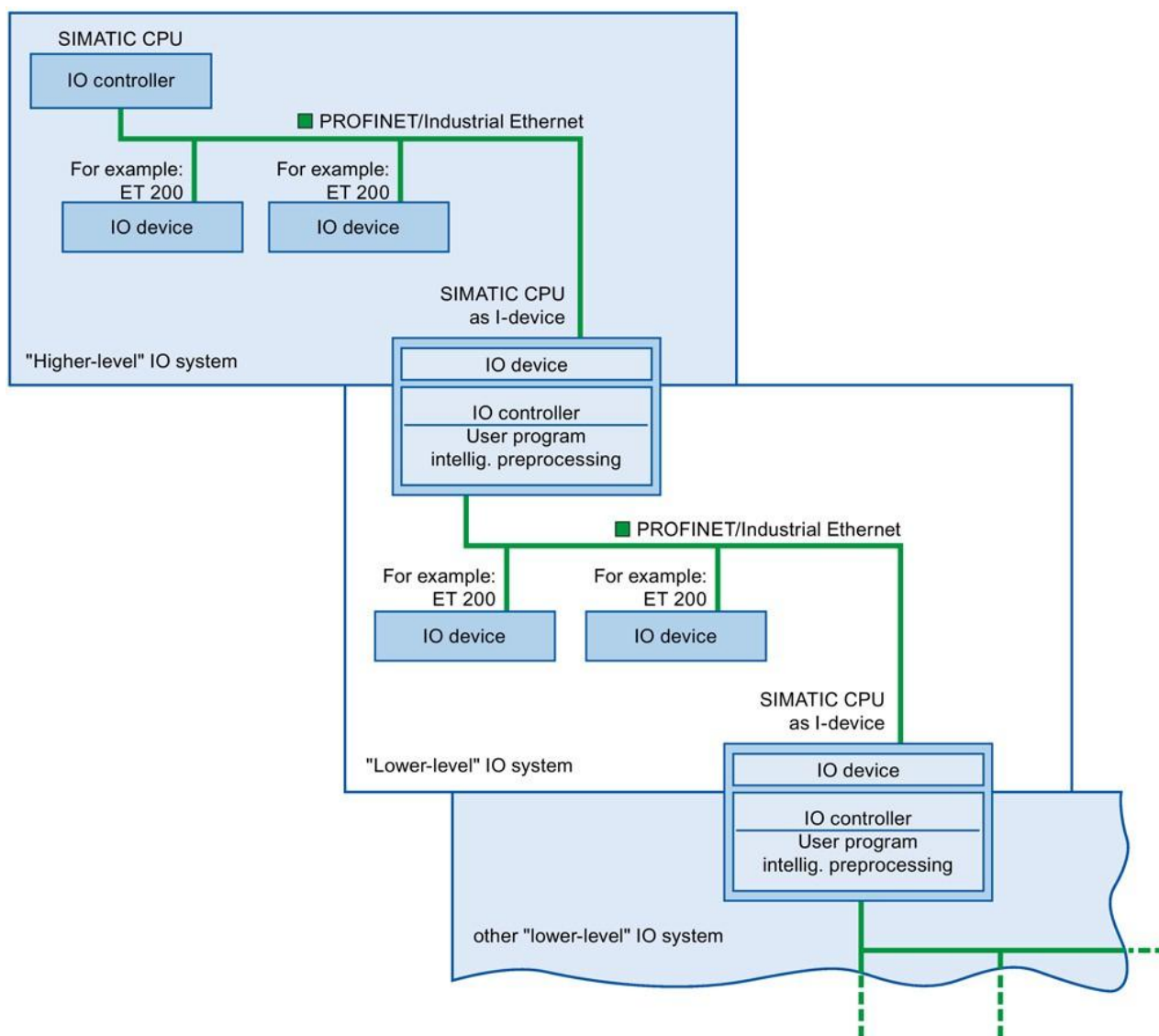
I-устройство с PROFINET IO системой более низкого уровня

В зависимости от конфигурации I-устройство может также быть IO-контроллером на PROFINET-интерфейсе в дополнение к роли IO-устройства.

Это означает, что I-устройство может быть частью IO-системы более высокого уровня посредством своего PROFINET-интерфейса и в качестве IO-контроллера может поддерживать свою собственную IO-систему более низкого уровня.

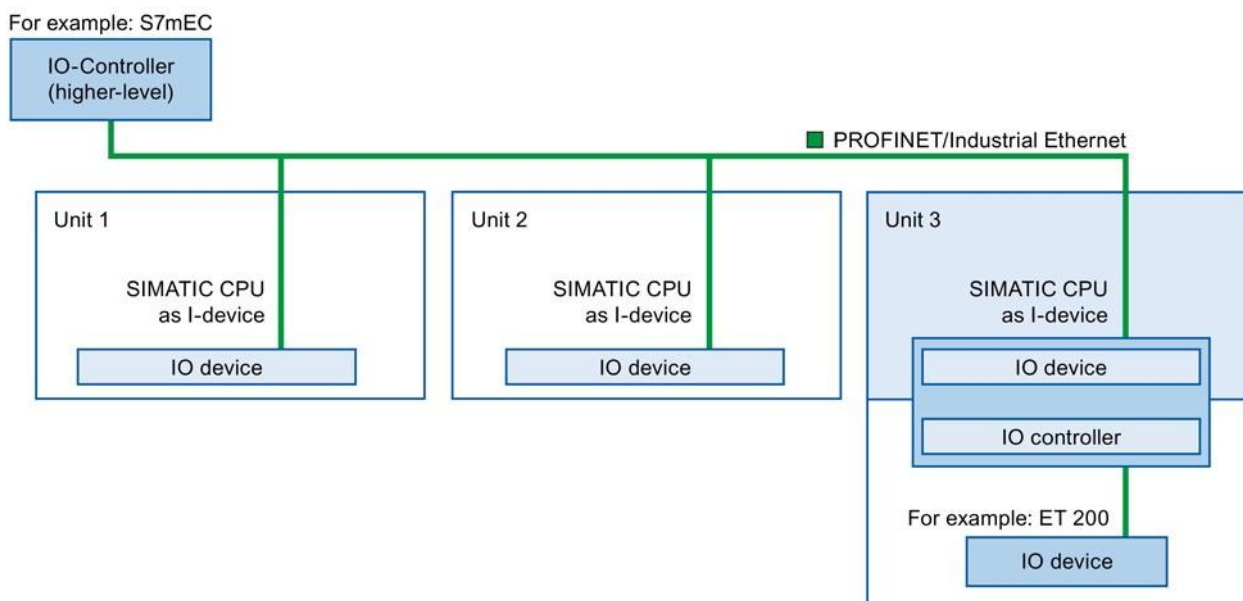
IO-система более низкого уровня может, в свою очередь, содержать I-устройства (см. рисунок ниже). Это делает возможным построение иерархически структурированных IO-систем.

В дополнение к его роли как IO-контроллера I-устройство может также использоваться через PROFIBUS-интерфейс в качестве ведущего устройства DP для системы PROFIBUS более низкого уровня.



Пример: I-устройство в качестве IO устройства и IO контроллера

Использование I-устройства в качестве IO устройства и IO контроллера объясняется на примере процесса печати. I-устройство управляет модулем (подпроцесс). Один модуль используется, например, чтобы вставить дополнительные листы, такие как флайеры или брошюры в пакет печатного материала.



Модуль 1 и модуль 2 каждый состоят из I-устройства с централизованным вводом-выводом. I-устройство вместе с распределенной системой ввода-вывода (например, ET 200) образуют модуль 3.

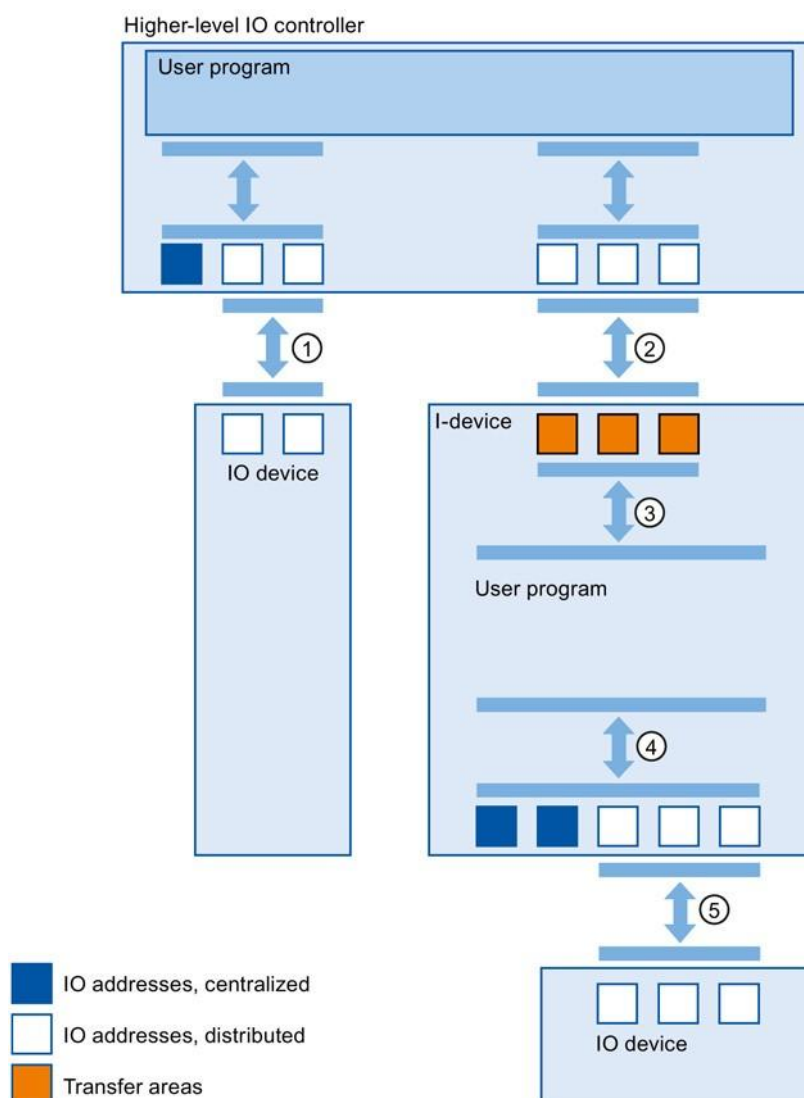
Пользовательская программа в I-устройстве ответственна за предварительную обработку данных процесса. Для этой задачи пользовательская программа I-устройства требует настроек по умолчанию (например, управляющая информация) от IO-контроллера верхнего уровня. I-устройство предоставляет IO-контроллеру верхнего уровня результаты (например, состояние его подзадачи).

11.2.13.4. Обмен данными между IO-системами верхнего и нижнего уровней

Областями передачи являются интерфейс к пользовательской программе ЦПУ I-устройства. Входы обрабатываются в пользовательской программе, а выходы представляют собой результат обработки в пользовательской программе.

Данные для обмена между IO-контроллером и I-устройством становятся доступными в областях передачи. Область передачи содержит информационный модуль, через который осуществляется последовательный обмен между IO-контроллером и I-устройством. Вы можете найти больше информации о конфигурации и использовании областей передачи в разделе "Конфигурирование I-устройства".

Следующий рисунок показывают обмен данными между IO-системами верхнего и нижнего уровней. Отдельные коммуникационные связи поясняются с помощью цифр:



- ① **Обмен данными между IO-контроллером верхнего уровня и обычным IO-устройством**
Таким образом IO-контроллер и IO-устройства обмениваются данными по PROFINET.
- ② **Обмен данными между IO-контроллером верхнего уровня и I-устройством**
Таким образом IO-контроллер и I-устройство обмениваются данными по PROFINET. Обмен данными между IO-контроллером верхнего уровня и I-устройством основан на стандартном взаимодействии IO-контроллер / IO-устройство.
Для IO-контроллера верхнего уровня области передачи I-устройств представляют субмодули предварительно сконфигурированной станции.
Выходные данные IO-контроллера являются входными данными I-устройства. Аналогично, входные данные IO-контроллера являются выходными данными I-устройства.
- ③ **Взаимодействие между пользовательской программой и областью передачи**
Таким образом, пользовательская программа и область передачи обмениваются входными и выходными данными.
- ④ **Обмен данными между пользовательской программой и вводом-выводом I-устройства**
Таким образом пользовательская программа и централизованный / распределенный ввод-вывод обмениваются входными и выходными данными.

- ⑤ **Обмен данными между I-устройством и IO-устройством нижнего уровня**
Таким образом I-устройство и его IO-устройства обмениваются данными. Передача данных осуществляется по PROFINET.

11.2.13.5. Конфигурирование I-устройства

Существует две возможности для конфигурирования:

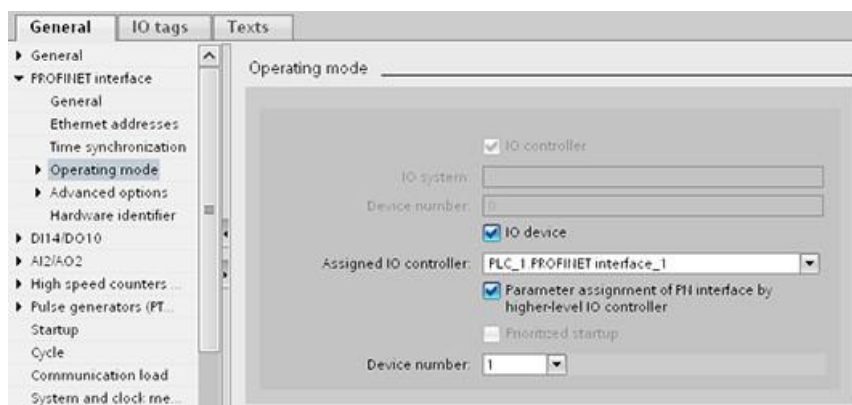
- Конфигурирование I-устройства в проекте
- Конфигурирование I-устройства, которое используется в другом проекте или в другой системе разработки.

STEP 7 позволяет Вам конфигурировать I-устройство для другого проекта или для другой системы разработки, экспортируя сконфигурированное I-устройство в файл GSD. Вы импортируете файл GSD в другие проекты или системы разработки, как и остальные GSD-файлы. Области для обмена данными, среди прочих данных, сохраняются в этом GSD-файле.

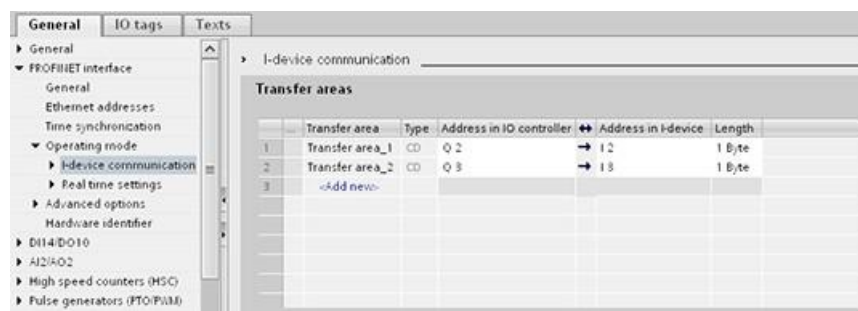
Конфигурирование I-устройства в проекте

1. Перетяните PROFINET ЦПУ из аппаратного каталога в сетевое представление.
2. Перетяните PROFINET ЦПУ, который может также быть сконфигурирован как IO устройство из аппаратного каталога в сетевое представление. Это устройство конфигурируется как I-устройство (например, ЦПУ 1215C).
3. Выберите PROFINET-интерфейс для I-устройства.
4. В окне инспектора в области навигации выберите "Operating mode" и установите флажок "IO device".
5. Теперь у Вас есть вариант выбора IO-контроллера в выпадающем списке "Assigned IO controller".

Как только Вы выбрали IO-контроллер, сетевые соединения и IO-система между обоими устройствами отображаются в сетевом представлении.



6. С помощью флажка "Parameter assignment of PN interface by higher-level IO controller" Вы определяете, будут ли интерфейсные параметры назначены самим I-устройством или IO-контроллером верхнего уровня.
Если Вы используете I-устройство с системой ввода-вывода нижнего уровня, то параметры PROFINET-интерфейса I-устройства (например, параметр порта) не могут быть назначены IO-контроллером верхнего уровня.
7. Сконфигурируйте области передачи. Области передачи находятся в области навигации в разделе "I-device communication".
 - Щелкните в первом поле столбца "Transfer area". STEP 7 назначает имя по умолчанию, которое Вы можете изменить.
 - Выберите тип коммуникационного отношения: Вы можете в настоящий момент выбрать только CD или F-CD.
 - Адреса задаются автоматически; Вы можете исправить адреса при необходимости, и определить размер области передачи, которая должна быть передана согласованно.



8. Отдельная запись создается в области навигации для каждой области передачи. Если Вы выбираете одну из этих записей, Вы можете скорректировать элементы области передачи, или исправить их и добавить комментарии к ним.

Конфигурирование I-устройства с помощью GSD-файла

Если Вы используете I-устройство в другом проекте, или если I-устройство используется в другой системе разработки, то сконфигурируйте IO-контроллер верхнего уровня и I-устройство, как описано выше.

При этом щелкните по кнопке "Export" после конфигурирования областей передачи, таким образом, из I-устройства создается новый GSD-файл. Этот GSD-файл представляет сконфигурированное I-устройство в других проектах.

Кнопка "Export" находится в разделе "I-device communication" окна инспектора.

Аппаратная конфигурация компилируется, и открывается диалоговое окно экспорта.

Назначьте имя для прокси I-устройства, а также описания в предоставленных полях. Нажмите кнопку "Export", чтобы завершить Ваш процесс.

В заключение импортируйте GSD-файл, например, в другом проекте.

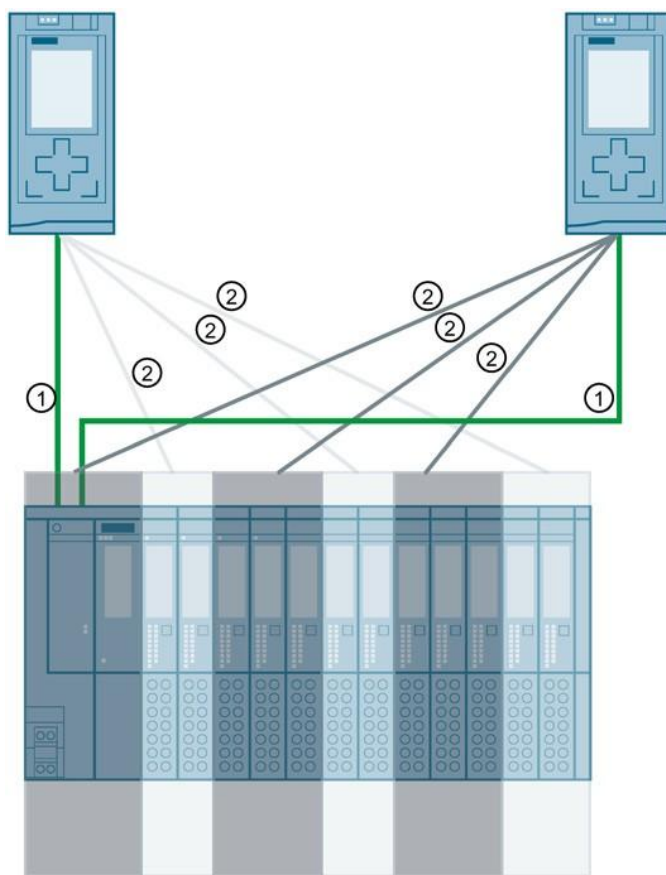
11.2.14. Устройства совместного использования

11.2.14.1. Функциональные возможности устройства совместного использования

Множество IO-контроллеров часто используются в больших или широко рассредоточенных системах.

Без функции "совместно используемого устройства" каждый модуль ввода-вывода IO-устройства назначен тому же IO-контроллеру. Если датчики, которые физически расположены близко друг к другу, должны предоставить данные различным IO-контроллерам, то требуется несколько IO-устройств.

Функция "совместно используемого устройства" позволяет модулям или субмодулям IO-устройства быть поделенными между различными IO-контроллерами. Это позволяет реализовать принцип гибкой автоматизации. У Вас есть, к примеру, возможность объединения модулей ввода-вывода, находящихся рядом в IO-устройстве.



- ① PROFINET
- ② Логическое назначение

Принцип

Доступ к субмодулям совместно используемого устройства в этом случае поделен между отдельными IO-контроллерами. Каждый субмодуль совместно используемого устройства назначен исключительно одному IO-контроллеру.

Необходимое условие (GSD конфигурация)

- STEP 7 V12 Service Pack 1 или выше
- ЦПУ с версией от FW 1.1 в качестве IO-контроллера
- IO-устройство поддерживает функцию совместно используемого устройства, например, интерфейсный модуль IM 155-5 PN ST
- GSD-файл для конфигурирования IO-устройства установлен
- ЦПУ S7-1200, сконфигурированный как I-устройство, поддерживает функциональность совместно используемого устройства. Вы должны экспортировать PROFINET GSD файл для I-устройства из STEP 7 (с V5.5) и затем импортировать его в STEP 7 (TIA Portal).

Конфигурирование доступа

IO-устройство должно присутствовать в нескольких проектах так, чтобы модули или submodule IO-устройства могли быть назначены различным IO-контроллерам. Отдельный проект требуется для каждого IO-контроллера.

Вы используете параметр "Shared device" интерфейсного модуля, чтобы определить модули или submodule, к которым IO-контроллер имеет доступ:

- Если локальный IO-контроллер обладает доступом к сконфигурированному модулю, выберите имя IO-контроллера из списка.
- Если IO-контроллер из другого проекта, а не локальный IO-контроллер должен обладать доступом к сконфигурированному модулю, выберите запись "---".

Конфигурация является согласованной относительно доступа, если каждый модуль или submodule назначен IO-контроллеру только в одном проекте.

Модуль или submodule назначены другому IO-контроллеру

Абзац ниже описывает следствия настройки "---" параметра "Shared device" с точки зрения локального IO-контроллера.

В этом случае у локального IO-контроллера нет доступа к модулю, сконфигурированному таким образом. В частности это означает, что:

- Отсутствие обмена данными с модулем или submodule
- Отсутствие приема аварийных или диагностических сигналов, что означает отсутствие отображения диагностики состояния в он-лайн представлении
- Невозможность назначения параметров модуля или submodule

Настройка свойств реального времени

STEP 7 вычисляет коммуникационную загрузку и вытекающее из этого время обновления. Вы должны ввести число внешних IO-контроллеров проекта, в котором PROFINET-интерфейс совместно используемого устройства назначен IO-контроллеру так, чтобы расчет был возможен с конфигурациями совместно используемых устройств.

Максимальное возможное число IO-контроллеров для совместно используемого устройства зависит от его типа. Это число сохранено в GSD-файле совместно используемого устройства.

Вы можете установить очень короткий период отправки часы с ЦПУ в качестве IO-контроллера. Период отправки может быть короче, чем наименьший период отправки, поддерживаемый совместно используемым устройством. В этом случае совместно используемое устройство управляется IO-контроллером с периодом отправки, который оно поддерживает (адаптация периода отправки).

Пример: ЦПУ поддерживает периоды отправки, начиная с 0.25 мс. Сконфигурированное IO-устройство также поддерживает период отправки, начиная с 0.25 мс; другое IO-устройство поддерживает период отправки, начиная с 1 мс. В этом случае у Вас есть возможность настройки для ЦПУ короткого периода отправки в 0.25 мс. ЦПУ управляет "медленным" IO-устройством с периодом отправки, например, в 1 мс.

Правила конфигурирования

- IO-контроллеры, которые задействуют совместно используемое устройство, создаются в различных проектах. В каждом проекте нужно уделить внимание тому, чтобы совместно используемое устройство было сконфигурировано идентично в каждой станции. Только один IO-контроллер может когда бы то ни было иметь полный доступ к субмодулю. Несоответствия в конфигурации приводят к отказу совместно используемого устройства.
- Адреса ввода-вывода модуля или субмодуля могут быть отредактированы только, если модуль или субмодуль назначены IO-контроллеру в том же проекте.
- У совместно используемого устройства должны быть одни и те же IP-параметры и имя устройства в каждом проекте.
- Период отправки должен быть идентичным для всех IO-контроллеров, которые обладают доступом к совместно используемому устройству.
- ID S7-подсети, к которой подключено совместно используемое устройство, должен быть идентичным во всех проектах.
- Следующие функции доступны только, если PROFINET-интерфейс совместно используемого устройства назначен локальному IO-контроллеру:
 - IRT режим
 - Перезапуск по приоритету
 - Назначение параметров свойств порта

Ограничения

Следующие ограничения имеют место, так как конфигурация с совместно используемым устройством распределена по нескольким проектам:

- Адреса модулей или субмодулей, которые не назначены этому IO-контроллеру, отсутствуют в обзоре адресов каждого IO-контроллера, у которого есть доступ к совместно используемому устройству.
- Модули или субмодули, которые не назначены, не учитываются при расчете предельной конфигурации для совместно используемого устройства во время проверки согласованности. Поэтому Вы должны самостоятельно убедиться, что максимальное количество субмодулей или максимальный объем циклических IO-данных для совместно используемого устройства не будут превышены. Для получения информации о максимальных значениях обратитесь к документации на устройства, которые Вы используете.
- Ошибки конфигурации, такие как назначение модуля или субмодуля нескольким IO-контроллерам не обнаруживаются в STEP 7.
- В ЦПУ, в которые загружены конфигурации с совместно используемым устройством, нет информации о том, является ли IO-устройство совместно используемым устройством. Модули или субмодули, которые назначены другим IO-контроллерам и, таким образом, другим ЦПУ, отсутствуют в загруженной конфигурации. Следовательно, данные модули или субмодули не отображаются ни в веб-сервере ЦПУ, ни на дисплее ЦПУ.

11.2.14.2. Пример: Конфигурирование совместно используемого устройства (GSD конфигурация)

Этот пример показывает, как сконфигурировать распределенную систему ввода-вывода в качестве совместно используемого устройства с помощью STEP 7 V13 SP1 или выше.

Возможна "распределенная" конфигурация с различными инженерными инструментами для различных семейств IO-контроллеров. Процедура, описанная ниже, основывается на STEP 7, начиная с V13 SP1 и ограничивается конфигурацией с двумя IO-контроллерами серий S7-1200, которые работают с совместно используемым устройством.

В примере создаются два проекта с одним IO-контроллером в каждом:

- Controller1
- Controller2

Вы должны создать совместно используемое устройство в обоих проектах, даже при том, что физически - это одно и то же IO-устройство.

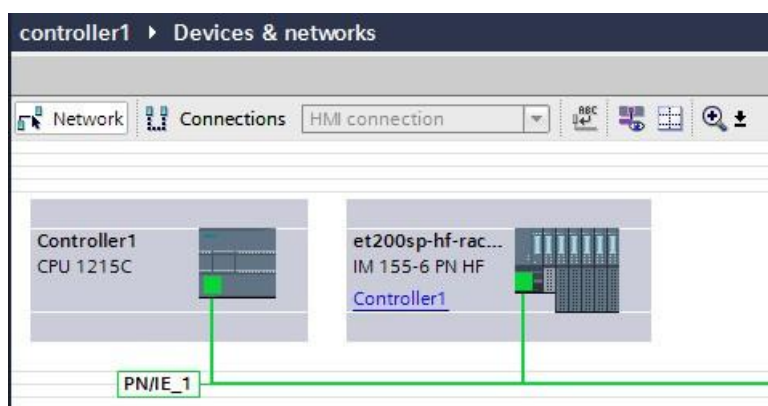
Требования

- STEP 7 V13 SP1 или выше
- IO-устройство поддерживает функциональность совместно используемого устройства (например, ET 200SP IM 155-6 PN HF V3.1).
- GSD-файл для конфигурирования IO-устройства в качестве совместно используемого устройства установлен.

Процедура: Создание проекта 1

Чтобы создать первый проект с совместно используемым устройством, выполните следующие шаги:

1. Запустите STEP 7.
2. Создайте новый проект с именем "Controller1".
3. Вставьте ЦПУ 1215C из аппаратного каталога в сетевом представлении. Назовите его "Controller1".
4. Вставьте IO-устройство с функцией "совместно используемого устройства" (например, ET 200SP) из аппаратного каталога (аппаратный каталог: Other field devices > PROFINET IO > I/O).
5. Назначьте IO-контроллер "Controller1" IO-устройству.

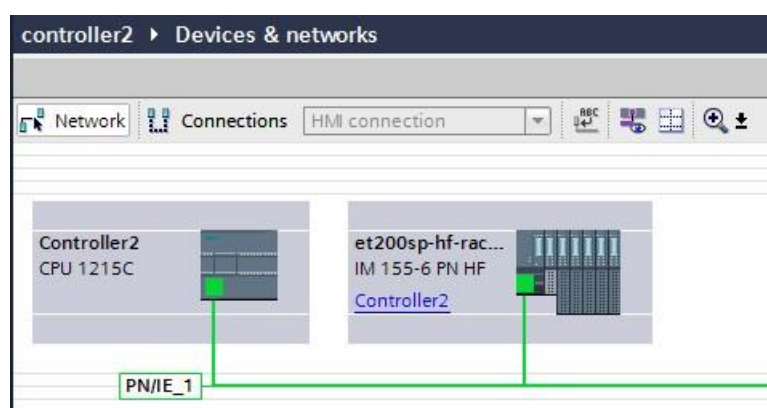


6. Дважды щелкните по IO-устройству и вставьте все требуемые модули и submodule из аппаратного каталога в таблице обзора устройств.
7. Назначьте параметры модулей.
8. Сохраните проект.

Процедура: Создание проекта 2

Чтобы создать второй проект с совместно используемым устройством, выполните следующие шаги:

1. Запустите STEP 7 еще раз
Открывается новый экземпляр STEP 7.
2. В новом экземпляре создайте новый проект с именем "Controller2".
3. Вставьте ЦПУ 1215C в сетевом представлении. Назовите его "Controller2".
4. Скопируйте IO-устройство из проекта "Controller1" и вставьте его в сетевое представление проекта "Controller2".
5. Назначьте IO-контроллер "Controller2" IO-устройству.



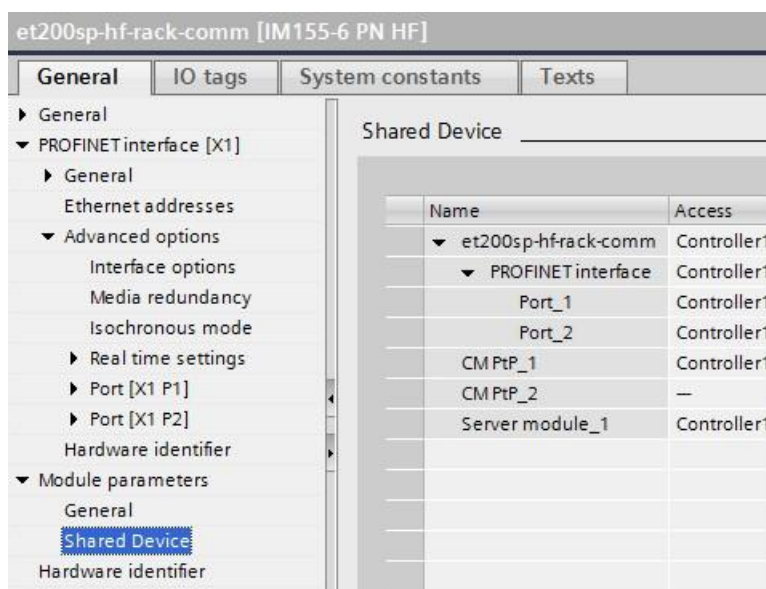
6. Сохраните проект.

В обоих проектах теперь есть тождественно структурированное IO-устройство, которое должно быть сконфигурировано на следующем шаге для других типов доступа от IO-контроллеров.

Процедура: Конфигурирование доступа к совместно используемому устройству

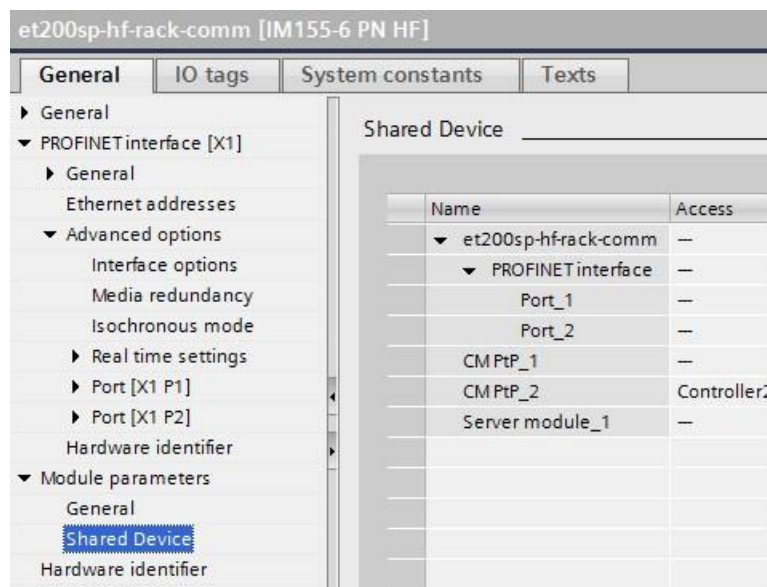
Модули и submodule, которые Вы вставляете в совместно используемое устройство, автоматически назначаются локальному ЦПУ. Чтобы изменить назначение, выполните следующие шаги:

1. Выберите интерфейсный модуль в сетевом представлении или представлении устройств для проекта "Controller1".
2. Выберите область "Shared Device" в окне инспектора.
Таблица показывает, у какого ЦПУ есть доступ к соответствующему модулю или submodule для всех сконфигурированных модулей. Настройкой по умолчанию является то, что у локального ЦПУ есть доступ ко всем модулям и submodule.
3. Сохраните настройку "Controller1" для всех модулей и submodule, которые должны остаться в диапазоне адресов локального ЦПУ.
Выберите настройку "---" для всех модулей и submodule, которые должны быть расположены в диапазоне адресов ЦПУ (Controller2) из проекта "Controller2". Это означает, что у IO-контроллера вне проекта должен быть доступ к модулю или submodule.



4. Выберите интерфейсный модуль в сетевом представлении или представлении устройств проекта "Controller2".
5. Выберите область "Shared Device" в окне инспектора.
Таблица показывает, у какого ЦПУ есть доступ к соответствующему модулю или submodule для всех сконфигурированных модулей.

6. Выберите настройку "---" для всех модулей и submodule, которые должны быть расположены в диапазоне адресов ЦПУ (Controller1) из проекта "Controller1".



7. В заключение, проверьте, "согласованны" ли настройки доступа для каждого модуля или submodule в обоих проектах. Это означает, что, если у локального ЦПУ есть доступ в одном проекте, то опция "---" должна быть установлена в другом проекте и наоборот.

Примечание: Опция "---" для PROFINET-интерфейса и, следовательно, для портов переводит связанные параметры в состояние только для чтения и делает их не изменяемыми. Параметры PROFINET-интерфейса и параметры порта могут быть отредактированы только в проекте, в котором PROFINET-интерфейс назначен локальному ЦПУ. Порты могут быть соединены в обоих проектах независимо от этого.

8. Проверьте, установлены ли одни и те же параметры IP-адреса и имя устройства для совместно используемого устройства во всех проектах.

Проверьте, установлен ли тот же ID S7-подсети во всех проектах для подсети, с которой соединено совместно используемое устройство (свойства подсети, область "General" в окне инспектора).

Примечание

Если Вы вносите изменения в совместно используемое устройство: Внесите одни те же изменения в каждом проекте. Удостоверьтесь, что только у одного IO-контроллера есть доступ к модулю или submodule.

Процедура: Корректировка параметров реального времени

Чтобы гарантировать, что все IO-контроллеры и совместно используемые устройства управляются надлежащими синхроимпульсами передачи и, что времена обновления правильно рассчитаны на основе коммуникационной загрузки, Вы должны скорректировать и проверить следующие настройки:

1. Выберите проект, в которых IO-контроллеру обладают доступом к PROFINET-интерфейсу и портам совместно используемого устройства.
2. Выберите интерфейсный модуль совместно используемого устройства в сетевом представлении.
3. В окне инспектора перейдите к области "PROFINET interface > Advanced options > Real time settings > IO cycle".
4. В области "Shared device" задайте количество внешних IO-контроллеров проекта. Максимальное количество зависит от IO-устройства (спецификация в GSD-файле).
5. Вы должны задать те же синхроимпульсы передачи для каждого IO-контроллера, у которого есть доступ к модулям и субмодулям совместно используемого устройства:
 - Если Вы конфигурируете IO-контроллер с помощью STEP 7 (TIA Portal):
 - Откройте соответствующий проект.
 - Выберите PROFINET-интерфейс IO-контроллера.
 - Выберите область "Advanced options > Real time settings > IO communication" в окне инспектора и задайте совместно используемые синхроимпульсы передачи.
 - Если Вы конфигурируете IO-контроллер с помощью другого инженерного инструмента:
 - Выберите PROFINET-интерфейс совместно используемого устройства в STEP 7 (TIA Portal) и определите сигнал синхронизации передачи совместно используемого устройства (область "Advanced options > Real time settings").
 - Введите сигнал синхронизации передачи в инженерном инструменте.

Примечание

Если Вы конфигурируете все IO-контроллеры которые обладают доступом к совместно используемому устройству в STEP 7 (TIA Portal), Вы можете установить для IO-контроллера сигналы синхронизации передачи более короткие, чем поддерживает совместно используемое устройство (адаптация синхроимпульсов передачи).

Компиляция и загрузка

Вы должны скомпилировать конфигурации для различных IO-контроллеров и загрузить их в ЦПУ одну за другой.

Из-за распределенной конфигурации с отдельными проектами, STEP 7 не выводит ошибки согласованности в случае назначения некорректного параметра доступа. Это примеры назначения параметра некорректного доступа:

- Несколько IO-контроллеров обладают доступом к одному и тому же модулю
- Параметры IP-адреса или сигналы синхронизации передачи не идентичны

Эти ошибки не отображаются вплоть до работы контроллера и выводятся как ошибки конфигурации.

11.2.14.3. Пример: Конфигурирование I-устройства, как совместно используемого устройства

Этот пример описывает, как сконфигурировать S7-1200 в качестве I-устройства с помощью STEP 7 версии V13 SP1 или выше и затем использовать его в двух проектах как совместно используемое устройство.

Возможна "распределенная" конфигурация с различными инженерными инструментами для различных семейств IO-контроллеров. Процедура, описанная ниже, основывается на STEP 7 V13 SP1 и ограничена конфигурацией с двумя IO-контроллерами семейства S7-1200, которые совместно используют области передачи I-устройства как совместно используемого устройства. Само I-устройство представляет собой ЦПУ 1215C.

В примере создается три проекта с одним IO-контроллером в каждом:

- S7-1200-I-устройство
- Controller1
- Controller2

Вы используете проект S7-1200-I-устройства, чтобы сконфигурировать I-устройство. Вы используете PROFINET GSD вариант S7-1200-I-устройства в проектах Controller1 и Controller2, чтобы назначить области передачи в соответствующем IO-контроллере верхнего уровня.

Концепция совместно используемого I-устройства

Концепция совместно используемого I-устройства требует как минимум трех отдельных проектов:

- Проект I-устройства: Вы конфигурируете и программируете I-устройство, чтобы решить определенную задачу автоматизации. Вы определяете области передачи как интерфейс ввода-вывода для контроллеров верхнего уровня и назначаете эти области передачи различным IO-контроллерам. Для соединения с IO-контроллерами верхнего уровня Вы обеспечиваете PROFINET GSD файл и используете области передачи, чтобы получить доступ к I-устройству.
- Контроллеры, которые совместно используют I-устройство (два проекта): Вы используете I-устройство в качестве PROFINET GSD вариант во время конфигурирования PROFINET IO системы и, в ходе этого процесса, определяете адреса ввода-вывода, по которым IO-контроллеры получают доступ к областям передачи.

I-устройство

Вы назначаете следующие параметры для ЦПУ S7-1200 в качестве I-устройства:

- Централизованный и распределенный ввод-вывод
- Желаемые области передачи
- Количество IO-контроллеров, имеющих доступ к этому I-устройству (всегда больше, чем 1 для совместно используемого устройства)

Примечание

Вы конфигурируете I-устройство без IO-контроллера верхнего уровня. В результате Вы можете использовать только локальные адреса ввода-вывода области передачи (= "Адрес в I-устройстве"), чтобы создать пользовательскую программу для редактирования адресов из области передачи. Вы загружаете I-устройство, полностью сконфигурированное за исключением соединения с IO-контроллером верхнего уровня в ЦПУ S7-1200.

Вы экспортируете PROFINET GSD файл из конфигурации I-устройства.

Контроллеры, которые совместно используют I-устройство

Вы должны установить PROFINET GSD файл, создаваемый из конфигурации I-устройства во всех инженерных системах, которые Вы используете при конфигурировании PROFINET IO системы с этим совместно используемым I-устройством. Если Вы конфигурируете все случаи использования этого I-устройства с помощью STEP 7 V13 SP1, достаточно установить GSD-файл в STEP 7.

Вы конфигурируете I-устройство как GSD вариант на PROFINET IO системе в задействованных проектах. В STEP 7 V13 SP1 Вы найдете это I-устройство в разделе Other field devices > PROFINET IO > PLCs & CPs" после установки.

В каждом из задействованных проектов Вы назначаете области передачи исключительно IO-контроллерам верхнего уровня (настройка по умолчанию: all). Вы задаете для других областей передачи "---" (не назначено). Если Вы поступаете подобным образом, то локальный IO-контроллер не может получить доступ к этой области передачи, и Вы можете назначить область передачи другому IO-контроллеру в другом проекте.

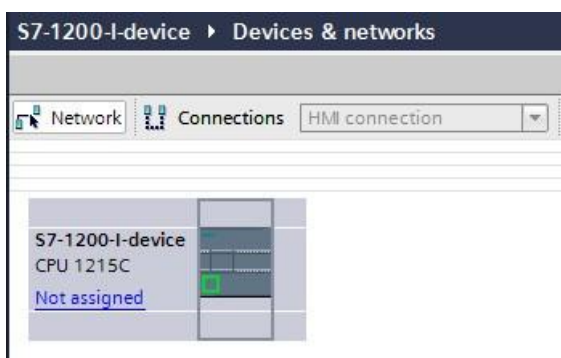
Требования

- STEP 7 V13 SP1 или выше
- IO-устройство поддерживает функциональность совместно используемого устройства (например, ET 200SP IM 155-6 PN HF V3.1).
- GSD-файл для конфигурирования IO-устройства как совместно используемого устройства установлен.

Процедура: Создание проекта S7-1200-I-устройства

Чтобы создать проект с совместно используемым I-устройством, выполните следующие шаги:

1. Запустите STEP 7.
2. Создайте новый проект с именем "S7-1200-I-device".
3. Вставьте ЦПУ 1215C из аппаратного каталога в сетевом представлении. Назначьте имя "S7-1200-I-device".



4. Дважды щелкните по IO устройству и сконфигурируйте все требуемые модули и submodule.

5. Назначьте параметры модуля. В частности Вы должны сконфигурировать следующие настройки для ЦПУ в области PROFINET-интерфейса [X1]:
 - Активируйте опцию "IO device" в области "Operating mode".

- Сконфигурируйте области передачи в области "Operating mode" > "I-device configuration".
Столбец "Address in IO controller" остается пустым, потому что не назначен никакой IO-контроллер.

> I-device communication

Transfer areas

	Transfer area	Type	Address in IO controller	↔ Address in I-device	Length
1	Transfer_area_1	CD		→ I 200...299	100 Byte
2	Transfer_area_2	CD		← Q 200...299	100 Byte
3	Transfer_area_3	CD		→ I 300...399	100 Byte
4	Transfer_area_4	CD		← Q 300...399	100 Byte
5	<Add new>				

Примечание: Чтобы изменить область ввода на область вывода, и наоборот, Вы должны перейти к соответствующей области передачи.

- Выберите количество IO-контроллеров (по крайней мере два), которые получают доступ к совместно используемому I-устройству в процессе работы (область "Operating mode" > "Real time settings", раздел "Shared Device").

> > IO cycle

Shared Device

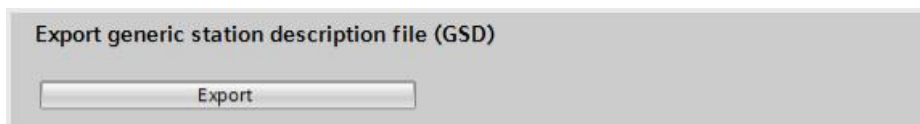
Number of IO controllers with access to this I-device

2

1

2

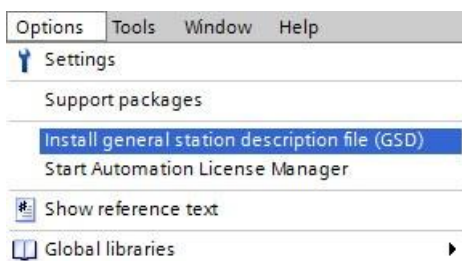
6. Сохраните проект.
7. Нажмите кнопку "Export" (область "Mode" > "I-device configuration", раздел "Export general station description file (GSD)"). Если Вы не изменяете имя в диалоговом окне экспорта, файл GSD использует назначенное имя формата (например, "GSDML-V2.31-#Siemens- PreConf_S7-1200-I-Device-20130925-123456").



Процедура: Создание проекта Controller1

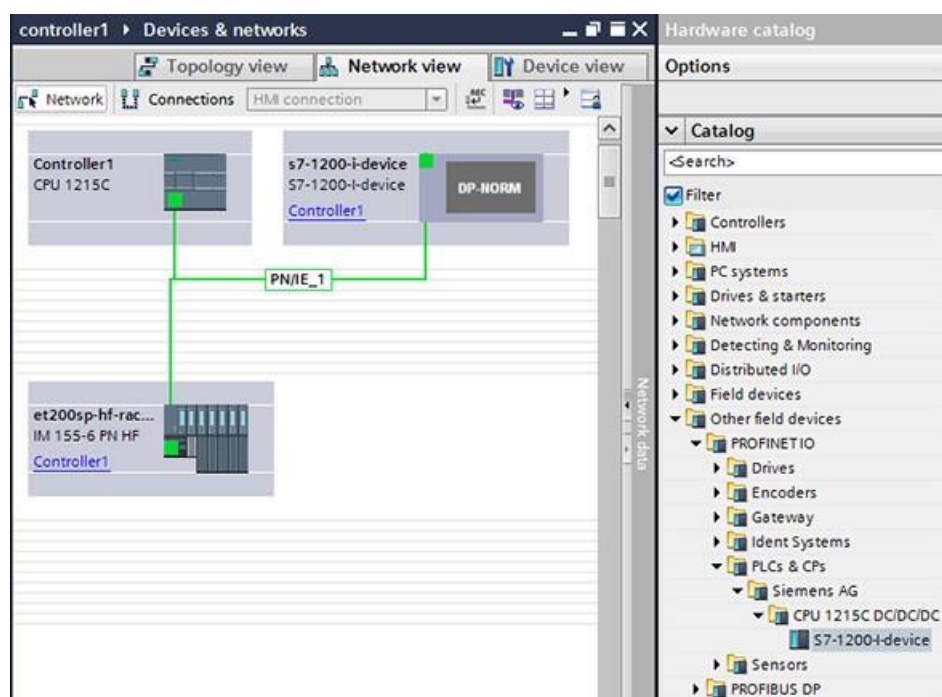
Чтобы создать первый проект с совместно используемым I-устройством, выполните следующие шаги:

1. Запустите STEP 7.
2. Установите PROFINET GSD файл, являющийся результатом экспорта ЦПУ в качестве I-устройства (S7-1200-I-Device).

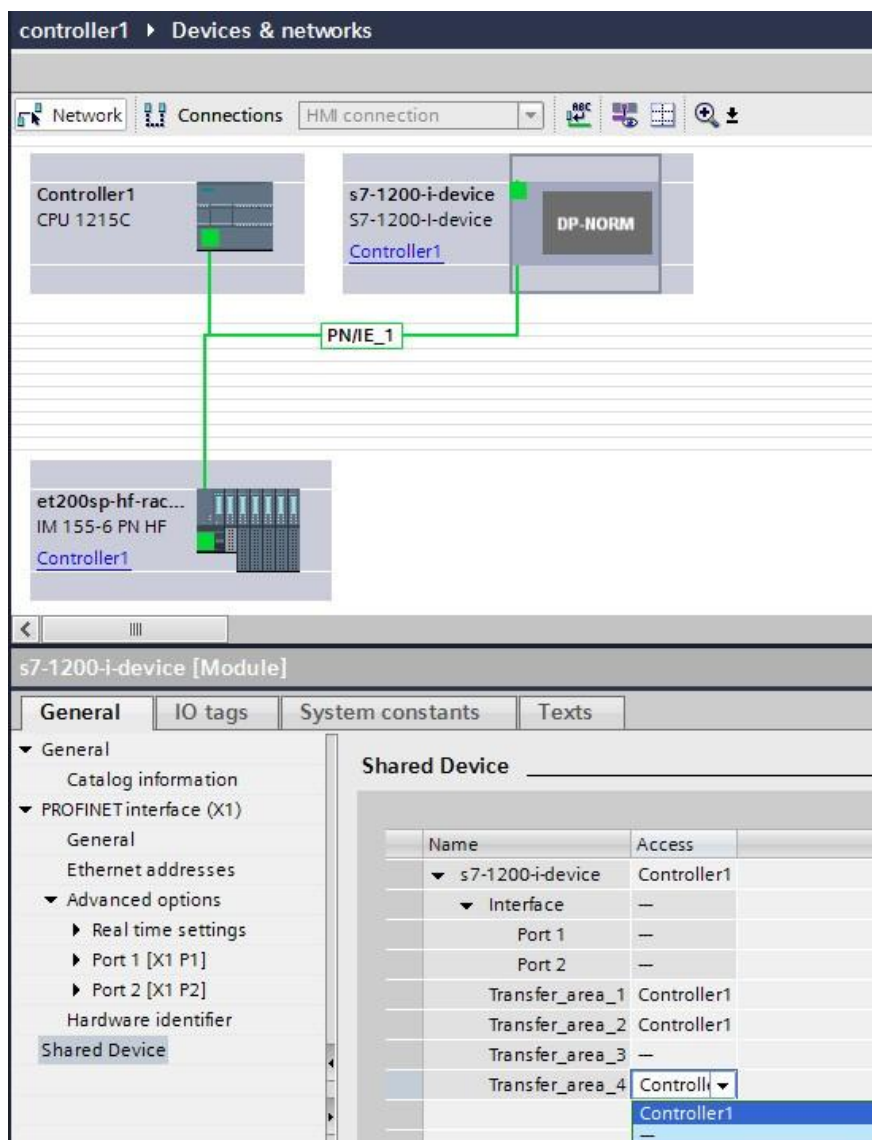


3. Создайте новый проект с именем "Controller1".
4. Вставьте ЦПУ 1215C в сетевом представлении. ЦПУ должен обладать именем "Controller1".
5. Вставьте I-устройство из аппаратного каталога (Аппаратный каталог: Other field devices > PROFINET IO > PLCs & CPs).

6. Назначьте IO-контроллер "Controller1" I-устройству.



7. Выберите область "Shared device" в свойствах I-устройства:
 - В таблице все области передачи и PROFINET-интерфейс назначены локальному IO-контроллеру (Controller1).
 - Определите области передачи, к которым у ЦПУ Controller1 **не** должно быть доступа. Выберите "---" запись для этих областей. Эти области передачи предназначены для Controller2.



- Вы можете адаптировать адреса из представления Device view для IO-контроллера в разделе Device overview. Чтобы открыть обзор устройства, дважды щелкните по I-устройству.

Module	Rack	Slot	I address	Q address	Type	Article number	Firmware	Access
s7-1200-i-device	0	1		256...355	S7-1200-i-device	6ES7 215-1AG40-0XB0	V4.1	Controller1
Transfer_area_1	0	1 1000		256...355	Transfer_area_1			Controller1
Transfer_area_2	0	1 1001	256...355		Transfer_area_2			Controller1
Transfer_area_3	0	1 1002			Transfer_area_3			—
Transfer_area_4	0	1 1003			Transfer_area_4			—
Interface	0	1 X1			s7-1200-i-device			—

- Сохраните проект.

Процедура: Создание проекта Controller2

Чтобы создать второй проект с совместно используемым I-устройством, выполните следующие шаги:

- Запустите STEP 7 еще раз.
Открывается новый экземпляр STEP 7.
- В новом экземпляре создайте новый проект с именем "Controller2".
- Вставьте ЦПУ 1215C в сетевом представлении. Назначьте имя "Controller2".
- Вставьте I-устройство из аппаратного каталога (Аппаратный каталог: Other field devices > PROFINET IO > PLCs & CPs).
- Назначьте IO-контроллер "Controller2" I-устройству.
- Адаптируйте доступ к областям передачи как в проекте Controller1. Гарантируйте отсутствие двойных назначений.
- Адаптируйте параметры PROFINET-интерфейса и подсети. Поскольку совместно используемое I-устройство внедряет одно и то же устройство в различные проекты, эти данные должны соответствовать.
- Сохраните проект.

Теперь в обоих проектах есть тождественно сконфигурированное совместно используемое I-устройство. Доступ IO-контроллера и параметры PROFINET-интерфейса должны по-прежнему быть проверены в различных проектах во время следующего шага.

Подведение итогов - Назначение параметров для доступа к совместно используемому устройству

Области передачи автоматически назначаются локальному IO-контроллеру. Чтобы изменить назначение, выполните следующие шаги:

- Щелкните по устройству "S7-1200-I-Device" в сетевом представлении проекта "Controller1" и выберите область "Shared device".
- Таблица показывает, какой ЦПУ обладает доступом к каждой из сконфигурированных областей передачи. Настройка по умолчанию является то, что у локального ЦПУ есть доступ ко всем модулям и submodule.
- Сохраните настройку "Controller1" для всех областей передачи, которые должны остаться в диапазоне адресов локального ЦПУ.
Выберите настройку "---" для всех областей передачи, которые должны быть расположены в диапазоне адресов ЦПУ "Controller2" из проекта "Controller2". Это означает, что IO-контроллер вне проекта должен обладать доступом к областям передачи.

4. Выполните ту же процедуру для оставшихся проектов.
5. В заключение проверьте, "дополняют" ли друг друга настройки доступа для каждого модуля или submodule в обоих проектах. Это означает, что, если у локального ЦПУ есть доступ в одном проекте, то опция "---" должна быть выбрана в другом проекте и наоборот.
Примечание: опция "---" для PROFINET-интерфейса и, следовательно, для портов делает связанные параметры не изменяемыми и переводит их в режим только для чтения. Параметры PROFINET-интерфейса и параметры порта могут быть отредактированы только в проекте, в котором PROFINET-интерфейс назначен локальному ЦПУ. Порты могут быть соединены в обоих проектах независимо от этого.
6. Проверьте, установлены ли те же параметры IP-адреса и имя устройства для совместно используемого устройства во всех проектах.
Проверьте, установлен ли тот же ID во всех проектах для S7 подсети, с которой соединено совместно используемое устройство (свойства подсети, область "General" в окне инспектора).

Примечание

Если Вы вносите изменения в I-устройство (например, изменяете количество или размер областей передачи), повторно экспортируйте I-устройство как GSD-файл. Переустановите GSD-файл в каждом проекте, который использует I-устройство в качестве совместно используемого устройства. Удостоверьтесь в том, что только один IO-контроллер обладает доступом к области передачи.

Процедура - корректировка настроек реального времени

Чтобы гарантировать, что все IO-контроллеры и совместно используемые устройства управляются надлежащими синхроимпульсами передачи и, что времена обновления правильно рассчитаны на основе коммуникационной загрузки, Вы должны скорректировать и проверить следующие настройки:

1. Вы должны задать те же синхроимпульсы передачи для каждого IO-контроллера, у которого есть доступ к модулям и submodule совместно используемого устройства:
 - Если Вы конфигурируете IO-контроллер с помощью STEP 7 (TIA Portal), выполните следующие шаги:
 - Откройте соответствующий проект.
 - Выберите PROFINET-интерфейс IO-контроллера.
 - Выберите область "Advanced options > Real time settings > IO communication" в окне инспектора и задайте совместно используемые синхроимпульсы передачи.
 - Если Вы конфигурируете IO-контроллер с помощью другого инженерного инструмента:
 - Выберите PROFINET-интерфейс совместно используемого устройства в STEP 7 (TIA Portal) и определите сигнал синхронизации передачи совместно используемого устройства (область "Advanced options > Real time settings").
 - Введите сигнал синхронизации передачи в инженерном инструменте.

Примечание

Если Вы конфигурируете все IO-контроллеры которые обладают доступом к совместно используемому устройству в STEP 7 (TIA Portal), Вы можете установить для IO-контроллера сигналы синхронизации передачи более короткие, чем поддерживает совместно используемое устройство (адаптация синхроимпульсов передачи).

Компиляция и загрузка

Вы должны скомпилировать конфигурации для различных IO-контроллеров и загрузить их в ЦПУ одну за другой.

Из-за распределенной конфигурации с отдельными проектами, STEP 7 не выводит ошибки согласованности в случае назначения некорректного параметра доступа. Это примеры назначения параметра некорректного доступа:

- Несколько IO-контроллеров обладают доступом к одному и тому же модулю
- Параметры IP-адреса или сигналы синхронизации передачи не идентичны

Эти ошибки не отображаются вплоть до работы контроллера и выводятся как ошибки конфигурации.

11.2.15. Диагностика

Обратитесь к теме "Организационные блоки (ОВ)" (стр. 88) для получения информации о том, как использовать организационные блоки (ОВ) для диагностики с этими коммуникационными сетями.

11.2.16. Инструкции для распределенного ввода-вывода

Обратитесь к теме "Распределенный ввод-вывод (PROFINET, PROFIBUS или AS-i)" (стр. 350) для получения информации о том, как использовать инструкции для распределенного ввода-вывода с этими коммуникационными сетями.

11.2.17. Диагностические инструкции

Обратитесь к теме (PROFINET или PROFIBUS)": "Диагностические инструкции" (стр. 384) для получения информации о том, как использовать упомянутые инструкции с этими коммуникационными сетями.

11.2.18. Диагностические события для распределенного ввода-вывода

Обратитесь к теме "Диагностика (PROFINET или PROFIBUS)": "Диагностические события для распределенного ввода-вывода" (стр. 384) для получения информации о том, как использовать данную диагностическую информацию с этими коммуникационными сетями.

11.3 PROFIBUS

Система PROFIBUS использует ведущее устройство шины, чтобы опросить ведомые устройства, распределенные многоточечным способом на последовательной шине RS485. Ведомым устройством PROFIBUS является любое периферийное устройство (преобразователь ввода-вывода, клапан, электропривод или другой измерительный прибор), который обрабатывает информацию и отправляет ее значение ведущему устройству. Ведомое устройство представляет пассивную станцию в сети, так как оно не имеет прав доступа к шине, и может только подтверждать полученные сообщения или отправлять ответные сообщения ведущему устройству по запросу. Все ведомые устройства PROFIBUS обладают одним и тем же приоритетом, и все сетевые коммуникации активируются ведущим устройством.

Ведущее устройство PROFIBUS представляет "активную станцию" в сети. PROFIBUS DP определяет два класса ведущих устройств. Ведущее устройство класса 1 (обычно центральный программируемый контроллер (ПЛК) или ПК, на котором работает специальное программное обеспечение), обрабатывает обычные коммуникации или обмен данными с ведомыми устройствами, назначенными ему. Ведущее устройство класса 2 (обычно устройство конфигурирования, такое как ноутбук или программная консоль, используемые для ввода в эксплуатацию, обслуживания или диагностики), является специальным устройством, используемым прежде всего для ввода в эксплуатацию ведомых устройств и в целях диагностики.

S7-1200 подключается к сети PROFIBUS как ведомое устройство DP с помощью коммуникационного модуля CM 1242-5. CM 1242-5 (ведомое устройство DP) модуль может быть коммуникационным партнером ведущих устройств DP V0/V1. Если Вы хотите сконфигурировать модуль в сторонней системе, то существует GSD-файл для CM 1242-5 (ведомое устройство DP) доступный на CD, который поставляется с модулем и в Интернете на странице поддержки клиентов автоматизации от Сименс (<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK72425DX300XE0&caller=view>).

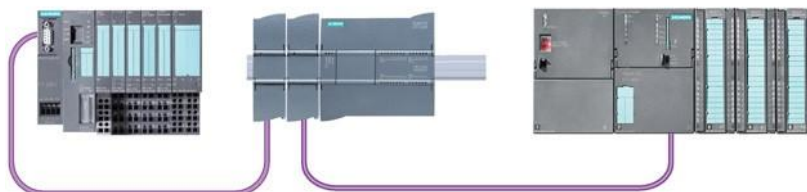
На рисунке ниже, S7-1200 является ведомым DP устройством контроллера S7-300:



S7-1200 подключается к сети PROFIBUS как ведущее устройство DP с помощью коммуникационного модуля CM 1243-5. CM 1243-5 (ведущее устройство DP) модуль может быть коммуникационным партнером ведомых устройств DP V0/V1. На рисунке ниже S7-1200 является ведущим устройством, управляющим ведомым DP устройством ET200S:



Если CM 1242-5 и CM 1243-5 установлены вместе, то S7-1200 может одновременно выступать и как ведомое устройство ведущей DP-системы верхнего уровня, и как ведущее устройство ведомой DP-системы нижнего уровня:



Для V4.0 Вы можете сконфигурировать максимум три PROFIBUS-CM на станцию, среди которых может быть любая комбинация CM ведущих и ведомых DP устройств. Ведущие устройства DP в V3.0 или более поздней версии встроенного ПО для ЦПУ могут каждый управлять максимумом 32 ведомыми устройствами.

Конфигурационные данные PROFIBUS-CM сохраняются на локальном ЦПУ. Это позволяет простую замену этих коммуникационных модулей при необходимости.

Чтобы использовать PROFIBUS с ЦПУ S7-1200 V4.0, Вы должны обновить встроенное ПО для CM ведущего устройства PROFIBUS до V1.3.

Вы можете выполнить это обновление, используя карту Secure Digital (SD).

Примечание

Рекомендуется, чтобы Вы всегда обновляли встроенное ПО PROFIBUS-CM до последней доступной на веб-сайте Обслуживания и Поддержки клиентов Сименс (<http://support.automation.siemens.com/WW/view/en/42131407>) версии.

11.3.1. Коммуникационные службы PROFIBUS-CM

PROFIBUS-CM используют протокол PROFIBUS DP-V1.

Типы коммуникаций с DP-V1

Следующие типы коммуникаций доступны с DP-V1:

- Циклический обмен (CM 1242-5 и CM 1243-5)
Оба PROFIBUS модуля поддерживают циклический обмен для передачи данных процесса между ведомым и ведущим DP устройствами.
Циклический обмен управляется операционной системой ЦПУ. Для этого не требуются никакие программные блоки. Данные ввода-вывода считываются или записываются непосредственно из / в образа процесса в ЦПУ.
- Ациклический обмен (только CM 1243-5)
Модуль ведущего DP устройства также поддерживает ациклический обмен с использованием программных блоков:
 - Инструкция "RALRM" доступна для обработки прерываний.
 - Инструкции "RDREC" и "WRREC" доступны для передачи диагностических и конфигурационных данных.

Функции, не поддерживаемые CM 1243-5: SYNC/FREEZE и Get_Master_Diag

Другие коммуникационные службы CM 1243-5

Модуль ведущего DP устройства CM 1243-5 поддерживает следующие дополнительные коммуникационные службы:

- S7 коммуникации
 - Службы PUT/GET
Ведущее устройство DP функционирует как клиент и сервер для запросов от других S7 контроллеров или ПК по PROFIBUS.
 - PG/OP коммуникации
Функции PG позволяют загрузку конфигурационных данных и пользовательских программ из программатора и передачу диагностических данных в программатор.
Возможными коммуникационными партнерами для OP-коммуникаций являются HMI-панели, панельные ПК SIMATIC WinCC flexible или SCADA-системами, которые поддерживают S7-коммуникации.

11.3.2. Ссылка на руководства пользователя для PROFIBUS-CM

Дополнительная информация

Вы можете найти подробную информацию о PROFIBUS-CM в руководствах для устройств. Вы можете найти их в Интернете на страницах Поддержки клиентов Автоматизации производства от Сименс по следующим регистрационным ID:

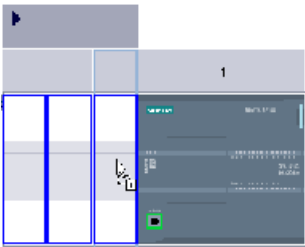
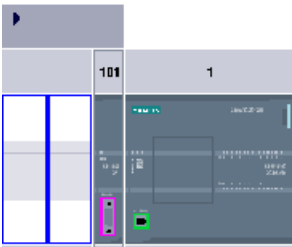
- CM 1242-5(<http://support.automation.siemens.com/WW/view/en/44632650>)
- CM 1243-5(<http://support.automation.siemens.com/WW/view/en/44632657>)

11.3.3. Конфигурирование ведущего и ведомого DP устройств

11.3.3.1. Добавление CM 1243-5 (ведущее устройство DP) модуля и ведомого устройства DP

В портале "Devices and networks" используйте аппаратный каталог, чтобы добавить PROFIBUS модули к ЦПУ. Эти модули подключаются с левой стороны от ЦПУ. Чтобы вставить модуль в аппаратную конфигурацию, выберите модуль в аппаратном каталоге и либо дважды щелкните, либо перетащите модуль в выделенный слот.

Таблица 11- 55 Добавление CM PROFIBUS 1243-5 (ведущее устройство DP) модуля в конфигурацию устройства

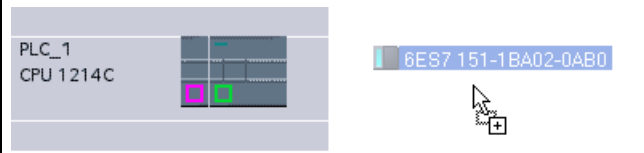
Модуль	Выберите модуль	Вставьте модуль	Результат
CM 1243-5 (ведущее устройство DP)			

Используйте аппаратный каталог, чтобы добавить также ведомые устройства DP. Например, чтобы добавить ведомое DP устройство ET200 S, раскройте в аппаратном каталоге следующие контейнеры:

- Distributed I/O
- ET200 S
- Interface modules
- PROFIBUS

Затем, выберите "6ES7 151-1BA02-0AB0" (IM151-1 HF) из списка номеров каталога и добавьте ведомое DP устройство ET200 S, как показано на рисунке ниже.

Таблица 11- 56 Добавление ведомого DP устройства ET200 S в конфигурацию устройства

Вставьте ведомое DP устройство	Результат
	

11.3.3.2. **Конфигурирование логических сетевых соединений между двумя PROFIBUS устройствами**

При конфигурировании логических сетевых соединений между двумя PROFIBUS устройствами. После того, как Вы сконфигурируете CM 1243-5 (ведущее устройство DP) модуль, Вы теперь готовы сконфигурировать свои сетевые соединения.

В портале "Devices and Networks" используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Чтобы создать PROFIBUS соединение, выберите фиолетовое поле (PROFIBUS) на первом устройстве. Перетащите линию к полю PROFIBUS на втором устройстве. Отпустите кнопку мыши, и Ваше PROFIBUS соединению будет выполнено.

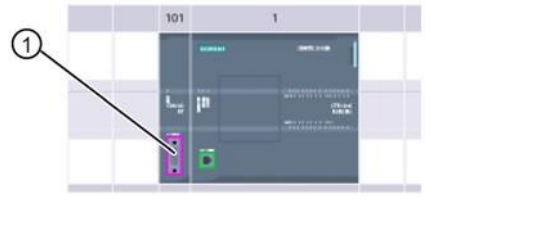
Обратитесь к теме "Конфигурация устройства: Создание сетевого соединения" (стр. 618) для получения более подробной информации.

11.3.3.3. **Назначение PROFIBUS адресов для CM 1243-5 модуля и ведомого устройства DP**

Конфигурирование PROFIBUS интерфейса

После того, как Вы сконфигурируете логические сетевые соединения между двумя PROFIBUS устройствами, Вы можете сконфигурировать параметры для PROFIBUS интерфейсов. Для этого щелкните по фиолетовому полю PROFIBUS на модуле CM 1243-5, и вкладка "Properties" в окне инспектора отобразит PROFIBUS интерфейс. PROFIBUS интерфейс ведомого устройства DP конфигурируется подобным образом.

Таблица 11- 57 Конфигурирование PROFIBUS-интерфейсов модуля CM 1243-5 (ведущее устройство DP) и ведомого DP устройства ET200 S

CM 1243-5 (ведущее DP устройство) модуль	Ведомое DP устройство ET200 S
	

① PROFIBUS-порт

Назначение PROFIBUS-адресов

В сети PROFIBUS каждому устройству назначен адрес PROFIBUS. Этот адрес может изменяться в диапазоне от 0 до 127, принимая во внимание следующие исключения:

- Адрес 0: Зарезервирован для конфигурации сети и/или инструментов программирования, присоединенных к шине
- Адрес 1: Зарезервирован Сименс для первого ведущего устройства
- Адрес 126: Зарезервирован для устройств от изготовителя, не имеющих установочных переключателей, адреса которых должны быть переназначены через сеть
- Адрес 127: Зарезервирован для широковещательных сообщений ко всем устройствам в сети и не может быть назначен работающим устройствам

Таким образом для работающих PROFIBUS устройств могут использоваться адреса со 2 по 125.

В окне свойств выберите конфигурационную запись "PROFIBUS address". STEP 7 отображает диалоговое окно конфигурирования PROFIBUS-адреса, которое используется, чтобы назначить PROFIBUS-адрес устройства.

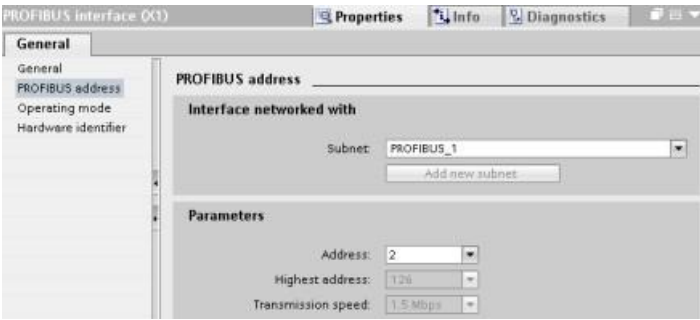


Таблица 11- 58 Параметры для PROFIBUS-адреса

Параметр		Описание
Subnet	Имя подсети, к которой подключено устройство. Нажмите кнопку "Add new subnet", чтобы создать новую подсеть. "Not connected" является значением по умолчанию. Возможны два типа подключения:	
	• Значение по умолчанию "Not connected" обеспечивает локальное соединение. • Подсеть требуется, когда в Вашей сети есть два или более устройств.	
Parameters	Address	Назначенный устройству PROFIBUS-адрес
	Highest address	Наибольший PROFIBUS-адрес основывается на активных станциях на PROFIBUS (например, ведущее устройство DP). Пассивные ведомые устройств DP независимо имеют PROFIBUS-адреса от 1 до 125, даже если наибольший PROFIBUS-адрес установлен, к примеру, на 15. Наибольший PROFIBUS-адрес важен для передачи маркера (перенаправление прав на передачу), и маркер передается только активным станциям. Определение наибольшего PROFIBUS-адреса оптимизирует шину.

Параметр		Описание
	Transmission rate	Скорость передачи сконфигурированной сети PROFIBUS: скорости передачи PROFIBUS колеблются от 9.6 Кбит/с до 12 Мбит/с. Настройка скорости передачи зависит от свойств используемых PROFIBUS-узлов. Скорость передачи не должна иметь значение больше, чем поддерживается самым медленным узлом. Скорость передачи обычно устанавливается для ведущего устройства в сети PROFIBUS со всеми ведомыми устройствами DP, автоматически использующими ту же самую скорость передачи (auto-baud).

11.3.4. Инструкции для распределенного ввода-вывода

Обратитесь к теме "Распределенный ввод-вывод (PROFINET, PROFIBUS или AS-i)" (стр. 350) для получения информации о том, как использовать инструкции для распределенного ввода-вывода с этими коммуникационными сетями.

11.3.5. Диагностические инструкции

Обратитесь к теме (PROFINET или PROFIBUS)": "Диагностические инструкции" (стр. 384) для получения информации о том, как использовать упомянутые инструкции с этими коммуникационными сетями.

11.3.6. Диагностические события для распределенного ввода-вывода

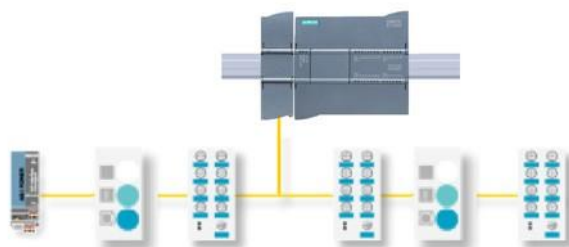
Обратитесь к теме "Диагностика (PROFINET или PROFIBUS)": "Диагностические события для распределенного ввода-вывода" (стр. 384) для получения информации о том, как использовать данную диагностическую информацию с этими коммуникационными сетями.

11.4 AS-i

CM 1243-2, являющийся ведущим устройством AS-i шины для S7-1200 позволяет подключение AS-i сети к ЦПУ S7-1200.

Интерфейс привода/датчика или AS-i является системой для подключения к сети самого нижнего уровня в системах автоматизации. CM 1243-2 служит ведущим устройством AS-i для сети. Используя единственный AS-i кабель, датчики и приводы (ведомые устройства AS-i) могут быть подключены к ЦПУ через CM 1243-2. CM 1243-2 осуществляет координацию AS-i сети и перенаправляет данные и информация о статусе от приводов и датчиков в ЦПУ через адреса ввода-вывода назначенные CM 1243-2. Вы можете получить доступ к двоичным или аналоговым значениям в зависимости от типа ведомого устройства. Ведомые устройства AS-i представляют собой каналы ввода и вывода AS-i системы и активны только в том случае, если вызываются CM 1243-2.

На рисунке ниже S7-1200 представляет собой ведущее AS-i устройство управляющее AS-i модульными ведомыми устройствами цифрового/аналогового ввода/вывода.



Чтобы использовать AS-i с ЦПУ S7-1200 V4.0, Вы должны обновить встроенное ПО CM ведущего устройства AS-i до V1.1.

Вы можете выполнить это обновление, используя веб-сервер или карту памяти SIMATIC.

Примечание

Для ЦПУ S7-1200 V4.0 при использовании веб-сервера или карты памяти SIMATIC, чтобы выполнить обновление встроенного ПО от V1.0 до V1.1 в ведущем AS-i устройстве CM 1243-2, Вы должны действовать согласно следующей процедуре:

1. Загрузите прошивку в ведущее AS-i устройство CM 1243-2.
2. Когда загрузка будет завершена, выполните цикл выключения и включения питания ЦПУ S7-1200, чтобы завершить процесс перепрошивки в ведущем AS-i устройстве CM 1243-2.
3. Повторите шаги 1 и 2 для каждого дополнительного ведущего AS-i устройства CM 1243-2. ПЛК S7-1200 позволяет подключить максимум три ведущих AS-i устройства CM 1243-2.

Примечание

Рекомендуется, чтобы Вы всегда обновляли встроенное ПО AS-i-CM до последней доступной на веб-сайте Обслуживания и Поддержки клиентов Сименс (<http://support.automation.siemens.com/WW/view/en/43416171>) версии.

11.4.1 Конфигурирование ведущего и ведомого AS-i устройств

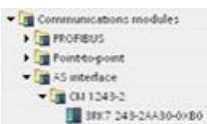
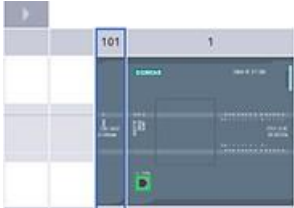
Ведущее AS-i устройство CM ASI 1243-2 интегрируется в систему автоматизации S7-1200 как коммуникационный модуль.

Вы можете найти подробную информацию о ведущем AS-i устройстве CM ASI 1243-2 в Руководстве (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

11.4.1.1 Добавление ведущего AS-i устройства CM 1243-2 и ведомого AS-i устройства

Используйте аппаратный каталог, чтобы добавить модули ведущего AS-i устройства CM1243-2 к ЦПУ. Эти модули присоединяются с левой стороны от ЦПУ, и могут быть использованы максимум три модуля ведущего AS-i устройства CM1243-2. Чтобы вставить модуль в аппаратную конфигурацию, выберите модуль в аппаратном каталоге и либо дважды щелкните по нему, либо перетащите модуль в выделенный слот.

Таблица 11- 59 Добавление модуля ведущего AS-i устройства CM1243-2 в конфигурацию устройства

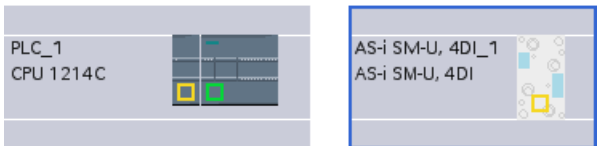
Модуль	Выберите модуль	Вставьте модуль	Результат
CM 1243-2 Ведущее AS-i устройство			

Также используйте аппаратный каталог, чтобы добавить ведомые AS-i устройства. Например, чтобы добавить ведомое устройство "модуль ввода-вывода, компактный цифровой ввод", раскройте в аппаратном Каталоге следующие контейнеры:

- Field devices
- AS-Interface slaves

Затем, выберите "3RG9 001-0AA00" (AS-i SM-U, 4DI) из списка заказных номеров и добавьте ведомое устройство "модуль ввода-вывода, компактный цифровой ввод", как показано на рисунке ниже.

Таблица 11- 60 Добавление ведомого AS-i устройства в конфигурацию устройства

Вставьте ведомое AS-i устройство	Результат
	

11.4.1.2 Конфигурирование логических сетевых соединений между двумя AS-i устройствами

После конфигурирования ведущего AS-i устройства CM1243-2 Вы теперь готовы сконфигурировать свои сетевые соединения.

В портале "Devices and Networks" используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Чтобы создать AS-i соединение, выберите желтое (AS-i) поле на первом устройстве. Перетащите линию к AS-i полю на втором устройстве.

Отпустите кнопку мыши, и Ваше AS-i соединение будет организовано.

Обратитесь к теме "Конфигурация устройства: Создание сетевого соединения" (стр. 618) для получения более подробной информации.

11.4.1.3 Конфигурирование свойств ведущего AS-i устройства CM1243-2

Чтобы сконфигурировать параметры для AS-i интерфейса, щелкните по желтому AS-i полю на модуле ведущего AS-i устройства CM1243-2, и вкладка "Properties" в окне инспектора отобразит AS-i интерфейс.

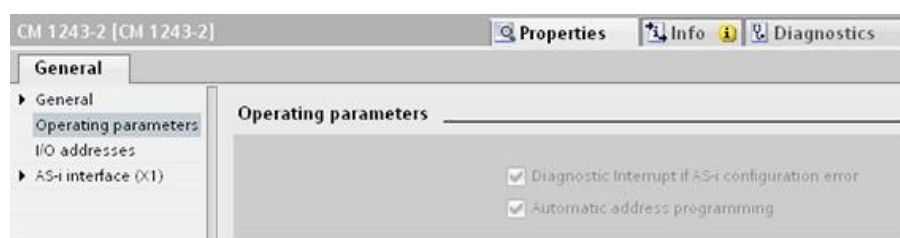
В STEP 7 окне инспектора Вы можете просмотреть, сконфигурировать и изменить общую информацию, адреса и рабочие параметры:

Таблица 11- 61 Свойства модуля ведущего AS-i устройства CM1243-2

Свойство	Описание
General	Имя ведущего AS-i устройства CM 1243-2
Operating parameters	Параметры для ответа ведущего AS-i устройства
I/O addresses	Адресное пространство для адресов ввода-вывода ведомых устройств
AS-i interface (X1)	Назначенная AS-i сеть

Примечание

Опции "Diagnostic interrupt for faults in the AS-i configuration" и "Automatic address programming" всегда активны и поэтому показаны в сером.



11.4.1.4 Назначение AS-i адреса ведомому AS-i устройству

Конфигурирование интерфейса ведомого AS-i устройства

Чтобы сконфигурировать параметры для AS-i интерфейса, щелкните по желтому AS-i полю на ведомом AS-i устройстве, и вкладка "Properties" в окне инспектора отобразит AS-i интерфейс.



① AS-i порт

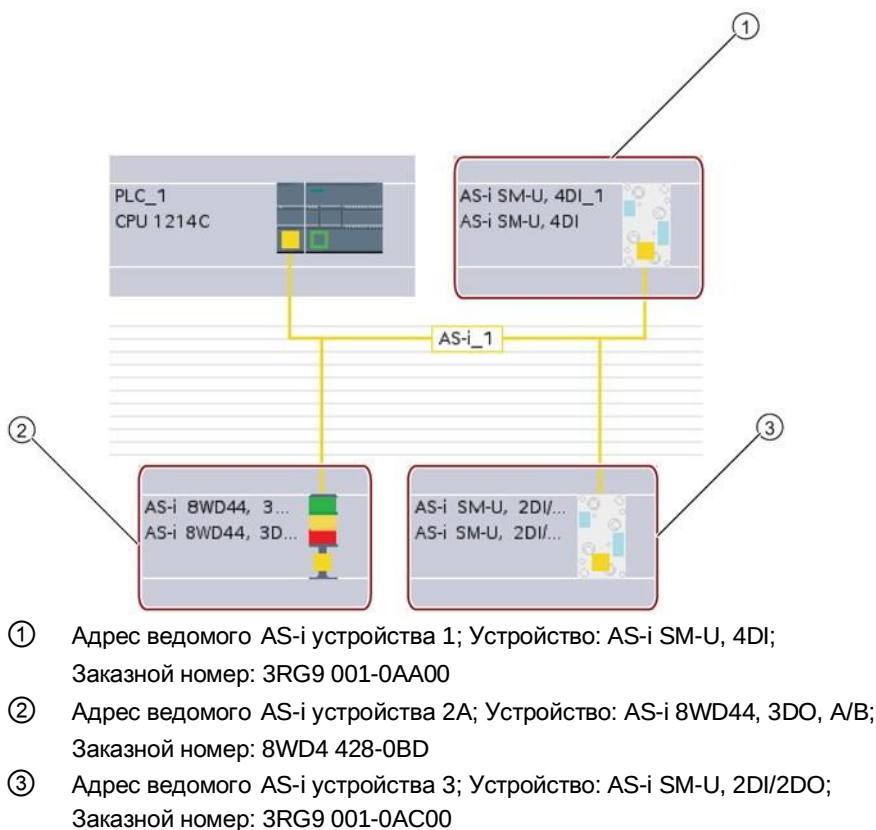
Назначение адреса ведомому AS-i устройству

В AS-i сети каждому устройству назначен адрес ведомого устройства. Этот адрес может изменяться от 0 до 31; однако, адрес 0 зарезервирован только для новых ведомых устройств. Ведомые адреса доступны от 1 (А или В) до 31 (А или В), что в общей сложности соответствует максимум 62 ведомым устройствам.

"Стандартные" AS-i устройства используют полный адрес, имея адрес без добавления А или В. AS-i устройства "А/В узла" используют А или В часть каждого адреса, позволяя каждому из 31 адресов использоваться дважды. Диапазон адресов распространяется от 1 А до 31 А плюс от 1В до 31В.

Любой адрес в диапазоне от 1 до 31 может быть назначен ведомому AS-i устройству; другими словами, не имеет значения, начинаются ли адреса ведомых устройств с адреса 21 или первому фактическому ведомому устройству назначают адрес 1.

В примере ниже, три AS-i устройства адресованы как "1" (устройство стандартного типа), "2 А" (устройство типа узла А/В), и "3" (устройство стандартного типа):



Введите здесь адрес ведомого AS-i устройства:

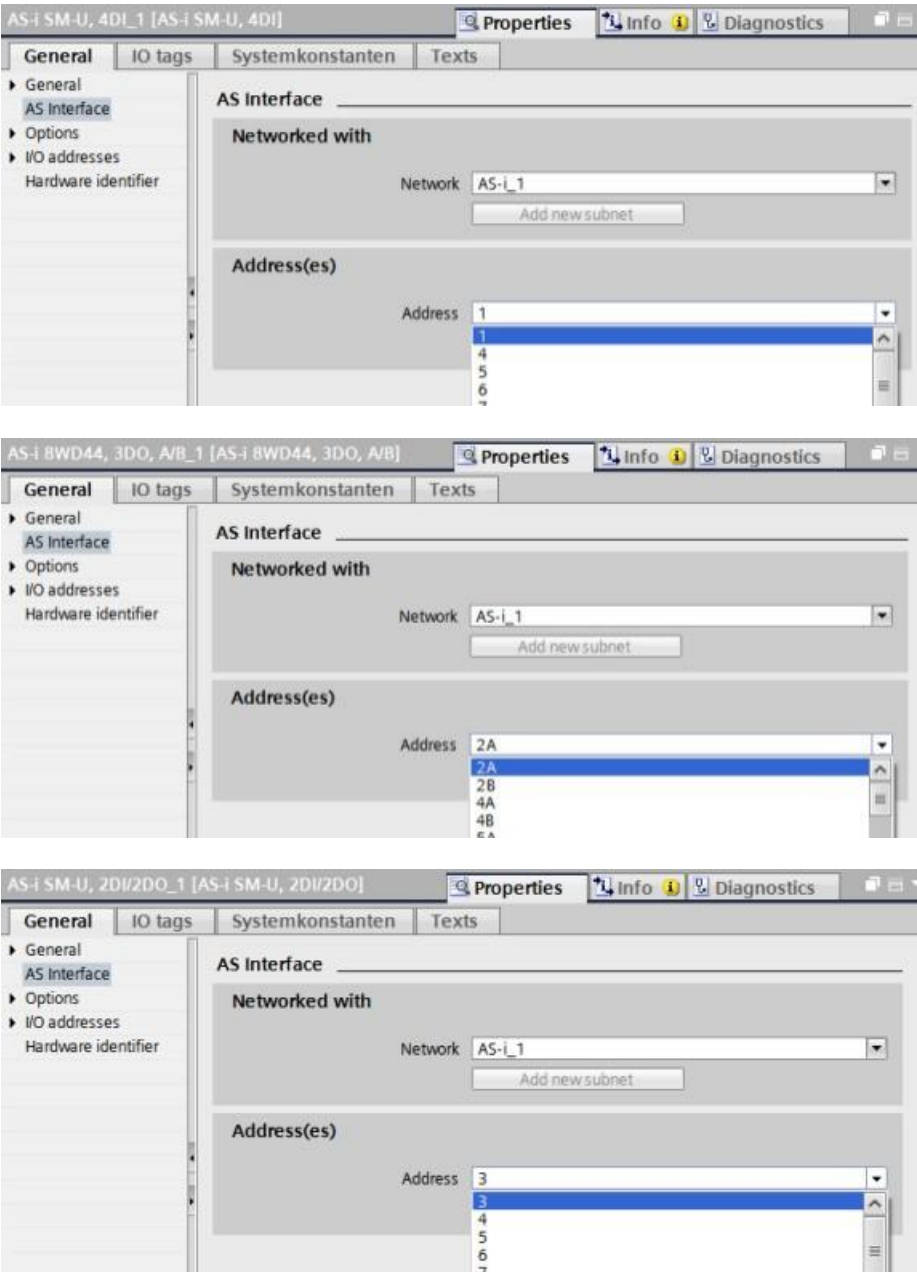


Таблица 11- 62 Параметры для AS-i интерфейса

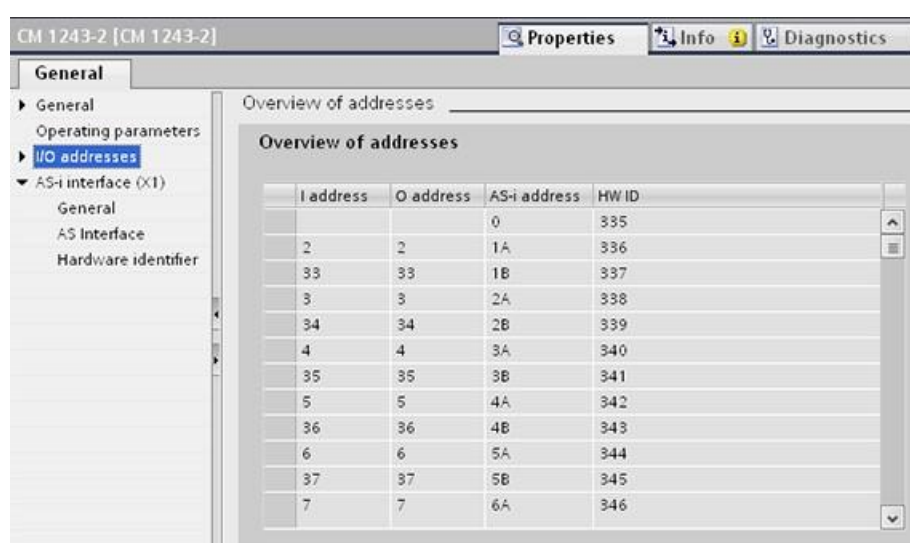
Параметр	Описание
Network	Имя сети, с которой соединено устройство
Address(es)	Назначенный AS-i адрес для ведомого устройства в диапазоне от 1 (A или B) до 31 (A или B), то есть в общей сложности для максимум 62 ведомых устройств

11.4.2 Обмен данными между пользовательской программой и ведомыми AS-i устройствами

11.4.2.1 Базовая конфигурация в STEP 7

Ведущее AS-i устройство резервирует 62-байтовую область данных в области ввода-вывода ЦПУ. Доступ к цифровым данным выполняется здесь в байтах; для каждого ведомого устройства есть один байт входных и один байт выходных данных.

Назначение AS-i соединений цифровых ведомых AS-i устройств битам данных назначенного байта отображается в окне инспектора ведущего AS-i устройства CM 1243-2.



The screenshot shows the 'Properties' window for 'CM 1243-2'. The 'General' tab is selected, and the 'I/O addresses' section is expanded. The 'Overview of addresses' table is displayed, showing the mapping of I address, O address, AS-i address, and HW ID.

I address	O address	AS-i address	HW ID
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

Вы можете получить доступ к данным ведомых AS-i устройств в пользовательской программе посредством отображаемых адресов ввода-вывода с помощью надлежащих битовых логических операций (например, "I") или битовых присвоений.

Примечание

"Назначение системой" автоматически активируется, если Вы не конфигурируете ведомые AS-i устройства с помощью STEP 7.

Если Вы не конфигурируете ведомые устройства, Вы должны сообщить ведущему AS-i устройству CM1243-2 о фактической конфигурации шины, используя онлайн-функцию, "ACTUAL > EXPECTED".

Дополнительная информация

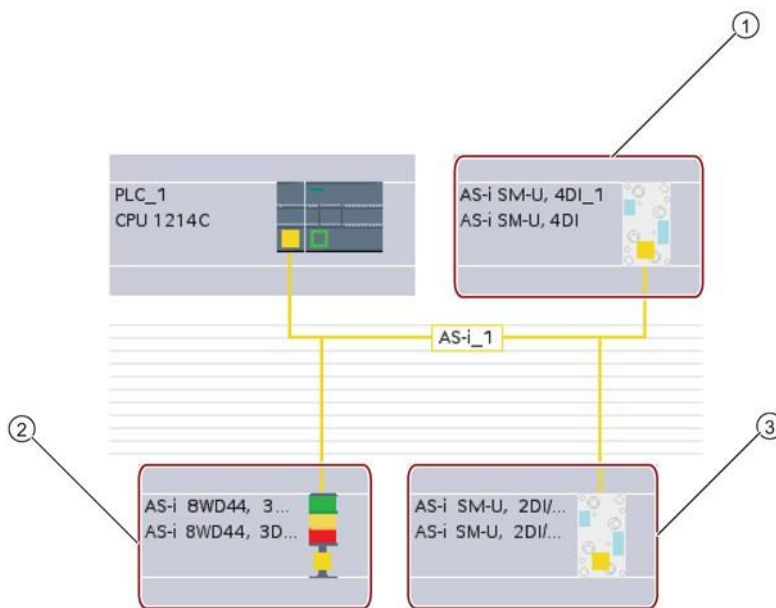
Вы можете найти подробную информацию о ведущем AS-i устройстве CM 1243-2 в руководстве "Ведущее AS-i устройство CM 1243-2 и AS-i модуле разъединения данных DCM 1271 для SIMATIC S7-1200"

(<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

11.4.2.2 Конфигурирование ведомых устройств с помощью STEP 7

Передача цифровых AS-i значений

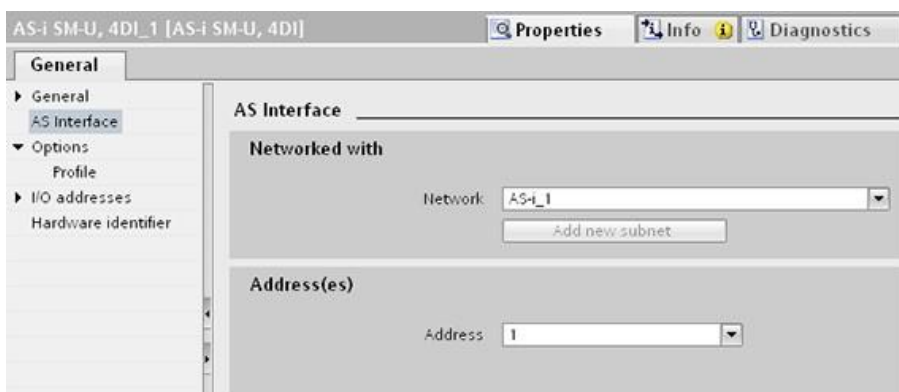
ЦПУ получает доступ к цифровым входам и выходам ведомых AS-i устройств через ведущее AS-i устройство SM1243-2 в ходе циклического процесса. К данным получают доступ через адреса ввода-вывода или посредством передачи записей данных.



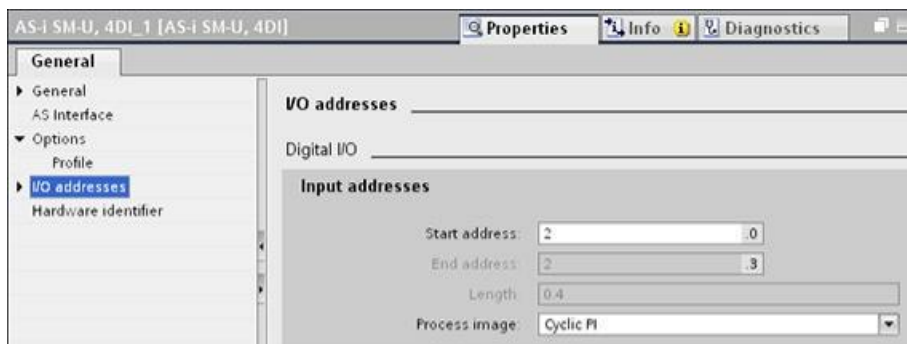
- ① Адрес ведомого AS-i устройства 1
- ② Адрес ведомого AS-i устройства 2A
- ③ Адрес ведомого AS-i устройства 3

Доступ к цифровым данным выполнен здесь в байтах (другими словами, один байт назначен каждому AS-i цифровому ведомому устройству). Когда Вы конфигурируете ведомые AS-i устройства в STEP 7, адрес ввода-вывода для доступа к данным из пользовательской программы отображается в окне инспектора для соответствующего ведомого AS-i устройства.

Цифровому входному модулю (AS-i SM-U, 4DI) в AS-i сети, показанной выше, был назначен, ведомый адрес 1. Щелкнув по цифровому входному модулю, вкладка "AS interface" в "свойствах" устройства отобразит ведомый адрес, как показано ниже:

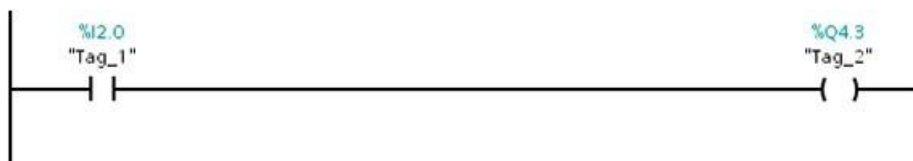


Цифровому входному модулю (AS-i SM-U, 4DI) в AS-i сети, показанной выше, был назначен адрес ввода-вывода 2. Щелкнув по цифровому входному модулю, вкладка "I/O addresses" в "свойствах" устройства отобразит адрес ввода-вывода, как показано ниже:



Вы можете получить доступ к данным ведомых AS-i устройств в пользовательской программе при помощи их адресов ввода-вывода посредством надлежащих битовых логических операций (например, "И") или битовых присвоений. Следующая простая программа иллюстрирует, как работает присвоение:

В этой программе опрашивается вход 2.0. В AS-i системе этот вход принадлежит slave 1 (Входной байт 2, бит 0). Выход 4.3, который затем устанавливается, соответствует AS-i slave 3 (Выходной байт 4, бит 3)



Передача аналоговых AS-i значений

Вы можете получить доступ к аналоговым данным ведомого AS-i устройства через образ процесса в ЦПУ, если Вы сконфигурировали это AS-i устройство в STEP 7 как аналоговое ведомое устройство.

Если Вы не конфигурировали аналоговое ведомое устройство в STEP 7, Вы можете получить доступ к данным ведомого AS-i устройства только через ациклические функции (интерфейс записи данных). В пользовательской программе ЦПУ AS-i вызовы считываются и записываются, используя RDREC (читать запись данных), и WRREC (записать запись данных) инструкции по работе с распределенным вводом-выводом.

Примечание

Конфигурация ведомых AS-i устройств, определенных с помощью STEP 7 и загруженных в S7 станцию, передается ЦПУ в ведущее AS-i устройство CM1243-2 во время запуска S7 станции. Любая существующая конфигурация, которая была определена через он-лайн функцию "Назначение системой" (стр. 765) ("ACTUAL -> EXPECTED"), будет перезаписана

Дополнительная информация

Вы можете найти подробную информацию о ведущем AS-i устройстве CM 1243-2 в руководстве "Ведущее AS-i устройство CM 1243-2 и AS-i модуле разъединения данных DCM 1271 для SIMATIC S7-1200"

(<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

11.4.3 Инструкции для распределенного ввода-вывода

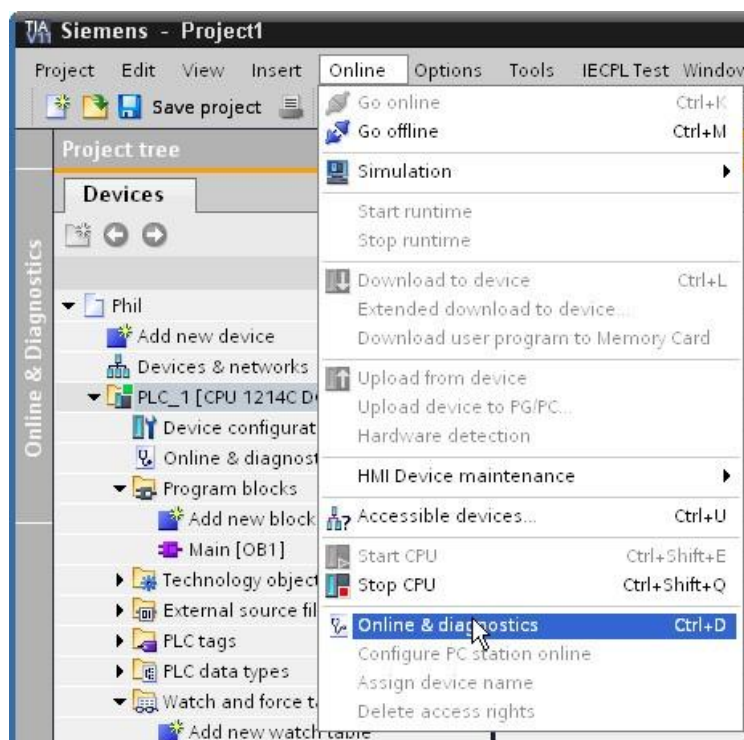
Обратитесь к теме "Распределенный ввод-вывод (PROFINET, PROFIBUS или ASI)" (стр. 350) для получения информации о том, как использовать инструкции для распределенного ввода-вывода с этими коммуникационными сетями.

11.4.4 Работа с AS-i он-лайн инструментами

Изменение AS-i рабочего режима он-лайн

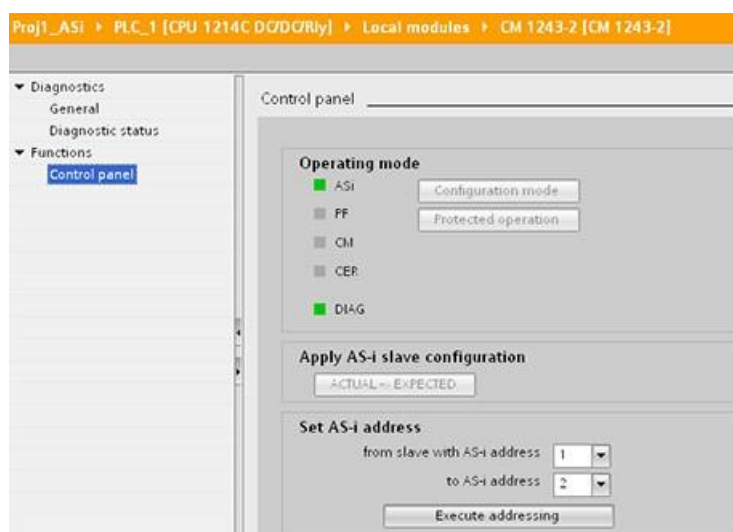
Вы должны перейти в режим он-лайн, чтобы просмотреть и изменить рабочие режимы AS-i.

Чтобы активировать он-лайн, Вы должны вначале перейти в "Конфигурацию устройства" с выбранным модулем ведущего устройством AS-i устройства CM1243-2, а затем нажать кнопку "Go online" на панели инструментов. Затем, выберите команду "Online and diagnostics" из меню "Online".



Существует два рабочих AS-i режима:

- Защищенный режим:
 - Вы не можете изменить ведомое AS-i устройство и адреса ввода-вывода ЦПУ.
 - Зеленый светодиод "CM" погашен.
- Режим конфигурирования:
 - Вы можете внести требуемые изменения в свое ведомое AS-i устройство и адреса ввода-вывода ЦПУ.
 - Зеленый светодиод "CM" LED зажат.



В поле "Set AS-i address" Вы можете изменить адрес ведомого AS-i устройства. Новое ведомое устройство, которому не был назначен адрес, всегда имеет адрес 0. Оно обнаруживается ведущим устройством как новое ведомое устройство без назначения адреса и не включается в нормальную коммуникацию, пока не будет назначен адрес.

Ошибка конфигурирования

Когда желтый светодиод "CER" зажат, то в конфигурации ведомого AS-i устройства есть ошибка. Нажмите кнопку "ACTUAL > EXPECTED", чтобы перезаписать конфигурацию ведомого устройства от модуля ведущего AS-i устройства CM1243-2 на конфигурацию ведомого устройства полевой AS-i шины.

11.5 S7 коммуникации

11.5.1 Инструкции GET и PUT (чтение и запись из удаленного ЦПУ)

Вы можете использовать инструкции GET и PUT, чтобы обмениваться данными с S7 ЦПУ через PROFINET и PROFIBUS соединения. Это возможно только в том случае, если функция "Permit access with PUT/GET communication" активирована для ЦПУ партнера в свойстве "Protection" локального ЦПУ:

- Доступ к данным в удаленном ЦПУ: ЦПУ S7-1200 может использовать только абсолютные адреса в поле ввода ADDR_x, чтобы адресовать переменные удаленных ЦПУ (S7-200/300/400/1200).
- Доступ к данным в стандартном DB: ЦПУ S7-1200 может использовать только абсолютные адреса в поле ввода ADDR_x, чтобы адресовать переменные DB в стандартном DB удаленного S7 ЦПУ.
- Доступ к данным в оптимизированном DB: ЦПУ S7-1200 не может получить доступ к переменным DB в оптимизированном DB удаленного ЦПУ S7-1200.
- Доступ к данным в локальном ЦПУ: ЦПУ S7-1200 может использовать или абсолютные или символьные адреса в качестве вводимых значений для полей ввода RD_x или SD_x инструкций GET или PUT, соответственно.

Примечание

В ЦПУ V4.0 программные операции GET/PUT автоматически не активированы

Программные операции ЦПУ V3.0 GET/PUT автоматически активированы в ЦПУ V4.0.

Однако, программные операции GET/PUT ЦПУ V4.0 автоматически не активированы в ЦПУ V4.0. Вы должны перейти для ЦПУ в "Конфигурацию устройства", окно инспектора вкладки "Properties", свойство "Protection", чтобы разрешить доступ GET/PUT (стр. 197).

Таблица 11- 63 Инструкции GET и PUT

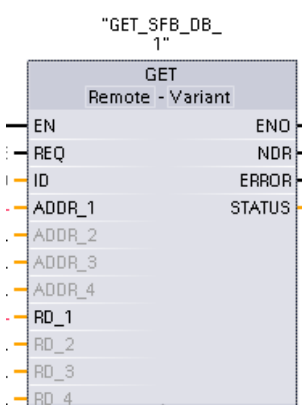
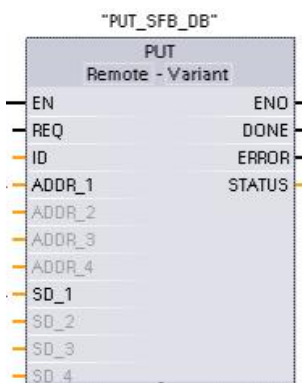
LAD / FBD	SCL	Описание
	<pre> "GET_DB" (req:=_bool_in_, ID:=_word_in_, ndr=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, addr_1:=_remote_inout_, [...addr_4:=_remote_inout_,] rd_1:=_variant_inout_ [...rd_4:=_variant_inout_]); </pre>	Используйте инструкцию GET, чтобы считать данные из удаленного S7 ЦПУ. Удаленный ЦПУ может быть в любом режиме RUN или STOP. STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.
	<pre> "PUT_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, er- ror=>_bool_out_, sta- tus=>_word_out_, addr_1:=_remote_inout_, [...addr_4:=_remote_inout_,] sd_1:=_variant_inout_, [...sd_4:=_variant_inout_]); </pre>	Используйте инструкцию PUT, чтобы записать данные в удаленный S7 ЦПУ. Удаленный ЦПУ может быть в любом режиме RUN или STOP. STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 11- 64 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	Input	Bool	Сигнал от низкого уровня к высокому (положительный фронт) запускает операцию.
ID	Input	CONN_PRG (Word)	ID S7-соединения (Hex)
NDR (GET)	Output	Bool	Новые данные готовы: 0: запрос еще не запущен или все еще обрабатывается 1: задача была выполнена успешно
DONE (PUT)	Output	Bool	DONE: 0: запрос еще не запущен или все еще обрабатывается 1: задача была выполнена успешно

Параметр и тип		Тип данных	Описание
ERROR STATUS	Output Output	Bool Word	- ERROR=0 Значение STATUS: 0000H: ни ошибок, ни предупреждений <> 0000H: Предупреждение, STATUS предоставляет подробную информацию - ERROR=1 Есть ошибка. STATUS предоставляет подробную информацию о характере ошибки
ADDR_1	InOut	Remote	Указатель на области памяти в удаленном ЦПУ, которые хранят данные, которые будут считаны (GET) или отправлены (PUT).
ADDR_2	InOut	Remote	
ADDR_3	InOut	Remote	
ADDR_4	InOut	Remote	
RD_1 (GET) SD_1 (PUT)	InOut	Variant	Указатель на области памяти в локальном ЦПУ, хранящие данные, которые будут считаны (GET, Допустимые типы данных: Bool (допустим только одиночный бит), Byte, Char, Word, Int, DWord, DInt или Real) или отправлены (PUT). Примечание: Если указатель обращается к DB, Вы должны определить абсолютный адрес, такой как: R# DB10.DBX5.0 Byte 10. В этом случае, 10 представляет число байтов, которые будут обработаны с помощью GET или PUT.
RD_2 (GET) SD_2 (PUT)	InOut	Variant	
RD_3 (GET) SD_3 (PUT)	InOut	Variant	
RD_4 (GET) SD_4 (PUT)	InOut	Variant	

Вы должны гарантировать, чтобы длина (число байтов) и типы данных для параметров ADDR_x (удаленный ЦПУ) и RD_x или SD_x (локальный ЦПУ) соответствовали. Число после идентификатора "Byte" является числом байтов, на которые ссылается параметр ADDR_x, RD_x или SD_x.

Примечание

Общее количество байтов, полученных инструкцией GET или общее количество байтов, отправленных инструкцией PUT, ограничено. Ограничения основаны на том, сколько из четырех возможных адресов и областей памяти Вы используете:

- Если Вы используете только ADDR_1 и RD_1/SD_1, инструкция GET может получить 222 байта, а инструкция PUT может отправить 212 байтов.
- Если Вы используете ADDR_1, RD_1/SD_1, ADDR_2 и RD_2/SD_2, инструкция GET может получить в общей сложности 218 байтов, а инструкция PUT может отправить в общей сложности 196 байтов.
- Если Вы используете ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3 и RD_3/SD_3, инструкция GET может получить в общей сложности 214 байтов, а инструкция PUT может отправить в общей сложности 180 байтов.
- Если Вы используете ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3, RD_3/SD_3, ADDR_4, RD_4/SD_4, инструкция GET может получить в общей сложности 210 байтов, а инструкция PUT может отправить в общей сложности 164 байта.

Сумма числа байтов каждого из Ваших параметров адреса и области памяти должна быть меньше или равна определенному пределу. Если Вы превышаете эти пределы, то инструкция GET или PUT возвращает ошибку.

По нарастающему фронту параметра REQ операция чтения (GET), или операция записи (PUT), загружает параметры ID, ADDR_1, и RD_1 (GET), или SD_1 (PUT).

- Для GET: удаленный ЦПУ возвращает запрошенные данные в области приема (RD_x), начиная со следующего скана. Когда операция чтения завершилась без ошибки, параметр NDR устанавливается в 1. Новая операция может быть запущена только после того, как предыдущая операция завершилась.
- Для PUT: локальный ЦПУ начинает отправлять данные (SD_x) в ячейки памяти (ADDR_x) в удаленном ЦПУ. Когда операция записи завершилась без ошибки, удаленный ЦПУ возвращает подтверждение выполнения. Параметр DONE инструкции PUT в этом случае устанавливается в 1. Новая операция записи может быть запущена только после того, как предыдущая операция завершилась.

Примечание

Чтобы гарантировать согласованность данных, всегда оценивайте завершение операции (NDR = 1 для GET, или DONE = 1 для PUT) прежде, чем обращаться к данным или инициировать другую операцию чтения или записи.

Параметры ERROR и STATUS предоставляют информацию о состоянии операции чтения (GET) или записи (PUT).

Таблица 11- 65 Информация об ошибках

ERROR	STATUS (десятичное)	Описание
0	11	• Новое задание не может вступить в силу, так как предыдущее задание еще не завершено. • Задание сейчас обрабатывается в более низком классе приоритета.
0	25	Коммуникация запущена. Задание обрабатывается.
1	1	Коммуникационная проблема, так как: • Описание соединения не загружено (локальный или удаленный) • Прерванное соединение (например: кабель, ЦПУ выключен, или CM/CB/CP находится в режиме STOP), • Соединение с партнером еще не установлено
1	2	Отрицательное подтверждение от устройства партнера. Задача не может быть выполнена.
1	4	Ошибки в указателях на области передачи (RD_x для GET, или SD_x для PUT), включая длину или тип данных.
1	8	Ошибка доступа к ЦПУ партнера
1	10	Доступ к локальной пользовательской памяти невозможен (например, попытка получить доступ к удаленному DB)
1	12	Когда вызывается SFB: • Был определен экземплярный DB, который не принадлежит, GET или PUT • Был определен не экземплярный DB, а глобальный DB • Не найден экземплярный DB (загрузка нового экземплярного DB)

ERROR	STATUS (десятичное)	Описание
1	20	• Превышено максимальное количество параллельных заданий/экземпляров • Экземпляры были перезагружены при переходе ЦПУ в RUN Это состояние возможно для первого выполнения инструкции GET или PUT
1	27	Отсутствует соответствующая инструкция GET или PUT в ЦПУ.

11.5.2 Создание S7-соединения

Механизмы соединения

Чтобы получить доступ к удаленным партнерам по соединению с помощью инструкции PUT/GET, пользователь также должен обладать разрешением.

По умолчанию опция "Permit access with PUT/GET communication" не активирована. В этом случае доступ для чтения и для записи к данным ЦПУ возможен только для коммуникационных соединений, которые требуют конфигурирования или программирования как для локального ЦПУ, так и для коммуникационного партнера. Доступ возможен, например, через инструкции BSEND/BRCV.

Соединения, для которых локальный ЦПУ является только сервером (подразумевается, что никакого сконфигурированного/запрограммированного соединения с коммуникационным партнером не существует в локальном ЦПУ), поэтому во время работы ЦПУ невозможны, например:

- PUT/GET, FETCH/WRITE или FTP доступ через коммуникационные модули
- PUT/GET доступ от других S7 ЦПУ
- HMI доступ через PUT/GET коммуникации

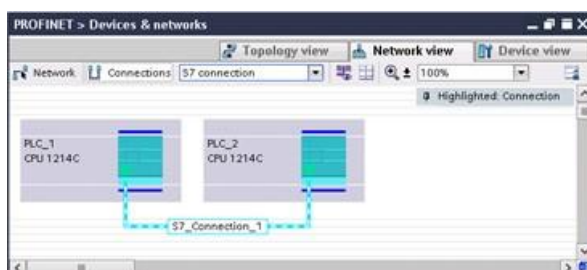
Если Вы хотите предоставить доступ к данным ЦПУ со стороны клиента, т.е. Вы не хотите ограничивать коммуникационные услуги ЦПУ, Вы можете сконфигурировать защиту доступа для ЦПУ S7-1200 (стр. 197) для этого уровня безопасности.

Типы соединения

Тип соединения, который Вы выбираете, создает коммуникационное соединение со станцией партнера. Соединение настраивается, устанавливается, и автоматически контролируется.

В портале "Devices and Networks" используйте представление "Network view", чтобы создать сетевые соединения между устройствами в Вашем проекте. Во-первых, щелкните по вкладке "Connections", а затем выберите тип соединения с помощью раскрывающегося списка, просто вправо (например, S7-соединение). Щелкните по зеленому полю (PROFINET) на первом устройстве и перетащите линию к полю PROFINET на втором устройстве. Отпустите кнопку мыши, и Ваше PROFINET соединение будет организовано.

Обратитесь к теме "Создание сетевого соединения" (стр. 618) для получения дополнительной информации.



Щелкните по кнопке "Highlighted: Connection", чтобы получить доступ к конфигурационному диалоговому окну "Properties" коммуникационной инструкции.

11.5.3 Конфигурирование пути соединения локальный/партнерский между двумя устройствами

Конфигурирование основных параметров

Вы определяете коммуникационные параметры в конфигурационном диалоговом окне "Properties" коммуникационной инструкции. Это диалоговое окно появляется внизу страницы каждый раз, когда Вы выбрали любую часть инструкции.

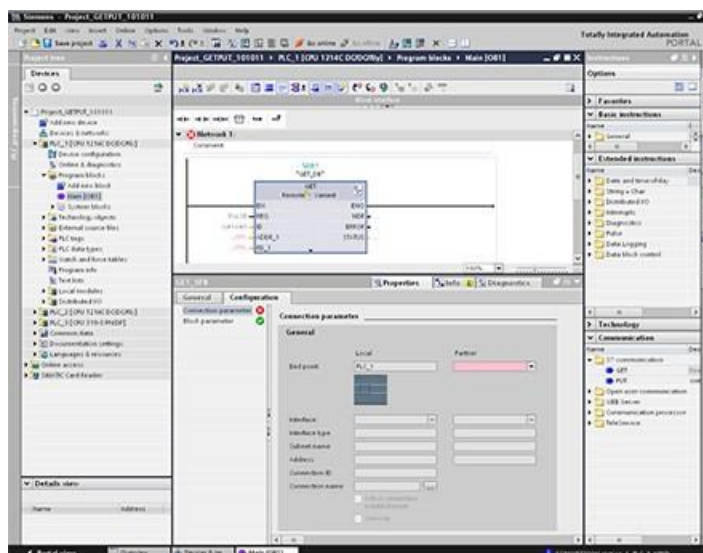
Обратитесь к теме "Конфигурация устройства: Конфигурирование пути соединения локальный/партнерский" (стр. 619) для получения дополнительной информации.

В разделе "Address Details" диалогового окна параметров соединения Вы определяете TSAP или порты, которые будут использоваться. TSAP или порт соединения в ЦПУ вводится в поле "Local TSAP". TSAP или порт, назначенный для соединения в Вашем партнерском ЦПУ, вводится в поле "Partner TSAP".

11.5.4 Назначение параметра соединения GET/PUT

Назначение параметра соединения инструкций GET/PUT является средством для пользователя по конфигурированию S7 коммуникационного соединения ЦПУ-ЦПУ.

STEP 7 после вставки блока GET или PUT отображает диалоговое окно назначения параметра соединения для инструкций GET/PUT:



Окно инспектора отображает свойства соединения каждый раз, когда Вы выбрали любую часть инструкции. Вы можете сконфигурировать коммуникационные параметры на вкладке "Configuration" для "свойств" коммуникационной инструкции.

Примечание

Программные операции GET/PUT ЦПУ V4.1 и выше автоматически не активированы

Программные операции ЦПУ V3.0 GET/PUT автоматически активированы в ЦПУ V4.1 и выше.

Однако, программные операции GET/PUT в ЦПУ V4.1 и выше автоматически не активированы. Вы должны перейти для ЦПУ в "Конфигурацию устройства", окно инспектора вкладку "Properties", свойство "Protection", чтобы разрешить доступ GET/PUT (стр. 197).

11.5.4.1 Параметры соединения

Страница "Connection parameters" позволяет Вам сконфигурировать необходимое S7-соединение и параметр "Connection ID", на который ссылается параметр блока GET/PUT "ID". Страница содержит информацию о локальной конечной точке и позволяет Вам определять локальный интерфейс. Вы можете также определить конечную точку партнера.

Страница "Block parameters" позволяет Вам сконфигурировать дополнительные параметры блока.

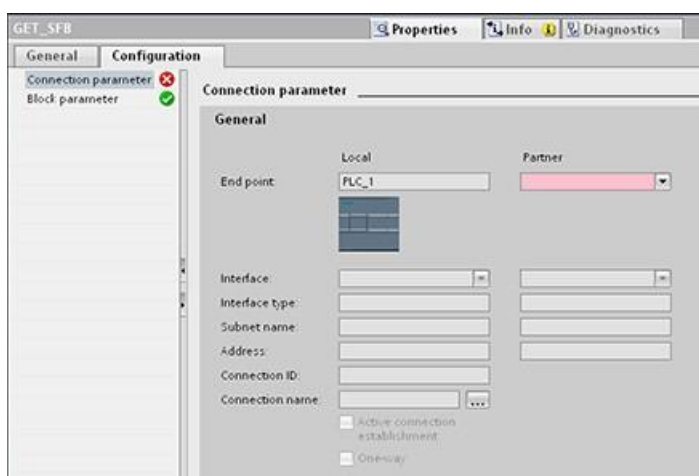


Таблица 11- 66 Параметры соединения: Основные определения

Параметр	Определение	
Connection parameter: General	End point	"Local End point": Имя, назначенное локальному ЦПУ "Partner End point": Имя, назначенное партнерскому (удаленному) ЦПУ Примечание: В выпадающем списке "Partner End point", система отображает всех потенциальных партнеров по S7-соединению текущего проекта, а также опцию "unspecified". Неуказанный партнер представляет собой коммуникационного партнера, который не описан в настоящее время в проекте STEP 7 (например, коммуникационный партнер в виде стороннего устройства).
	Interface	Имя, назначенное интерфейсам Примечание: Вы можете изменить соединение, изменив локальный и партнерский интерфейсы
	Interface type	Тип интерфейса
	Subnet name	Имя, назначенное подсетям
	Address	Назначенные IP адреса Примечание: Вы можете определить удаленный адрес стороннего устройства для "неуказанного" коммуникационного партнера.
	Connection ID	Номер ID: Автоматически генерируется при назначении параметра соединения GET/PUT
	Connection name	Расположение хранилища данных в локальном и партнерском ЦПУ: Автоматически генерируется при назначении параметра соединения GET/PUT
	Active connection establishment	Флажок выбора локального ЦПУ, как активного участника соединения

Параметр		Определение
	One-way	Флажок для определения одностороннего или двухстороннего соединения; только для чтения Примечание: В PROFINET GET/PUT соединении, как локальное, так и партнерское устройства могут действовать как сервер или клиент. Это позволяет двухстороннее соединение, и флажок "One-way" снят. В PROFIBUS GET/PUT соединениях, в некоторых случаях, устройство партнера может действовать только как сервер (например, S7-300), и флажок "One-way" установлен.

Параметр Connection ID

Есть три способа изменить определенные системой ID соединения:

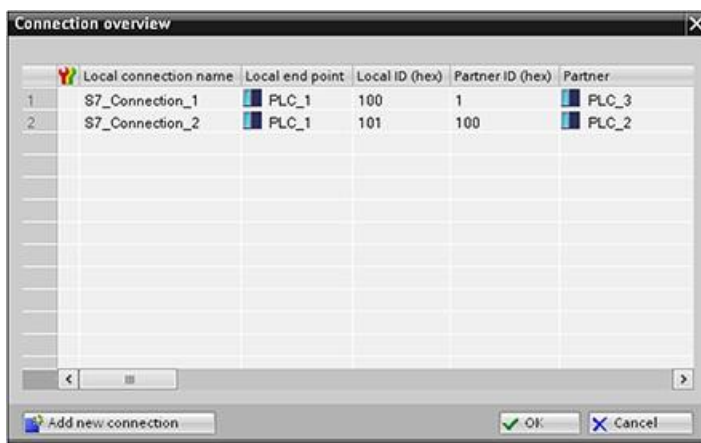
1. Пользователь может изменить текущий ID непосредственно на блоке GET/PUT. Если новый ID принадлежит уже существующему соединению, соединение изменится.
2. Пользователь может изменить текущий ID непосредственно на блоке GET/PUT, но новый ID еще не существует. Новое S7-соединение создается системой.
3. Пользователь может изменить текущий ID через диалоговое окно "Connection overview": ввод данных пользователем синхронизируется с параметром ID на соответствующем блоке GET/PUT.

Примечание

Параметр "ID" блока GET/PUT не является именем подключения, а числовым выражением, которое записывается как следующий пример: W#16#1

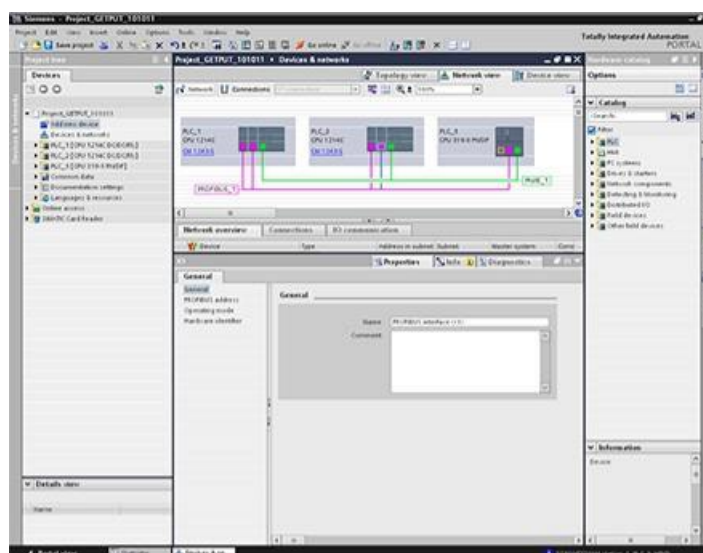
Параметр Connection name

Имя подключения доступно для редактирования посредством специального пользовательского органа управления, диалогового окна "Connection overview". Это диалоговое окно предлагает все доступные S7-соединения, которые могли бы быть выбраны, как альтернатива для существующих GET/PUT коммуникаций. Пользователь может создать абсолютно новое соединение в этой таблице. Нажмите кнопку справа от поля "Connection name", чтобы открыть диалоговое окно "Connection overview".



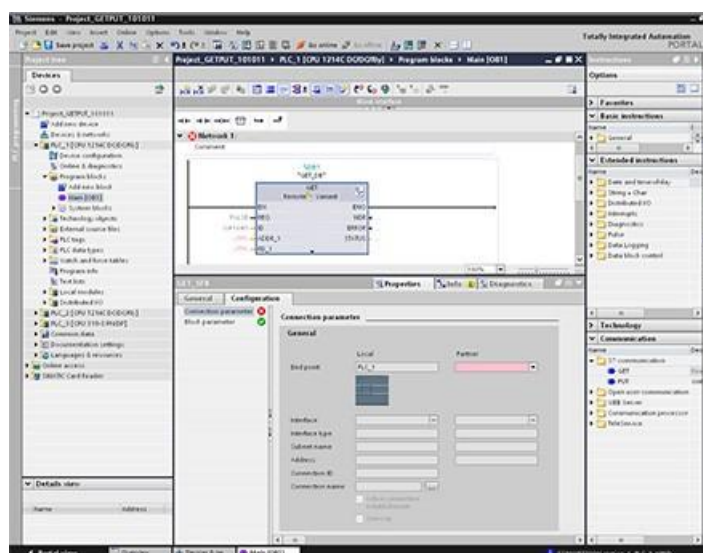
11.5.4.2 Конфигурирование S7-соединения ЦПУ-ЦПУ

Учитывая конфигурацию PLC_1, PLC_2 и PLC_3, как показано на рисунке ниже, вставьте блоки GET или PUT для "PLC_1".



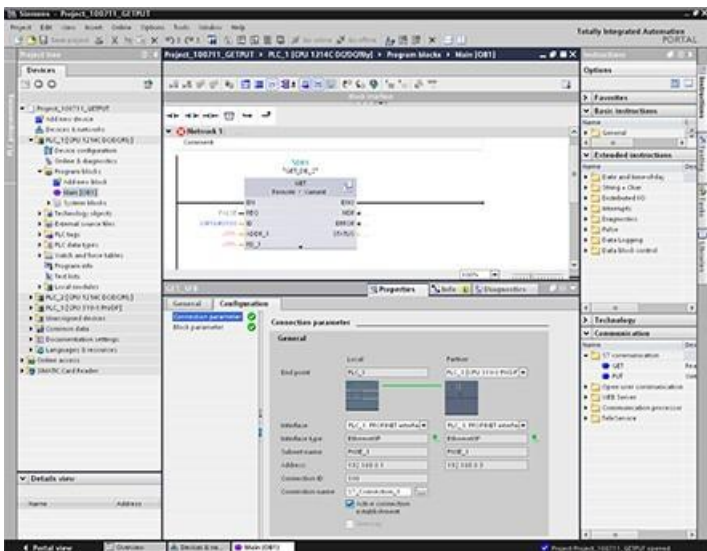
Для инструкции GET или PUT, вкладка "Properties" автоматически отображается в окне инспектора со следующими элементами меню:

- "Configuration"
- "Connection parameters"



Конфигурирование PROFINET S7-соединения

Для "Partner End point" выберите "PLC_3".



Система реагирует следующими изменениями:

Таблица 11- 67 Параметры соединения: Значения в разделе General

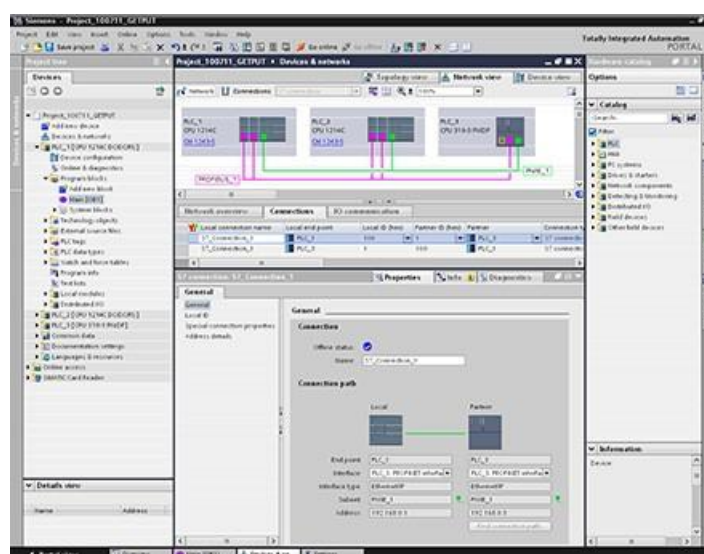
Параметр		Определение
Connection parameter: General	End point	"Local End point" содержит "PLC_1", "только для чтения". "Partner End point" содержит "PLC_3[CPU319-3PN/DP]": - Цвет переключается из красного на белый - Показано изображение устройства "Partner" . - Соединительная линия появляется между изображениями устройств PLC_1-и PLC_3 (зеленая линия Ethernet).
	Interface	"Local Interface" содержит "CPU1214C DC/DC/DC, PROFINET interface (R0/S1)". "Partner Interface" содержит: "CPU319-3PN/DP, PROFINET interface (R0/S2)".
	Interface type	"Local Interface type" содержит "Ethernet/IP"; "только для чтения". "Partner Interface type" содержит "Ethernet/IP"; "только для чтения". Изображения типа интерфейса показаны справа от локального и партнерского "Типа интерфейса " (зеленый значок Ethernet).
	Subnet name	"Local Subnet name" содержит "PN/IE_1"; "только для чтения". "Partner Subnet name" содержит "PN/IE_1"; "только для чтения".
	Address	"Local Address" содержит локальный IP адрес; "только для чтения". "Partner Address" содержит партнерский IP адрес; "только для чтения".
	Connection ID	"Connection ID" содержит "100". В программном редакторе, в блоке Main [OB1] вызов блока GET/PUT также содержит значение "Connection ID" равное "100".

Параметр	Определение
Connection name	"Connection name" содержит имя соединения по умолчанию (например, "S7 Connection 1"); изменения возможны.
Active connection establishment	Флажок может быть установлен и позволяет выбрать локальный ЦПУ в качестве активного участника соединения.
One-way	Находится в состоянии "только для чтения" и флажок снят. Примечание: "PLC_1" (S7-1200 ЦПУ 1214CDC/DC/Rly) и "PLC_3" (S7-300 ЦПУ 319-3PN/DP) могут оба действовать и как сервер, и как клиент в PROFINET GET/PUT соединении, позволяя двухстороннее соединение.

Значок GET/PUT в дереве просмотра свойств также меняет цвет с красного на зеленый.

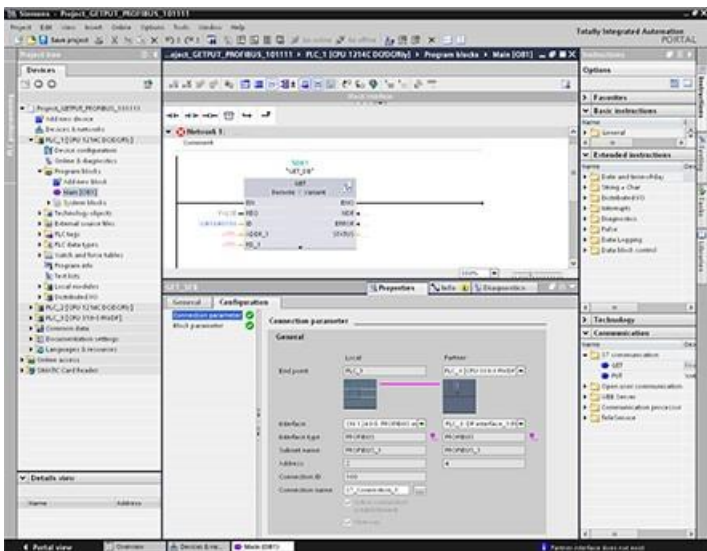
Завершенное PROFINET S7-соединение

В представлении "Network view" двухстороннее S7-соединение между "PLC_1" и "PLC_3" показано в таблице "Connections".



Конфигурирование PROFIBUS S7-соединения

Для "Partner End point" выберите "PLC_3".



Система реагирует следующими изменениями:

Таблица 11- 68 Параметры соединения: Значения в разделе General

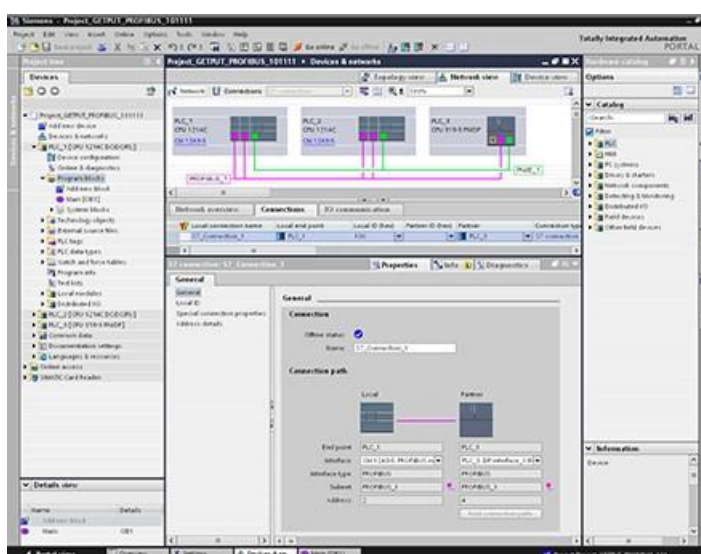
Параметр		Определение
Connection parameter: General	End point	"Local End point" содержит "PLC_1", "только для чтения". "Partner End point" содержит "PLC_3 [CPU319-3PN/DP]": - Цвет переключается из красного на белый - Показано изображение устройства "Partner" . - Соединительная линия появляется между изображениями устройств PLC 1-и PLC_3 (фиолетовая линия PROFIBUS).
	Interface	"Local Interface" содержит "CPU1214C DC/DC/DC, PROFINET interface (R0/S1)". "Partner Interface" содержит: "CPU319-3PN/DP, PROFIBUS interface (R0/S2)".
	Interface type	"Local Interface type" содержит "PROFIBUS "; "только для чтения". "Partner Interface type" содержит "PROFIBUS "; "только для чтения". Изображения типа интерфейса показаны справа от локального и партнерского "Типа интерфейса " (фиолетовый значок PROFIBUS).
	Subnet name	"Local Subnet name" содержит "PROFIBUS _1"; "только для чтения". "Partner Subnet name" содержит " PROFIBUS _1"; "только для чтения".
	Address	"Local Address" содержит локальный IP адрес; "только для чтения". "Partner Address" содержит партнерский IP адрес; "только для чтения".
	Connection ID	"Connection ID" содержит "100". В программном редакторе, в блоке Main [OB1] вызов блока GET/PUT также содержит значение "Connection ID" равное "100".

Параметр	Определение
Connection name	"Connection name" содержит имя соединения по умолчанию (например, "S7 Connection 1"); изменения возможны.
Active connection establishment	Флажок может быть установлен и позволяет выбрать локальный ЦПУ в качестве активного участника соединения.
One-way	Находится в состоянии "только для чтения" и флажок снят. Примечание: "PLC_3" (S7-300 ЦПУ319-3PN/DP) может действовать только как сервер (не может быть клиентом) в PROFIBUS GET/PUT соединении, позволяя только одностороннее соединение.

Значок GET/PUT в дереве просмотра свойств также меняет цвет с красного на зеленый.

Завершенное PROFIBUS S7-соединение

В представлении "Network view" одностороннее S7-соединение между "PLC_1" и "PLC_3" показано в таблице "Connections".



Веб-сервер

Веб-сервер для S7-1200 обеспечивает доступ посредством веб-страницы к данным Вашего ЦПУ и данным техпроцесса.

Вы можете получить доступ к Веб-страницам S7-1200 из ПК или из мобильного устройства. Веб-сервер отображает страницы в формате и размере, совместимом с устройством, которое Вы используете для доступа. Веб-сервер поддерживает минимальное разрешение 240 x 240 пикселей.

Вы используете Веб-браузер, чтобы обратиться к IP-адресу ЦП S7-1200 или адресу IP-адресу разрешенного веб-сервером CP модуля (коммуникационный процессор) (стр. 793) в локальной стойке ЦПУ для установления соединения. S7-1200 поддерживает несколько одновременных соединений.



Стандартные веб-страницы

S7-1200 содержит стандартные Веб-страницы, к которым Вы можете обратиться из Веб-браузера Вашего ПК (стр. 791) или из мобильного устройства (стр. 792):

- Введение (стр. 800) - отправная точка для стандартных Веб-страниц
- Начальная страница (стр. 801) - общая информация о ЦПУ
- Идентификация (стр. 802) - подробная информация о ЦПУ, включая серийный, заказной номера и номер версии
- Информация о модуле (стр. 804) - информация о модулях в локальной стойке и возможности обновить встроенное ПО
- Коммуникации (стр. 807) - информация о сетевых адресах, физических свойствах коммуникационных интерфейсов и коммуникационной статистике
- Диагностический Буфер (стр. 803) - диагностический буфер
- Состояние переменных (стр. 808) - переменные ЦПУ и ввод-вывод, доступный по адресу или по имени ПЛК тега
- Файловый браузер (стр. 810) - браузер для файлов, хранящихся внутри ЦПУ или на карте памяти, например, журналы данных и рецепты
- Вход в систему (стр. 796) - вход в систему под другим пользователем или выход из системы.

Эти страницы встроены в ЦПУ S7-1200 и доступны на английском, немецком, французском, испанском, итальянском и упрощенном китайском. Некоторые страницы требуют дополнительных пользовательских полномочий (стр. 789), которые Вы конфигурируете в STEP 7 для того, чтобы просматривать страницы. Для получения дополнительной информации о стандартных Веб-страницах, и о том, как получить доступ к ним, обратитесь к разделу Стандартные Веб-страницы (стр.794).

Примечание

ЦПУ S7-1200 не содержат отдельную стандартную Веб-страницу обновления встроенного ПО. Функция обновления встроенного ПО включена в страницу информации о модуле.

Пользовательские Веб-страницы

S7-1200 также предоставляет поддержку по созданию Вами пользовательских Веб-страниц, которые могут получить доступ к данным ЦПУ. Вы можете разработать эти страницы с помощью ПО для создания HTML-документов по Вашему выбору и включать готовые "AWP" команды (Automation Web Programming) в Ваш HTML-код, чтобы получить доступ к данным ЦПУ. Обратитесь к главе Пользовательские Веб-страницы (стр. 813) для получения определенной информации о разработке пользовательских Веб-страниц и связанным с ней конфигурированием и программированием в STEP 7.

Вы можете получить доступ к пользовательским страницам посредством ПК или мобильного устройства из стандартных Веб-страниц.

Требования к веб-браузеру

Веб-сервер поддерживает следующие веб-браузеры для ПК:

- Internet Explorer 8.0
- Internet Explorer 9.0
- Mozilla Firefox 17.0.1
- Google Chrome 23.0
- Apple Safari 5.1.7 (Windows)
- Apple Safari 6.0.2 (Mac)

Веб-сервер поддерживает следующие веб-браузеры мобильных устройств:

- Internet Explorer 6.0 и ранее для HMI панелей
- Mobile Safari 7534.48.3 (iOS 5.0.1)
- Mobile Android Browser 2.3.4
- Mobile Google Chrome 23.0

Для получения информации о связанных с браузерами ограничениях, которые могут помешать отображению стандартных или пользовательских веб-страниц, см. раздел Ограничения (стр. 855).

12.1 Активация веб-сервера

Вы активируете веб-сервер в STEP 7 из Конфигурации устройства для ЦПУ, к которому Вы намереваетесь подключиться.

Чтобы активировать веб-сервер, выполните следующие шаги:

1. Выберите CPU в представлении Device Configuration.
2. В окне инспектора выберите "Web server" в свойствах ЦПУ.
3. Установите флажок для "Activate web server on all modules of this device".
4. Для повышения безопасности выберите "Permit access only with HTTPS", чтобы затребовать безопасный доступ к веб-серверу.

ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к ЦПУ через веб-сервер

Несанкционированный доступ к ЦПУ или изменение переменных ПЛК на недопустимые значения могли бы нарушить работу процесса и привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку активация веб-сервера позволяет авторизованным пользователям выполнять изменения рабочего режима, запись в данные ПЛК и обновления встроенного ПО, Сименс рекомендует, чтобы Вы соблюдали следующие меры безопасности:

- Разрешите доступ к веб-серверу только по протоколу HTTPS.
- Защитите сильными паролями идентификаторы пользователей веб-сервера (стр. 789). Сильные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.
- Не расширяйте минимальные по умолчанию полномочия пользователя веб-сервера "Everybody".
- Выполняйте проверку ошибок и проверку диапазонов Ваших переменных в Вашей программной логике, так как пользователи Веб-страницы могут изменить переменные ПЛК на недопустимые значения.
- Используйте безопасную Виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу ПЛК S7-1200, находясь вне Вашей защищенной сети.

После того, как Вы загрузите конфигурацию устройства, Вы можете использовать стандартные веб-страницы, чтобы получить доступ к ЦПУ. Если Вы выбираете "Enable" для "Automatic update", стандартные веб-страницы обновляются каждые десять секунд.

Если Вы создали и активировали пользовательские веб-страницы (стр. 813), Вы можете получить доступ к ним из стандартного меню веб-страницы.

Примечание

Замена устройства: замена ЦПУ V3.0 на ЦПУ V4.1

Если Вы заменяете существующий ЦПУ V3.0 на ЦПУ V4.1 (стр. 1287) и конвертируете Ваш проект V3.0 в проект V4.1, обратите внимание на то, что STEP 7 и ЦПУ V4.1 сохраняют параметры настройки веб-сервера для:

- "Activate web server on this module"
 - "Permit access only with HTTPS"
-

Примечание

Если выполняется "Загрузка в RUN" (стр. 1087), то стандартные и пользовательские веб-страницы не обновляют значения данных и не разрешают Вам изменять значения данных, пока загрузка не завершена. Веб-сервер отклоняет любые попытки записать значения данных во время выполнения загрузки.

Активация других языков для веб-сервера

Вы можете также дополнительно выбрать другие языки для отображения стандартных веб-страниц. Выберите "User interface languages" на вкладке Properties Вашей конфигурации устройства, и Вы можете в этом случае назначить в качестве языка проекта STEP 7 один из шести языков, которые поддерживает веб-сервер. После того, как Вы загрузите конфигурацию устройства, стандартные веб-страницы обеспечивают селектор для языка пользовательского интерфейса. Если Вы не выбираете языки, то выбором по умолчанию является английский язык.

12.2 Конфигурирование пользователей веб-сервера

Вы можете конфигурировать пользователей с различными уровнями привилегий для доступа к ЦПУ через веб-сервер.

Чтобы сконфигурировать пользователей веб-сервера и связанные с ними привилегии, выполните следующие шаги:

1. Выберите ЦПУ в представлении Конфигурации Устройства.
2. В окне инспектора выберите "Web server" в свойствах ЦПУ и активируйте веб-сервер (стр. 787).
3. Выберите "User management" в свойствах веб-сервера.
4. Введите имена пользователей, уровни доступа и пароли для сеансов пользователей, которых Вы хотите создать.

После того, как Вы загружаете конфигурацию в ЦПУ, только зарегистрированные пользователи могут получить доступ к функциям веб-сервера, для которых у них есть привилегии.

Уровни доступа к веб-серверу

STEP 7 обеспечивает пользователя по умолчанию с именем "Everybody" без пароля. По умолчанию этот пользователь не имеет никаких дополнительных привилегий и может просматривать только стандартные веб-страницы Старт (стр. 801) и Введение (стр. 800). Вы можете, однако, предоставить дополнительные привилегии пользователю "Everybody", а также другим пользователям, которых Вы конфигурируете:

- Диагностика посредством запросов
- Чтение тегов
- Запись тегов
- Чтение статуса тега
- Запись статуса тега
- Открывание пользовательских страниц
- Запись в пользовательские страницы
- Чтение файлов
- Запись/удаление файлов
- Изменение рабочего режима
- Мигание светодиодами
- Выполнение обновления встроенного ПО
- Изменение системных параметров
- Изменение параметров приложения

⚠ ПРЕДУПРЕЖДЕНИЕ**Доступ к веб-сервер**

Предоставление привилегий пользователю "Everybody" позволяет авторизоваться на веб-сервер без пароля. Несанкционированный доступ к ЦПУ или изменение ПЛК переменных на недействительные значения могли бы нарушить работу процесса и привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку пользователь "Everybody", когда ему предоставлены достаточные привилегии может выполнить изменения рабочего режима, запись данных ПЛК и обновления встроенного ПО без пароля, Сименс рекомендует, чтобы Вы соблюдали следующие меры безопасности:

- Разрешите доступ к веб-серверу только по протоколу HTTPS.
- Защитите сильными паролями идентификаторы пользователей веб-сервера (стр. 789). Сильные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.
- Не расширяйте минимальные по умолчанию полномочия пользователя веб-сервера "Everybody".
- Выполняйте проверку ошибок и проверку диапазонов Ваших переменных в Вашей программной логике, так как пользователи Веб-страницы могут изменить переменные ПЛК на недопустимые значения.
- Используйте безопасную Виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу ПЛК S7-1200, находясь вне Вашей защищенной сети.

12.3 Доступ к веб-страницам от ПК

Вы можете получить доступ к стандартным веб-страницам S7-1200 из ПК или из мобильного устройства с помощью IP-адреса ЦПУ S7-1200 или IP-адреса любого разрешенного веб-сервером CP (стр. 793) в локальной стойке.

Чтобы получить доступ к стандартным веб-страницам S7-1200 из ПК, выполните следующие шаги:

1. Обеспечьте, чтобы S7-1200 и ПК находились в общей Ethernet-сети или соединены друг с другом непосредственно с помощью стандартного Ethernet-кабеля.
2. Откройте веб-браузер и войдите в URL "https://www.xx.yy.zz", где "www.xx.yy.zz" соответствует IP-адресу ЦПУ S7-1200 или IP-адресу CP в локальной стойке.

Веб-браузер открывает страницу Введение.

Примечание

Используйте безопасную Виртуальную частную сеть (VPN), чтобы соединиться с веб-сервером S7-1200 ПЛК, находясь вне Вашей защищенной сети. Примите также к сведению любые ограничения (стр. 855), которые могли бы налагать Ваша Веб-среда или операционная система.

В качестве альтернативы, Вы можете адресовать определенную стандартную веб-страницу своего веб-браузера. Чтобы сделать это, введите URL в виде "https://www.xx.yy.zz/<page>.html", где <page> соответствует одной из стандартных веб-страниц:

- start (стр. 801) - общая информация о ЦПУ
- identification (стр. 802) - подробная информация о ЦПУ, включая серийный, заказной номера и номер версии
- module (стр. 804) - информация о модулях в локальной стойке и возможность обновить встроенное ПО
- communication (стр. 807) - информация о сетевых адресах, физических свойствах коммуникационных интерфейсов и коммуникационной статистике
- diagnostic (стр. 803) - диагностический буфер
- variable (стр. 808) - переменные ЦПУ и ввод/вывод, доступные по адресу или имени ПЛК тега
- filebrowser (стр. 810) - браузер для доступа к файлам системного журнала данных или файлам рецептов, хранящимся внутри ЦПУ или на карте памяти
- index (стр. 800) - вводная страница, чтобы перейти к стандартным веб-страницам
- login (стр. 796) - страница для входа под другим пользователем или выхода. (Обратите внимание на то, что окно входа доступно из каждой стандартной веб-страницы ПК, но страница login необходима для входа из мобильного устройства.)

Например, если Вы вводите "https://www.xx.yy.zz/communication.html", браузер отображает страницу коммуникаций.

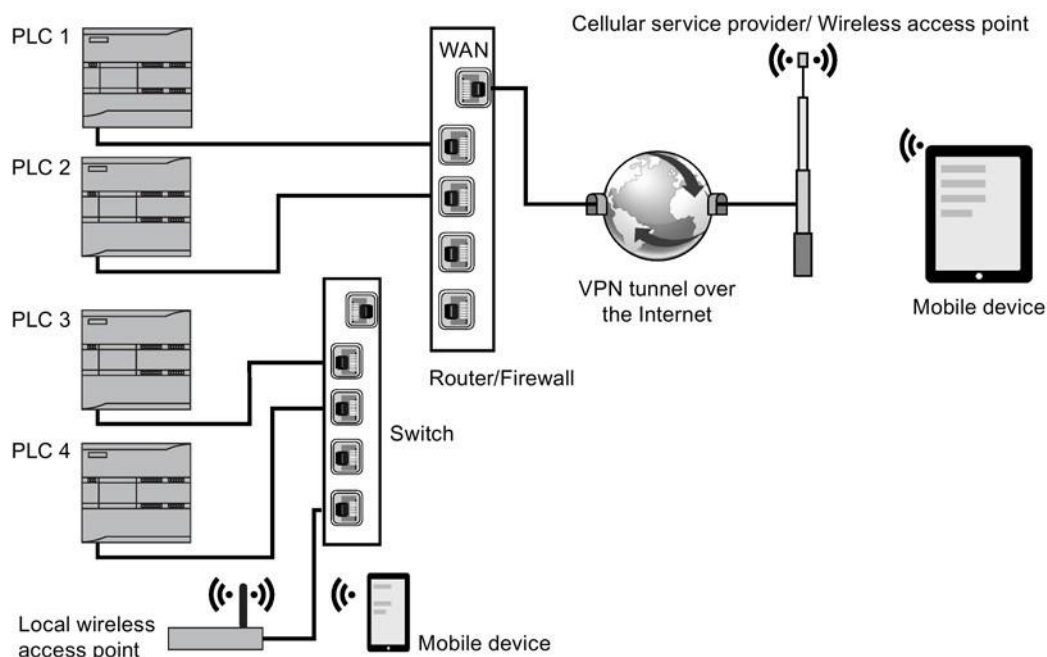
Защищенный доступ

Используйте безопасную Виртуальную частную сеть (VPN), чтобы соединиться с веб-сервером ПЛК S7-1200, находясь вне Вашей защищенной сети. Предусмотрите и используйте `https://` вместо `http://` для защищенного доступа (стр. 787) к стандартным веб-страницам. Когда Вы соединяетесь с S7-1200 посредством `https://`, веб-сайт шифрует сессию с помощью цифрового сертификата. Веб-сервер передает данные защищено, и они недоступны для просмотра кем-либо. Вы, как правило, получаете предупреждение системы безопасности, что Вы можете подтвердить с помощью "Да" переход к стандартным веб-страницам. Чтобы избежать предупреждения системы безопасности при каждом защищенном доступе, Вы можете импортировать сертификат ПО Сименс своему веб-браузеру (стр. 857).

12.4 Доступ к веб-страницам из мобильного устройства

Чтобы получить доступ к S7-1200 из мобильного устройства, Вы должны соединить свой ПЛК с сетью, которая соединяется с Интернетом или с локальной беспроводной точкой доступа. Используйте безопасную Виртуальную частную сеть (VPN), чтобы соединить мобильное устройство с веб-сервером ПЛК S7-1200. Вы можете использовать переадресацию порта в беспроводном маршрутизаторе, чтобы привязать IP-адрес ПЛК к адресу, по которому мобильное устройство может получить доступ к нему из Интернета. Чтобы сконфигурировать перенаправление порта, следуйте инструкциям по конфигурированию ПО Вашего маршрутизатора. Вы можете соединить столько ПЛК и переключающих устройств сколько поддерживает Ваш маршрутизатор.

Без перенаправления порта Вы можете соединиться с ПЛК, но только локально в пределах зоны действия радиосигнала.



В этом примере мобильное устройство, которое находится в зоне действия локальной беспроводной точки доступа, может соединиться с ПЛК 3 и ПЛК 4 по их IP-адресам. Из Интернета, вне зоны действия локальной беспроводной точки доступа мобильное устройство может соединиться с ПЛК 1 и ПЛК 2, используя перенаправленный адрес для каждого ПЛК.

Чтобы получить доступ к стандартным веб-страницам, у Вас должен быть доступ к сотовой связи или беспроводной точке доступа. Чтобы получить доступ к ПЛК из Интернета, введите адрес порта перенаправления в веб-браузере Вашего мобильного устройства, чтобы получить доступ к ПЛК, например `http://www.xx.yy.zz:rrrr` или `https://www.xx.yy.zz:rrrr`, где `www.xx.yy.zz` - адрес маршрутизатора и `rrrr`, назначение порта для определенного ПЛК.

Для локального доступа через беспроводную точку доступа введите IP-адрес ЦПУ S7-1200 или разрешенного веб-сервером CP (стр. 793) в локальной стойке: `http://www.xx.yy.zz` или `https://www.xx.yy.zz`. Вы можете также перейти к определенной веб-странице по имени, как описано в Доступе к веб-страницам из ПК (стр. 791).

Для повышенной безопасности сконфигурируйте только безопасный доступ (HTTPS) к веб-серверу (стр. 787).

12.5 Использование CP модуля для доступа к веб-страницам

Независимо от того, получаете ли Вы доступ к веб-серверу из ПК или мобильного устройства, Вы можете соединиться со стандартными веб-страницами через один из следующих CP модулей, если Вы сконфигурировали его в STEP 7 и установили его в локальной стойке с ЦПУ S7-1200:

- CP 1242-7 GPRS V2
- CP 1243-7 LTE-EU

Вы используете стандартную веб-страницу Start (стр. 801), чтобы получить доступ к веб-страницам через эти CP модули. Страница Start показывает все сконфигурированные и установленные CP модули, которые находятся в Вашей локальной стойке, но Вы можете получить доступ к веб-страницам только из упомянутых выше модулей.

Примечание

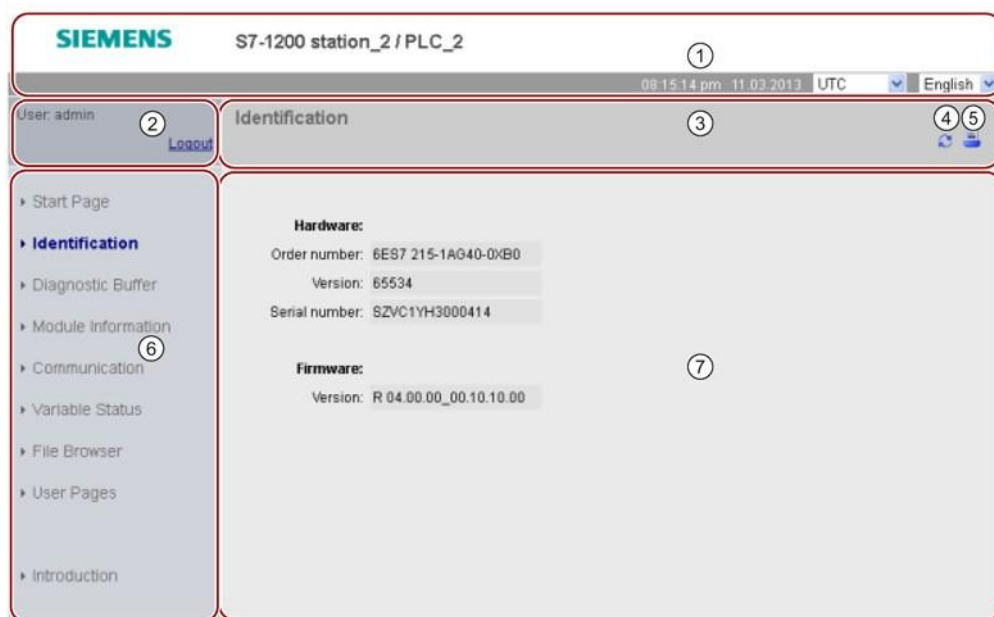
Доступ к стандартным веб-страницам, когда разрешенные веб-сервером CP находятся в локальной стойке

Вы могли бы наблюдать задержки до одной или двух минут, соединяясь со стандартными веб-страницами S7-1200, когда разрешенные веб-сервером CP находятся в локальной стойке. Если Вам кажется, что страницы недоступны, или Вы получаете ошибки, просто подождите одну или две минуты и обновите страницу.

12.6 Стандартные веб-страницы

12.6.1. Компоновка стандартных веб-страниц

Все стандартные веб-страницы S7-1200 имеют общую компоновку с навигационными ссылками и элементы управления страницами. Независимо от того, просматриваете ли Вы страницу на ПК или на мобильном устройстве, каждая страница имеет ту же самую область содержимого, но расположение и органы навигации варьируются на основании размера экрана и разрешении устройства. На стандартном ПК или большом мобильном устройстве компоновка стандартной веб-страницы выглядит следующим образом:



- ① Заголовок веб-сервера с селектором для отображения локального времени ПЛК или времени UTC и селектора языка экрана (стр. 162)
- ② Авторизация или завершение сеанса
- ③ Стандартный заголовок веб-страницы с именем страницы, которую Вы просматриваете. В данном случае - это страница идентификации ЦПУ. Некоторые стандартные веб-страницы, такие как информация о модуле, также отображают здесь путь, если можно получить доступ к нескольким экранам этого типа.
- ④ Символ обновления: для страниц с автоматическим обновлением, разрешает или запрещает функцию автоматического обновления; для страниц без автоматического обновления, вызывает обновление страницы с текущими данными
- ⑤ Символ печати: готовит и показывает версию для печати информации, доступной на отображаемой странице
- ⑥ Область навигации для переключения на другую страницу
- ⑦ Область содержимого для определенной стандартной веб-страницы, которую Вы просматриваете. В данном случае - это страница идентификации ЦПУ.

Компоновка для мобильного устройства

В устройстве с шириной меньше 768 пикселей, веб-сервер отображает мобильную версию каждой страницы. Страница не содержит области навигации, авторизации и заголовка, и содержит кнопки для продвижения назад и вперед по веб-страницам и кнопку Домашней страницы, которая отправляет Вас к странице Navigation. Вы можете также использовать органы навигации, предоставляемые Вашим мобильным устройством. Например, на мобильном устройстве с шириной экрана меньше чем 768 пикселей страница Identification отображается следующим образом с вертикальной ориентацией:



Обратите внимание на то, что иллюстрации к стандартным веб-страницам в этой главе показывают стандартное представление веб-страницы на ПК. У каждой стандартной веб-страницы есть эквивалентное мобильное представление страницы.

Примечание

Стандартные веб-страницы CP модуля

Определенные CP модули (стр. 793) обеспечивают стандартные веб-страницы, которые подобны по внешнему представлению и функциональности стандартным веб-страницам ЦПУ S7-1200. Обратитесь к своей документации по CP для получения описаний стандартных веб-страниц CP.

12.6.2. Авторизация и привилегии пользователей

Каждая из стандартных веб-страниц ПК обеспечивает окно авторизации над областью навигации. По соображениям экономии пространства мобильные веб-страницы предлагают отдельную страницу авторизации. S7-1200 поддерживает многопользовательскую авторизацию с различными уровнями доступа (полномочиями):

- Диагностика посредством запросов
- Чтение тегов
- Запись тегов
- Чтение статуса тега
- Запись статуса тега
- Открывание пользовательских страниц
- Запись в пользовательские страницы
- Чтение файлов
- Запись/удаление файлов
- Изменение рабочего режима
- Мигание светодиодами
- Выполнение обновления встроенного ПО
- Изменение системных параметров
- Изменение параметров приложения

Вы определяете пользовательские роли, связанные с ними уровни доступа (полномочия) и пароли (стр. 789) в свойствах управления пользователями веб-сервера в STEP 7 конфигурации ЦПУ.

Авторизация

STEP 7 предоставляет пользователя по умолчанию с именем "Everybody" без пароля. По умолчанию этот пользователь не имеет никаких дополнительных полномочий и может только просматривать стандартные веб-страницы Start (стр. 801) и Introduction (стр. 800). Вы можете, однако, предоставить дополнительные полномочия пользователю "Everybody", а также другим пользователям, которых Вы конфигурируете:

ПРЕДУПРЕЖДЕНИЕ

Доступ к веб-сервер

Предоставление привилегий пользователю "Everybody" позволяет авторизоваться на веб-сервер без пароля. Несанкционированный доступ к ЦПУ или изменение ПЛК переменных на недействительные значения могли бы нарушить работу процесса и привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку пользователь "Everybody", когда ему предоставлены достаточные привилегии может выполнить изменения рабочего режима, запись данных ПЛК и обновления встроенного ПО без пароля, Сименс рекомендует, чтобы Вы соблюдали следующие меры безопасности:

- Разрешите доступ к веб-серверу только по протоколу HTTPS.
- Защитите сильными паролями идентификаторы пользователей веб-сервера (стр. 789). Сильные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Держите пароль в секрете и изменяйте его часто.
- Не расширяйте минимальные по умолчанию полномочия пользователя веб-сервера "Everybody".
- Выполняйте проверку ошибок и проверку диапазонов Ваших переменных в Вашей программной логике, так как пользователи Веб-страницы могут изменить переменные ПЛК на недопустимые значения.
- Используйте безопасную Виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу ПЛК S7-1200, находясь вне Вашей защищенной сети.

Чтобы выполнить определенные действия, такие как изменение рабочего режима контроллера, запись значений в память и обновления встроенного ПО ЦПУ, у Вас должны быть требуемые полномочия. Обратите внимание на то, что, если Вы установили уровень защиты ЦПУ (стр. 197) на "Complete protection (no access)", пользователь "Everybody" не может получить доступ к веб-серверу.



Авторизуйтесь во фрагменте рядом с левым верхним углом на каждой из стандартных веб-страниц, когда она отображается на экране ПК или широкого мобильного устройства.



Авторизуйтесь на отдельной странице для небольших мобильных устройств, которая доступна с домашней страницы.

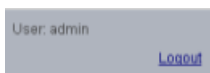
Чтобы зарегистрироваться выполните следующие шаги:

1. Введите имя пользователя в поле Username.
2. Введите пароль в поле Password.

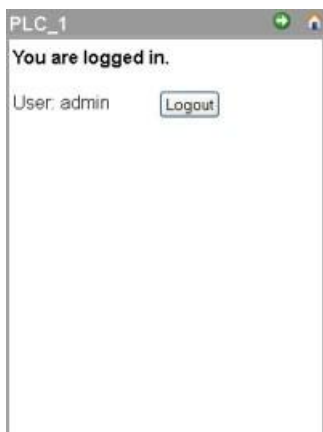
Срок Вашей авторизации истекает после отсутствия активности в течение тридцати минут. Если загруженная в настоящее время страница постоянно обновляется, срок авторизации не истекает.

Если Вы испытываете какие-либо проблемы при авторизации, возвратитесь к странице Introduction (стр. 800) и загрузите сертификат безопасности Siemens (стр. 857). Вы можете затем авторизоваться без ошибок.

Регистрация выхода



Чтобы зарегистрировать выход из системы, просто щелкните по ссылке "Logout" на любой странице при просмотре на ПК или широком мобильном устройстве.



Для небольших мобильных устройств перейдите к странице регистрации от домашней страницы и коснитесь кнопки "Logout".

После того, как Вы выполните выход, Вы можете только обратиться и просмотреть стандартные Веб-страницы согласно полномочиям пользователя "Everybody". Каждое из описаний стандартных веб-страниц определяет требуемые полномочия для этой страницы.

Примечание

Регистрируйте выход, прежде чем закрывать веб-сервер

Если Вы вошли авторизовались на веб-сервере, убедитесь, что Вы зарегистрировали выход из системы до закрытия Вашего Веб-браузера. Веб-сервер поддерживает максимум семь параллельных авторизаций.

12.6.3. Страница Introduction

Страница Introduction - это экран приветствия для входа в стандартные веб-страницы S7-1200.



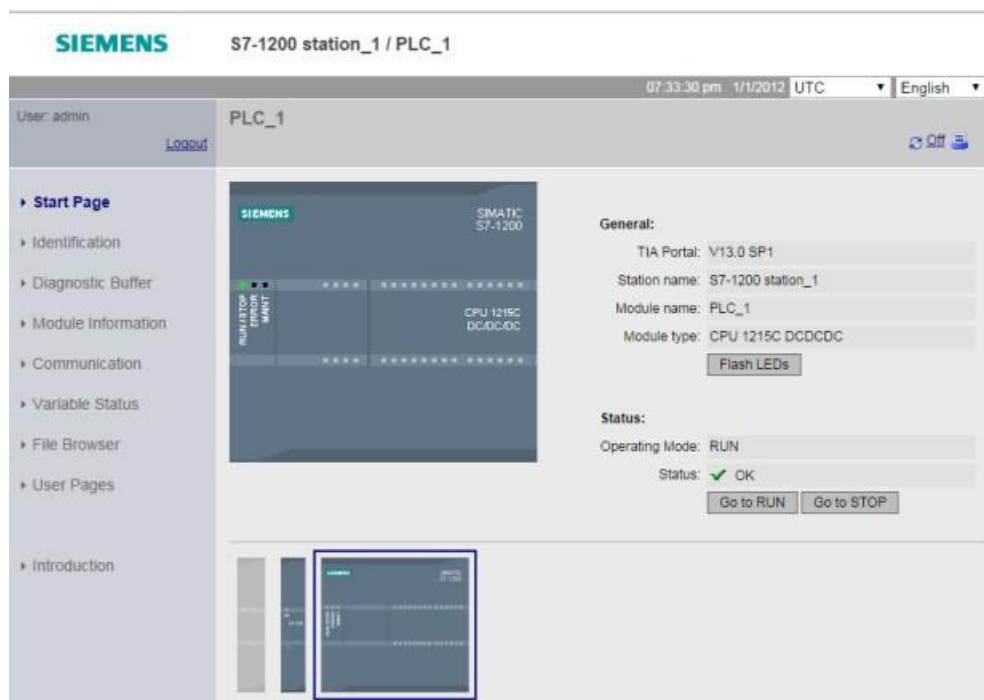
Из этой страницы Вы нажимаете "Enter", чтобы получить доступ к стандартным Веб-страницам S7-1200. Вверху экрана находятся ссылки на полезные веб-ресурсы Siemens, а также ссылка для загрузки сертификата безопасности Siemens (стр. 857). Вы можете также принять решение и пропускать вводную страницу при будущих обращениях к веб-серверу.

12.6.4. Страница Start

Страница Start содержит представление ЦПУ или CP, с которым Вы соединены, и приводит общую информацию об устройстве. Для ЦПУ Вы можете использовать кнопки, чтобы изменить рабочий режим и заставить мигать светодиоды, если Вы авторизовались (стр. 796) с определенными полномочиями.

Нижняя часть экрана является видимой, если Вы сконфигурировали и установили разрешенный на веб-сервере CP модуль (стр. 793) в локальной стойке ЦПУ S7-1200. Вы можете навести курсор и щелкнуть по разрешенному в веб-сервере CP модулю, чтобы получить доступ к стандартным Веб-страницам. Обратитесь к документации для Вашего CP модуля для получения информации о его веб-страницах. Вы видите имя CP модуля, когда Вы наводите на него курсор.

Веб-сервер также показывает любые другие CM и CP модули в локальной стойке, но Вы не можете щелкнуть по ним, поскольку они не содержат веб-страниц. Представление модуля для этих CM и CP светло-серое (неактивное), чтобы указать на то, что они только отображаются и не активируются по щелчку мыши.

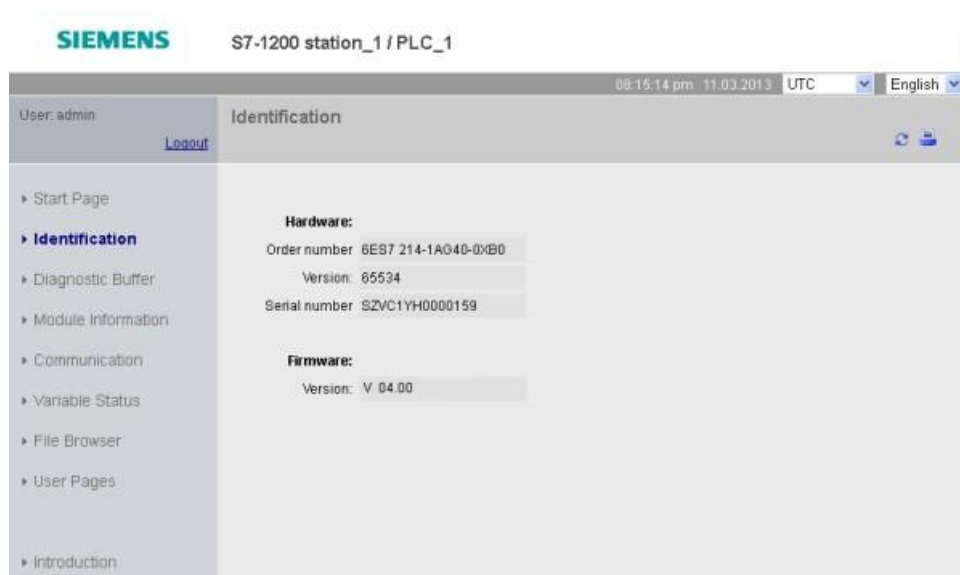


Обратите внимание на то, что отказоустойчивые центральные процессоры S7-1200 отображают на этой странице дополнительные данные, связанные с функциональной безопасностью.

12.6.5. Страница Identification

Страница Identification отображает индивидуальные характеристики ЦПУ:

- Серийный номер
- Заказной номер
- Номер версии



Просмотр страницы Identification требует привилегии "диагностика посредством запросов" (стр. 789).

12.6.6. Страница Diagnostic Buffer

Страница Diagnostic buffer отображает диагностические события. С помощью селектора слева, Вы можете выбрать диапазон отображаемых записей в диагностическом буфере, 1 - 25 или 26 - 50. С помощью селектора справа, Вы можете выбрать, отображаемое время: UTC или локальное время ПЛК. Верхняя часть страницы показывает диагностические записи со временем и датой связанного с ними события.

Вверху страницы Вы можете выбрать любую отдельную запись, чтобы показать подробную информацию об этой записи в нижней части страницы.

SIEMENS S7-1200 station_1 / PLC_1

User: admin Logout 08:15:14 pm 11.03.2013 UTC English

Diagnostic Buffer

Diagnostic buffer entries 1-25

Number	UTC Time	UTC Date	Event
1	08:43:47:939 pm	25.02.2013	CPU info: Follow-on operating mode change
2	08:43:47:836 pm	25.02.2013	CPU info: Follow-on operating mode change
3	08:43:47:734 pm	25.02.2013	CPU info: Follow-on operating mode change
4	08:43:46:789 pm	25.02.2013	CPU info: Power on
5	08:43:38:783 pm	25.02.2013	CPU info: Power off
6	08:43:22:748 pm	25.02.2013	CPU info: Follow-on operating mode change
7	08:43:22:645 pm	25.02.2013	CPU info: Communication initiated request: WARM RESTART
8	08:43:22:645 pm	25.02.2013	CPU info: New startup information

Details: 1 Event ID: 16# 02:400C

CPU info: Follow-on operating mode change
Power-on mode set: WARM RESTART to RUN (if CPU was in RUN before power off)

Pending startup inhibit(s):
- No startup inhibit set

> CPU changes from STARTUP to RUN mode

Просмотр страницы Diagnostic Buffer требует привилегии "диагностика посредством запросов" (стр. 789).

12.6.7. Страница Module Information

Страница Module information предоставляет информацию обо всех модулях в локальной стойке. Верхний раздел экрана отображает сводку модулей, а нижний раздел показывает состояние, идентификационные данные и информацию о версии ПО выбранного модуля. Страница Module information также обеспечивает возможность выполнить обновление встроенного ПО.

Просмотр страницы Module Information требует привилегии "диагностика посредством запросов" (стр. 789).

Страница Module information: вкладка Status

Вкладка Status в нижнем разделе страницы Module information показывает описание текущего статуса модуля, который выбран в верхнем разделе.



SIEMENS S7-1200 station_1 / PLC_1

User: admin Logout 05.08.02 am 18.01.1970 UTC English

Module Information

S7-1200 station_1 - S7-1200 station_1 Slot Filter

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	PLC 1	Details 6ES7 214-1AG40-0XB0	0	0	
2	✓	DI16/DQ16 x 24VDC 1	Details 6ES7 223-1BL30-0XB0	8	8	

Status Identification Firmware

The module is parameterized, and it exchanges data with the CPU. The module did not detect any parameter assignment errors, channel errors, or hardware related errors.

Примечание

Страница Module information в мобильном устройстве отображает "I адрес", "Q адрес" и информацию "Comment" на вкладке Identification, а не столбцы из основной таблицы информации о модуле.

Развертывание информации

Вы можете выбрать ссылку в верхнем разделе, чтобы развернуть информацию об определенном модуле. У модулей с субмодулями есть ссылки для каждого субмодуля. Тип выводимой информации меняется в зависимости от выбранного модуля. Например, диалоговое окно информации о модуле первоначально отображает имя станции SIMATIC 1200, индикатор состояния и комментарий. Если Вы выполняете развертку для ЦПУ, то выводится информации об именах модулей цифрового и аналогового ввода и вывода, которые содержит ЦПУ (например, "DI14/DO10", "AI2"), адреса ввода-вывода, индикаторы состояния, номера слотов и комментарии.

Slot	Status	Name	Order number	I address	Q address	Comment
1.16	✓	HSC_1	Details 6ES7 214-1AG40-0XB0	1000	---	
1.17	✓	HSC_2	Details 6ES7 214-1AG40-0XB0	1004	---	
1.18	✓	HSC_3	Details 6ES7 214-1AG40-0XB0	1008	---	
1.19	✓	HSC_4	Details 6ES7 214-1AG40-0XB0	1012	---	
1.20	✓	HSC_5	Details 6ES7 214-1AG40-0XB0	1016	---	
1.21	✓	HSC_6	Details 6ES7 214-1AG40-0XB0	1020	---	
1.2	✓	AI2_1	Details 6ES7 214-1AG40-0XB0	64	---	
1.1	✓	DI14/DO10_1	Details 6ES7 214-1AG40-0XB0	0	0	
1.32	✓	Pulse_1	Details 6ES7 214-1AG40-0XB0	---	1000	

При развертывании информации страница информации о модуле показывает путь, по которому Вы следовали. Вы можете щелкнуть по любой ссылке в этом пути, чтобы вернуться на более высокий уровень.



Поля сортировки

Когда в списке представлено несколько модулей, Вы можете щелкнуть по заголовку столбца поля, чтобы отсортировать его вверх или вниз по этому полю.

Примечание: Эта функция еще не доступна для страницы Module Information на китайском языке.

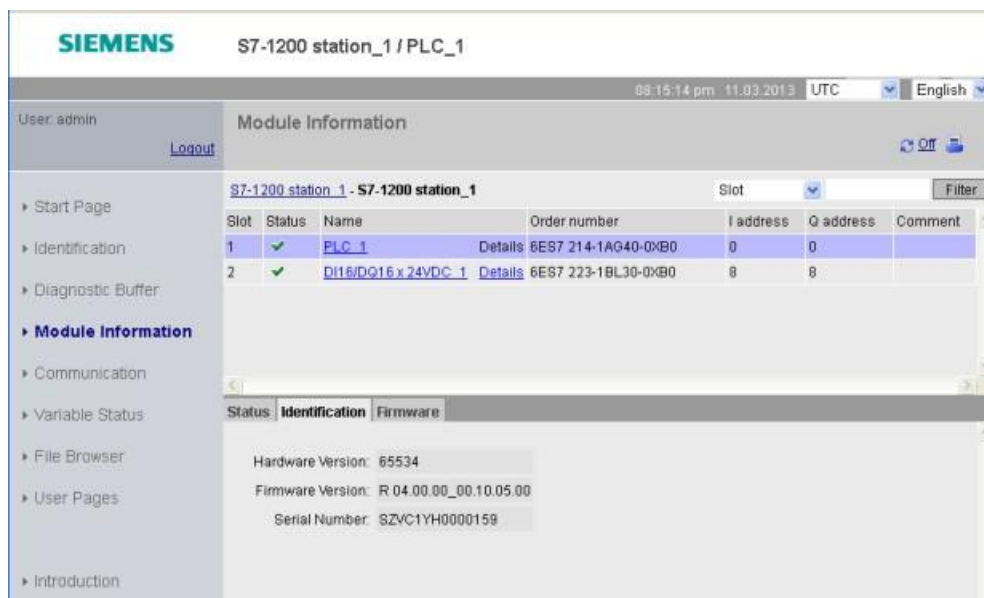
Name ▼	
AI2_1	Details
DI14/DO10_1	Details
HSC_1	Details
HSC_2	Details
HSC_3	Details
HSC_4	Details
HSC_5	Details
HSC_6	Details
Pulse_1	Details

Фильтрация информации о модуле

Вы можете фильтровать любое поле в списке информации о модуле. Из выпадающего списка выберите имя поля, для которого Вы хотите фильтровать данные. Введите текст в связанном текстовом поле и щелкните по ссылке Filter. Список обновляется, чтобы показать Вам модули, которые соответствуют Вашим критериям фильтрации.

Страница Module information: вкладка Identification tab

Вкладка Identification отображает серийный номер и номер версии выбранного модуля.



Страница Module information: вкладка Firmware

Вкладка Firmware страницы Module information отображает информации о модуле о встроенном ПО выбранного модуля. Если Вы обладает привилегией "выполнение обновления ПО" (стр. 789), Вы можете также выполнить обновление встроенного ПО ЦПУ или других модулей в локальной стойке, которые поддерживают обновление.

Примечание

Вы можете обновить с помощью функции Update Firmware только ЦПУ S7-1200 версии 3.0 и выше.



ЦПУ должен быть в режиме STOP, чтобы выполнить обновление встроенного ПО. Когда ЦПУ будет в режиме STOP, щелкните по кнопке просмотра, чтобы перейти и выбрать файл встроенного ПО. Обновления встроенного ПО доступны на веб-сайте поддержки клиентов (<http://support.automation.siemens.com>).

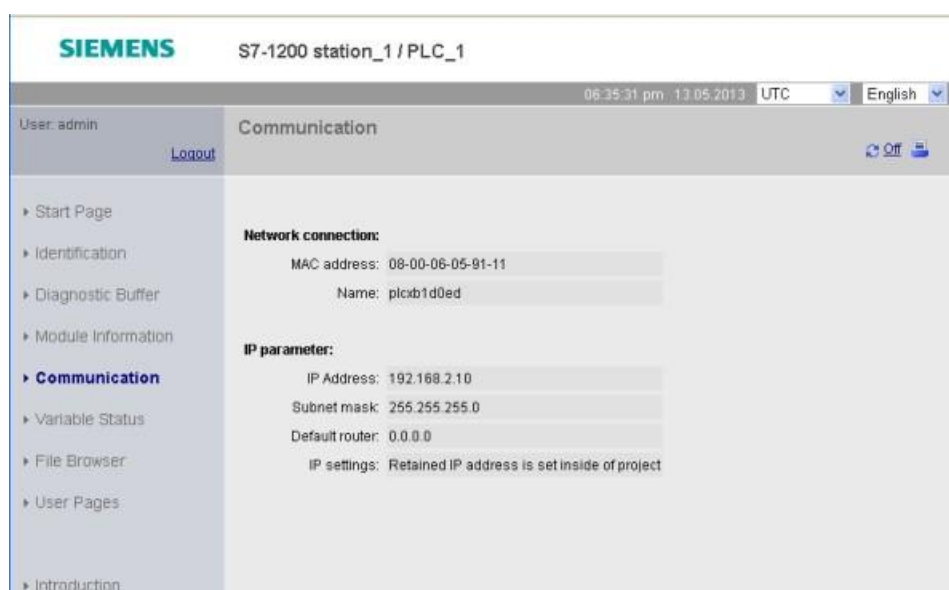
Во время обновления, страница отображает сообщение о ходе обновления. После того, как обновление завершено, страница отображает артикульный номер и номер версии обновленного встроенного ПО. Если Вы обновили встроенное ПО для ЦПУ или сигнальной платы, веб-сервер перезапускает ЦПУ.

Примечание

Вы можете также выполнить обновление встроенного ПО из STEP 7 (стр. 1074) или при помощи карты памяти (стр. 141).

12.6.8. Страница Communication

Страница Communication отображает параметры подключенного ЦПУ, включая MAC-адрес, IP-адрес и IP-настройки ЦПУ.



Просмотр страницы Communication требует привилегии "диагностика посредством запросов".

12.6.9. Страница Variable Status

Страница Variable Status позволяет Вам просматривать любой ввод-вывод или данные памяти в Вашем ЦПУ. Вы можете ввести абсолютный адрес (такой как I0.0), имя тега ПЛК или тег из определенного блока данных. Для тегов блока данных Вы включаете имя блока данных в двойные кавычки. Для каждого контролируемого значения Вы можете выбрать формат отображения для данных. Вы можете продолжать вводить и определять значения, пока Вам позволяют ограничения для страницы. Контролируемые значения обнаруживаются автоматически и обновляются по умолчанию, если Вы не щелкаете по значку "Off" в верхней правой части страницы. Когда обновление остановлено, Вы можете щелкнуть "On", чтобы повторно активировать автоматическое обновление.

Просмотр страницы Variable Status требует привилегий "чтение состояния переменной". Для возможности просмотра тега на странице Variable Status, Вы должны сконфигурировать его как "Accessible from HMI" в STEP 7.

Если Вы авторизуетесь как пользователь с привилегией "запись состояния переменной" (стр. 796), Вы также можете изменять значения данных. Введите любые значения, которые Вы хотите задать в соответствующем поле "Modify Value". Нажмите кнопку "Go" рядом со значением, чтобы записать все значение в ЦПУ. Вы можете также ввести несколько значений и нажать "Modify All Values", чтобы записать все значения в ЦПУ. Кнопки и метки столбцов для изменения появляются только в том случае, если у Вас есть привилегия "запись состояния переменной".

SIEMENS S7-1200 station_1 / PLC_1

User: admin Logout

08:15:14 pm 11.03.2013 UTC English

Variable Status

Enter the address of a tag here which you want to monitor/modify

Address / Display Format	Monitor Value / Modify Value	Modify
Q0.1 BOOL	true	Go
I0.1 BOOL	false	Go
Conveyor_speed DEC	145	Go
Mixer_On BOOL	false	Go
'Data block 1' location CHARACTER	16#00000000	Go
Tag_1 FLOATING POINT	17.2	Go
New variable BIN		

Monitor Value Modify All Values

Если Вы покидаете страницу Variable Status и возвращаетесь к ней, то Ваши записи не сохраняются. Вы можете отметить страницу и возвратиться к закладке, чтобы просматривать те же записи. Если Вы не отмечаете страницу, Вы должны повторно ввести переменные.

Примечание

Обратите внимание на следующие проблемы при использовании стандартной страницы Variable Status:

- Закрывайте все измененные значения для строк в одиночные кавычки.
 - Страница Variable Status не может контролировать или изменять теги, которые содержат любой из следующих символов: &, <, (, +, ,(запятая), [,], \$ или %. Например, Вы не можете контролировать тег "Clock_2.5Hz".
 - Страница Variable Status не позволяет Вам изменять строку длиннее 198 символов.
 - При использовании экспоненциального представления, чтобы ввести значение для типа данных Real или LReal на странице Variable Status:
 - Вводите значение вещественного числа (Real или LReal) с положительной экспонентой (такое как +3.402823e+25) в любом из следующих форматов:
+3.402823e25
+3.402823e+25
 - Вводите значение вещественного числа (Real или LReal) с отрицательной экспонентой (такое как +3.402823e-25) следующим образом:
+3.402823e-25
 - Убедитесь, что часть мантиссы вещественного значения в экспоненциальной записи включает десятичную точку. Отказ добавить десятичную точку приводит к изменению значения на неожиданное целочисленное значение. Например, введите-1.0e8, а не-1e8.
 - Страница Variable Status поддерживает только 15 цифр для значения LReal (независимо от положения десятичной точки). Ввод больше чем 15 цифр дает погрешность округления.
-

Ограничения для страницы Variable Status:

- Максимальное количество записей переменных на страницу составляет 50.
 - Максимальное количество символов для URL, соответствующего странице Variable Status, является 2083. Вы видите URL, который представляет Вашу текущую страницу переменных в поисковой строке Вашего браузера.
 - Для формата отображения character страница показывает шестнадцатеричные значения, если фактические значения ЦПУ не являются допустимыми ASCII символами, что интерпретируется браузером.
-

Примечание

Если имя тега содержит специальные символы, которые отклоняются при записи на странице Variable Status, Вы можете заключить имя тега в двойные кавычки. В большинстве случаев страница Variable Status тогда распознает имя тега.

12.6.10. Страница File Browser

Страница File Browser обеспечивает доступ к файлам во внутренней загрузочной памяти ЦПУ или на карте памяти (внешняя загрузочная память). Страница файлового браузера первоначально отображает корневую папку загрузочной памяти, которая содержит папки "DataLogs" и "Recipes", но также и выводит на экран любые другие папки, которые Вы, возможно, создали при использовании карты памяти.

Ваш тип доступа к файлам и папкам, зависит от Ваших пользовательских привилегий (стр. 789). Любой пользователь с привилегиями "чтения файлов" может просмотреть файлы и папки с помощью файлового браузера. Вы не можете удалить папку DataLogs или папку Recipes независимо от Ваших привилегий авторизации, но если Вы создали пользовательские папки на карте памяти, то Вы можете удалить эти папки, если Вы авторизовались в качестве пользователя с привилегиями "записи/удаления файлов".

Щелкните по папке, чтобы получить доступ к отдельным файлам в папке.



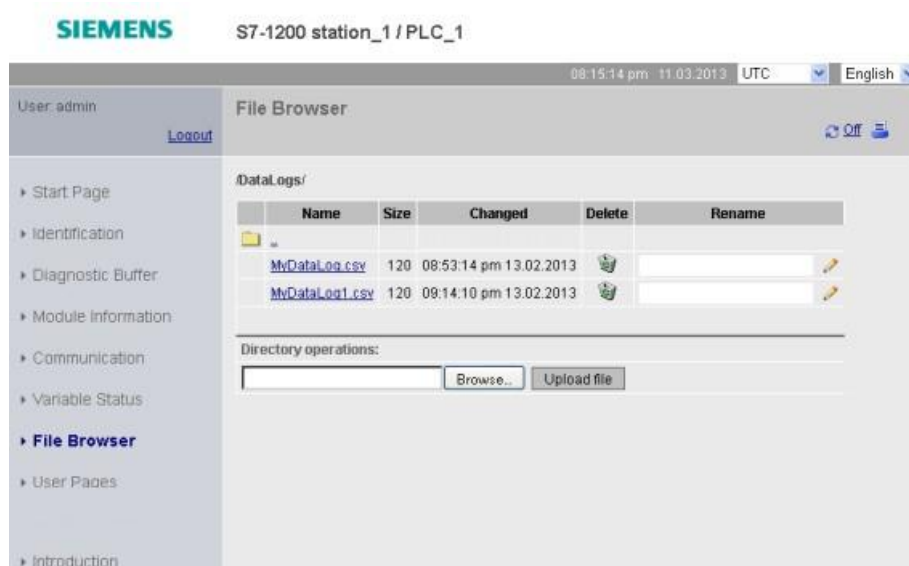
Папка Data logs

Из папки "Data Logs" Вы можете открыть любой из файлов журнала данных. Если Вы авторизовались с привилегией "записи/удаление файлов" (стр. 789), Вы можете также удалить, переименовать и загрузить файлы. Файлы журнала данных хранятся в формате файла с разделением значений запятыми (CSV). Вы можете сохранить их на своем компьютере или открыть их в Microsoft Excel (по умолчанию) или другой программе.

Примечание

Метки времени для регистрации данных

Веб-сервер отображает метки времени для регистрации данных в формате UTC или как локальное время ПЛК в зависимости от Вашего выбора в верхней части страницы.



Примечание: Опции "Delete" и "Rename" недоступны, если Вы не авторизовались с привилегией "записи/удаление файлов".

Примечание

Управление журналами данных

Храните не более 1000 журналов данных в файловой системе. Превышение этого числа может препятствовать тому, чтобы веб-сервер имел достаточно ресурсов ЦПУ для отображения журналов данных.

Если Вы замечаете, что веб-страница файлового браузера не в состоянии отобразить журналы данных, то Вы должны перевести ЦПУ в режим STOP, чтобы отобразить и удалить журналы данных.

Управляйте своими журналами данных, чтобы гарантировать, чтобы Вы сохранили только необходимое количество, и не превышаете значение в 1000 журналов данных.

Работа с журналом данных в Excel

Файл журнала данных имеет формат USA/UK с разделением значений запятыми (CSV). Чтобы открыть его с помощью Excel в не USA/UK системах, Вы должны импортировать его в Excel с определенными настройками (стр. 859).

Файлы рецептов

Как и папка журналов данных, папка рецептов отображает любые файлы рецептов, которые присутствуют в загрузочной памяти. Файлы рецептов также хранятся в формате CSV, и Вы можете открыть их в Microsoft Excel или другой программе. Как и для журналов данных, Вы должны обладать привилегиями для того, чтобы удалить, изменить и сохранить, переименовать или загрузить файлы рецептов.

Загрузка файлов и автоматическое обновление страницы

Если Вы начинаете загрузку файла, операция загрузки продолжается, пока Вы остаетесь на веб-странице файлового браузера. Если Вы разрешили автоматическое обновление страниц веб-сервера каждые десять секунд, то каждый раз, когда происходит обновление страницы, Вы видите инкрементный индикатор операции загрузки файла. Например, если Вы загружаете файл на 2 Мб, Вы могли бы видеть обновления, которые показывают размер файла в байтах в 2500, 5000, 10000, 15000, и 20000, во время загрузки файла.

Если Вы оставляете страницу File Browser, прежде чем загрузка завершится, Вы не получите полный файл. Когда Вы возвращаетесь страница File Browser отображает имя файла и размер файла на момент остановки загрузки. Вы не видите других указаний на то, что файл неполный. Чтобы быть уверенным, что Вы выгружаете полный файл, оставайтесь на странице File Browser, пока отображаемый размер файла не достигнет фактического размера файла.

Дополнительная информация

Для получения информации о программировании инструкций журналов данных, а также импорта (стр. 417) и экспорта (стр. 415) рецептов, смотри главу Рецепты и журналы данных (стр. 411).

12.7 Пользовательские веб-страницы

Веб-сервер S7-1200 также предоставляет Вам средства для создания Ваших собственных связанных с приложением HTML-страниц, которые включают в себя данные ПЛК.

ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к ЦПУ через пользовательские веб-страницы

Несанкционированный доступ к ЦП через пользовательские веб-страницы мог бы нарушить работу процесса, что в свою очередь могло бы привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

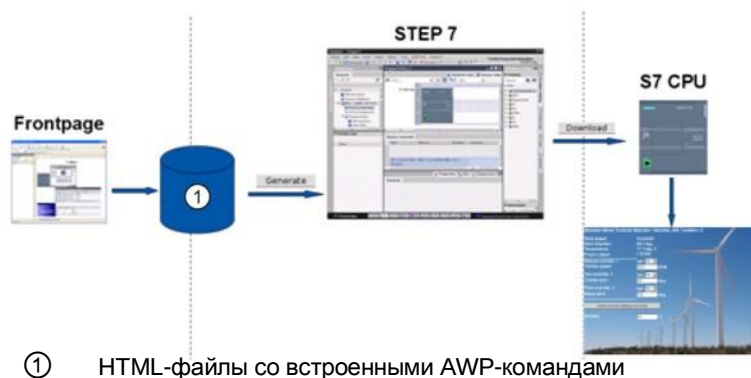
Небезопасное кодирование пользовательских веб-страниц представляет уязвимости с точки зрения безопасности, такие как межсайтовый скриптинг (XSS), внедрение кода и другие.

Защитите свой ЦПУ S7-1200 от несанкционированного доступа, введя его в эксплуатацию безопасным образом, как описано в Практических рекомендациях, которые можно найти веб-сайте Промышленной безопасности (<http://www.siemens.com/industrialsecurity>).

Вы создаете пользовательские веб-страницы, используя HTML-редактор по Вашему выбору и загружаете их в ЦПУ, где они доступны из меню стандартных веб-страниц. Этот процесс включает в себя несколько задач:

- Создание HTML-страниц с помощью HTML-редактора, такого как Microsoft Frontpage (стр. 814)
- Включение AWP-команд в HTML-комментарии в HTML-коде (стр. 815): AWP-команды - это фиксированный набор команд, который Siemens предусматривает для доступа к информации о ЦПУ.
- Конфигурирование STEP 7 для чтения и обработки HTML-страниц (стр. 830)
- Генерация блоков из HTML-страниц (стр. 830)
- Программирование STEP 7 для управления использованием HTML-страниц (стр. 831)
- Компиляция и загрузка блоков в ЦПУ (стр. 832)
- Доступ к пользовательским веб-страницам от ПК (стр. 833)

Этот процесс проиллюстрирован ниже:



① HTML-файлы со встроенными AWP-командами

12.7.1. Создание HTML-страниц

Вы можете использовать пакет программного обеспечения по Вашему выбору, чтобы создавать Ваши собственные HTML-страницы для использования с веб-сервером. Убедитесь, что Ваш HTML-код совместим с HTML-стандартами W3C (Консорциум World Wide Web). STEP 7 не выполняет проверку Вашего HTML-синтаксиса.

Вы можете использовать пакет программного обеспечения, который позволяет Вам проектировать в WYSIWYG или проектировать режим компоновки, но Вам необходимо редактировать свой код HTML в чистой HTML-форме. Большинство веб-инструментов разработки обеспечивает этот тип редактирования; в противном случае, Вы можете всегда использовать простой текстовый редактор, чтобы отредактировать HTML-код. Включайте следующую строку в свою HTML-страницу, чтобы выбрать для страницы набор символов UTF-8:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Также убедитесь, что сохраняете файл в редакторе в кодировке UTF-8.

Вы используете STEP 7, чтобы скомпилировать все содержимое Ваших HTML-страниц в блоки данных STEP 7. Эти блоки данных состоят из одного управляющего блока, который руководит отображением веб-страниц и одного или более блоков данных фрагментов, которые содержат скомпилированные веб-страницы. Примите к сведению, что обширные наборы HTML-страниц, особенно с большим количеством изображений, требуют существенного объема загрузочной памяти (стр. 833) для DB фрагментов. Если внутренняя загрузочная память Вашего ЦПУ недостаточна для Ваших пользовательских веб-страниц, используйте карту памяти (стр. 132), чтобы обеспечить внешнюю загрузочную память.

Чтобы создать Ваш HTML-код, использующий данные S7-1200, Вы вставляете AWP-команды (стр. 815) в качестве HTML-комментариев. По окончании сохраните свои HTML-страницы в Вашем ПК и отметьте путь к папке, где Вы сохраняете их.

Примечание

Предел размера файла для HTML-файлов, содержащих AWP-команду, составляет 64 килобайта. Вы не должны превышать этот предельный размер файла.

Обновление пользовательских веб-страниц

Пользовательские веб-страницы автоматически не обновляются. Ваш выбор, программировать ли HTML для обновления страницы или нет. Для страниц, которые отображают данные ПЛК, периодическое обновление поддерживает актуальность данных. Для HTML-страниц, которые служат формами для ввода данных, обновление может помешать вводу данных пользователем. Если Вы хотите, чтобы вся Ваша страница автоматически обновлялась, Вы можете добавить следующую строку к своему HTML-заголовку, где "10" - это количество секунд между обновлениями:

```
<meta http-equiv="Refresh" content="10">
```

Вы также можете использовать JavaScript или другие HTML-средства, чтобы управлять обновлением данных или страниц. Для этого обратитесь к документации по HTML и JavaScript.

12.7.2. AWP-команды, поддерживаемые веб-сервером S7-1200

Веб-сервер S7-1200 обеспечивает AWP-команды, которые Вы встраиваете в свои пользовательские веб-страницы, как HTML-комментарии в следующих целях:

- Чтение переменных (стр. 817)
- Запись переменных (стр. 818)
- Чтение специальных переменных (стр. 820)
- Запись специальных переменных (стр. 821)
- Определение типов перечислений (стр. 823)
- Назначение переменных типам перечислений (стр. 824)
- Создание блоков данных фрагментов (стр. 826)

Общий синтаксис

За исключением команды чтения переменной, AWP-команды имеют следующий синтаксис:

<!-- AWP_ <command name and parameters> -->

Вы используете AWP-команды в сочетании с типичными командами HTML-форм, чтобы выполнять запись в переменные в ЦПУ.

Описания AWP-команд на следующих страницах используют следующие соглашения:

- Элементы, заключенные в скобки [], являются дополнительными.
- Элементы, заключенные в угловые скобки < >, являются значениями параметров, которые будут определены.
- Кавычки отмечают литеральную часть команды. Они должны присутствовать, как обозначено.
- Специальные символы в тегах или именах блока данных, в зависимости от использования, нужно оставить без кавычек или заключить в кавычки (стр. 828).

Используйте текстовый редактор или режим HTML-редактирования, чтобы вставить AWP-команды в Ваши страницы.

Примечание

Ожидаемый синтаксис AWP-команд

Пробел после "<!--" и пробел до "-->" в формулировке AWP-команды важны для надлежащей компиляции команды. Пропуск пробелов может помешать компилятору сгенерировать надлежащий код. Компилятор в этом случае не отображает ошибку.

Обзор AWP-команд

Детальные сведения по использованию каждой из AWP-команд содержатся в последующих разделах, здесь же представлен краткий обзор команд:

Чтение переменных

`:=<Varname>:`

Запись переменных

`<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->`

Эта AWP-команда просто объявляет, что переменная в компоненте Name перезаписываемая. Ваш HTML-код выполняет запись в переменную по имени из `<input>`, `<select>` или других HTML-операторов в HTML-форме.

Чтение специальных переменных

`<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->`

Запись специальных переменных

`<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->`

Определение типов перечислений

`<!--`

`AWP_Enum_Def Name='<Enum type name>' Values='<Value>, <Value>, ... '`
`-->`

Ссылка на типы перечислений

`<!-- AWP_In_Variable Name='<Varname>' Enum="<Enum type name>" -->`

`<!-- AWP_Out_Variable Name='<Varname>' Enum="<Enum type name>" -->`

Создание фрагментов

`<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>][ID=<id>] -->`

Импортирование фрагментов

`<!-- AWP_Import_Fragment Name='<Name>' -->`

12.7.2.1. Чтение переменных

Пользовательские веб-страницы могут читать переменные (теги ПЛК) и теги блока данных из ЦПУ при условии, что Вы сконфигурировали доступ к тегам от HMI.

Синтаксис

`:=<Varname>:`

Параметры

<Varname>	Переменная, которую необходимо прочесть, которая может быть именем ПЛК тега из Вашей программы STEP 7, тегом блока данных, каналом ввода-вывода или ячейкой адресуемой памяти. Для адресов из памяти или ввода-вывода или псевдонимов не заключайте в одиночные кавычки (стр. 828) имя тега. Имена ПЛК тегов заключайте в двойные кавычки. Для тегов блока данных заключайте в двойные кавычки только имя блока. Имя тега находится вне кавычек. Обратите внимание на то, что Вы используете имя, а не номер блока данных.
-----------	--

Примеры

```

:= "Conveyor_speed" := "My_Data_Block".flag1:
:= I0.0:
:= MW100:

```

Пример чтения переменных-псевдонимов

```

<!-- AWP_Out_Variable Name='flag1' Use='My_Data_Block'.flag1 -->
:= flag1:

```

Примечание

Определение псевдонимов для ПЛК тегов и тегов блока данных описано в теме Использование псевдонимов для ссылки на переменную (стр. 823).

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

12.7.2.2. Запись переменных

Пользовательские страницы могут записывать данные в ЦПУ. Это достигается с помощью AWP-команды, идентифицируя переменную в ЦПУ, которая может быть записана из HTML-страницы. Переменная должна быть определена именем ПЛК тега или именем тега блока данных. Вы можете объявить несколько имен переменной в одном операторе. Чтобы записать данные в ЦПУ, Вы используете стандартные HTTP POST команды.

Типичным использованием является разработка формы на Вашей HTML-странице с полями ввода текста или списками выбора, которые соответствуют перезаписываемым переменным ЦПУ. Как со всеми пользовательскими страницами, Вы затем генерируете блоки из STEP 7, таким образом, чтобы они были включены в Вашу STEP 7 программу.

Когда пользователь с привилегиями по изменению переменных впоследствии получает доступ к этой странице и вводит данные в поля ввода или выбирает значение из списка, веб-сервер преобразовывает введенное значение в надлежащий тип данных для переменной и записывает значение в переменную в ЦПУ. Обратите внимание на то, что элемент имени для HTML полей ввода и HTML списков выбора использует синтаксис, типичный для элемента имени команды AWP_In_Variable. Обычно, заключайте имя в одиночные кавычки и, если Вы ссылаетесь на блок данных, заключайте имя блока данных в двойные кавычки.

Для получения дополнительной информации по управлению формой обратитесь к документации для HTML.

Синтаксис

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Параметры

<Varname1>	Если не добавлен элемент использования, то Varname1 является переменной, которая будет записана. Это может быть имя ПЛК тега из Вашей STEP 7 программы или тег из определенного блока данных. Если элемент использования добавлен, то Varname1 является альтернативным именем для переменной, упомянутой в <Varname2> (стр. 823). Это локальное имя для HTML-страницы.
<Varname2>	Если элемент использования добавлен, то Varname2 является переменной, которая будет записана. Это может быть имя ПЛК тега из Вашей STEP 7 программы или тег из определенного блока данных.

Как для элемента имени, так и для элемента использования, полное имя должно быть заключено в одиночные кавычки. В одиночных кавычках используйте двойные кавычки для выделения ПЛК тега и имени блока данных. Заключайте в двойные кавычки имя блока данных, но не имя тега блока данных. Обратите внимание на то, что для тегов блока данных, Вы используете имя, а не номер блока данных.

Примеры использования HTML поля ввода

```
<!-- AWP_In_Variable Name='"Target_Level"' -->
<form method="post">
<p>Input Target Level: <input name='"Target_Level"' type="text" />
</p>
</form>

<!-- AWP_In_Variable Name='"Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name='"Data_block_1".Braking' type="text" />
%</p>
</form>

<!-- AWP_In_Variable Name='"Braking"' Use='"Data_block_1".Braking'
-
->
<form method="post">
<p>Braking: <input name='"Braking"' type="text" /> %</p>
</form>
```

Примеры использования HTML списка выбора

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable' -->
<form method="post">
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=:"Data_block_1".ManualOverrideEnable:> </option>
<option value=1>Yes</option>
<option value=0>No</option>
</select><input type="submit" value="Submit setting" /></form>
```

Примечание

Только пользователь с привилегиями по изменению переменных может записать данные в ЦПУ. Веб-сервер игнорирует команды, если пользователь не обладает привилегиями по изменению.

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

12.7.2.3. Чтение специальных переменных

Веб-сервер обеспечивает возможность считать значения из ПЛК для сохранения в специальных переменных в HTTP-заголовке ответа. Вы, возможно, захотели бы прочитать путь из ПЛК тега, чтобы перенаправить URL в другое расположение, используя специальную переменную HEADER:Location.

Синтаксис

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Параметры

<Type>	Один из следующих типов специальной переменной: HEADER COOKIE_VALUE COOKIE_EXPIRES
<Name>	Обратитесь к документации по HTTP для списка всех имен переменных HEADER. Несколько примеров приведены ниже: Status: код отклика Location: путь для перенаправления Retry-After: ожидаемое время недоступности службы для запрашивающего клиента Для типов COOKIE_VALUE и COOKIE_EXPIRES, <Name> является именем определенного куки. COOKIE_VALUE:name: значение именованного куки COOKIE_EXPIRES:name: время истечения срока именованного куки в секундах Элемент Name должен быть заключен в одиночные или двойные кавычки. Если элемент Use добавлен, имя специальной переменной соответствует имени ПЛК тега. Заключите весь элемент Name в одиночные кавычки, а ПЛК тег в двойные кавычки. Имя специальной переменной и имя ПЛК тега должны точно совпадать.
<Varname>	Имя ПЛК тега или тега блока данных, в который должна быть считана переменная Varname должно быть заключено в одиночные кавычки. В одиночных кавычках используйте двойные кавычки вокруг ПЛК тега или имени блока данных. В двойные кавычки заключайте имя блока данных, но не имя тега блока данных. Обратите внимание на то, что для тегов блока данных, Вы используете имя, а не номер блока данных.

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

Пример: Чтение специальной переменной без элемента Use

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' -->
```

В этом примере специальная переменная HTTP "HEADER:Status" получает значение ПЛК тега "HEADER:Status". Имя в таблице ПЛК тегов должно точно соответствовать имени специальной переменной, если элемент Use не определен.

Пример: Чтение специальной переменной с элементом Use

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Status"' -->
```

В этом примере специальная переменная HTTP "HEADER:Status" получает значение ПЛК тега "Status".

12.7.2.4. Запись специальных переменных

Веб-сервер обеспечивает возможность записывать значения в ЦПУ из специальных переменных в заголовке HTTP-запроса. Например, Вы можете сохранять информацию в STEP 7 о куки, связанном с пользовательской Веб-страницей, пользователя, который обращается к странице, или данные заголовка. Веб-сервер обеспечивает доступ к определенным специальным переменным, которые Вы можете записывать в ЦПУ, если авторизовались в качестве пользователя с привилегиями по изменению переменных.

Синтаксис

```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Параметры

<Type>	Один из следующих типов специальных переменных: HEADER SERVER COOKIE_VALUE
<Name>	Специфическая переменная одного из типов, определенных выше, как показано в настоящих примерах: HEADER:Accept: допустимые типы контента HEADER:User-Agent: информация о программном агенте, порождающем запрос SERVER:current_user_id: id текущего пользователя; 0, если нет зарегистрированных пользователей SERVER:current_user_name: имя текущего пользователя COOKIE_VALUE:<name>: значение именованного куки Заключите элемент Name в одиночные кавычки. Если элемент Use не определен, то имя специальной переменной соответствует имени переменной ПЛК. Заключите весь элемент Name в одиночные кавычки, а ПЛК тега в двойные кавычки. Имя специальной переменной и имя ПЛК тега должны точно совпадать. Обратитесь к документации по HTTP для получения списка всех имен переменных HEADER.
<Varname>	Имя переменной в Вашей STEP 7 программе, в которую Вы хотите записать специальную переменную, которая может быть именем ПЛК тега или тегом блока данных. Varname должно быть заключено в одиночные кавычки. В одиночных кавычках используйте двойные кавычки вокруг ПЛК тега или имени блока данных. В двойные кавычки заключайте имя блока данных, но не имя тега блока данных. Обратите внимание на то, что для тегов блока данных, Вы используете имя, а не номер блока данных.

Примеры

```
<!-- AWP_In_Variable Name='\"SERVER:current_user_id\"' -->
```

В этом примере веб-страница записывает значение специальной HTTP-переменной "SERVER:current_user_id" в ПЛК тег с именем "SERVER:current_user_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id' Use='\"my_userid\"' -->
```

В этом примере веб-страница записывает значение специальной HTTP-переменной "SERVER:current_user_id" в ПЛК тег с именем "my_userid".

Примечание

Только пользователь с привилегиями по изменению переменных может записать данные в ЦПУ. Веб-сервер игнорирует команды, если пользователь не обладает привилегиями по изменению.

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

12.7.2.5. Использование псевдонимов для ссылки на переменные

Вы можете использовать псевдоним на своей пользовательской веб-странице для In_Variable или Out_Variable. Например, Вы можете использовать на своей HTML-странице символьное имя, отличное от того, которое используется в ЦПУ, либо Вы можете приравнять переменную в ЦПУ к специальной переменной. AWP элемент Use обеспечивает такую возможность.

Синтаксис

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

Параметры

<Varname1>	Имя псевдонима или имя специальной переменной Varname1 должно быть заключено в одиночные или двойные кавычки.
<Varname2>	Имя ПЛК переменной, которой Вы хотите присвоить псевдоним. Переменная может быть ПЛК тегом, тегом блока данных или специальной переменной. Varname2 должно быть заключено в одиночные кавычки. В одиночных кавычках используйте двойные кавычки вокруг ПЛК тега, специальной переменной или имени блока данных. В двойные кавычки заключайте имя блока данных, но не имя тега блока данных. Обратите внимание на то, что для тегов блока данных, Вы используете имя, а не номер блока данных.

Примеры

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use='"Data_Block_10".server_user' -->
```

В этом примере специальная переменная SERVER:current_user_id записывается в тег "server_user" в блоке данных "Data_Block_10".

```
<-- AWP_Out_Variable Name='Weight'
Use='"Data_Block_10".Tank_data.Weight' -->
```

В этом примере на значение элемента структуры блока данных Data_Block_10.Tank_data.Weight можно сослаться просто с помощью "Weight" повсюду в остальной части пользовательской веб-страницы.

```
<-- AWP_Out_Variable Name='Weight' Use='"Raw_Milk_Tank_Weight"' -->
```

В этом примере на значение ПЛК тега "Raw_Milk_Tank_Weight" можно сослаться просто с помощью "Weight" повсюду в остальной части пользовательской веб-страницы.

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

12.7.2.6. Определение типов перечислений

Вы можете определить типы перечислений на своих пользовательских страницах и назначить элементы в AWP-команде.

Синтаксис

```
<!-- AWP_Enum_Def Name='<Enum type name>' Values='<Value>,
<Value>, ... ' -->
```

Параметры

<Enum type name>	Имя типа перечисления, заключенного в одиночные или двойные кавычки.
<Value>	<constant>:<name> Константа указывает числовое значение для назначения типа перечисления. Общее количество не ограничено. Имя - это значение, назначенное элементу перечисления.

Обратите внимание на то, что вся строка назначений перечисления заключена в одиночные кавычки, а каждое отдельное назначение элемента типа перечисления заключено в двойные кавычки. Сфера действия определения типа перечисления является глобальной для пользовательских веб-страниц. Если Вы скомпоновали свои пользовательские веб-страницы в языковые папки (стр. 847), то определение типа перечисления является глобальным для всех страниц языковой папки.

Пример

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is full", 2:"Tank is empty"' -->
```

12.7.2.7. Ссылка на переменные ЦПУ с типом перечислений

Вы можете присвоить переменной в ЦПУ тип перечисления. Эта переменная может использоваться в других местах Вашей пользовательской веб-страницы в операции чтения (стр. 817) или операции записи (стр. 818). В операции чтения веб-сервер заменит числовое значение, которое считано из ЦПУ на соответствующее текстовое значение перечисления. В операции записи веб-сервер заменит текстовое значение на целочисленное значение перечисления, которое соответствует тексту прежде, чем записать значение в ЦПУ.

Синтаксис

```
<!-- AWP_In_Variable Name='<Varname>' Enum="<EnumType>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<EnumType>" -->
```

Параметры

<Varname>	Имя ПЛК тега или тега блока данных для установления соответствия с типом перечисления или псевдоним для ПЛК тега (стр. 823), если объявлен. Varname должно быть заключено в одиночные кавычки. В одиночных кавычках используйте двойные кавычки вокруг ПЛК тега или имени блока данных. Обратите внимание на то, что для тегов блока данных, Вы используете имя, а не номер блока данных. Закрывайте в двойные кавычки имя блока данных, но не имя тега блока данных.
<EnumType>	Имя типа перечисления, которое должно быть заключено в одиночные или двойные кавычки

Сферой действия ссылки на тип перечисления является текущий фрагмент.

Пример использования при чтении переменной

```
<!-- AWP_Out_Variable Name='Alarm' Enum='AlarmEnum' -->...  
<p>The current value of "Alarm" is := "Alarm":</p>
```

Если значение "Alarm" в ЦПУ равняется 2, то HTML-страница отображает 'The current value of "Alarm" is Tank is empty', поскольку определение типа перечисления (стр. 823) назначает текстовую строку "Tank is empty" числовому значению 2.

Пример использования при записи переменной

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank  
is full", 2:"Tank is empty"' -->  
<!-- AWP_In_Variable Name='Alarm' Enum='AlarmEnum' -->...  
<form method="POST">  
<p><input type="hidden" name="Alarm" value="Tank is full" /></p>  
<p><input type="submit" value='Set Tank is full' /><p>  
</form>
```

Так как определение типа перечисления (стр. 823) назначает "Tank is full" числовому значению 1, значение 1 записывается в ПЛК тег с именем "Alarm" в ЦПУ.

Обратите внимание на то, что элемент Enum в объявлении AWP_In_Variable должен точно соответствовать элементу Name в объявлении AWP_Enum_Def.

Пример использования при записи переменной с использованием псевдонима

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank  
is full", 2:"Tank is empty"' -->  
<!-- AWP_In_Variable Name='Alarm' Enum='AlarmEnum'  
Use='Data_block_4.Motor1.Alarm'-->...  
<form method="POST">  
<p><input type="hidden" name="Alarm" value="Tank is full" /></p>  
<p><input type="submit" value='Set Tank is full' /><p>  
</form>
```

Так как определение типа перечисления (стр. 823) назначает "Tank is full" числовому значению 1, значение 1 записывается в псевдоним "Alarm", который соответствует ПЛК тегу с именем "Motor1.Alarm" в блоке данных "Data_Block_4" в ЦПУ.

Если имя тега или имя блока данных включает в себя специальные символы, Вы должны использовать дополнительные одиночные кавычки или символы ESC, как описано в теме Обработка имен тегов, содержащих специальные символы (стр. 828).

Примечание

Предыдущие выпуски требовали отдельного объявления AWP_Enum_Ref для установления соответствия с переменной с определенным типом перечисления. STEP 7 и S7-1200 поддерживают существующий код с AWP_Enum_Ref объявлениями; однако, необходимости в этой команде больше нет.

12.7.2.8. Создание фрагментов

STEP 7 преобразовывает и хранит пользовательские веб-страницы как управляющие DB и DB фрагментов, когда Вы нажимаете "Generate blocks" в свойствах ЦПУ для веб-сервера. Вы можете установить определенные фрагменты для определенных страниц или для разделов определенных страниц. Вы можете идентифицировать эти фрагменты по имени и номеру с помощью AWP-команды "Start_Fragment". Все, находящееся на странице после команды AWP_Start_Fragment, принадлежит этому фрагменту, пока не использована другая команда AWP_Start или пока не достигнут конец файла.

Синтаксис

```
<!-- AWP_Start_Fragment Name='<Name>'
[Type=<Type>][ID=<id>][Mode=<Mode>] -->
```

Параметры

<Name>	Текстовая строка: имя DB фрагмента Имена фрагмента должны начинаться с буквы или подчеркивания и состоять из букв, цифр и подчеркиваний. Имя фрагмента является регулярным выражением формы: [a-zA-Z_][a-zA-Z_0-9]*
<Type>	"manual" или "automatic" manual: Программа STEP 7 должна запросить этот фрагмент и может ответить соответствующим образом. Работой фрагмента нужно управлять с помощью переменных STEP 7 и управляющего DB. automatic: Веб-сервер обрабатывает фрагмент автоматически. Если Вы не определяете параметр типа, значением по умолчанию является "automatic".
<id>	Целочисленный идентификационный номер. Если Вы не определяете параметр ID, веб-сервер назначает число по умолчанию. Для ручных фрагментов, установите для ID небольшое число. ID является средством, с помощью которого STEP 7 программа управляет ручным фрагментом.
<Mode>	"visible" или "hidden" visible: Содержимое фрагмента отображается на пользовательской веб-странице. hidden: Содержимое фрагмента не отображается на пользовательской веб-странице. Если Вы не определяете параметр типа, значением по умолчанию является "visible".

Ручные фрагменты

Если Вы создаете ручной фрагмент для пользовательской веб-страницы или части страницы, то Ваша STEP 7 программа должна осуществлять управление, когда фрагмент послан. STEP 7 программа должна установить надлежащие параметры в управляющем DB для пользовательской страницы при ручном управлении и затем вызвать инструкцию WWW с управляющим DB после изменений. Для понимания структуры управляющего DB и того, как управлять отдельными страницами и фрагментами, смотрите тему Расширенное управление пользовательскими веб-страницами (стр. 851).

12.7.2.9. Импортрование фрагментов

Вы можете создать именованный фрагмент из части Вашего HTML-кода и затем импортировать этот фрагмент в другое место в Вашем наборе пользовательских веб-страниц. Например, представьте ряд пользовательских веб-страниц, у которого есть начальная страница, а также несколько других HTML-страниц, доступных с помощью ссылок на начальной странице. Предположим, что каждая из отдельных страниц должна отображать логотип компании. Вы могли бы реализовать это, создав фрагмент (стр. 826), который загружает изображение логотипа компании. Каждая отдельная HTML-страница могла бы в этом случае импортировать этот фрагмент, чтобы отобразить логотип компании. С этой целью Вы используете команду AWP Import_Fragment. HTML-код для фрагмента существует только в одном фрагменте, но Вы можете импортировать этот DB фрагмента столько раз сколько необходимо на веб-страницах по Вашему выбору.

Синтаксис

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

Параметры

<Name>	Текстовая строка: имя DB фрагмента, подлежащего импортированию
--------	--

Пример

Выборка из HTML-кода, которая создает фрагмент вывода изображения:

```
<!-- AWP_Start_Fragment Name='My_company_logo' --><p></p>
```

Выборка из HTML-кода в другом .html файле, импортирующем фрагмент, выводящий изображение логотипа:

```
<!-- AWP_Import_Fragment Name='My_company_logo' -->
```

Оба .html файла (один, который создает фрагмент и второй, который импортирует его) находятся в структуре каталога, которую Вы определяете, когда конфигурируете пользовательские страницы в STEP 7 (стр. 830).

12.7.2.10. Объединение определений

При объявлении переменных для использования в пользовательских веб-страницах Вы можете объединить объявление переменной и псевдоним для переменной (стр. 823). Вы можете также объявить несколько In_Variables в одном операторе и несколько Out_Variables в одном операторе.

Примеры

```
<!-- AWP_In_Variable Name='"Level"', Name='"Weight"', Name='"Temp"'  
-->  
<--! AWP_Out_Variable Name='HEADER:Status', Use='"Status"',  
Name='HEADER:Location', Use="Location", Name='COOKIE_VALUE:name',  
Use="my_cookie" -->  
<!-- AWP_In_Variable Name='Alarm' Use='"Data_block_10".Alarm' -->
```

12.7.2.11. Обработка имен тегов, содержащих специальные символы

При определении имен переменных на пользовательских веб-страницах Вы должны проявить особенную осторожность, если имена тега содержат символы, имеющие особое предназначение.

Чтение переменных

Вы используете следующий синтаксис для чтения переменных (стр. 817):

:=<Varname>:

Следующие правила применимы к чтению переменных:

- Для имен переменных из таблицы ПЛК тегов заключайте имя тега в двойные кавычки.
- Для имен переменных, которые являются тегами блока данных, заключайте имя блока данных в двойные кавычки. Тег располагается вне кавычек.
- Для имен переменных, которые являются прямыми адресами ввода-вывода, адресами памяти или псевдонимами, не используют кавычки вокруг читаемой переменной .
- Для имен тегов или имен тегов блока данных, которые содержат наклонную черту влево, предваряйте наклонную черту влево другой наклонной чертой влево.
- Если имя тега или имя тега блока данных содержит двоеточие, знак меньше, знак больше, или амперсанд определяйте псевдоним, у которого нет специальных символов для переменной чтения, и читайте переменную, используя псевдоним. Предваряйте двоеточия в именах тега в элементе Use наклонной чертой влево.

Таблица 12- 1 Примеры переменных чтения

Имя блока данных	Имя тега	Команда чтения
не используется	ABC:DEF	<code><!--AWP_Out_Variable Name='special_tag' Use ="ABC:DEF" -->:=special_tag:</code>
не используется	T\	<code>:= "T\\":</code>
не используется	A \B 'C :D	<code><!--AWP_Out_Variable Name='another_special_tag' Use='\"A \\B \\C :D\"' -->:=another_special_tag:</code>
не используется	a<b	<code><!--AWP_Out_Variable Name='a_less_than_b' Use='\"a<b\"' -->:=a_less_than_b:</code>
Data_block_1	Tag_1	<code>:= "Data_block_1".Tag_1:</code>
Data_block_1	ABC:DEF	<code><!-- AWP_Out_Variable Name='special_tag' Use='\"Data_block_1\".ABC\ :DEF' -->:=special_tag:</code>
DB A' B C D\$ E	Tag	<code>:= "DB A' B C D\$ E".Tag:</code>
DB:DB	Tag:Tag	<code><!--AWP_Out_Variable Name='my_tag' Use ="DB:DB\".Tag\ :Tag' -->:=my_tag:</code>

Элементы Name и Use

У AWP-команд AWP_In_Variable, AWP_Out_Variable, AWP_Enum_Def, AWP_Enum_Ref, AWP_Start_Fragment и AWP_Import_Fragment есть элемент Name. У команд HTML-формы таких, как <input> и <select> также есть элемент Name.

У AWP_In_Variable и AWP_Out_Variable может дополнительно быть элемент Use. Независимо от команды синтаксис для элементов Name и Use относительно обработки специальных символов один и тот же:

- Текст, который Вы предусматриваете для элементов Name или Use, должен быть заключен в одиночные кавычки. Если вложенное имя является ПЛК тегом или именем блока данных, используйте одиночные кавычки для всего элемента.
- Внутри элемента Name или Use имена блока данных и ПЛК тега должны быть заключены в двойные кавычки.
- Если имя тега или блока данных содержит символ одинарной кавычки или наклонную черту влево, изолируйте этот символ с помощью наклонной черты влево. Наклонная черта влево является символом перехода в компиляторе AWP-команд.

Таблица 12- 2 Примеры элементов Name

Имя блока данных	Имя тега	Опции элемента Name
не используется	ABC'DEF	Name=" 'ABC\ 'DEF' "
не используется	A \B 'C :D	Name=" 'A \\B \\ 'C :D' "
Data_block_1	Tag_1	Name=" 'Data_block_1'.Tag_1' "
Data_block_1	ABC'DEF	Name=" 'Data_block_1'.ABC\ 'DEF' "
Data_block_1	A \B 'C :D	Name=" 'Data_block_1'.A \\B \\ 'C :D' "
DB A' B C D\$ E	Tag	Name=" 'DB A\ ' B C D\$ E".Tag' "

Элементы Use следуют тем же соглашениям, что и элементы Name.

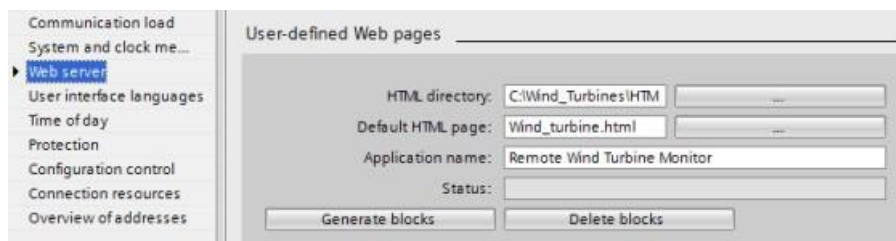
Примечание

Независимо от того, какие символы Вы используете на своей HTML-странице , установите для HTML-страницы набор символов UTF-8 и сохраняйте страницу в редакторе с кодировкой UTF-8.

12.7.3. Конфигурирование использования пользовательских веб-страниц

Чтобы сконфигурировать пользовательские веб-страницы из STEP 7, выполните следующие шаги:

1. Выберите ЦПУ в представлении Device Configuration.
2. Выведите на экран свойства "Web server" в окне инспектора для ЦПУ.



3. Если не выбрано ранее, установите флажок для "Activate Web server on this module".
4. Выберите "Permit access only with HTTPS", чтобы гарантировать, что веб-сервер использует зашифрованный обмен и усилить безопасность Вашего доступного по сети ЦПУ.
5. Введите или найдите каталог на Вашем ПК, где Вы сохранили HTML-страницу по умолчанию (начальная страница).
6. Введите имя страницы по умолчанию.
7. Присвойте имя своему приложению (по желанию). Веб-сервер использует имя приложения, чтобы далее разбить на подкатегории или сгруппировать веб-страницы. Когда Вы присваиваете имя приложению, веб-сервер создает URL для Вашей пользовательской страницы в следующем формате: `http[s]://www.xx.yy.zz/awp/<имя приложения>/<имя страницы>.html`.
Избегайте специальных символов в имени приложения. Некоторые символы могут привести к отсутствию способности веб-сервера отображать пользовательские страницы.
8. Введите расширения файлов, которые содержат AWP-команды. По умолчанию STEP 7 анализирует файлы с расширениями .htm, .html или .js. Если у Вас есть файлы с дополнительными расширениями, добавьте их.
9. Сохраните значение по умолчанию для номера веб-DB или введите свой номер. Это номер DB управления, который управляет отображением веб-страниц.
10. Сохраните значение по умолчанию для стартового DB фрагмента или введите свой номер. Это первый из DB фрагментов, которые содержат веб-страницы.

Генерация программных блоков

Когда Вы щелкаете по "Generate blocks", STEP 7 генерирует блоки данных из HTML-страниц в исходном HTML-каталоге, который Вы определили и управляющий блок для управления Вашими веб-страницами. Вы можете установить эти атрибуты по мере необходимости для Вашего приложения (стр. 831). STEP 7 также генерирует ряд блоков данных фрагментов, содержащих представление всех Ваших HTML-страниц. Когда Вы генерируете блоки данных, STEP 7 обновляет свойства, чтобы отобразить номер управляющего блока данных и номер первого из блоков данных фрагментов. После того, как Вы генерируете блоки данных, Ваши пользовательские веб-страницы становятся частью Вашей STEP 7 программы. Блоки, соответствующие этим страницам, появляются в папке веб-сервера, которая находится в папке System blocks под Program blocks в дереве проекта.

Удаление программных блоков

Чтобы удалить блоки данных, которые Вы ранее сгенерировали, щелкните по "Delete data blocks". STEP 7 удаляет управляющей блок данных и все блоки данных фрагментов из Вашего проекта, которые соответствуют пользовательским веб-страницам.

12.7.4. Программирование инструкции WWW для пользовательских веб-страниц

Ваша пользовательская STEP 7 программа должна содержать и обрабатывать инструкцию WWW для пользовательских веб-страниц, чтобы они стали доступными из стандартных веб-страниц. Управляющий блок данных является входным параметром инструкции WWW и определяет содержание страниц, как представлено в блоках данных фрагментов, а также информации состояния и управления. STEP 7 создает управляющий блок данных, когда Вы щелкаете по "Create blocks" в конфигурации пользовательских веб-страниц (стр. 830).

Программирование инструкции WWW

STEP 7 программа должна обрабатывать инструкцию WWW для пользовательских веб-страниц, чтобы они были доступными из стандартных веб-страниц. Вы могли бы захотеть организовать доступ к пользовательским веб-страницам только при определенных условиях, как продиктовано требованиями и предпочтениями Вашего приложения. В этом случае Ваша программная логика может управлять вызовом инструкции WWW.

Таблица 12- 3 Инструкция WWW

LAD / FBD	SCL	Описание
	<pre>ret_val := WWW(ctrl_db:=_uint_in_);</pre>	Обеспечивает доступ к пользовательским веб-страницам из стандартных веб-страниц.

Вы должны обеспечить входной параметр управляющего блока данных (CTRL_DB), который соответствует целочисленному номеру DB управления. Вы можете найти этот номер управляющего DB (называемого номеров веб-DB) в свойствах веб-сервера ЦПУ после того, как Вы создадите блоки для пользовательских веб-страниц. Введите целочисленный номер DB как параметр CTRL_DB инструкции WWW. Возвращаемое значение (RET_VAL) содержит результат функции. Обратите внимание на то, что инструкция WWW выполняется асинхронно и что у вывода RET_VAL могло бы быть начальное значение 0, хотя ошибка может произойти позже. Программа может проверять состояние DB управления, чтобы гарантировать, что приложение запустилось успешно, или проверять RET_VAL после вызова WWW.

Таблица 12- 4 Выходной параметр Return value

RET_VAL	Описание
0	Без ошибок
16#00ух	х: Запрос, представленный соответствующим битом, находится в состоянии ожидания: х=1: запрос 0 х=2: запрос 1 х=4: запрос 2 х=8: запрос 3 Значения х могут быть логически сложены, чтобы представить состояния ожидания нескольких запросов. Если , например, х = 6, то запросы 1 и 2 находятся в состоянии ожидания. у: 0: без ошибок; 1: ошибка существует, и "last_error" был установлен в управляющем DB (стр. 851).
16#803a	Управляющий DB не загружен.
16#8081	Управляющий DB имеет неправильный тип, формат или версию.
16#80C1	Нет доступных ресурсов для инициализации веб-приложения.

Использование управляющего DB

STEP 7 создает управляющий блок данных, когда Вы щелкаете по "Generate blocks", и отображает номер управляющего DB в свойствах пользовательских веб-страниц. Вы можете также найти управляющий DB в папке Program blocks в дереве проекта.

Как правило, Ваша STEP 7 программа использует управляющий DB непосредственно после создания посредством "Generate blocks" без дополнительной обработки. Однако пользовательская STEP 7 программа может установить глобальные команды в управляющем DB для деактивирования веб-сервера или его последующего включения. Кроме того, для пользовательских страниц, которые Вы создаете как ручные DB фрагментов (стр. 830), пользовательская STEP 7 программа должна управлять их поведением посредством таблицы запросов в управляющем DB. Для получения информации об этих дополнительных задачах смотри тему Расширенное управление пользовательскими веб-страницами (стр. 851).

12.7.5. Загрузка программных блоков в ЦПУ

После того, как Вы сгенерировали блоки для пользовательских веб-страниц, они являются частью Вашей STEP 7 программы точно так же, как любые другие программные блоки. Вы загружаете программные блоки в ЦПУ обычным порядком. Обратите внимание на то, что Вы можете загрузить только программные блоки пользовательских веб-страниц, когда ЦПУ находится в режиме STOP.

12.7.6. Доступ к пользовательским веб-страницам

Вы получаете доступ к своим пользовательским Веб-страницам из стандартных Веб-страниц (стр. 791). Стандартные Веб-страницы отображают ссылку "User Pages" в меню слева, где появляются ссылки на другие страницы. Страница навигации мобильного устройства также обеспечивает ссылку "User Pages".

Когда Вы щелкаете по ссылке "User Pages", Ваш Веб-браузер переходит к странице, которая обеспечивает ссылку на Вашу страницу по умолчанию. Из пользовательских страниц навигация выполняется согласно тому, как Вы спроектировали свои определенные страницы.



12.7.7. Ограничения, связанные с пользовательскими веб-страницами

Ограничения для стандартных веб-страниц (стр. 855) также применимы к пользовательским веб-страницам. Кроме того пользовательские веб-страницы обладают некоторыми специфическими ограничениями.

Объем загрузочной памяти

Ваши пользовательские веб-страницы становятся блоками данных, когда Вы щелкаете по "Generate blocks", которым необходимо пространство в загрузочной памяти. Если у Вас установлена карта памяти, то в Вашем распоряжении имеется объем вплоть до емкости установленной карты памяти, являющейся внешней загрузочной памятью для пользовательских веб-страниц.

Если у Вас отсутствует карта памяти, эти блоки занимают внутреннюю загрузочную память, которая ограничена в соответствии с моделью Вашего ЦПУ.

Вы можете проверить объем используемой и доступной загрузочной памяти из онлайн и диагностических инструментов в STEP 7. Вы можете также просмотреть свойства отдельных блоков, которые STEP 7 генерирует из Ваших пользовательских веб-страниц и увидеть потребность в загрузочной памяти.

Примечание

Если Вам необходимо уменьшить объем, требуемый для Ваших пользовательских веб-страниц, уменьшите, если возможно, использование изображений.

Кавычки в текстовых строках

Избегайте использования текстовых строк, которые содержат встроенные одиночные или двойные кавычки в тегах блока данных, которые Вы используете с любой целью в пользовательских веб-страницах. Поскольку синтаксис HTML часто использует одиночные или двойные кавычки в качестве разделителей, кавычки в текстовых строках могут помешать отображению пользовательских веб-страниц.

Для тегов блока данных с типом String, которые Вы используете в пользовательских веб-страницах, соблюдайте следующие правила:

- Не вводите одиночные или двойные кавычки в строковое значение тега блока данных в STEP 7.
- Не позволяйте пользовательской программе выполнять присвоения строк, содержащих кавычки, этим тегам блока данных.

12.7.8. Пример пользовательской веб-страницы

12.7.8.1. Веб-страница для контроля и управления ветряной турбиной

В качестве примера пользовательской веб-страницы, рассмотрите веб-страницу, которая используется, чтобы удаленно контролировать и управлять ветряной турбиной:

Remote Wind Turbine Monitor: Turbine #5

Wind speed:	7.5 km/h
Wind direction:	23.5 deg.
Temperature:	17.2 deg. C
Power output:	1000 KW
Manual override: On	Set: Yes
Turbine speed:	15 RPM
Yaw override: On	Set: Yes
Turbine yaw:	52 deg.
Pitch override: On	Set: Yes
Blade pitch:	45 deg.

Submit override settings and values

Braking: 25 %

Описание

В этом приложении каждая ветряная турбина на ветряной электростанции оборудована управлением на базе S7-1200. В STEP 7 программе у каждой ветряной турбины есть определенный блок данных.

Пользовательская веб-страница обеспечивает удаленный доступ к турбине от ПК. Пользователь может подключиться к стандартным веб-страницам ЦПУ определенной ветряной турбины и получить доступ к пользовательской странице "Remote Wind Turbine Monitor", чтобы просмотреть данные этой турбины. Пользователь с привилегиями по изменению переменных может также перевести турбину в ручной режим и управлять переменными для скорости турбины, угла поворота и наклона из веб-страницы. Пользователь с привилегиями по изменению переменных может также установить величину тормозного усилия независимо от того, находится ли турбина под ручным или автоматическим управлением.

STEP 7 программа проверила бы булевы значения для коррекции автоматического управления, и, если они установлены, стала бы использовать введенные пользователями значения для скорости турбины, тангажа и рыскания. В противном случае, программа проигнорировала бы эти значения.

Используемые файлы

Этот пример пользовательской веб-страницы состоит из трех файлов:

- **Wind_turbine.html:** Это - HTML-страница , которая реализует дисплей, показанный выше, используя AWP-команды, чтобы получить доступ к данным контроллера.
- **Wind_turbine.css:** Это - каскадная таблица стилей, которая содержит стили форматирования для HTML-страницы. Использование каскадной таблицы стилей является необязательным, но оно может упростить разработку HTML-страницы.
- **Wind_turbine.jpg:** Это - фоновое изображение, которое использует HTML-страница. Использование изображений в пользовательских Веб-страницах, конечно, необязательное, и требует дополнительного объема памяти в ЦПУ.

Этим файлы не предоставлены с Вашей установкой, но описаны в качестве примера.

Внедрение

HTML-страница использует AWP-команды, чтобы прочесть значения из ПЛК (стр. 817) для полей вывода и записать значения (стр. 818) данных, поступающих из пользовательских полей ввода, в ПЛК. Эта страница также использует AWP-команды для определения типа перечисления (стр. 823) и ссылку (стр. 824) для обработки параметров настройки ВКЛЮЧЕНИЯ - ВЫКЛЮЧЕНИЯ.

Первая часть страницы представляет строку заголовка, которая включает номер ветряной турбины.

Remote Wind Turbine Monitor: Turbine #5

Следующая часть страницы отображает атмосферные условия для ветряной турбины. Ввод вывод на месте установки турбины обеспечивает скорость ветра, направление ветра и фактическую температуру.

Wind speed:	7.5 km/h
Wind direction:	23.5 deg.
Temperature:	17.2 deg. C

Затем, страница отображает выходную мощность турбины, которая считывается из S7-1200.

Power output:	1000 KW
---------------	---------

Следующие разделы допускают ручное управление турбиной, переопределяя обычное автоматическое управление S7-1200.

- Manual override: разрешает ручную коррекцию турбины. Пользовательская STEP 7 программа требует, чтобы настройка для ручной коррекции имела значение истина прежде, чем разрешать использование любой из ручных настроек для скорости турбины, тангажа или рыскания.

Manual override: On	Set: Yes
Turbine speed:	15 RPM

- Yaw override: разрешает и вводит ручную коррекцию настройки угла поворота. Пользовательская STEP 7 программа требует, чтобы и ручная коррекция и коррекция угла поворота имели значение истина, чтобы применить настройку угла поворота.

Yaw override: On	Set: Yes
Turbine yaw:	52 deg.

- Pitch override: разрешает ручную коррекцию наклона лопастей. Пользовательская STEP 7 программа требует, чтобы и ручная коррекция и коррекция наклона имели значение истина, чтобы применить настройку наклона лопастей.

Pitch override: On	Set: Yes
Blade pitch:	15 deg.

HTML-страница содержит кнопку подтверждения, чтобы отправить настройки коррекции в контроллер.

Submit override settings and values

Пользовательское поле ввода торможения обеспечивает ручную настройку процента торможения. Пользовательской STEP 7 программе не требуется ручная коррекция для принятия величины торможения.

Braking:	25 %
----------	------

Кроме того, HTML-страница использует AWP-команду для записи специальной переменной (стр. 821), содержащей идентификатор пользователя, который получает доступ к странице, в тег из таблицы ПЛК тегов.

12.7.8.2. Чтение и отображение данных контроллера

HTML-страница "Remote Wind Turbine Monitor" использует множество AWP-команд для чтения данных из контроллера (стр. 817) и отображения их на странице. Например, проанализируйте HTML-код для отображения выходной мощности, как показано в этой части Веб-страницы:

Power output:	1000 KW
---------------	---------

Пример HTML-кода

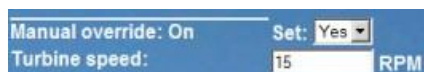
Следующая выборка из HTML-страницы "Remote Wind Turbine Monitor" отображает текст "Power Output:" в левой ячейке строки таблицы, читает переменную для выходной мощности и отображает значение в правой ячейке строки таблицы вместе с текстовым сокращением для киловатт, kW.

AWP-команда := "Data_block_1".PowerOutput: выполняет операцию чтения. Обратите внимание на то, что на блоки данных ссылаются по имени, а не по номеру блока данных (т.е. "Data_block_1", а не "DB1").

```
<tr style="height:2%;>
<td>
<p>Power output:</p>
</td>
<td>
<p style="margin-bottom:5px;"> := "Data_block_1".PowerOutput: kW</p>
</td>
</tr>
```

12.7.8.3. Использование типа перечисления

HTML-страница "Remote Wind Turbine Monitor" использует типы перечисления для трех экземпляров там, где на HTML-странице отображается "ON" или "OFF" для булевого значения, и там, где пользователь устанавливает булево значение. Тип перечисления для "ON" дает значение 1, а тип перечисления для "OFF" дает значение 0. Например, рассмотрите HTML-код для чтения и записи настройки Manual Override Enable в тег "Data_block_1".ManualOverrideEnable, используя тип перечисления:



Пример HTML-кода

Следующие выборки из HTML-страницы "Remote Wind Turbine Monitor" Они показывают, как объявить тип перечисления с именем "OverrideStatus" со значениями "Off" и "On" для 0 и 1, а затем организовать ссылку на тип перечисления OverrideStatus для булевого тега ManualOverrideEnable в блоке данных с именем "Data_block_1".

```
<!-- AWP_In_Variable Name=' "Data_block_1".ManualOverrideEnable'
Enum="OverrideStatus" -->

<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
```

Там где HTML-страница содержит поле отображения в ячейке таблицы для текущего состояния ManualOverrideEnable, она использует обычную команду чтения переменной, но с использованием ранее объявленного и упомянутого типа перечисления, на странице отображается "Off" или "On", а не 0 или 1.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: := "Data_block_1".ManualOverrideEnable:</p>
</td>
```

HTML-страница содержит раскрывающийся список для пользователя, чтобы изменить значение ManualOverrideEnable. Раскрывающийся список использует для отображения тексты "Yes" и "No". С помощью использования типа перечисления, "Yes" связано со значением "On" для типа перечисления, а "No" - соответственно, "Off". Пустой выбор оставляет значение ManualOverrideEnable таким как оно есть.

```
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=':"Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

Раскрывающийся список включен в форму на HTML-странице. Когда пользователь нажимает кнопку подтверждения, страница отправляет форму, которая пишет значение "1" в булевый тер ManualOverrideEnable в Data_block_1, если пользователь выбрал "Yes", или "0", если пользователь выбрал "No".

12.7.8.4. Запись введенных пользователем данных в контроллер

HTML-страница "Remote Wind Turbine Monitor" содержит несколько AWP-команд для записи данных в контроллер (стр. 818). HTML-страница объявляет AWP_In_Variables для булевых переменных так, чтобы пользователь с привилегиями по изменению переменных мог перевести ветряную турбину на ручное управление и разрешить ручную коррекцию скорости турбины, угла поворота и/или наклона лопастей. Страница также использует AWP_In_Variables, чтобы позволить пользователю с привилегиями по изменению переменных выполнить последующую установку вещественных значений для скорости турбины, угла поворота, наклона и процента торможения. Страница использует команду передачи HTTP-формы, чтобы записать AWP_In_Variables в контроллер.

Например, рассмотрите HTML-код для задания вручную значения торможения:



Пример HTML-кода

Следующая выдержка из HTML-страницы "Remote Wind Turbine Monitor" вначале объявляет AWP_In_Variable для "Data_block_1", которая позволяет HTML-странице выполнять запись в любые теги в блоке данных "Data_block_1". Страница отображает текст "Braking:" в левой ячейке строки таблицы. В правой ячейке строки таблицы находится поле, которое принимает введенные пользователем данные для тега "Braking" в "Data_block_1". Это введенное значение находится в HTML-форме, которая использует метод HTTP "POST", чтобы отправить введенные текстовые данные в ЦПУ. Страница затем читает фактическое значение торможения из контроллера и отображает его в поле ввода данных.

Пользователь с привилегиями по изменению переменных может впоследствии использовать эту страницу, чтобы записать значение торможения в блок данных в ЦПУ, который управляет торможением.

```
<!-- AWP_In_Variable Name='"Data_block_1"' -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
```

```
<p><input name='\"Data_block_1\".Braking' size='10' type='text'>
%</p>
</form>
</td>
</tr>
```

Примечание

Обратите внимание на то, что, если у пользовательской страницы есть поле ввода данных для перезаписываемого тега блока данных, который является строковым типом данных, пользователь должен заключить строку в одиночные кавычки при вводе строкового значения в поле.

Примечание

Обратите внимание на то, что, если Вы объявляете целый блок данных в объявлении AWP_In_Variable, таком как <!-- AWP_In_Variable Name="Data_block_1" -->, тогда каждый тег в этом блоке данных может быть записан из пользовательской веб-страницы. Используйте это, когда Вы намереваетесь сделать все теги в блоке данных перезаписываемыми. В противном случае, если Вы хотите, чтобы только определенные теги блока данных были перезаписываемыми из пользовательской веб-страницы, объявите их частным образом, например, <!-- - Имя AWP_In_Variable = "Data_block_1".Braking'--->

12.7.8.5. Запись специальной переменной

Веб-страница "Remote Wind Turbine Monitor" записывает специальную переменную SERVER:current_user_id в ПЛК тег в ЦП при условии, что пользователь обладает привилегиями по изменению. В этом случае значение ПЛК тега содержит идентификатор пользователя, который получает доступ к веб-странице "Remote Wind Turbine Monitor".

Веб-страница записывает специальную переменную в ПЛК и не требует никакого пользовательского интерфейса.

Пример HTML-кода

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID" -->
```

12.7.8.6. Справка: Распечатка HTML-кода веб-страницы удаленного контроля ветряной турбины

Wind_turbine.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--
This test program simulates a Web page to monitor and control a
Wind Turbine
Required PLC tags and Data Block Tags in STEP 7:

PLC Tag:
User_ID: Int

Data Blocks: Data_block_1

Tags in Data_Block_1:

TurbineNumber: Int
WindSpeed: Real
WindDirection: Real Tem-
perature: Real
PowerOutput: Real
ManualOverrideEnable:
Bool TurbineSpeed: Real
YawOverride: Bool
Yaw: Real
PitchOverride:
Bool Pitch: Real
Braking: Real
The user-defined Web page displays current values for the PLC data,
and provides a select list to set the three Booleans using an enu-
merated type assignment. The "Submit" button posts the selected
Boolean values as well as the data entry fields for TurbineSpeed,
Yaw, and Pitch. The value for Braking can be set without use of the
"Submit" button.

No actual STEP 7 program is required to use this page. Theoretical-
ly, the STEP 7 program would only act on the values of
TurbineSpeed, Yaw, and Pitch, if the associated Booleans were set.
The only STEP 7 requirement is to call the WWW instruction with the
DB number of the generated data blocks for this page.
-->
<!-- AWP_In_Variable Name='"Data_block_1"' -->
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name='"Data_block_1".PitchOverride'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name='"Data_block_1".YawOverride'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->

<html>
```

```

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-
8"><link rel="stylesheet" href="Wind_turbine.css">
<title>Remote Wind Turbine Monitor</title>
</head>
<body>
<table cellpadding="0" cellspacing="2">
<tr style="height: 2%;">
<td colspan="2">
<h2>Remote Wind Turbine Monitor: Turbine
#:="Data_block_1".TurbineNumber:</h2>
</td>

<tr style="height: 2%;"><td style="width: 25%;"><p>Wind
speed:</p></td>
<td><p> :="Data_block_1".WindSpeed: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Wind direction:</p></td>
<td><p> :="Data_block_1".WindDirection: deg.</p></td>
</tr>

<tr style="height: 2%;"><td style="width:
25%;"><p>Temperature:</p></td>
<td><p> :="Data_block_1".Temperature: deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Power output:</p></td>
<td><p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput:
kW</p>
</td>
</tr>

<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=25%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".ManualOverrideEnable'>
<option value=' :="Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
25%;"><p>Turbine speed:</p></td>
<td>

```

```

<p style="margin-bottom:5px;"><input
name='"Data_block_1".TurbineSpeed' size="10" val-
ue=':="Data_block_1".TurbineSpeed:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Yaw override: :="Data_block_1".YawOverride: </p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".YawOverride'>
<option value=':="Data_block_1".YawOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Turbine yaw:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name='"Data_block_1".Yaw'
size="10" value=':="Data_block_1".Yaw:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Pitch override: :="Data_block_1".PitchOverride: </p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".PitchOverride'>
<option value=':="Data_block_1".PitchOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width=25%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Blade pitch:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name='"Data_block_1".Pitch'
size="10" value=':="Data_block_1".Pitch:' type="text"> deg.</p>
</td>

```

```
</tr>
<tr style="height: 2%;">
<td colspan="2">
<input type="submit" value="Submit override settings and values">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;"><p>Braking:</p></td>
<td>
<form method="POST" action="">
<p> <input name="'Data_block_1'.Braking' size="10" val-
ue=':="Data_block_1".Braking:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>
```

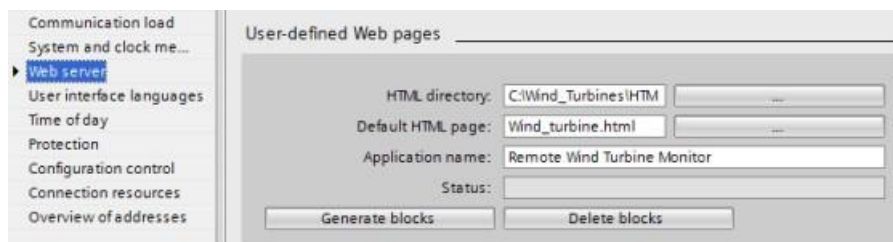

Wind_turbine.css

```
BODY {
background-image: url('../Wind_turbine.jpg');
background-position: 0% 0%;
background-repeat: no-repeat;
background-size: cover;
}
H2 {
font-family: Arial;
font-weight: bold;
font-size: 14.0pt;
color: #FFFFFF;
margin-top:0px;
margin-bottom:10px;
}
P {
font-family: Arial;
font-weight: bold;
color: #FFFFFF; font-size: 12.0pt;
margin-top:0px;
margin-bottom:0px;
}
TD.Text {
font-family: Arial;
font-weight: bold;
color: #FFFFFF;
font-size: 12.0pt;
margin-top:0px;
margin-bottom:0px;
}
```

12.7.8.7. Пример конфигурирования веб-страницы в STEP 7

Чтобы включать HTML-страницу "Remote Wind Turbine Monitor" как пользовательскую веб-страницу для S7-1200, Вы конфигурируете данные о HTML-странице в STEP 7 и создаете блоки данных из HTML-страницы.

Обратитесь к свойствам ЦПУ для S7-1200, который управляет ветряной турбиной и введите конфигурационную информацию в User-defined web pages свойства для веб-сервера:



Поля конфигурирования

- **HTML directory:** Это поле определяет полный путь к папке на компьютере, где расположена страница по умолчанию (домашняя страница или начальная страница). Кнопка "..." позволяет Вам найти путь к папке, которая Вам необходима.
- **Default HTML page:** Это поле определяет имя файла страницы по умолчанию или домашней страницы HTML-приложения. Кнопка "..." позволяет Вам выбирать необходимый файл. Для этого примера WindTurbine.html является HTML-страницей по умолчанию. Пример по удаленному контролю ветряной турбины состоит только из одной страницы, но в других пользовательских приложениях страница по умолчанию может вызывать дополнительные страницы с помощью ссылок. В HTML-коде страница по умолчанию должна сослаться на другие страницы относительно исходной HTML-папки.
- **Application name:** Это дополнительное поле содержит имя, которое веб-браузер включает в поле адреса при отображении страницы. Для данного примера - это "Remote Wind Turbine Monitor", но Вы можете использовать любое имя.

Никакие другие поля не требуют конфигурации.

Завершающие шаги

Чтобы использовать удаленный контроль ветряной турбиной, как он сконфигурирован, сгенерируйте блоки, запрограммируйте инструкцию WWW (стр. 831) с номером сгенерированного управляющего DB в качестве входного параметра, загрузите программные блоки и переведите ЦПУ в режим исполнения.

Когда оператор впоследствии обращается к стандартным веб-страницам S7-1200, который управляет ветряной турбиной, веб-страница "Remote Wind Turbine Monitor" доступна по ссылке "User Pages" на панели навигации. Эта страница теперь обеспечивает средства по контролю и управлению ветряной турбиной.

12.7.9. Создание пользовательских веб-страниц на нескольких языках

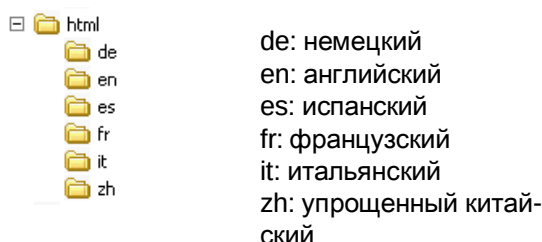
Веб-сервер предоставляет средства для создания пользовательских веб-страниц на следующих языках:

- Немецкий (de)
- Английский (en)
- Испанский (es)
- Французский (fr)
- Итальянский (it)
- Упрощенный китайский (zh)

Вы делаете это, создавая Ваши HTML-страницы в структуре папок (стр. 847), которая соответствует языкам, и устанавливая определенный куки "siemens_automation_language" из Ваших страниц (стр. 847). Веб-сервер отвечает на этот куки и переключается на страницу по умолчанию в соответствующей языковой папке.

12.7.9.1. Создание структуры папок

Чтобы обеспечить пользовательские веб-страницы на нескольких языках, Вы организуете структуру папок в соответствии со своим HTML-каталогом. Имена папок из двух букв являются специфическими, и папки должны быть названы как показано ниже:



На том же уровне Вы можете также добавлять любые другие папки, которые нужны Вашим страницам, например, папки для изображений или сценариев.

Вы можете включать любое подмножество языковых папок. Вы не должны включать все шесть языков. В языковых папках Вы создаете и программируете свои HTML-страницы на соответствующем языке.

12.7.9.2. Программирование переключателя языка

Веб-сервер выполняет переключение между языками с помощью куки с именем "siemens_automation_language". Это куки, определенный и установленный на HTML-страницах и интерпретируемый веб-сервером, чтобы отобразить страницу на соответствующем языке из языковой папки с тем же именем. HTML-страница должна включать JavaScript, чтобы установить этот куки на один из заранее определенных идентификаторов языка: "de", "en", "es", "fr", "it", или "zh".

Например, если HTML-страница устанавливает куки на "de", то веб-сервер переключается на "de" папку и отображает страницу с именем HTML-страницы по умолчанию, как определено в конфигурации STEP 7 (стр. 851).

Пример

Следующий пример использует HTML-страницу по умолчанию с именем "langswitch.html" в каждой из языковых папок. Также в HTML-каталоге находится папка с именем "script". Папка сценария содержит файл JavaScript с именем "lang.js". Каждая langswitch.html страница использует этот JavaScript, чтобы установить языковой куки - "siemens_automation_language".

HTML для "langswitch.html" в папке "en"

Заголовок HTML-страницы устанавливает язык на английский, устанавливает набор символов на UTF-8 и устанавливает путь к JavaScript файлу lang.js.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en"> <meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page
</title>
<script type="text/javascript" src="script/lang.js" ></script>
```

Тело файла использует список выборки для пользователя, чтобы выбрать между немецким и английским языками. Предварительно выбран английский язык ("en"). Когда пользователь изменяет язык, страница вызывает JavaScript функцию DoLocalLanguageChange () со значением выбранной опции.

```
<!-- Language Selection -->
<table>
<tr>
<td align="right" valign="top" nowrap>
<!-- change language immediately on selection change
-->
<select name="Language"
onchange="DoLocalLanguageChange(this)"
size="1">
<option value="de" >German</option>
<option value="en" selected >English</option>
</select>
</td>
</tr>
</table><!-- Language Selection End-->
```

HTML для "langswitch.html" в папке "de"

Заголовок для немецкой страницы langswitch.html такой же как и для английского языка, за исключением установки немецкого языка.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta
http-equiv="Content-Type" content="text/html; char-
set=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>
```

HTML на немецкой странице идентичен коду английской страницы, за исключением того, что языком по умолчанию является немецкий ("de").

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top"
        nowrap>
      <!-- change language immediately on change of the selection
      -
    ->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        <size="1">
          <option value="de" selected >Deutsch</option>
          <option value="en" >Englisch</option>
        </select>
      </td>
    </tr>
  </table><!-- Language Selection End-->
```

JavaScript "lang.js" в папке "script"

Функция "DoLocalLanguageChange()" находится в файле lang.js. Она вызывает функцию "SetLangCookie()" и затем перезагружает окно, которое отображает HTML-страницу.

Функция "SetLangCookie()" создает присвоение, которое присваивает куки документа "siemens_automation_language" значение из списка выборки. Она также устанавливает путь к приложению так, чтобы переключаемая страница, а не запрашивающая страница получила значение куки.

Дополнительно, в разделе комментариев, страница могла бы установить значение истечения срока действия для куки.

```
function DoLocalLanguageChange(oSelect) {
  SetLangCookie(oSelect.value); top.window.location.reload();
}
function SetLangCookie(value) {
  var strval = "siemens_automation_language=";
  // This is the cookie by which the Web server
  // detects the desired language
  // This name is required by the Web server.
  strval = strval + value;
  strval = strval + "; path=/ ";
  // Set path to the application, since otherwise
  // path would be set to the requesting page
  // and this page would not get the cookie.
  /* OPTIONAL
   use expiration if this cookie should live longer
   than the current browser session:
   var now = new Date();
   var endttime = new Date(now.getTime() + expiration);
   strval = strval + "; expires=" +
               endttime.toGMTString() + ";";
  */
  document.cookie = strval;
}
```

Примечание

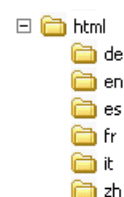
Если реализация Вашей пользовательской веб-страницы содержит HTML-файлы в определенных языковых папках (например en, de), а также HTML-файлы, которые не находятся в определенных языковых папках, обратите внимание на то, что Вы не можете определить типы перечисления с помощью команды AWP_Enum_Def в файлах, находящихся в обоих расположениях. Если Вы используете перечисления, Вы должны определить их в файлах внутри или вне определенных языковых папок. Вы не можете выполнить объявления перечислений в файлах обоих расположений.

12.7.9.3. Конфигурирование STEP 7 для использования многоязыковой структуры страницы

Процедура для конфигурирования многоязыковых пользовательских веб-страниц подобна общему процессу конфигурирования пользовательских веб-страниц (стр. 830). У Вас есть папки для языков, однако, Вы выбираете в качестве HTML-каталога папку, которая содержит отдельные языковые папки. Вы не выбираете в качестве HTML-каталога одну из языковых папок.

Когда Вы выбираете HTML-страницу по умолчанию, Вы перемещаетесь в языковую папку и выбираете HTML-страницу, которая должна быть начальной страницей. Когда Вы впоследствии генерируете блоки и загружаете блоки в ЦПУ, веб-сервер отображает начальную страницу в языковой папке, которую Вы сконфигурировали.

Например, если бы структура папок, показанная здесь, была бы в C:\, настройкой для HTML-каталога была бы C:\html, и, если страница должна была бы изначально отображаться на английском языке, Вы переместились бы к en\langswitch.html как к HTML-странице по умолчанию.



12.7.10. Расширенное управление пользовательской веб-страницей

Когда Вы генерируете блоки данных для своих пользовательских веб-страниц, STEP 7 создает управляющий DB, который он использует, чтобы управлять отображением и взаимодействием с пользовательскими страницами. STEP 7 также создает набор DB фрагментов, которые представляют отдельные страницы. При нормальных обстоятельствах Вы не должны знать структуру управляющего DB или как работать с ним.

Если Вы, например, хотите включать или выключать веб-приложение или управлять отдельными ручными фрагментами, Вы используете теги управляющего DB и инструкцию WWW.

Структура управляющего DB

Управляющий DB - это большая структура данных, которая доступна при программировании пользовательской STEP 7 программы. Здесь описаны лишь некоторые теги управляющего блока данных.

Структура Commandstate

"Commandstate" является структурой, которая содержит глобальные команды и глобальные состояния веб-сервера.

Глобальные состояния в структуре "Commandstate"

Глобальные команды применяются к веб-серверу в целом. Вы можете деактивировать веб-сервер или перезапустить его с помощью параметров управляющего DB.

Тег блока	Тип данных	Описание
init	BOOL	Оценить управляющий DB и инициализировать веб-приложение
deactivate	BOOL	Деактивировать веб-приложение

Глобальные состояния в структуре Commandstate

Глобальные состояния применяются к веб-серверу в целом и содержат информацию о статусе веб-приложения.

Тег блока	Тип данных	Описание
initializing	BOOL	Веб-приложение читает управляющий DB
error	BOOL	Веб-приложение не могло быть инициализировано
deactivating	BOOL	Веб-приложение завершается
deactivated	BOOL	Веб-приложение завершено
initialized	BOOL	Веб-приложение инициализировано
last_error	INT	Последняя ошибка, возвращенная вызовом инструкции WWW (стр. 831), когда код возврата WWW равен 16#0010: 16#0001: структура DB фрагмента не согласована 16#0002: имя приложения уже существует 16#0003: нет ресурсов (памяти) 16#0004: структура DB управления не согласована 16#0005: DB фрагмента недоступен 16#0006: DB фрагмента не для AWP 16#0007: данные перечисления не согласованы 16#000D: конфликт размера управляющего DB

Таблица запросов

Таблица запросов представляет собой массив структур, содержащих команды и состояния, которые применяются к отдельному DB фрагмента. Если бы создали фрагменты с командой AWP_Start_Fragment (стр. 826) типа "ручной", то пользовательская STEP 7 программа должна управлять этими страницами с помощью управляющего DB. Состояния запроса имеют статус только для чтения и предоставляют информацию о текущем фрагменте. Вы используете команды запроса, чтобы управлять текущим фрагментом.

Тег блока	Тип данных	Описание
requesttab	ARRAY [1 .. 4] OF STRUCT	Массив структур для индивидуального управления DB фрагментов. Веб-сервер может обработать до четырех фрагментов за один раз. Индекс массива для определенного фрагмента произволен, когда веб-сервер обрабатывает несколько фрагментов или фрагменты из нескольких

Компоненты структуры requesttab

Тег блока	Тип данных	Описание
page_index	UINT	Номер текущей веб-страницы
fragment_index	UINT	Номер текущего фрагмента - может быть определен для другого фрагмента
// Request Commands		
continue	BOOL	Разрешает передачу текущих страницы / фрагмента и продолжение на следующем фрагменте
repeat	BOOL	Разрешает повторную передачу текущих страницы / фрагмента и продолжение на следующем фрагменте
abort	BOOL	Закрывает http соединение без передачи
finish	BOOL	Передать этот фрагмент; страница завершена – не обрабатывать дополнительные фрагменты
// Состояния запроса		Состояния запроса имеют статус только для чтения
idle	BOOL	Ничего не делать, но быть активным
waiting	BOOL	Фрагмент ожидает разрешения
sending	BOOL	Фрагмент передается
aborting	BOOL	Пользователь прервал текущий запрос

Работа

Каждый раз, когда Ваша программа вносит изменения в управляющий DB, она должна вызывать инструкцию WWW с номером измененного управляющего DB в качестве параметра. Глобальные команды и команды запроса вступают в силу, когда пользовательская STEP 7 программа выполняет инструкцию WWW (стр. 831).

Пользовательская STEP 7 программа может установить fragment_index явно, таким образом заставив веб-сервер обработать указанный фрагмент по команде запроса. В противном случае, веб-сервер обрабатывает текущий фрагмент для текущей страницы, когда выполняется инструкция WWW.

Возможные методы использования fragment_index следующие:

- Обработка текущего фрагмента: Оставьте fragment_index неизменным и установите команду продолжения.
- Пропустить текущий фрагмент: Установите fragment_index на 0 и установите команду продолжения.
- Заменить текущий фрагмент другим фрагментом: Установите fragment_index на новый ID фрагмента и установите команду продолжения.

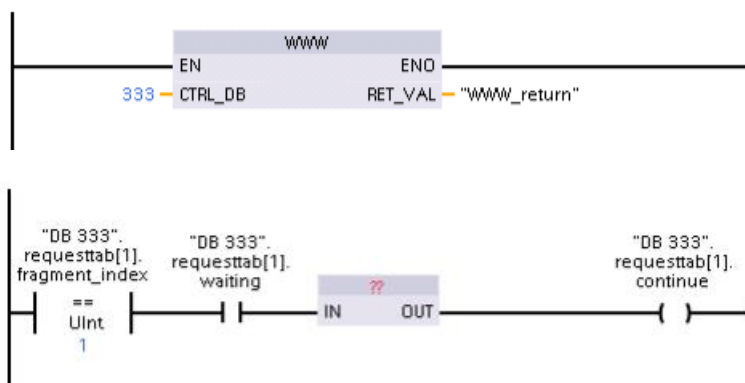
Чтобы проверить глобальные состояния или состояния запроса, которые, возможно, изменились, пользовательская STEP 7 программа должна вызвать инструкцию WWW, чтобы оценить текущие значения этих состояний. Типовым использованием мог бы быть периодический вызов инструкции WWW, пока не наступит определенное состояние.

Примечание

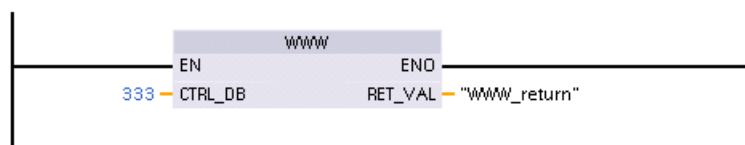
Если пользовательская STEP 7 программа выставляет больше чем одну команду запроса, инструкция WWW обрабатывает только одну в следующем порядке очередности: abort, finish, repeat, continue. Инструкция WWW сбрасывает все из команд запроса после обработки.

Примеры

Следующий пример показывает пользовательскую STEP 7 программу, которая проверяет фрагмент с ID 1 на состояние ожидания, после предшествующего вызова инструкции WWW. Это могло бы также ожиданием других определенных условий. Затем она выполняет, любую необходимую обработку фрагмента, такую как установка тегов блока данных, выполнение вычислений или другие специализированные задачи. Впоследствии, она устанавливает флаг continue, чтобы веб-сервер исполнил этот фрагмент.



Когда программа вызывает инструкцию WWW с этим измененным управляющим DB, пользовательская веб-страница с этим фрагментом может быть отображена веб-браузером.



Обратите внимание на то, что это упрощенный пример; проверяемый фрагмент мог бы находиться в любой из четырех структур requesttab в массиве.

12.8 Ограничения

Следующие факторы, имеющие отношение к информационным технологиям, могут влиять на использование веб-сервера:

- Как правило, Вы должны использовать IP-адрес ЦПУ, чтобы получить доступ к стандартным веб-страницам или пользовательским веб-страницам или IP-адрес беспроводного маршрутизатора с номером порта. Если Ваш веб-браузер не позволяет соединяться непосредственно с IP-адресом, обратитесь к своему системному администратору. Если Ваша локальная политика поддерживает DNS, Вы можете соединиться с IP-адресом посредством записи DNS в этот адрес.
- Брандмауэры, настройки для прокси и другие специфичные для сайта ограничения могут также ограничить доступ к ЦПУ. Обратитесь к своему системному администратору, чтобы решить эти вопросы.
- Стандартные Веб-страницы используют JavaScript и куки. Если Ваши настройки веб-браузера запрещают JavaScript или куки, измените их. Если Вы не можете изменить их, некоторые функции будут ограничены (стр. 856). Использование JavaScript и куки в пользовательских веб-страницах необязательное. Если они используются, Вы должны активировать их в Вашем браузере.
- Веб-сервер поддерживает Уровень защищенных сокетов (SSL). Вы можете получить доступ к стандартным веб-страницам и пользовательским веб-страницам с помощью URL, по-другому `http://www.xx.yy.zz` или `https://www.xx.yy.zz`, где "www.xx.yy.zz" представляет IP-адрес ЦПУ.
- Сименс обеспечивает сертификат безопасности для безопасного доступа к веб-серверу. Со стандартной веб-страницы Introduction (стр. 800) Вы можете загрузить и импортировать сертификат в интернет-опции Вашего веб-браузера (стр. 857). Если Вы примете решение не импортировать сертификат, то Вы будете получать приглашение на проверку безопасности каждый раз, когда Вы обратитесь к веб-серверу с помощью `https://`.

Количество соединений

Веб-сервер поддерживает максимум 30 активных HTTP-соединений. Различные действия используют эти 30 соединений, в зависимости от веб-браузера, который Вы используете и числа различных объектов на странице (.css файлы, изображения, дополнительные .html файлы). Некоторые соединения сохраняются, в то время как веб-сервер отображает страницу; другие соединения не сохраняются после начального соединения.

Если, например, Вы используете Mozilla Firefox 8, который поддерживает максимум шесть постоянных соединений, Вы могли бы использовать пять браузеров или экземпляров вкладок браузера, прежде чем веб-сервер начнет сбрасывать соединения. В случае, если страница не использует все шесть соединений, Вы смогли бы использовать дополнительный браузер или экземпляры вкладок браузера.

Также примите к сведению, что число активных соединений может влиять на производительность страницы.

Примечание

Завершайте сеанс прежде, чем закрывать веб-сервер

Если Вы проходили авторизацию на веб-сервере, убедитесь что Вы завершили сеанс до закрытия Вашего веб-браузера. Веб-сервер поддерживает максимум семь параллельных авторизаций.

12.8.1. Ограничения функциональности, когда интернет-опции отключают JavaScript

Стандартные веб-страницы используют HTML, JavaScript и куки. Если Ваш сайт ограничивает использование JavaScript и куки, то разрешите их для надлежащего функционирования страниц. Если Вы не можете включить JavaScript для своего веб-браузера, то функции, использующие средства управления JavaScript, не смогут работать.

Общие сведения

Страницы не обновляют динамически. Вы должны вручную обновить страницу с помощью пиктограммы Refresh (стр. 794), чтобы просмотреть новые данные.

Страница Diagnostic Buffer

Страница Diagnostic Buffer использует JavaScript следующим образом:

- Отображение деталей события: С JavaScript Вы выбираете строку в диагностическом буфере, чтобы видеть детали в нижнем окне. Без JavaScript Вы должны щелкнуть по гиперссылке для записи о событии в диагностическом буфере, чтобы просмотреть данные о событии в нижнем разделе.
- Изменение диапазона просмотра записей в диагностическом буфере: С JavaScript Вы используете выпадающий список наверху, чтобы выбрать диапазон записей в диагностическом буфере, и страница автоматически обновляется. Без JavaScript Вы используете выпадающий список наверху, чтобы выбрать диапазон записей в диагностическом буфере, но Вы должны затем щелкнуть по ссылке "Go", чтобы получить диапазон записей диагностического буфера, который Вы выбрали из выпадающего списка.

Обратите внимание на то, что гиперссылка "Go" и гиперссылки события видны только тогда, когда JavaScript не активирован. В них нет необходимости, и поэтому они отсутствуют, когда JavaScript включен.

Страница Module Information

При неактивном JavaScript применимы следующие ограничения:

- Вы не можете фильтровать данные.
- Вы не можете сортировать поля.

Страница Variable Status

При неактивном JavaScript применимы следующие ограничения:

- После того, как Вы вводите каждую переменную, Вы должны вручную установить фокус на строку "New variable", чтобы ввести новую переменную.
- Выбор формата отображения не преобразовывает автоматически значение данных в выбранный формат. Вы должны нажать кнопку "Monitor value", чтобы обновить отображение в новом формате.

12.8.2. Ограничения функциональности, когда интернет-опции отключают куки

Если Вы отменили куки в Вашем веб-браузере, применимы следующие ограничения:

- Вы не можете авторизоваться.
- Вы не можете изменить языковую настройку.
- Вы не можете переключиться с времени с UTC на время ПЛК. Без куки действует UTC время.

12.8.3. Импорт сертификата безопасности Сименс

Вы можете импортировать сертификат безопасности Сименс в свои интернет-опции так, чтобы Вам не предлагали проверку безопасности, когда Вы вводите `https://www.xx.yy.zz` в Вашем веб-браузере, где `"www.xx.yy.zz"` является IP-адресом устройства. Если Вы используете `http:// URL`, а не `https:// URL`, то Вы не должны загружать и устанавливать сертификат.

Загрузка сертификата

Вы используете гиперссылку "download certificate" на странице Introduction (стр. 800), чтобы загрузить сертификат безопасности Сименс в Ваш ПК. Процедура варьируется, согласно используемому Вами веб-браузеру:

Импорт сертификата в Internet Explorer

1. Щелкните по ссылке "download certificate" на странице Introduction. Появляется диалоговое окно "File Download - Security Warning".
2. В диалоговом окне "File Download - Security Warning" нажмите "Open", чтобы открыть файл. Появляется диалоговое окно "Certificate".
3. В диалоговом окне "Certificate" нажмите кнопку "Install Certificate", чтобы запустить Мастер импорта Сертификата.
4. Следуйте за диалоговыми окнами "Certificate Import Wizard", чтобы импортировать сертификат, позволяя операционной системе автоматически выбрать хранилище сертификата.

Импорт сертификата в Mozilla Firefox

1. Щелкните по ссылке "download certificate" на странице Intro. Появляется диалоговое окно "Opening MiniWebCA_Cer.crt" .
2. Нажмите "Save file" в диалоговом окне "Opening MiniWebCA_Cer.crt" . Появляется диалоговое окно "Downloads" появляется.
3. В диалоговом окне "Downloads" дважды щелкните по "MiniWebCA_Cer.crt". Если Вы делали попытку загрузки несколько раз, обнаруживаются несколько копий. Просто дважды щелкните по любой из записей "MiniWebCA_Cer.crt".
4. Нажмите "OK", при запросе, чтобы открыть исполняемый файл.
5. Щелкните по "Open" в диалоговом окне "Open File - Security Warning" , если оно появляется. Появляется диалоговое окно "Certificate" .
6. В диалоговом окне "Certificate" нажмите кнопку "Install Certificate".
7. Следуйте за диалоговыми окнами "Certificate Import Wizard", чтобы импортировать сертификат, позволяя операционной системе автоматически выбрать хранилище сертификата.
8. Если возникает диалоговое окно "Security Warning" , нажмите "Yes", чтобы подтвердить установку сертификата.

Другие браузеры

Следуйте соглашениям своего веб-браузера, чтобы импортировать и установить сертификат Сименс.

После того, как Вы установили сертификат безопасности Сименс "Семейство контроллеров S7-1200", в интернет-опциях для контента Вашего веб-браузера, у Вас пропадет необходимость подтверждать проверку безопасности, когда Вы обращаетесь к веб-серверу с помощью <https://www.xx.yy.zz>.

Примечание

Сертификат безопасности остается постоянным, несмотря на перезагрузки ЦПУ; однако, если Вы изменяете IP-адрес устройства, Вы должны загрузить новый сертификат, если Вы используете браузер отличный от Internet Explorer или Mozilla Firefox.

12.8.4. Импорт журналов данных в CSV формате в non-USA/UK версии Microsoft Excel

Файлы журналов данных имеют формат файла значений разделенных запятой (CSV). Вы можете открыть эти файлы непосредственно в Excel со страницы Data Logs, когда Ваша система использует американскую или британскую версию Excel. В других странах, однако, этот формат используется не так широко, потому что запятые часто применяют в разделении разрядов чисел.

Чтобы открыть файл журнала данных, который Вы сохранили, выполните следующие шаги для версии Excel отличной от американской / британской:

1. Откройте Excel и создайте пустую рабочую книгу.
2. Из меню "Data > Import External Data" выберите команду "Import Data".
3. Перейдите и выберите файл журнала данных, который Вы хотите открыть. Запускается Мастер Импорта Текста.
4. В Мастере Импорта Текста измените опцию по умолчанию для "Original data type" с "Fixed width" на "Delimited".
5. Нажмите кнопку Next.
6. В диалоговом окне Шага 2 установите флажок "Comma", чтобы изменить тип разделителя с "Tab" на "Comma".
7. Нажмите кнопку Next.
8. В диалоговом окне Шага 3 Вы можете дополнительно изменить формат даты с MDY (месяц/день/год) на другой формат.
9. Завершите оставшиеся шаги Мастера Импорта Текста, чтобы импортировать файл.

13.1 Применение последовательных коммуникационных интерфейсов

Два коммуникационных модуля (СМ) и одна коммуникационная плата (СВ) обладают интерфейсом для PtP-коммуникаций:

- СМ 1241 RS232 (стр. 1263)
- СМ 1241 RS422/485 (стр. 1264)
- СВ 1241 RS485 (стр. 1260)

Вы можете подключить до трех СМ (любого типа) плюс СВ для получения, в общей сложности, четырех коммуникационных интерфейсов. Установите СМ слева от ЦПУ или другого СМ. Установите СВ на передней стороне ЦПУ. Обратитесь к инструкциям по установке (стр. 64) для получения информации об установке и снятии модуля.

Последовательные интерфейсы имеют следующие характеристики:

- обладают изолированным портом;
- поддерживают протоколы "точка-к-точке";
- конфигурируются и программируются через инструкции коммуникационного процессора "точка-к-точке";
- отображают активность, связанную с передачей и приемом посредством светодиодов;
- обладают диагностическим светодиодом (только СМ);
- получают питание от ЦПУ: не требуют какого-либо внешнего питания.

Обратитесь к спецификациям для коммуникационных интерфейсов (стр. 1250).

Светодиодные индикаторы

Коммуникационные модули имеют три светодиодных индикатора:

- **Диагностический светодиод (DIAG):** Этот светодиод мигает красным, пока ЦПУ назначает адрес для СМ. После того, как на ЦПУ подается питание, он проверяет наличие СМ и назначает им адреса. Диагностический светодиод начинает мигать зеленым. Это означает, что ЦПУ назначил адрес СМ, но еще не обеспечил его конфигурирование. ЦПУ загружает конфигурацию в СМ при загрузке в ЦПУ программы. После загрузки в ЦПУ диагностический светодиод коммуникационного модуля, должен светиться зеленым непрерывно.
- **Светодиод передачи (Tx):** загорается, когда коммуникационный порт передает данные.
- **Светодиод приема (Rx):** этот светодиод загорается, когда коммуникационный порт получает данные.

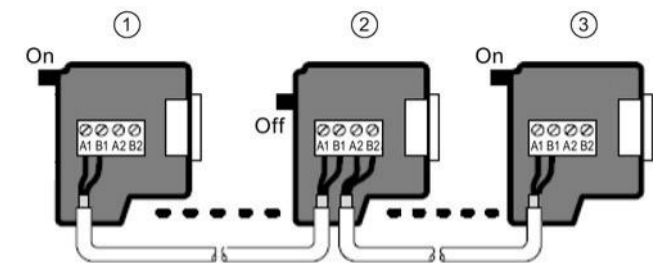
Коммуникационная плата обладает светодиодами передачи (TxD), и приема(RxD). У нее нет диагностического светодиода.

13.2 Смещение и конечная нагрузка в сетевом RS485-разъеме

Сименс предлагает сетевой RS485-разъем (стр. 1280), который Вы можете использовать для простого подключения нескольких устройств в RS485-сеть. Соединитель имеет две группы клемм, которые позволяют Вам подключать приходящие и уходящие сетевые кабели. Соединитель также обладает переключателями для того, чтобы выборочно осуществлять подачу смещения и подключать конечную нагрузку сети.

Примечание

Вы подключаете конечную нагрузку и подаете смещение только с двух концов RS485-сети. Промежуточные устройства, находящиеся между этими двумя конечными устройствами, не требуют этого. Экран разделанного кабеля должен контактировать с металлическими направляющими в местах всех подключений на участке в 12 мм (1/2 дюйма).



- ① Положение переключателя = On: Подключение конечной нагрузки и подача смещения
- ② Положение переключателя = Off: Без подключения нагрузки и подачи смещения
- ③ Положение переключателя = On: Подключение конечной нагрузки и подача смещения

Таблица 13- 1 Подключение конечной нагрузки и подача смещения для RS485-разъема

Оконечное устройство (смещение ВКЛ.)	Промежуточное устройство (смещение ВЫКЛ.)

- ① Номер контакта
- ② Сетевой разъем
- ③ Экран кабеля

13.2 Смещение и концевая нагрузка в сетевом RS485-разъеме

СВ 1241 обладает внутренними резисторами для завершения и смещения сети. Чтобы завершить и выполнить смещение для подключения, соедините TRA с TA и TRB с TB, чтобы включить внутренние резисторы в схему. У СВ 1241 нет 9-контактного разъема. Следующая таблица показывает соединения с 9-контактным разъемом в коммуникационном партнере.

Таблица 13- 2 Подключение концевой нагрузки и подача смещения для СВ 1241

Оконечное устройство (смещение ВКЛ.)	Промежуточное устройство (смещение ВЫКЛ.)

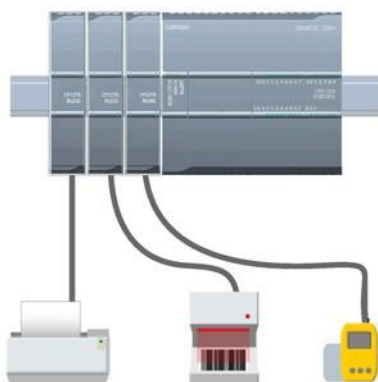
- ① Соедините M с экраном кабеля
- ② A = TxD/RxD - (зеленый провод / контакт 8)
- ③ B = TxD/RxD + (красный провод / контакт 3)

13.3 Коммуникации точка-к-точке (PtP)

ЦПУ поддерживает следующие типы коммуникации "точка-к-точке" (PtP) для символично-ориентированных последовательных протоколов:

- PtP
- USS (стр. 911)
- Modbus (стр. 931)

PtP обеспечивает максимальную свободу и гибкость, но требует значительных усилий для реализации в пользовательской программе.



PtP предоставляет большое разнообразие возможностей:

- Возможность отправить информацию непосредственно во внешнее устройство, такое как принтер
- Возможность получить информацию от других устройств, таких как устройства считывания штрих-кода, RFID-считыватели, сторонняя камера, видеосистемы и множество других типов устройств
- Возможность обмениваться информацией, отправляя и получая данные, с другими устройствами, такими как устройства GPS, сторонняя камера, видеосистемы, радио-модемы и множество других типов устройств

Этот тип PtP-коммуникаций является последовательным обменом, который использует стандартные UART для поддержки множества опций, связанных с четностью и скоростью передачи. RS232 и RS422/485 коммуникационные модули (CM 1241) и коммуникационная плата RS485 (CB 1241) имеют электрические интерфейсы для выполнения PtP-коммуникаций.

PtP по PROFIBUS или PROFINET

Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможность PtP, используя распределенную стойку ввода-вывода PROFINET или PROFIBUS для связи с различными устройствами (RFID-считыватели, устройство GPS и другие):

- PROFINET (стр. 616): вы соединяете Ethernet-порт ЦПУ S7-1200 с интерфейсным модулем PROFINET. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательные коммуникации с PtP-устройствами.
- PROFIBUS (стр. 752): Вы устанавливаете коммуникационный модуль PROFIBUS слева от ЦПУ S7-1200. Вы соединяете коммуникационный модуль PROFIBUS со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные PtP-модули в стойке с интерфейсным модулем могут в этом случае обеспечить последовательные коммуникации с PtP-устройствами.

По этой причине S7-1200 поддерживает два набора PtP-инструкций:

- Наследованные инструкции "точка-к-точке" (стр. 977): Эти инструкции существовали до версии V4.0 S7-1200, и работали только с последовательными коммуникациями с использованием коммуникационного модуля CM 1241 или коммуникационной платы CB 1241.
- Инструкции "точка-к-точке" (стр. 877): Эти инструкции обеспечивают всю функциональность наследованных инструкций плюс возможность подключения через распределенный ввод-вывод PROFINET и PROFIBUS. Инструкции "точка-к-точке" позволяют Вам конфигурировать коммуникации между коммуникационными модулями PtP в стойке распределенного ввода-вывода и PtP-устройствами.

Примечание

С версией V4.1 S7-1200 Вы можете использовать инструкции "точка-к-точке" для всех типов коммуникаций: последовательные, последовательные по PROFINET и последовательные по PROFIBUS. STEP 7 предлагает наследованные инструкции "точка-к-точке" только для поддержки существующих программ. Наследованные инструкции, однако, по-прежнему работают с центральными процессорами версий V4.1, V4.0, а также более ранними версиями ЦПУ. Вы не должны преобразовывать ранее созданные программы от одного набора инструкций к другому.

13.3.1. Конфигурирование коммуникационных портов

Вы можете использовать любой из следующих методов, чтобы сконфигурировать коммуникационные интерфейсы:

- Используйте конфигурацию устройства в STEP 7, чтобы задать параметры порта (скорость и четность), параметры передачи и приема. ЦПУ хранит настройки конфигурации устройства и применяет настройки после цикла подачи питания и перехода из RUN в STOP.
- Используйте Port_Config (стр. 879), Send_Config (стр. 882) и Receive_Config (стр. 884) инструкции, чтобы настроить параметры. Настройки порта, установленные инструкциями, действительны, пока ЦПУ находится в режиме RUN. Настройки порта возвращаются к настройкам конфигурации устройства после перехода в STOP или цикла подачи питания.

После конфигурирования аппаратных модулей (стр. 145), Вы задаете параметры для коммуникационных интерфейсов, выбирая один из CM в Вашей стойке или CB, если они были сконфигурированы.



Вкладка "Properties" окна инспектора показывает параметры выбранного CM или CB. Выберите "Port configuration", чтобы отредактировать следующие параметры:

- Скорость передачи
- Четность
- Количество битов на символ
- Количество стоповых битов
- Контроль передачи (только RS232)
- Время ожидания

13.3 Коммуникации точка-к-точке (PtP)

Для CM 1241 RS232 и CB RS485 (за исключением управления передачей (стр. 867), которую поддерживает только CM 1241 RS232), параметры конфигурации порта являются теми же независимо от того, конфигурируете ли Вы RS232 или RS485 коммуникационный модуль или RS485 коммуникационную плату. Значения параметров могут отличаться.

Для CM 1241 RS422/485 у Вас есть дополнительные опции для конфигурации порта как показано ниже. Режим 422 модуля RS422/485 CM 1241 также поддерживают программное управление передачей.



Выберите "Port configuration", чтобы отредактировать следующие RS422/485 параметры:

- "Operating mode" (рабочий режим):
 - Full duplex (дуплекс) (RS422) четырехпроводной режим (подключение точка-к-точке)
 - Full duplex (дуплекс) (RS422) четырехпроводной режим (многоточечный ведущий)
 - Full duplex (дуплекс) (RS422) четырехпроводной режим (многоточечный ведомый)
 - Half duplex (полудуплекс) (RS485) двухпроводной режим
- "Receive line initial state" (принять начальное состояние линии):
 - None (нет)
 - Forward bias (прямое смещение) (сигнал R(A) 0В, сигнал R(B) 5В)

Пользовательская программа STEP 7 может также сконфигурировать порт или изменить существующую конфигурацию инструкцией Port_Config (стр. 879). Раздел описания инструкции предлагает детальную информацию о режиме работы и начальном состоянии линии, а также других параметрах.

Параметр	Описание
Baud rate	Значение по умолчанию для скорости передачи составляет 9.6 кбит/с. Допустимый выбор: 300 бод, 600 бод, 1.2 кбит/с, 2.4 кбит/с, 4.8 кбит/с, 9.6 кбит/с, 19.2 кбит/с, 38.4 кбит/с, 57.6 кбит/с, 76.8 кбит/с и 115.2 кбит/с.
Parity	Значение по умолчанию для четности является по parity. Допустимый выбор: по parity, even, odd, mark (бит четности всегда установлен в 1) и space (бит четности всегда установлен в 0).
Data bits per character	Число битов данных в символе. Варианты выбора 7 или 8.
Number of stop bits	Число стоповых битов может равняться 1 или 2. Значением по умолчанию является 1.
Flow control	Для коммуникационного модуля RS232 Вы можете выбрать аппаратное или программное управление передачей (стр. 867). Если Вы выбираете аппаратное управление передачей, Вы можете выбрать, включен ли RTS-сигнал всегда, или он переключается. Если Вы выбираете программное управление передачей, то Вы можете определить символы XON и XOFF. Коммуникационные интерфейсы RS485 не поддерживают управление передачей. Режим 422 модуля CM 1241 RS422/485 поддерживают программное управление передачей.
Wait time	Время ожидания определяет время, когда CM или CB ожидают получения CTS после утверждения RTS, или для приема XON после приема XOFF, в зависимости от типа управления потоком. Если время ожидания истекает, прежде чем коммуникационный интерфейс получает ожидаемый CTS или XON, CM или CB прерывают операцию передачи и

	возвращают ошибку пользовательской программе. Вы определяете время ожидания в миллисекундах. Диапазон от 0 до 65535 миллисекунд.
Operating mode	Здесь Вы выбираете рабочий режим RS422 или RS485 и конфигурации сети.
Receive line initial state	Здесь Вы выбираете опции смещения. Допустимые значения: без смещения, прямое смещение и обратное смещение. Обратное смещение используется, чтобы позволить обнаружение обрыва кабеля.

13.3.1.1. Организация управления потоком

Управление передачей относится к механизму для балансирования отправки и получения данных так, чтобы никакие данные не были потеряны. Управление передачей гарантирует, что устройство передачи не отправляет информации больше, чем приемное устройство может обработать. Управление передачей может быть реализовано через аппаратные средства или через программное обеспечение. RS232 CM поддерживает как аппаратное так и программное управления передачей. RS485 CM и CB не поддерживают управление передачей. Режим 422 модуля CM 1241 RS422/485 поддерживает программное управление передачей. Вы определяете тип управления передачей либо при конфигурировании порта (стр. 865), либо с помощью инструкции PORT_CFG (стр. 977).

Аппаратное управление передачей работает через Request-to-send (RTS) и Clear-to-send (CTS) коммуникационные сигналы. С RS232 CM сигнал RTS выводится на контакт 7, а сигнал CTS приходит через контакт 8. RS232 CM является DTE-устройством (оконечное оборудование данных), которое утверждает RTS как выходной сигнал и контролирует CTS как входной сигнал.

Аппаратное управление передачей: переключение RTS

Если Вы включаете аппаратное управление передачей с переключением RTS для RS232 CM, модуль устанавливает сигнал RTS в активное состояние при отправке данных. Он контролирует сигнал CTS, чтобы определить, может ли приемное устройство принять данные. Когда сигнал CTS активен, модуль может передать данные, пока сигнал CTS остается активным. Если сигнал CTS становится неактивным, передача должна прекратиться.

Передача возобновляется, когда сигнал CTS становится активным. Если сигнал CTS неактивен в течение заданного времени ожидания, модуль прерывает передачу и возвращает ошибку пользовательской программе. Вы задаете время ожидания в конфигурации порта (стр. 865).

Управление передачей с переключением RTS полезно для устройств, которым требуется сигнал активации передачи. Примером мог бы послужить радиомодем, который использует RTS в качестве "ключевого" сигнала для подачи питания на радиопередатчик. Управление передачей с переключением RTS не будет работать со стандартными телефонными модемами. Используйте для телефонных модемов вариант RTS всегда включен.

Аппаратное управление передачей: RTS всегда включен

В режиме RTS всегда включен, CM 1241 устанавливает RTS активным по умолчанию. Устройство, такое как телефонный модем, контролирует сигнал RTS от CM и использует этот сигнал как clear-to-send. Модем лишь передает данные в CM, когда RTS активен, т.е. когда телефонный модем видит активный CTS. Если RTS неактивен, телефонный модем не передает данные в CM.

Чтобы позволить модему отправлять данные в CM в любое время, выбирайте аппаратное управление передачей "RTS always on". CM, таким образом, держит сигнал RTS активным постоянно. CM не деактивирует RTS, даже если модуль не может принять символы. Устройство передачи должно гарантировать, что не переполнит приемный буфер CM.

Использование сигналов готовности терминала (DTR) и готовности данных (DSR)

CM устанавливает DTR активным для любого типа аппаратного управления передачей. Модуль выполняет передачу только в том случае, если сигнал DSR становится активным. Состояние DSR оценивается только в начале операции передачи. Если DSR станет неактивным после того, как передача начата, передача не будет приостановлена.

Программное управление передачей

Когда передающее устройство получает символ XOFF от приемного устройства, оно прекращает передачу. Передача возобновляется, когда передающее устройство получает символ XON. Если оно не получает символ XON по истечении времени ожидания, которое определено в конфигурации порта (стр. 865), CM прерывает передачу и возвращает ошибку пользовательской программе.

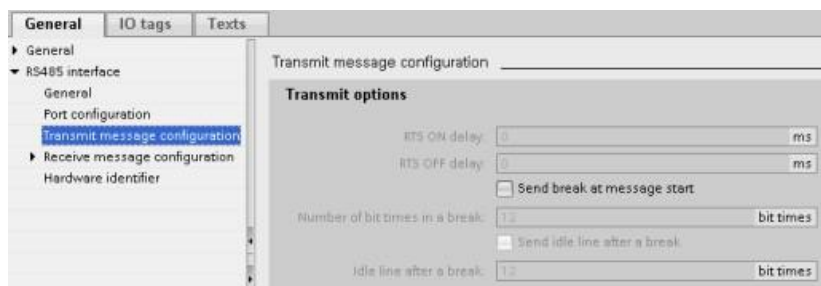
Программное управление передачей требует дуплексных коммуникаций, поскольку принимающая сторона должна быть в состоянии отправить XOFF передающей стороне, во время передачи. Программное управление передачей возможно только с использованием сообщений, содержащих только ASCII-символы. Двоичные протоколы не могут использовать программное управление передачей.

13.3.2. Конфигурирование параметров передачи (отправки) и приема

Прежде чем ЦПУ сможет участвовать в PtP-коммуникациях, Вы должны сконфигурировать параметры для передачи (или отправки) сообщений и приема сообщений. Эти параметры определяют процедуру коммуникаций, когда сообщения передаются или принимаются от целевого устройства.

13.3.2.1. Конфигурирование параметров передачи (отправки)

В конфигурации ЦПУ Вы определяете, как коммуникационный интерфейс передает данные, настраивая свойства "Transmit message configuration" для выбранного интерфейса.



Вы можете также динамически конфигурировать или изменять параметры передачи сообщения из пользовательской программы при помощи инструкции Send_Config (стр. 882).

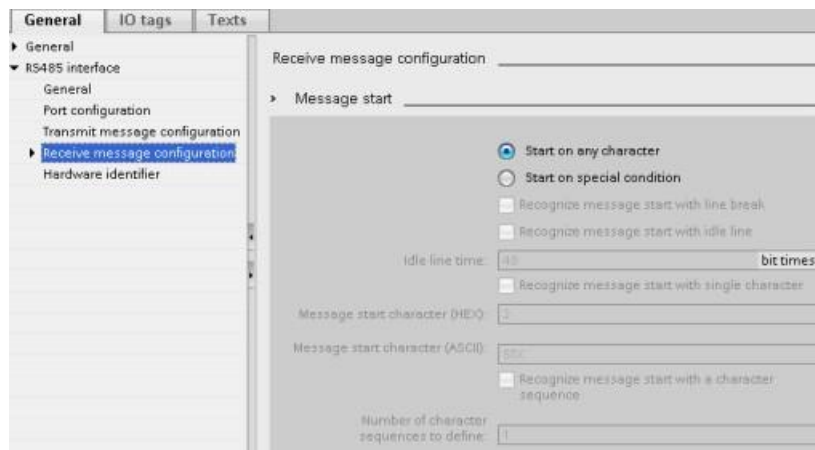
Примечание

Значения параметров установленные инструкцией Send_Config в пользовательской программе переопределяют свойства "Transmit message configuration". Обратите внимание на то, что ЦПУ не сохраняет параметры, установленные инструкцией Send_Config в случае выключения питания.

Параметр	Описание
RTS On delay	Определяет время ожидания после активирования RTS, прежде чем будет инициирована передача. Диапазон от 0 до 65535 мс со значением по умолчанию равным 0. Этот параметр допустим только, когда конфигурация порта (стр. 865) определяет аппаратное управление передачей. CTS оценивается после того, как истекло RTS On delay time. Этот параметр применим только для модулей RS232.
RTS Off delay	Определяет время ожидания деактивирования RTS после завершения передачи. Диапазон от 0 до 65535 мс со значением по умолчанию равным 0. Этот параметр допустим только, когда конфигурация порта (стр. 865) определяет аппаратное управление передачей. Этот параметр применим только для модулей RS232.
Send break at message start Number of bit times in a break	Определяет что с момента запуска каждого сообщения, сигнал окончания передачи будет отправлен после того, как RTS On delay (если сконфигурировано) истекло, и CTS активен. Вы определяете, сколько битовых интервалов составляет сигнал окончания передачи, когда линия удерживается в режиме интервала. Значение по умолчанию составляет 12, а максимумом является 65535, до ограничения равного 8 секундам.
Send idle line after a break Idle line after a break	Определяет, что сигнал свободной линии будет отправлен перед началом сообщения. Он отправляется после сигнала разрыва линии, если таковой сконфигурирован. Параметр "Idle line after a break" определяет, сколько битовых интервалов содержит сигнал свободной линии, когда линия удерживается в режиме маркировки. Значение по умолчанию составляет 12, а максимумом является 65535, до ограничения равного 8 секундам.

13.3.2.2. Конфигурирование параметров приема

В конфигурации устройства ЦПУ Вы определяете, как коммуникационный интерфейс принимает данные, и как он распознает начало и конец сообщения. Вы задаете эти параметры в свойствах "Receive message configuration" для выбранного интерфейса.



Вы можете также динамически сконфигурировать или изменить параметры приема сообщения из пользовательской программы при помощи инструкции Receive_Config (стр. 884).

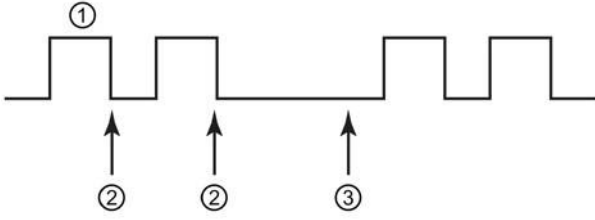
Примечание

Значения параметров установленные инструкцией Receive_Config в пользовательской программе переопределяют свойства "Receive message configuration". Обратите внимание на то, что ЦПУ не сохраняет параметры, установленные инструкцией RCV_CFG в случае выключения питания или перехода в режим STOP.

Условия начала сообщения

Вы можете определить, как коммуникационный интерфейс распознает начало сообщения. Стартовые символы и символы, составляющие сообщение, поступают в приемный буфер, пока не встречается сконфигурированное условие окончания.

Вы можете определить несколько условий начала. Если Вы определили более одного условия начала, то все условия должны быть соблюдены, прежде чем сообщение считают начавшимся. Например, если Вы сконфигурируете время свободной линии и определенный начальный символ, то СМ или СВ будут сначала ожидать соблюдения требования свободной линии, а затем СМ будет искать указанный начальный символ. Если будет принят какой-либо другой символ (не являющийся начальным символом), то СМ или СВ перезапустят поиск начала сообщения, ожидая свободную линию.

Параметр	Описание
Start on Any Character	Условие Any Character определяет, что любой успешно принятый символ указывает на начало сообщения. Этот символ является первым символом сообщения.
Line Break	Условие Line Break определяет, что операция приема сообщения запускается после того, как получен символ разрыва линии.
Idle Line	Условие Idle Line определяет, что прием сообщения запускается, как только приемная линия была незанятой или тихой в течение указанного количества битовых интервалов. Как только это условие наступает, запускается начало приема сообщения.  ① Символы ② Перезапуск таймера свободной линии ③ Свободная линия обнаружена, и запущен прием сообщения
Special condition: Recognize message start with single character	Определяет, что определенный символ указывает на начало сообщения. Этот символ в этом случае является первым символом в сообщении. Любой символ, который получен перед этим определенным символом отбрасывается. Символом по умолчанию является STX.
Special condition: Recognize message start with a character sequence	Указывает на то, что определенная символьная последовательность вплоть до четырех сконфигурированных последовательностей указывает на начало сообщения. Для каждой последовательности Вы можете определить до пяти символов. Для каждой символьной позиции Вы либо определяете конкретный шестнадцатеричный символ, либо символ игнорируется в соответствующей последовательности (подстановочный символ). Последний определенный символ последовательности завершает эту последовательность условия начала сообщения. Входящие последовательности оцениваются относительно сконфигурированных условий запуска, пока условие начала не будет соблюдено. Как только последовательность начала была соблюдена, иницируется оценка условия окончания. Вы можете сконфигурировать до четырех определенных символьных последовательностей. Вы используете несколько последовательностей условия начала, когда различные последовательности символов могут указать на начало сообщения. Если какая-либо из символьных последовательностей соответствует, сообщение начато.

Порядок проверки условий начала следующий:

- свободная линия;
- разрыв линии;
- символы или последовательности символов.

При проверке на несколько условий начала, если одно из условий не соблюдено, СМ или СВ перезапустят проверку, начиная с первого требуемого условия. После того, как СМ или СВ устанавливает, что условия начала соблюдены, начинается оценка условия окончания.

Пример конфигурации: Начинать сообщение по одной из двух последовательностей символов

Рассмотрите следующую конфигурацию условия начала сообщения:

The screenshot displays the configuration interface for message start sequences. At the top, there is a checkbox labeled "Recognize message start with a character sequence" which is checked. Below it, a text field "Number of character sequences to define:" contains the value "2".

The main section is titled "5-character message start sequences". Under this, "Message start sequence 1" is configured. It features five rows, each for a character in the sequence:

- Character 1:** "Inspect character 1" is checked. The "Character value (HEX)" is "6A" and "Character value (ASCII)" is empty.
- Character 2:** "Inspect character 2" is unchecked. The "Character value (HEX)" is "0" and "Character value (ASCII)" is "any".
- Character 3:** "Inspect character 3" is unchecked. The "Character value (HEX)" is "0" and "Character value (ASCII)" is "any".
- Character 4:** "Inspect character 4" is unchecked. The "Character value (HEX)" is "0" and "Character value (ASCII)" is "any".
- Character 5:** "Inspect character 5" is checked. The "Character value (HEX)" is "1C" and "Character value (ASCII)" is "FS".

Message start sequence 2

☐ Inspect character 1

Character value (HEX): 0

Character value (ASCII): any

☒ Inspect character 2

Character value (HEX): 6A

Character value (ASCII):

☒ Inspect character 3

Character value (HEX): 6A

Character value (ASCII):

☐ Inspect character 4

Character value (HEX): 0

Character value (ASCII): any

☐ Inspect character 5

Character value (HEX): 0

Character value (ASCII): any

С этой конфигурацией условие начала удовлетворено, когда принят любой из образцов:

- Когда принята последовательность из пяти символов, в которой первый символ - 0x6A, а пятый символ - 0x1C. Символы в позициях 2, 3 и 4 могут быть любым символом с данной конфигурацией. После того, как получен пятый символ, начинается оценка условий окончания.
- Когда приняты два последовательных символа 0x6A, предшествовавшие любым символам. В этом случае оценка условий окончания начинается после того, как принят второй 0x6A (3 символа). Символ, предшествующий первому 0x6A, включен в условие начала.

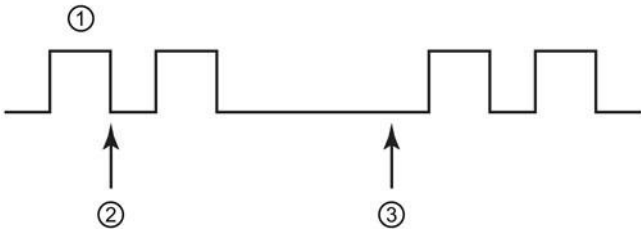
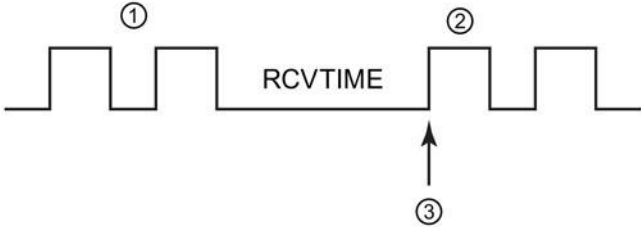
Примеры последовательностей, которые удовлетворили бы этому условию начала:

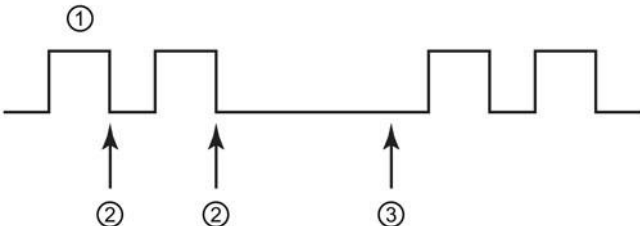
- <любой символ> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

Условия завершения сообщения

Вы также конфигурируете то, как коммуникационный интерфейс распознает завершение сообщения. Вы можете сконфигурировать несколько условий завершения сообщения. Если происходит какое-либо из сконфигурированных условий, то сообщение завершается.

Например, Вы могли бы определить условие завершения с истечением тайм-аута сообщения в 300 мс, межсимвольного тайм-аута в 40 битовых интервалов и максимальной длины в 50 байтов. Сообщение закончится, если прием сообщения займет больше чем 300 мс или если разрыв между какими-либо двумя символами превышает 40 битовых интервалов или если получены 50 байтов.

Параметр	Описание
Recognize message end by message timeout	Завершение сообщения имеет место, когда истекло сконфигурированное время ожидания завершения сообщения. Отсчет тайм-аута сообщения начинается, когда было удовлетворено условие начала сообщения. Значение по умолчанию равно 200 мс, а диапазон составляет от 0 до 65535 мс.  ① Принятые символы ② Условие начала сообщения соблюдено: таймер сообщения запускается ③ Таймер сообщения истек и прерывает сообщение
Recognize message end by response timeout	Завершение сообщения имеет место, когда истекает сконфигурированное время ожидания ответа, прежде чем принята допустимая последовательность начала. Отсчет тайм-аута ответа начинается, когда передача завершается, и СМ или СВ начинают прием. Тайм-аут ответа по умолчанию равен 200 мс, а диапазон составляет от 0 до 65535 мс. Если символ не получен в течение периода времени отклика, RCVTIME, то ошибка возвращается соответствующей инструкции RCV_PTP. Тайм-аут ответа не определяет специфическое условие завершения. Он определяет только то, что символ должен быть успешно принят в течение требуемого времени. Вы должны сконфигурировать другое условие завершения, чтобы указать на фактический конец сообщения.  ① Переданные символы ② Принятые символы ③ Первый символ должен быть успешно принят за это время.

Параметр	Описание
Recognize message end by inter-character gap	Завершение сообщения имеет место, когда истек максимальный сконфигурированный тайм-аут между любыми двумя последовательными символами сообщения. Значение по умолчанию для межсимвольного разрыва составляет 12, а максимальное количество - 65535 битовых интервалов, до максимума равного 8 секундам.  ① Принятые символы ② Перезапуск межсимвольного таймера ③ Межсимвольный таймер истек и прерывает сообщение
Recognize message end by receiving a fixed number of characters	Завершение сообщения имеет место, когда было получено определенное количество символов. Допустимый диапазон для фиксированной длины составляет от 1 до 4096. Обратите внимание на то, что для S7-1200, это условие завершения допустимо только для ЦПУ V4.0 или выше.
Recognize message end by max length	Завершение сообщения имеет место, когда было получено сконфигурированное максимальное количество символов. Допустимый диапазон для максимальной длины составляет от 1 до 1023. Это условие может использоваться, чтобы предотвратить ошибку переполнения буфера сообщения. Когда это условие завершения объединено с условиями тайм-аута завершения, и условие тайм-аута возникает, любые допустимые полученные символы используются, пока не достигнута максимальная длина. Это позволяет поддержку протоколов переменной длины, когда известна только максимальная длина.
Read message length from message	Сообщение само определяет длину. Завершение сообщения имеет место, когда было получено сообщение указанной длины. Метод для определения и интерпретации длины сообщения описан ниже.
Recognize message end with a character	Завершение сообщения имеет место, когда принят указанный символ.
Recognize message end with a character sequence	Завершение сообщения имеет место, когда принята указанная символьная последовательность. Вы можете определить последовательность до пяти символов. Для каждой символьной позиции Вы определяете либо определенный шестнадцатеричный символ, либо символ игнорируется в соответствующей последовательности. Начальные проигнорированные символы не являются частью условия завершения. Замыкающие проигнорированные символы являются частью условия завершения.

Пример конфигурации: Завершать сообщение по последовательности символов

Рассмотрите следующую конфигурацию условия завершения сообщения:

Recognize message end with a character sequence

5-character message end sequence

☐ Check character 1

Character value (HEX):

Character value (ASCII): ANY

☒ Check character 2

Character value (HEX): 6A

Character value (ASCII):

☒ Check character 3

Character value (HEX): 6A

Character value (ASCII):

☐ Check character 4

Character value (HEX):

Character value (ASCII): ANY

☐ Check character 5

Character value (HEX):

Character value (ASCII): ANY

В этом случае условие завершения удовлетворено, когда приняты два последовательных символа 0x6A, сопровождаемые любыми двумя символами. Символ, предшествующий образцу 0x6A 0x6A, не является частью символьной последовательности завершения. Два символа после образца 0x6A 0x6A требуются, чтобы дополнить символьную последовательность завершения. Значения, принятые в символьных позициях 4 и 5, не важны, но они должны быть приняты, чтобы удовлетворить условию завершения.

Примечание

Если Вы хотите, чтобы Ваша символьная последовательность указала на завершение сообщения, поместите последовательность в последние знаковые позиции. В примере выше, если бы Вы захотели, чтобы 0x6A 0x6A завершали сообщение без замыкающих символов, Вам следовало бы сконфигурировать 0x6A в символьных позициях 4 и 5.

Спецификация длины сообщения в сообщении

При выборе специального условия, когда длина сообщения включена в сообщение, Вы должны обеспечить три параметра, которые определяют информацию о длине сообщения.

Фактическая структура сообщения варьируется согласно используемому протоколу. Это следующие три параметра:

- n: символьная позиция (по отношению к 1) в сообщении, с которой начинается спецификатор длины;
- Length size: количество байтов (один, два, или четыре) спецификатора длины;
- Length m: количество символов после спецификатора длины, которые не включены в расчет длины.

Конечные символы не должны быть смежными. Значение "Length m" может использоваться, чтобы определить длину поля контрольной суммы, размер которого не включен в поле длины.

Эти поля появляются в конфигурации приема сообщения в свойствах устройства:

Пример 1: Рассмотрите сообщение, структурированное согласно следующему протоколу:

STX	Len (n)	Символы с 3 по 14, подсчитываемые посредством длины											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Сконфигурируйте параметры длины принятого сообщения следующим образом:

- n = 2 (Длина сообщения начинается с байта 2.)
- Length size = 1 (Длина сообщения определена в одном байте.)
- Length m = 0 (Нет никаких дополнительных символов после спецификатора длины, которые не включены в значение длины. Двенадцать символов следуют за спецификатором длины.)

В этом примере символы с 3 по 14 включительно являются символами подсчитываемыми с помощью Len (n).

Пример 2: Рассмотрите другое сообщение, структурированное согласно следующему протоколу:

SD1	Len (n)	Len (n)	SD2	Символы с 5 по 10, подсчитываемые посредством длины						FCS	ED
				DA	SA	FA	Элемент данных=3 байта				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Сконфигурируйте параметры длины принятого сообщения следующим образом:

- $n = 3$ (Длина сообщения начинается с байта 3.)
- Length size = 1 (Длина сообщения определена в одном байте.)
- Length m = 3 (Есть три символа после спецификатора длины, которые не включены в значение длины. В протоколе этого примера символы SD2, FCS и ED не включены в значение длины. Другие шесть символов учтены в значении длины; поэтому общее количество символов после спецификатора длины равняется девяти.)

В этом примере символы с 5 по 10 включительно являются символами подсчитываемыми с помощью Len (n).

13.3.3. Инструкции точка-к-точке

13.3.3.1. Общие параметры для инструкций точка-к-точке

Таблица 13- 3 Общие входные параметры для PTP-инструкций

Параметр	Описание
REQ	Многие из PtP-инструкций используют вход REQ, чтобы инициировать работу по положительному фронту. На входе REQ должен быть высокий уровень (ИСТИНА) для однократного выполнения инструкции, но на входе REQ может оставаться сигнал ИСТИНА столько, сколько необходимо. Инструкция не иницирует новое выполнение, пока она не будет вызвана с сигналом ЛОЖЬ на входе REQ так, чтобы инструкция могла сбросить состояние истории входа REQ. Это требуется, чтобы инструкция могла обнаружить положительный фронт, чтобы инициировать следующее выполнение. Когда Вы помещаете PtP-инструкцию в свою программу, STEP 7 предлагает Вам идентифицировать экземплярный DB. Используйте уникальный DB для каждого вызова PtP-инструкции. Это гарантирует, что каждая инструкция должным образом обрабатывает входы, такие как REQ.
PORT	Адрес порта присваивается во время конфигурации коммуникационного устройства. После конфигурации символьное имя порта по умолчанию может быть выбрано из выпадающего списка ассистента параметра. Присвоенное значение порта CM или CB является аппаратным свойством "hardware identifier" в конфигурации устройства. Символьное имя порта присваивается на вкладке "Constants" таблицы тегов ПЛК.
Bit time resolution	Несколько параметров определяются за множество битовых интервалов на сконфигурированной скорости передачи. Определение параметра в битовых интервалах позволяет параметру быть независимым от скорости передачи. Все параметры, которые определены в битовых интервалах, могут иметь максимальное значение 65535. Однако максимальное количество времени, которое могут измерить CM или CB, составляет восемь секунд.

13.3 Коммуникации точка-к-точке (PtP)

Выходные параметры PtP-инструкций DONE, NDR, ERROR и STATUS обеспечивают статус завершения выполнения для PtP-операций.

Таблица 13- 4 Выходные параметры DONE, NDR, ERROR и STATUS

Параметр	Тип данных	По умолчанию	Описание
DONE	Bool	FALSE	Устанавливается в TRUE на один скан, чтобы указать на то, что последний запрос завершен без ошибок; в противном случае, FALSE.
NDR	Bool	FALSE	Устанавливается в TRUE на один скан, чтобы указать на то, что требуемое действие завершилось без ошибок, и что новые данные были получены; в противном случае, FALSE.
ERROR	Bool	FALSE	Устанавливается в TRUE на один скан, чтобы указать на то, что последний запрос завершен с ошибками с соответствующим кодом ошибки на выходе STATUS; в противном случае, FALSE.
STATUS	Word	0	Статус результата: - Если биты DONE или NDR установлены, то выходу STATUS присвоен 0 или информационный код. - Если установлен бит ERROR, то на выходе STATUS содержится код ошибки. - Если ни один из вышеупомянутых битов не установлен, то инструкция возвращает статус результата, который описывает текущее состояние функции. STATUS сохраняет свое значение на время выполнения функции.

Примечание

Параметры DONE, NDR и ERROR устанавливаются только на один скан. Ваша программная логика должна сохранить временные значения состояния выхода в ячейках данных, таким образом, Вы можете обнаружить изменения состояния на последующих сканах программы.

Таблица 13- 5 Общие коды состояний

STATUS(W#16#....)	Описание
0000	Без ошибок
7000	Функция не занята
7001	Функция занята первым вызовом.
7002	Функция занята последующими вызовами (опросы после первого вызова).
8x3A	Недопустимый указатель в параметре x
8070	Вся внутренняя экземплярная память используется, слишком многих параллельных обрабатываемых инструкций
8080	Номер порта недопустим.
8081	Тайм-аут, ошибка модуля или другая внутренняя ошибка
8082	Параметризация перестала работать, потому что параметризация происходит в фоновом режиме.
8083	Переполнение буфера: СМ или СВ возвратили полученное сообщение с длиной, больше, чем допустимый параметр длины.

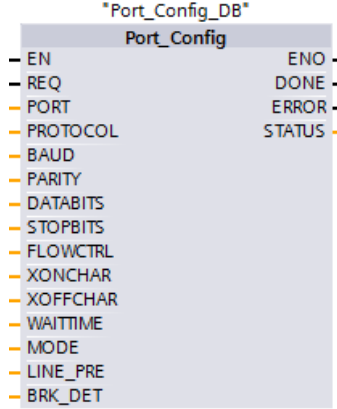
STATUS (W#16#....)	Описание
8090	Внутренняя ошибка: Неверная длина сообщения, неправильный субмодуль или недопустимое сообщение. Обратитесь в службу поддержки клиентов.
8091	Внутренняя ошибка: Неправильная версия в сообщении параметризации. Обратитесь в службу поддержки клиентов.
8092	Внутренняя ошибка: Неверная длина записи в сообщении параметризации. Обратитесь в службу поддержки клиентов.

Таблица 13- 6 Общие классы ошибок

Описание класса	Классы ошибок	Описание
Конфигурация порта	16#81Ax	Используется для определения общих ошибок конфигурации порта
Transmit configuration	16#81Bx	Используется для определения общих ошибок конфигурации передачи
Receive configuration	16#81Cx 16#82Cx	Используется для определения общих ошибок конфигурации приема
Процесс передачи	16#81Dx	Используется для определения общих ошибок процесса передачи
Процесс приема	16#81Ex	Используется для определения общих ошибок процесса приема
Обработка сигнала	16#81Fx	Используется для определения общих ошибок, связанных с обработкой сигнала
Ошибки указателя	от 16#8p01 до 16#8p51	Используется для ошибок указателя ANY, где "p" - номер параметра инструкции
Ошибки встроенного протокола	16#848x 16#858x	Используется для ошибок встроенного протокола

13.3.3.2. Инструкция Port_Config (динамическое конфигурирование коммуникационных параметров)

Таблица 13- 7 Инструкция Port_Config (конфигурирование порта)

LAD / FBD	SCL	Описание
	<pre> "Port_Config_DB"(REQ:=_bool_in_, PORT:=_word_in_, PROTOCOL:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, DATABITS:=_uint_in_, STOPBITS:=_uint_in_, FLOWCTRL:=_uint_in_, XONCHAR:=_char_in_, XOFFCHAR:=_char_in_, WAITTIME:=_uint_in_, MODE:=_uint_in_, LINE_PRE:=_uint_in_, BRK_DET:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	Port_Config позволяет Вам изменять параметры порта, такие как скорость обмена из Вашей программы. Вы можете настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. Вы можете обрабатывать инструкцию Port_Config в Вашей программе, чтобы изменить конфигурацию..

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

13.3 Коммуникации точка-к-точке (PtP)

ЦПУ не хранит постоянно значения, которые Вы устанавливаете с помощью инструкции Port_Config. ЦПУ восстанавливает параметры, заданные в конфигурации устройства, когда ЦПУ переходит из RUN в STOP, а также после цикла включения и выключения питания. Смотрите разделы Конфигурирование коммуникационных портов (стр. 865) и Организация управления потоком (стр. 867) для получения дополнительной информации.

Таблица 13- 8 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
PROTOCOL	IN	UInt	0 –Коммуникационный протокол Point-to-Point (Значение по умолчанию) 1..n - будущее определение для специфических протоколов
BAUD	IN	UInt	Скорость обмена порта (Значение по умолчанию: 6): 1 = 300 бод, 2 = 600 бод, 3 = 1200 бод, 4 = 2400 бод, 5 = 4800 бод, 6 = 9600 бод, 7 = 19200 бод, 8 = 38400 бод, 9 = 57600 бод, 10 = 76800 бод, 11 = 115200 бод
PARITY	IN	UInt	Контроль четности порта (Значение по умолчанию: 1): 1 = Без контроля, 2 = На четность, 3 = На нечетность, 4 = По единичному биту, 5 = По нулевому биту
DATABITS	IN	UInt	Битов на символ (Значение по умолчанию: 1): 1 = 8 битов данных, 2 = 7 битов данных.
STOPBITS	IN	UInt	Стоповые биты (Значение по умолчанию: 1): 1 = 1 стоповый бит, 2 = 2 стоповых бита
FLOWCTRL	IN	UInt	Управление потоком (Значение по умолчанию: 1): 1 = Без управления, 2 = XON/XOFF, 3 = Аппаратное: RTS всегда ON, 4 = Аппаратное: с переключением RTS
XONCHAR	IN	Char	Определяет символ, который используется в качестве символа XON. Это обычно символ DC1 (16#11). Этот параметр оценивается только, если включено управление потоком. (Значение по умолчанию: 16#11)
XOFFCHAR	IN	Char	Определяет символ, который используется в качестве символа XOFF. Это обычно символ DC3 (16#13). Этот параметр оценивается только, если управление потоком включено. (Значение по умолчанию: 16#13)
WAITTIME	IN	UInt	Определяет время ожидания символа XON после получения символа XOFF, или время ожидания сигнала CTS после активации RTS (от 0 до 65535 мс). Этот параметр оценивается только, если управление потоком включено. (Значение по умолчанию: 2000)
MODE	IN	UInt	Определяет выбор рабочего режима модуля. - Режим RS232 (по умолчанию для RS232 CM или CB) - RS422 Point-to-Point, передатчик всегда активен - RS422 Multi-Point Master, передатчик всегда активен - RS422 Multi-Point Slave, передатчик активен при отправке - Режим RS485 (полудуплекс, 2-проводное подключение) (по умолчанию для RS422/RS485 CM или CB)

Параметр и тип		Тип данных	Описание
LINE_PRE	IN	UInt	Определяет неактивное (свободное) состояние линии. Для RS422 и RS485 модулей состояние свободной линии устанавливается, приложением напряжение смещения к R (A) и R (B) сигналам. Возможны следующие варианты: - Без смещения (без предварительных установок) (по умолчанию) - Со смещением $R(A) > R(B) \geq 0V$; только RS422 - Со смещением $R(B) > R(A) \geq 0V$; RS422 и RS485
BRK_DET	IN	UInt	Разрешает/запрещает обнаружение обрыва коммуникационного кабеля. Разрешение обнаружения разрыва кабеля заставляет модуль указывать на отказ, когда коммуникационный кабель не соединен с модулем. В режиме RS422 "точка-к-точке" обнаружение обрыва кабеля возможно только когда Используется предварительная установка приемной линии, со смещением, приложенным так, чтобы $R(A) > R(B) \geq 0V$. - Без обнаружения обрыва кабеля (по умолчанию) - Обнаружение обрыва кабеля разрешено
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Таблица 13-9 Коды состояния

STATUS (W#16#....)	Описание
81A0	Специфический протокол не существует.
81A1	Специфическая скорость обмена не существует.
81A2	Специфическая опция четности не существует.
81A3	Специфическое число битов данных не существует.
81A4	Специфическое число стоповых битов не существует.
80A5	Специфический тип управления потоком не существует.
81A6	Время ожидания - 0, и управление потоком разрешено
81A7	XON и XOFF - недопустимые значения (например, одно то же значение)
81A8	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
81A9	Реконфигурирование отклонено, так как выполняется конфигурирование
81AA	Недопустимый режим работы RS422/RS485
81AB	Недопустимая предварительная установка приемной линии для обнаружения обрыва
81AC	Недопустимая RS232 обработка обрыва
8280	Уведомление об отказе при чтении модуля
8281	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны

Таблица 13- 10 Инструкция Send Config (конфигурация передачи)

LAD / FBD	SCL	Описание
*Send_Config_DB" Send_Config <pre> - EN ENO - - REQ DONE - - PORT ERROR - - RTSONLY STATUS - - RTSOFFDLY - BREAK - IDLELINE - USR_END - APP_END </pre>	<pre> "Send_Config_DB"(REQ:=_bool_in_, PORT:=_word_in_, RTSONLY:=_uint_in_, RTSOFFDLY:=_uint_in_, BREAK:=_uint_in_, IDLELINE:=_uint_in_, USR_END:=_string_in_, APP_END:=_string_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STATUS=>_word_out_); </pre>	Send_Config позволяет динамическую конфигурацию параметров последовательного передатчика для PtP коммуникационного порта. Любые сообщения в очереди в СМ или СВ сбрасываются, когда выполняется Send_Config .

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Вы можете установить начальную статическую конфигурацию порта в свойствах конфигурации устройства, или просто использовать значения по умолчанию. Вы можете выполнить инструкцию `Send Config` в своей программе, чтобы изменить конфигурацию.

ЦПУ постоянно не хранит значения, которые Вы устанавливаете инструкцией `Send_Config`. ЦПУ восстанавливает параметры, заданные в конфигурации устройства, когда ЦПУ переходит из RUN в STOP, а также после цикла включения и выключения питания. Смотрите раздел **Конфигурирование параметров передачи (отправки)** (стр. 868).

Таблица 13- 11 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
RTSONDLY	IN	UInt	Число миллисекунд для ожидания после активации RTS прежде чем произойдет какая-либо передача Tx данных. Этот параметр допустим только, если разрешено аппаратное управление потоком. Допустимый диапазон 0 - 65535 мс. Значение 0 отключает опцию. (Значение по умолчанию: 0).
RTSOFFDLY	IN	UInt	Число миллисекунд для ожидания после того, как произойдет какая-либо передача Tx данных, прежде чем RTS будет запрещен. Этот параметр допустим только, если разрешено аппаратное управление потоком. Допустимый диапазон 0 - 65535 мс. Значение 0 отключает опцию. (Значение по умолчанию: 0).
BREAK	IN	UInt	Этот параметр определяет, что разрыв будет отправлен при запуске каждого сообщения для конкретного количества битовых интервалов. Максимум составляет 65535 битовых интервалов, то есть 8 секунд максимум. Значение 0 отключает опцию. (Значение по умолчанию: 12)

Параметр и тип		Тип данных	Описание
IDLELINE	IN	UInt	Этот параметр определяет, что линия останется свободной в течение указанного числа битовых интервалов перед запуском каждого сообщения. Максимум составляет 65535 битовых интервалов, то есть 8 секунд максимум. Значение 0 отключает опцию. (Значение по умолчанию: 0)
USR_END*	IN	STRING[2]	Определяет число и символы в конечном ограничителе. Конечный ограничитель включен в буфер передачи (только символы) и отмечает конец переданного сообщения (символы передаются, пока не распознается конечный ограничитель). Конечный ограничитель добавляется в конце сообщения. - STRING[2,0,xx,yy] –ограничитель не используется (по умолчанию) - STRING[2,1,xx,yy] – ограничитель является одиночным символом - STRING[2,2,xx,yy] – ограничитель состоит из двух символов USR_END или APP_END должны иметь нулевую длину.
APP_END*	IN	STRING[5]	Определяет число и символы, которые будут добавлены к переданному сообщению (только добавляемые символы). STRING[5,0, aa, bb, cc, dd, ee] – конечный символ не используется (значение по умолчанию) - STRING[5,1,aa,bb,cc,dd,ee] – передать один конечный символ - STRING[5,2,aa,bb,cc,dd,ee] – передать два конечных символов - STRING[5,3,aa,bb,cc,dd,ee]– передать три конечных символов - STRING[5,4,aa,bb,cc,dd,ee] – передать четыре конечных символов - STRING[5,5,aa,bb,cc,dd,ee] – передать пять конечных символов
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

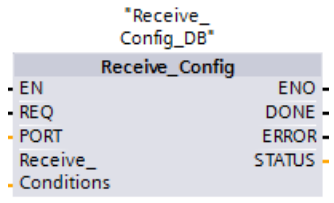
* Не поддерживается для CM и CB 1241; Вы должны использовать пустую строку ("") для параметра.

Таблица 13- 12 Коды состояния

STATUS (W#16#....)	Описание
81B0	Конфигурация прерывания передачи недопустима. Обратитесь в службу поддержки.
81B1	Разрыв больше, чем максимально допустимое значение.
81B2	Время простоя больше, чем максимальное допустимое значение.
81B3	Ошибка в заголовке блока, например, неправильный тип блока или неправильная длина блока
81B4	Реконфигурирование отклонено, так как выполняется конфигурирование
81B5	Число определенных конечных ограничителей больше, чем два, или число конечных символов больше, чем пять
81B6	Передача конфигурации отклонена, когда сконфигурированы включенные во встроенное ПО протоколы
8280	Уведомление об отказе при чтении модуля
8281	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.3.4. Инструкция Receive_Config (динамическое конфигурирование параметров последовательного приема)

Таблица 13- 13 Инструкция Receive_Config (конфигурация приема)

LAD / FBD	SCL	Описание
	<pre>"Receive_Config_DB"(REQ:=_bool_in_, PORT:=_uint_in_, Re- ceive_Conditions:=_struct_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_);</pre>	Receive_Config выполняет динамическую конфигурацию параметров последовательного приемника для PtP коммуникационного порта. Эта инструкция конфигурирует условия, которые сигнализируют о начале и конце принятого сообщения. Любые сообщения в очереди в СМ или СВ сбрасываются, когда выполняется Receive_Config.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Вы можете установить начальную статическую конфигурацию порта в свойствах конфигурации устройства, или просто использовать значения по умолчанию. Вы можете выполнить инструкцию Receive_Config в своей программе, чтобы изменить конфигурацию.

ЦПУ постоянно не хранит значения, которые Вы устанавливаете инструкцией Send_Config. ЦПУ восстанавливает параметры, заданные в конфигурации устройства, когда ЦПУ переходит из RUN в STOP, а также после цикла включения и выключения питания. Смотрите раздел Конфигурирование параметров приема (стр. 869) для получения более подробной информации.

Таблица 13- 14 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
CONDITIONS	IN	CONDITIONS	Структура данных Conditions определяет условия начала и окончания сообщения, как описано ниже.
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Условия начала для инструкции Receive_P2P

Инструкция Receive_P2P использует конфигурацию, определенную инструкцией Receive_Config, чтобы определить начало и окончание коммуникационных сообщений "точка-к-точке". Начало сообщения определяется условиями начала. Начало сообщения может быть определено одним или комбинацией условий. Если определено больше чем одно условие начала, то все условия должны быть удовлетворены, прежде чем сообщение будет начато.

Смотрите раздел Конфигурирование параметров приема (стр. 869) для получения описания условий начала сообщения.

Структура данных параметра CONDITIONS, Часть 1 (условия начала)

Таблица 13- 15 Структура CONDITIONS для условий начала

Параметр и тип		Тип данных	Описание
STARTCOND	IN	UInt	Определяет условие начала (Значение по умолчанию: 1) • 01H – Стартовый символ • 02H – Любой символ • 04H – Разрыв линии • 08H – Свободная линия • 10H - Последовательность 1 • 20H - Последовательность 2 • 40H - Последовательность 3 • 80H - Последовательность 4
IDLETIME	IN	UInt	Число битовых интервалов, требуемых для тайм-аута свободной линии. (Значение по умолчанию: 40). Используется только с условием свободной линии. От 0 до 65535
STARTCHAR	IN	Byte	Начальный символ, используемый с условием начального символа. (Значение по умолчанию: V#16#2)
STRSEQ1CTL	IN	Byte	Управление игнорированием/сравнением для каждого символа последовательности 1: (Значение по умолчанию: V#16#0) Это биты разрешения для каждого символа в последовательности начала • 01H - Символ 1 • 02H - Символ 2 • 04H - Символ 3 • 08H - Символ 4 • 10H - Символ 5 Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
STRSEQ1	IN	Char[5]	Стартовые символы последовательности 1 (5 символов). Значение по умолчанию: 0
STRSEQ2CTL	IN	Byte	Управление игнорированием/сравнением для каждого символа последовательности 2: (Значение по умолчанию: V#16#0)
STRSEQ2	IN	Char[5]	Стартовые символы последовательности 2 (5 символов). Значение по умолчанию: 0
STRSEQ3CTL	IN	Byte	Управление игнорированием/сравнением для каждого символа последовательности 3: (Значение по умолчанию: V#16#0)
STRSEQ3	IN	Char[5]	Стартовые символы последовательности 3 (5 символов). Значение по умолчанию: 0

Параметр и тип		Тип данных	Описание
STRSEQ4CTL	IN	Byte	Управление игнорированием/сравнением для каждого символа последовательности 4: (Значение по умолчанию: В#16#0)
STRSEQ4	IN	Char[5]	Стартовые символы последовательности 4 (5 символов). Значение по умолчанию: 0

Пример

Рассмотрите следующее принятое шестнадцатеричное кодированное сообщение: "68 10 aa 68 bb 10 aa 16" и сконфигурированные последовательности начала, показанные в таблице ниже. Последовательности начинают оцениваться, когда первый 68H символ успешно получен. После успешного получения четвертого символа (второе 68H) условие 1 удовлетворено. Как только условия начала удовлетворены, начинается оценка условий окончания.

Обработка последовательности начала может быть завершена из-за различных ошибок четности, синхронизации кадров или символов. Эти ошибки исключают какое-либо принятое сообщение, так как условие начала не было удовлетворено.

Таблица 13- 16 Условия начала

Условия начала	Первый символ	Первый символ +1	Первый символ +2	Первый символ +3	Первый символ +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Условия завершения для инструкции Receive_P2P

Конец сообщения определяется спецификацией условий завершения. Конец сообщения определяется первым появлением одного или нескольких сконфигурированных условий завершения. Раздел "Условия завершения сообщения" в теме "Конфигурирование параметров приема (стр. 869)", описывает условия завершения, которые Вы можете сконфигурировать в инструкции Receive_Config.

Вы можете сконфигурировать условия завершения в свойствах коммуникационного интерфейса в конфигурации устройства или с помощью инструкции Receive_Config. Каждый раз, когда ЦПУ переходит из STOP в RUN, параметры приема (и условия начала и завершения) возвращаются к параметрам конфигурации устройства. Если пользовательская STEP 7 программа обрабатывает Receive_Config, то настройки изменяются на условия, заданные в Receive_Config.

Структура данных параметра CONDITIONS, Часть 2 (условия завершения)

Таблица 13- 17 Структура CONDITIONS для условий завершения

Параметр	Тип параметра	Тип данных	Описание
ENDCOND	IN	UInt 0	Этот параметр определяет условие завершения сообщения: • 01H – Время отклика • 02H – Время сообщения • 04H – Межсимвольный интервал • 08H – Максимальная длина • 10H - N + LEN + M • 20H - Последовательность
MAXLEN	IN	UInt 1	Максимальная длина сообщения: используется только, когда выбрано условие завершения по максимальной длине. От 1 до 1024 байтов.
N	IN	UInt 0	Позиция байта в сообщении поля длины. Используется только с условием завершения N + LEN + M. От 1 до 1022 байтов.
LENGTHSIZE	IN	UInt 0	Размер поля длины (1, 2 или 4 байта). Используется только с условием завершения N + LEN + M.
LENGTHM	IN	UInt 0	Определяет число символов после поля длины, которые не включены в значение поля длины. Используется только с условием завершения N + LEN + M. От 0 до 255 байтов.
RCVTIME	IN	UInt 200	Определяет время ожидания первого, подлежащего приему символа. Операция приема будет завершена с ошибкой, если символ не будет успешно получен за установленное время. Используется только с условием времени отклика. (От 0 до 65535 битовых интервалов, то есть до 8 секунд максимум). Этот параметр не является условием завершения сообщения, так как оценка прерывается, когда первый символ отклика получен. Это является условием завершения только в том смысле, что приводит к завершению работы приемника при отсутствии какого-либо ответа, когда ответ ожидается. Вы должны выбрать отдельное условие завершения.
MSGTIME	IN	UInt 200	Определяет время ожидания всего сообщения, которое будет полностью получено, с момента получения первого символа. Этот параметр используется только, когда выбрано условие тайм-аута сообщения. (От 0 до 65535 миллисекунд).
CHARGAP	IN	UInt 12	Определяет число битовых интервалов между символами. Если число битовых интервалов между символами превысит указанное значение, то условие завершения будет удовлетворено. Используется только с условием межсимвольного интервала. (От 0 до 65535 битовых интервалов, то есть до 8 секунд максимум).

13.3 Коммуникации точка-к-точке (PtP)

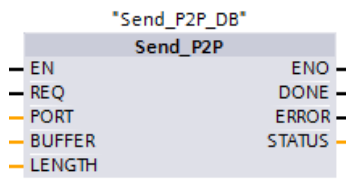
Параметр	Тип параметра	Тип данных	Описание
ENDSEQ1CTL	IN	Byte B#16#0	Управление игнорированием/сравнением для каждого символа: Это биты активации для каждого символа в последовательности завершения. Символ 1 является битом 0, символ 2 является битом 1, ..., символ 5 является битом 4. Деактивация бита, связанного с символом, означает, что любой символ будет соответствовать в этой позиции последовательности.
ENDSEQ1	IN	Char[5] 0	Начальные символы последовательности 1 (5 символов)

Таблица 13- 18 Коды состояния

STATUS (W#16#...)	Описание
81C0	Выбрано недопустимое условие начала
81C1	Выбрано недопустимое условие завершения, не выбрано условие завершения
81C2	Разрешено прерывание приема, и это недопустимо.
81C3	Разрешено условие завершения по максимальной длине, и максимальная длина равна 0 или > 1024.
81C4	Разрешена расчетная длина, и $N \geq 1023$.
81C5	Разрешена расчетная длина, и длина не равна 1, 2 или 4.
81C6	Разрешена расчетная длина, и значение $M > 255$.
81C7	Разрешена расчетная длина, и она > 1024.
81C8	Разрешен тайм-аут отклика, и он равен нулю.
81C9	Разрешен тайм-аут межсимвольного интервала, и он равен нулю.
81CA	Разрешен тайм-аут свободной линии, и он равен нулю.
81CB	Последовательность завершения разрешена, но все символы, "игнорируются".
81CC	Последовательность начала (любая из 4) разрешена, но все символы, "игнорируются".
81CD	Ошибка выбора защиты перезаписи неверно принятого сообщения
81CE	Ошибка выбора обработки буфера неверно принятого сообщения при переходе из STOP в RUN
81CF	Ошибка в заголовке блока, например, неправильный тип блока или неправильная длина блока
8281	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны
82C0	Реконфигурирование отклонено, так как выполняется конфигурирование
82C1	Заданное значение для числа сообщений, которые модуль может поместить в буфер, больше, чем максимальное разрешенное значение.
82C2	Передача конфигурации отклонена, когда сконфигурированы включенные во встроенное ПО протоколы
8351	Тип данных недопустим в этом указателе Variant

13.3.3.5. Инструкция Send_P2P (отправить данные из приемного буфера)

Таблица 13- 19 Инструкция Send_P2P (отправить данные точка-к-точке)

LAD / FBD	SCL	Описание
	<pre>"Send_P2P_DB"(REQ:=_bool_in_, PORT:=_word_in_, BUFFER:=_variant_in_, LENGTH:=_uint_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_);</pre>	Send_P2P инициирует передачу данных и передает назначенный буфер в коммуникационный интерфейс. Программа ЦПУ продолжается, в то время как СМ или СВ отправляют данные с заданной скоростью передачи. Только одна операция отправки может обрабатываться в данный момент времени. СМ или СВ возвращают ошибку, если вторая Send_P2P обрабатывается, в то время как СМ или СВ уже передают сообщение.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 20 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация требуемой передачи по нарастающему фронту на этом входе. Иницирует передачу содержимого буфера в коммуникационный интерфейс "точка-к-точке". (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальное положение буфера передачи. (Значение по умолчанию: 0) Примечание: Булевы данные или булевы массивы не поддерживаются.
LENGTH	IN	UInt	Длина переданного кадра в байтах (Значение по умолчанию: 0) При передаче сложной структуры всегда используйте длину 0. Когда длина 0, инструкция передает весь кадр.
DONE	OUT	Bool	TRUE на один скан после последнего запроса завершенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса завершенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Во время передачи, выходы DONE и ERROR находятся в состоянии FALSE. Когда передача будет завершена, или DONE или ERROR выходы будут установлены в TRUE, чтобы показать состояние передачи. Когда выходы DONE или ERROR находятся в TRUE, выход STATUS допустим.

Инструкция возвращает состояние 16#7001, если коммуникационный интерфейс принимает переданные данные. Последующее выполнение Send_P2P возвращает 16#7002, если СМ или СВ все еще заняты передачей. Когда передача завершена, СМ или СВ возвращают состояние процесса передачи как 16#0000 (при отсутствии ошибок). Последующее выполнение Send_P2P с REQ в состоянии низкого уровня возвращает состояние 16#7000 (незанято).

13.3 Коммуникации точка-к-точке (PtP)

Следующие диаграммы показывают отношение выходных значений к REQ. Это предполагает, что инструкция вызывается периодически, чтобы проверить состояние процесса передачи. На диаграмме ниже, предполагается, что инструкция вызывается на каждом скане (представлено значениями STATUS).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Следующая диаграмма показывает, что параметры DONE и STATUS допустимы только для одного скана, если вход REQ управляется импульсно (для одного скана), чтобы инициировать процесс передачи.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Следующая диаграмма показывает отношение параметров DONE, ERROR и STATUS, когда есть ошибка.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

Значения DONE, ERROR и STATUS допустимы только, пока Send_P2P выполняется снова с тем же экземплярным DB.

Таблица 13- 21 Коды состояния

STATUS (W#16#....)	Описание
81D0	Новый запрос, в то время как передатчик активен
81D1	Передача прервана из-за отсутствия CTS во время ожидания
81D2	Передача прервана из-за отсутствия DSR от DCE-устройства
81D3	Передача прервана из-за переполнения очереди (передача более 1024 байтов)
81D5	Сигнал обратного смещения (условие обрыва провода)
81D6	Запрос передачи отклонен, так как конечный ограничитель не был найден в буфере пе-
81D7	Внутренняя ошибка / ошибка в синхронизации между FB и CM
81D8	Попытка передачи отклонена, так как порт не был сконфигурирован
81DF	CM сбросил интерфейс к FB по одной из следующих причин: • Модуль перезапустился (цикл включения питания); • ЦПУ достиг контрольной точки и установил ODIS (вывод запрещен); • Модуль был повторно запаараметрирован. В каждом случае модуль указывает этот код в параметре Status. Модуль сбрасывает Status и Error в нуль после первой принятой записи для SEND_P2P.
8281	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны
8301	Недопустимый синтаксис ID в указателе ANY
8322	Ошибка длины диапазона при чтении параметра
8324	Ошибка диапазона при чтении параметра
8328	Ошибка выравнивания при чтении параметра
8332	Параметр содержит номер DB, который больше, чем наибольшее разрешенное число (ошибка номера DB).
833A	DB для параметра BUFFER не существует.

Взаимодействие параметров LENGTH и BUFFER

у, которые могут быть переданы инструкцией SEND_P2P, составляет один байт. Параметр BUFFER определяет размер данных, которые будут переданы. Вы не можете использовать тип данных Bool или массивы Bool для параметра BUFFER.

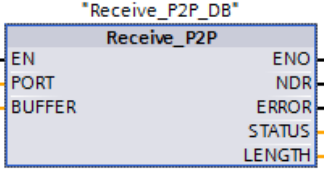
Вы можете всегда устанавливать параметр LENGTH на 0 и гарантировать, что SEND_P2P отправляет всю структуру данных, представленную параметром BUFFER. Если Вы хотите отправить только часть структуры данных из параметра BUFFER, Вы можете установить LENGTH следующим образом:

Таблица 13- 22 Параметры LENGTH и BUFFER

LENGTH	BUFFER	Описание
= 0	Не используется	Полные данные отправляются, как определено в параметре BUFFER. Вы не обязаны определять число передаваемых байтов когда LENGTH = 0.
> 0	Элементарный тип данных	Значение LENGTH должно содержать количество байтов этого типа данных. Например, для значения Word, LENGTH должно равняться двум. Для Dword или Real, LENGTH должна равняться четырем. В противном случае, ничего не передается, и возвращается ошибка 8088H.
	Структура	Значение LENGTH может содержать количество байтов меньше, чем полная длина структуры в байтах, в этом случае инструкция отправляет только первые n байтов структуры из BUFFER, где n = LENGTH. Так как внутренняя байтовая организация структуры не может быть всегда определенной, Вы могли бы получить неожиданные результаты. В этом случае используйте LENGTH равную 0, чтобы посылать полную структуру.
	Массив	Значение LENGTH должно содержать число байтов, которое меньше или равно полной байтовой длине массива и которое должно быть кратным числу байтов элемента данных. Например, параметр LENGTH для массива слов должен быть кратным двум, а для массива вещественных чисел - четырем. Когда LENGTH определен, инструкция передает число элементов массива, которые соответствуют значению LENGTH в байтах. Если Ваш BUFFER, например, содержит массив из 15 элементов Dword (всего 60 байтов), и Вы определяете LENGTH как 20, то передаются первые пять элементов массива. Значение LENGTH должно быть кратным числу байтов в элементе данных. В противном случае, STATUS = 8088H, ERROR = 1, и передача не выполняется.
	Строка	Параметр LENGTH содержит число символов, которые необходимо передать. Передаются только символы из данных String. Максимальная и фактическая длина String в байтах не передаются.

13.3.3.6. Инструкция Receive_P2P (разрешить прием сообщений)

Таблица 13- 23 Инструкция Receive_P2P (принять данные точка-к-точке)

LAD / FBD	SCL	Описание
	<pre>"Receive_P2P_DB"(PORT:=_word_in_, BUFFER:=_variant_in_, NDR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, LENGTH=>_uint_out_);</pre>	Receive_P2P выполняет проверку на сообщения, которые были получены в СМ или СВ. Если сообщение будет доступно, то оно будет передано из СМ или СВ в ЦПУ. Ошибка возвращает соответствующее значение в параметре STATUS.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 24 Типы данных для параметров

Параметр и тип		Тип данных	Описание
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
BUFFER	IN	Variant	Этот параметр указывает на начальное расположение приемного буфера. Этот буфер должен быть достаточно большим, чтобы получить сообщение максимальной длины. Булевы данные или булевы массивы не поддерживаются. (Значение по умолчанию: 0)
NDR	OUT	Bool	TRUE на один скан, когда новые данные готовы и операция завершена без ошибок.
ERROR	OUT	Bool	TRUE на один скан после последнего запроса завершенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)
LENGTH	OUT	UInt	Длина возвращенного сообщения в байтах (Значение по умолчанию: 0)

Значение STATUS допустимо, когда либо NDR, либо ERROR установлены в состояние TRUE. Значение STATUS описывает причину завершения операции приема в СМ или СВ. Обычно это положительное значение, указывающее на то, что операция приема была успешна и что процесс приема завершен обычным образом. Если STATUS имеет отрицательное значение (старший значащий бит шестнадцатеричного значения установлен), операция приема была завершена в состоянии ошибки, таких как ошибки четности, синхронизации кадров или переполнения.

Каждый PtP коммуникационный интерфейс может поместить в буфер максимум 1024 байта. Это могло быть одним большим сообщением или несколькими меньшими сообщениями. Если больше чем одно сообщение доступно в СМ или СВ, инструкция Receive_P2P возвращает самое старое доступное сообщение. Последующее выполнение инструкции Receive_P2P возвращает следующее самое старое доступное сообщение.

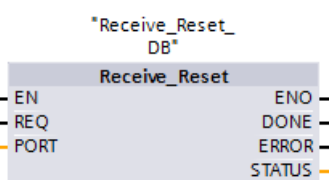
13.3 Коммуникации точка-к-точке (PtP)

Таблица 13- 25 Коды состояния

STATUS (W#16#...)	Описание
0000	Не существует буфера
0094	Сообщение прервано из-за максимальной длины в символах
0095	Сообщение прервано из-за тайм-аута сообщения
0096	Сообщение прервано из-за межсимвольного тайм-аута
0097	Сообщение прервано из-за тайм-аута ответа
0098	Сообщение прервано, так как было удовлетворено условие длины "N+LEN+M"
0099	Сообщение прервано, так как была распознана последовательность завершения
8085	Параметр LENGTH имеет значение 0 или больше, чем 4 КБ.
8088	Параметр LENGTH или принятая длина длиннее, чем область, определенная в BUFFER или принятая длина длиннее, чем область, определенная в BUFFER.
8090	Сообщение неверной конфигурации, неправильная длина сообщения, неправильный субмодуль, недопустимое сообщение
81E0	Сообщение прервано, так как приемный буфер полон
81E1	Сообщение прервано из-за ошибки четности
81E2	Сообщение прервано из-за ошибки кадровой синхронизации
81E3	Сообщение прервано из-за ошибки переполнения
81E4	Сообщение прервано, так как расчетная длина превышает размер буфера
81E5	Сигнал обратного смещения (условие обрыва провода)
81E6	Очередь сообщений полна. Об этой ошибке сообщают без данных. Если это происходит, модуль переключается между безошибочной передачей данных и этой ошибкой.
81E7	Внутренняя ошибка, ошибка в синхронизации между инструкцией и CM: устанавливается при обнаружении ошибки последовательности
81E8	Сообщение прервано, межсимвольный тайм-аут истек, прежде чем был удовлетворен критерий завершения сообщения
81E9	Обнаружена ошибка Modbus CRC (используется только модулями, которые поддерживают генерацию/проверку CRC для протокола Modbus)
81EA	Телеграмма Modbus слишком коротка (используется только модулями, которые поддерживают генерацию/проверку CRC для протокола Modbus)
81EB	Сообщение прервано, так как превышен максимальный размер сообщения
8201	Недопустимый синтаксис ID в указателе ANY
8223	Ошибка длины диапазона при записи параметра. Параметр расположен либо полностью, либо частично вне диапазона адреса или длина битового диапазона не кратна 8 в указателе ANY.
8225	Ошибка диапазона при записи параметра. Параметр расположен в диапазоне, который недопустим для системной функции.
8229	Ошибка выравнивания при записи параметра. Параметр, на который ссылаются, расположен в битовом адресе, который не равен 0.
8230	Параметр размещается в глобальном DB с защитой от записи (read only).
8231	Параметр размещается в экземплярном DB с защитой от записи (read only).
8232	Параметр содержит номер DB, который больше, чем наибольшее разрешенное число (ошибка номера DB).
823A	DB для параметра BUFFER не существует.
8280	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.3.7. Инструкция Receive_Reset (удалить приемный буфер)

Таблица 13- 26 Инструкция Receive_Reset (сбросить приемник)

LAD / FBD	SCL	Описание
	<pre>"Receive_Reset_DB"(REQ:=_bool_in_, PORT:=_word_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	Receive_Reset очищает приемный буфер в СМ или СВ.

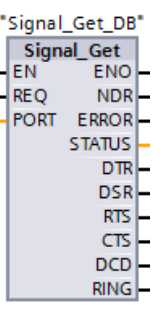
¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 27 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активирует сброс приемника по нарастающему фронту этого разрешающего входа (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
DONE	OUT	Bool	При установке в TRUE на один скан извещает о том, что последний запрос был завершен без ошибок.
ERROR	OUT	Bool	Установка в TRUE извещает о том, что последний запрос был завершен с ошибками. Таким образом, когда этот выход установлен в TRUE, выход STATUS будет содержать соответствующую информацию об ошибке.
STATUS	OUT	Word	Код ошибки (Значение по умолчанию: 0)

13.3.3.8. Инструкция Signal_Get (запросить RS-232 сигналы)

Таблица 13- 28 Инструкция Signal_Get (получить RS232 сигналы)

LAD / FBD	SCL	Описание
	<pre>"Signal_Get_DB"(REQ:=_bool_in_, PORT:=_uint_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DTR=>_bool_out_, DSR=>_bool_out_, RTS=>_bool_out_, CTS=>_bool_out_, DCD=>_bool_out_, RING=>_bool_out_);</pre>	Signal_Get читает текущие состояния RS232 коммуникационных сигналов. Эта функция допустима только для RS232 СМ.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

13.3 Коммуникации точка-к-точке (PtP)

Таблица 13- 29 Типы данных для параметров

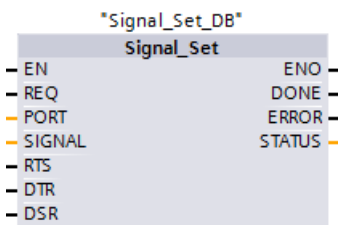
Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Получение состояний RS232 состояний по нарастающему фронту этого входа (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
NDR	OUT	Bool	TRUE на один скан, когда новые данные готовы, и операция завершена без ошибок
ERROR	OUT	Bool	TRUE на один скан после того, как операция была завершена с ошибками
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)
DTR	OUT	Bool	Терминал данных готов, модуль готов (выход). Значение по умолчанию: False
DSR	OUT	Bool	Источник данных готов, коммуникационный партнер готов (вход). Значение по умолчанию: False
RTS	OUT	Bool	Запрос на передачу, модуль, готов к передаче (выход). Значение по умолчанию: False
CTS	OUT	Bool	Разрешение на передачу, коммуникационный партнер может получить данные (вход). Значение по умолчанию: False
DCD	OUT	Bool	Обнаружен носитель данных, уровень приемного сигнала (всегда False, не поддерживается)
RING	OUT	Bool	Кольцевой индикатор, индикация входящего вызова (всегда False, не поддерживается)

Таблица 13- 30 Коды состояния

STATUS (W#16#...)	Описание
81F0	СМ или СВ в режиме RS485, и никакие сигналы не доступны
81F4	Ошибка в заголовке блока, например, неправильный тип блока или неправильная длина блока
8280	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.3.9. Инструкция Signal_Set (установить RS-232 сигналы)

Таблица 13- 31 Инструкция Signal_Set (установить RS232 сигналы)

LAD / FBD	SCL	Описание
	<pre>"Signal_Set_DB"(REQ:=_bool_in_, PORT:=_word_in_, SIGNAL:=_byte_in_, RTS:=_bool_in_, DTR:=_bool_in_, DSR:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	Signal_Set устанавливает состояния RS232 коммуникационных сигналов. Эта функция доступна только для RS232 CM.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

Таблица 13- 32 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Запускает операцию установки RS232 сигналов по нарастающему фронту на этом входе (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
SIGNAL	IN	Byte	Выбирает какой сигнал установить: (допустимо несколько). Значение по умолчанию: 0 • 01H = Установить RTS • 02H = Установить DTR • 04H = Установить DSR
RTS	IN	Bool	Запрос на передачу, модуль готов отправить значение для установки (true или false), Значение по умолчанию: False
DTR	IN	Bool	Терминал данных готов, модуль готов отправить значение для установки (true или false), Значение по умолчанию: False
DSR	IN	Bool	Сигнал готовности данных (применим только для интерфейсов DCE-типа), не используется.
DONE	OUT	Bool	TRUE на один скан после того, как последний запрос был завершен без ошибок
ERROR	OUT	Bool	TRUE на один скан после того, как последний запрос был завершен с ошибками
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)

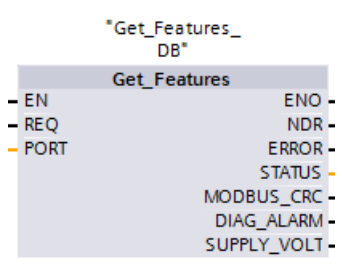
13.3 Коммуникации точка-к-точке (PtP)

Таблица 13- 33 Коды состояния

STATUS (W#16#...)	Описание
81F0	СМ или СВ в режиме RS485, и никакие сигналы не могут быть установлены
81F1	Сигналы не могут быть установлены из-за аппаратного управления потоком
81F2	Нельзя установить DSR, так как модуль - DTE
81F3	Нельзя установить DTR, так как модуль - DCE
81F4	Ошибка в заголовке блока, например, неправильный тип блока или неправильная длина
8280	Уведомление об отказе при чтении модуля
8281	Уведомление об отказе при записи модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.3.10. Get_Features

Таблица 13- 34 Инструкция Get_Features (получить расширенные свойства)

LAD / FBD	SCL	Описание
	<pre>"Get_Features_DB"(REQ:=_bool_in_, PORT:=_word_in_, NDR:=_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MODBUS_CRC=>_bool_out_, DIAG_ALARM=>_bool_out_, SUPPLY_VOLT=>_bool_out_);</pre>	Get_Features выполняет чтение расширенных функций модуля.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

Используйте инструкцию Get_Features для чтения расширенных функций модуля.

Таблица 13- 35 Типы данных для параметров

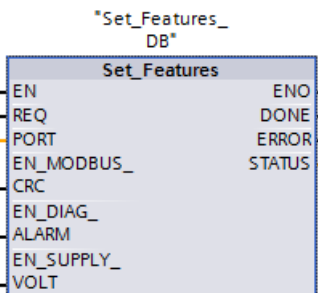
Параметр и тип	Тип данных	Описание
REQ	IN	Bool
PORT	IN	PORT
NDR	OUT	Bool
ERROR	OUT	Bool
STATUS	OUT	Word
MODBUS_CRC*	OUT	Bool

Параметр и тип		Тип данных	Описание
DIAG_ALARM*	OUT	Bool	Генерация диагностических тревог
SUPPLY_VOLT*	OUT	Bool	Диагностика для отсутствующего напряжения питания L + доступна

* Get_Features возвращает TRUE (1), если функция доступна, FALSE (0), если функция недоступна

13.3.3.11. Set_Features

Таблица 13- 36 Инструкция Set_Features (установить расширенные свойства)

LAD / FBD	SCL	Описание
	<pre>"Set_Features_DB"(REQ:=_bool_in_, PORT:=_word_in_, EN_MODBUS_CRC:=_bool_in_, EN_DIAG_ALARM:=_bool_in_, EN_SUPPLY_VOLT:=_bool_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STATUS=>_word_out_);</pre>	Set_Features разрешает расширенные функции, которые поддерживает модуль.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию

Используйте инструкцию Set_Features для установки расширенных функций модуля.

Таблица 13- 37 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК
EN_MODBUS_CRC	IN	Bool	Разрешение генерации и проверки MODBUS CRC: 0: вычисление CRC отключено (по умолчанию) 1: вычисление CRC включено Примечание: Только СМ V2.1, ЦПУ V4.1 с СВ и СМ модулями PtP для распределенного ввода-вывода поддерживают этот пара-
EN_DIAG_ALARM	IN	Bool	Разрешение генерации диагностических тревог: 0: Диагностическая тревога отключена 1: Диагностическая тревога включена (по умолчанию)

13.3 Коммуникации точка-к-точке (PtP)

Параметр и тип		Тип данных	Описание
EN_SUPPLY_VOLT	IN	Bool	Разрешение диагностики отсутствующего напряжения питания L+: 0: Диагностика напряжения питания запрещена (по умолчанию) 1: Диагностика напряжения питания разрешена
DONE	OUT	Bool	Указывает на то, что настройка функций выполнена
ERROR	OUT	Bool	TRUE на один скан после того, как последний запрос был завершен с ошибкой
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)

13.3.4. Программирование PtP-коммуникаций

STEP 7 обеспечивает расширенные инструкции, которые позволяют пользовательской программе выполнять коммуникации точка-к-точке с протоколом, разработанным и реализованным в пользовательской программе. Эти инструкции попадают в две категории:

- Инструкции конфигурирования
- Инструкции коммуникаций

Инструкции конфигурирования

Прежде чем Ваша пользовательская программа сможет участвовать в PtP-коммуникациях, Вы должны сконфигурировать коммуникационный порт интерфейса и параметры для отправки и получения данных.

Вы можете выполнить конфигурацию порта и сообщения для каждого СМ или СВ через конфигурацию устройства или через эти инструкции в Вашей пользовательской программе:

- Port_Config (стр. 879)
- Send_Config (стр. 882)
- Receive_Config (стр. 884)

Инструкции коммуникаций

PtP-коммуникационные инструкции позволяют пользовательской программе отправлять и принимать сообщения от коммуникационных интерфейсов. Для получения информации о передаче данных с помощью этих инструкций смотрите раздел по согласованности данных (стр. 185).

Все PtP-функции работают асинхронно. Пользовательская программа может использовать архитектуру опроса, чтобы определить состояние процессов передачи и приема. Send_P2P и Receive_P2P могут обрабатываться одновременно. Коммуникационные модули и буфер коммуникационная плата помещает в буфер передаваемые и принятые сообщения по мере необходимости до максимального заполнения буфера (1024 байта).

СМ и СВ отправляют и принимают сообщения от фактических устройств "точка-к-точке". Протокол сообщения находится в буфере, который либо принят из, либо отправлен в определенный коммуникационный порт. Буфер и порт являются параметрами инструкций передачи и приема:

- Send_P2P (стр. 889)
- Receive_P2P (стр. 893)

Дополнительные инструкции обеспечивают возможность сброса приемного буфера, а также получения и установки определенных RS232-сигналов:

- Receive_Reset (стр. 895)
- Signal_Get (стр. 895)
- Signal_Set (стр. 897)

13.3.4.1. Архитектура опроса

Пользовательская STEP 7 программа должна вызвать S7-1200 инструкции точка-к-точке циклически/периодически, чтобы выполнить проверку на принятые сообщения. Опрос отправления извещает пользовательскую программу о завершении передачи.

Архитектура опроса: ведущее устройство

Типичная последовательность для ведущего устройства является следующей:

1. Инструкция Send_P2P (стр. 889) инициирует передачу в СМ или СВ.
2. Инструкция Send_P2P обрабатывается в течение последовательных сканов, чтобы выполнять опрос относительно завершения передачи.
3. Когда инструкция Send_P2P указывает на то, что передача завершена, пользовательский код может выполнить подготовку к приему ответа.
4. Инструкция Receive_P2P (стр. 893) выполняется повторно, чтобы проверить наличие ответа. Когда СМ или СВ приняли ответное сообщение, инструкция Receive_P2P копирует ответ в ЦПУ и указывает на то, что новые данные были приняты.
5. Пользовательская программа может обработать ответ.
6. Перейти к шагу 1 и повторить цикл.

Архитектура опроса: ведомое устройство

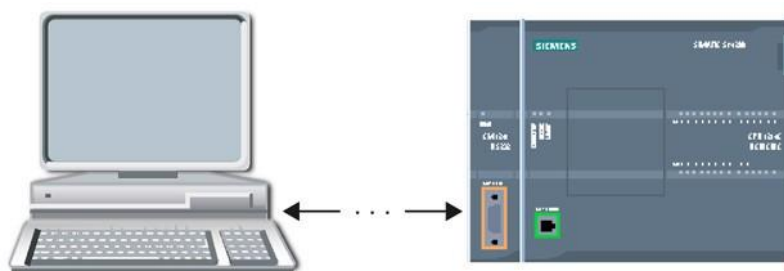
Типичная последовательность для ведомого устройства является следующей:

1. Пользовательская программа обрабатывает инструкцию Receive_P2P каждый скан.
2. Когда СМ или СВ получили запрос, инструкция Receive_P2P указывает на то, что новые данные готовы, и запрос скопирован в ЦПУ.
3. Пользовательская программа обслуживает запрос и генерирует ответ.
4. Использовать инструкцию Send_P2P, чтобы передать ответ ведущему устройству.
5. Повторно обрабатывайте Send_P2P, чтобы быть уверенными, что передача выполняется.
6. Перейти к шагу 1 и повторить цикл.

Ведомое устройство должен быть ответственно за вызов Receive_P2P с достаточной частотой, чтобы принять передачу от ведущего устройства, до наступления таймаута в ведущем устройстве при ожидании ответа. Чтобы выполнить эту задачу, пользовательская программа может вызывать RCV_PTP из циклического ОВ, в котором время цикла достаточно для приема передачи от ведущего устройства, прежде чем наступит тайм-аут. Если Вы устанавливаете время цикла для ОВ, предусматривающее два вызова за уставку тайм-аута ведущего устройства, то пользовательская программа сможет принять передачи, не пропустив никакое из них.

13.3.5. Пример: Коммуникации точка-к-точке

В этом примере ЦПУ S7-1200 обменивается данными с ПК с эмулятором терминала через модуль CM 1241 RS232. Конфигурация точка-к-точке и STEP 7 программа в этом примере иллюстрируют, как ЦПУ может принять сообщение от ПК и повторить сообщение назад в ПК.



Вы должны подключить коммуникационный интерфейс модуля CM 1241 RS232 к интерфейсу RS232 в ПК, который обычно является портом COM1. Поскольку оба этих порта являются терминальным оборудованием (DTE), Вы должны перебросить приемные и передающие контакты (2 и 3) при соединении этих двух портов, что можно выполнить любым из следующих способов:

- Используйте адаптер нуль-модема, чтобы перебросить контакты 2 и 3 вместе со стандартным RS232-кабелем .
- Используйте нуль-модемный кабель, у которого уже переброшены контакты 2 и 3. Вы можете обычно идентифицировать нуль-модемный кабель, как кабель с двумя 9-штырьковыми гнездовыми D-соединителями.

13.3.5.1. Конфигурирование коммуникационного модуля

Вы можете сконфигурировать CM 1241 в конфигурации устройства в STEP 7 или с помощью инструкций пользовательской программы. Данный пример использует метод конфигурации устройства.

- Конфигурация порта: щелкните по коммуникационному порту CM модуля в конфигурации устройства и сконфигурируйте порт как показано:

Port configuration

Operating mode

- ☐ Full duplex (RS422) four wire mode (point-to-point connection)
- ☐ Full duplex (RS422) four wire mode (multipoint master)
- ☐ Full duplex (RS422) four wire mode (multipoint slave)
- ☒ Half duplex (RS485) two wire mode

Receive line initial state

- ☒ None
- ☐ Forward bias (Signal R(A) 0V, signal R(B) 5V)

Transmission rate: 9.6 kbps

Parity: No parity

Data bits: 8 bits per character

Stop bits: 1

Flow control: None

XON character (HEX): 0

XON character (ASCII): NUL

XOFF character (HEX): 0

XOFF character (ASCII): NUL

Wait time: 1 ms

Примечание

Параметры конфигурации для "Operating mode" и "Receive line initial state", применимы только для CM 1241 (RS422/RS485) модуля. У другого CM 1241 модуля нет этих параметров конфигурации порта. Обратитесь к разделу Конфигурирование RS422 и RS485 (стр. 905).

- Конфигурация сообщения передачи: Примите значение по умолчанию для конфигурации сообщения передачи. Никакой разрыв не должен быть отправлено при начале сообщения.

13.3 Коммуникации точка-к-точке (PtP)

- Конфигурация начала приемного сообщения: Сконфигурируйте СМ 1241, для начала приема сообщения, когда линия связи будет неактивной в течение по крайней мере 50 битовых интервалов (приблизительно 5 миллисекунд при 9600 бодах = $50 * 1/9600$):

- Конфигурация завершения приемного сообщения: Сконфигурируйте СМ 1241 так, чтобы завершить сообщение, когда он получит максимум 100 байтов или символ перевода строки (10 десятичных или шестнадцатеричных чисел). Последовательность завершения позволяет наличие до пяти конечных символов в последовательности. Пятый символ в последовательности является символом перевода строки. Предшествующие четыре символа завершения последовательности, "игнорируются" или не выбираются. СМ 1241 не оценивает "игнорируемые" символы, но ищет символ перевода строки, которому предшествует нуль, или больше "игнорируемых" символов, чтобы указать на завершение сообщения.

5-character message end sequence

☐ Check character 1	
Character value (HEX):	0
Character value (ASCII):	ANY
☐ Check character 2	
Character value (HEX):	0
Character value (ASCII):	ANY
☐ Check character 3	
Character value (HEX):	0
Character value (ASCII):	ANY
☐ Check character 4	
Character value (HEX):	0
Character value (ASCII):	ANY
☒ Check character 5	
Character value (HEX):	A
Character value (ASCII):	LF

13.3.5.2. Рабочие режимы RS422 и RS485

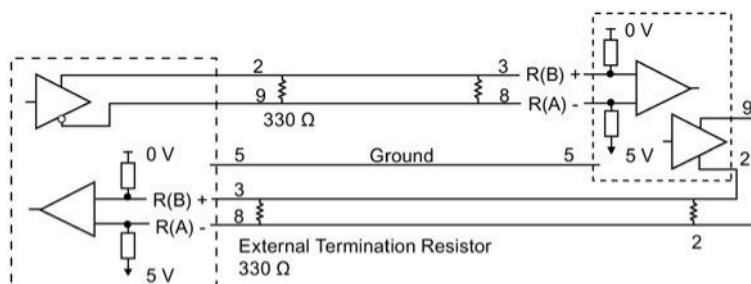
Конфигурирование RS422

Для режима RS422 существует три рабочих режима в зависимости от Вашей конфигурации сети. Выберите один из этих рабочих режимов в зависимости от устройств в Вашей сети. Различные варианты выбора для начального состояния приемной линии, ссылаются на случаи, представленные ниже для получения дополнительной информации.

- Полный дуплекс (RS422) четырехпроводный режим (соединение точка-к-точке): выберите эту опцию, когда в Вашей сети будет два устройства. В начальном состоянии приемной линии:
 - Выберите none, когда Вы предоставляете смещение и завершение (Случай 3).
 - Выберите forward bias, чтобы использовать внутреннее смещение и завершение (Случай 2).
 - Выберите reverse bias, чтобы использовать внутреннее смещение и завершение, и разрешить обнаружение обрыва кабеля для обоих устройств (Случай 1).
- Полный дуплекс (RS422) четырехпроводный режим (многоточечное ведущее устройство): выберите эту опцию для ведущего устройства, когда у Вас будет сеть с одним ведущим и несколькими ведомыми устройствами. В начальном состоянии приемной линии:
 - Выберите none, когда Вы предоставляете смещение и завершение (Случай 3).
 - Выберите forward bias, чтобы использовать внутреннее смещение и завершение (Случай 2).
 - Обнаружение обрыва кабеля невозможно в этом режиме.
- Полный дуплекс (RS422) четырехпроводный режим (многоточечное ведомое устройство): Выберите эту опцию для всех ведомых устройств когда у Вас будет сеть с одним ведущим и несколькими ведомыми устройствами. В начальном состоянии приемной линии:
 - Выберите none, когда Вы предоставляете смещение и завершение (Случай 3).
 - Выберите forward bias, чтобы использовать внутреннее смещение и завершение (Случай 2).
 - Выберите reverse bias, чтобы использовать внутреннее смещение и завершение, и разрешить обнаружение обрыва кабеля для ведомых устройств (Случай 1).

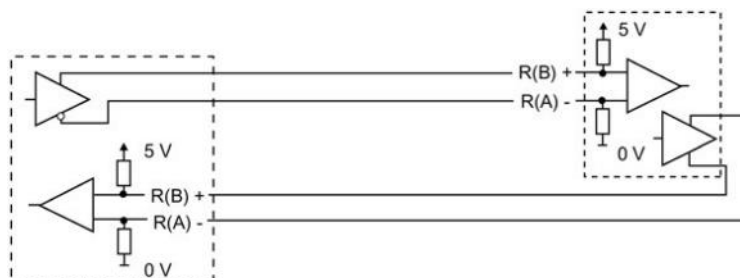
Случай 1: RS422 с обнаружением обрыва кабеля

- Режим работы: RS422
- Начальное состояние приемной линии: Reverse bias (смещение с $R(A) > R(B) > 0\text{ В}$)
- Обрыв кабеля: Обнаружение обрыва кабеля активно (передатчик всегда активен)



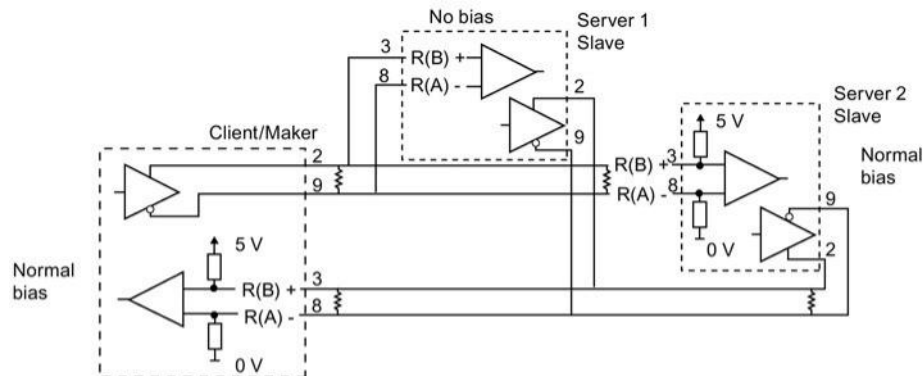
Случай 2: RS422 без обнаружения обрыва кабеля, прямое смещение

- Режим работы: RS422
- Начальное состояние приемной линии: Forward bias (смещение с $R(B) > R(A) > 0\text{ В}$)
- Обрыв кабеля: Без обнаружения обрыва кабеля (передатчик разрешен только при передаче)



Случай 3: RS422: Без обнаружения обрыва кабеля, без смещения

- Режим работы: RS422
- Начальное состояние приемной линии: no bias
- Обрыв кабеля: Без обнаружения обрыва кабеля (передатчик разрешен только при передаче). Смещение и завершение добавляются пользователем в конечных узлах сети.

**Конфигурирование RS485**

Для режима RS485 есть только один рабочий режим. Различные варианты выбора для начального состояние приемной линии, ссылаются на случаи, представленные ниже для получения дополнительной информации.

- Полудуплекс (RS485) двухпроводный режим. В начальном состоянии приемной линии:
 - Выберите none, когда Вы предоставляете смещение и завершение (Случай 5).
 - Выберите forward bias, чтобы использовать внутреннее смещение и завершение (Случай 4).

Случай 4: RS485: Прямое смещение

- Режим работы: RS485
- Начальное состояние приемной линии: Forward bias (смещение с $R(B) > R(A) > 0 \text{ V}$)



Случай 5: RS485: Без смещения (внешнее смещение)

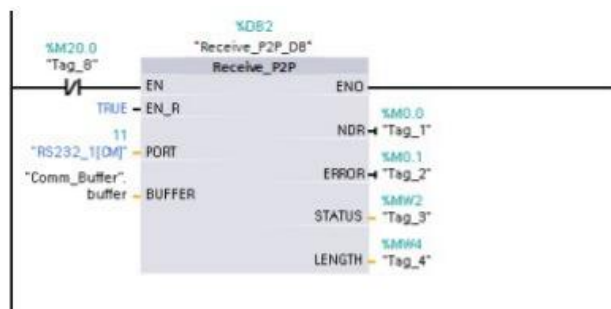
- Режим работы: RS485
- Начальное состояние приемной линии: No bias (требуется внешнее смещение)

**13.3.5.3. Создание STEP 7 программы**

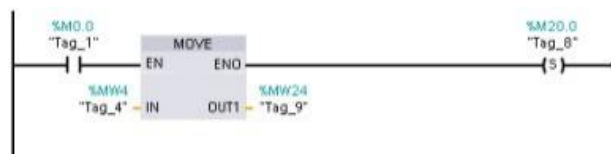
Программа из примера использует глобальный блок данных для коммуникационного буфера, инструкцию RCV_PTP (стр. 987), чтобы получить данные от эмулятора терминала и инструкцию SEND_PTP (стр. 985), чтобы переслать буфер назад в эмулятор терминала. Чтобы запрограммировать пример, добавьте конфигурацию блока данных и основной программный блок OB 1, как описано ниже.

Глобальный блок данных "Comm_Buffer": Создайте глобальный блок данных (DB) и назовите его "Comm_Buffer". Создайте один элемент в блоке данных, названном "буфером" с типом данных "array [0 .. 99] of byte".

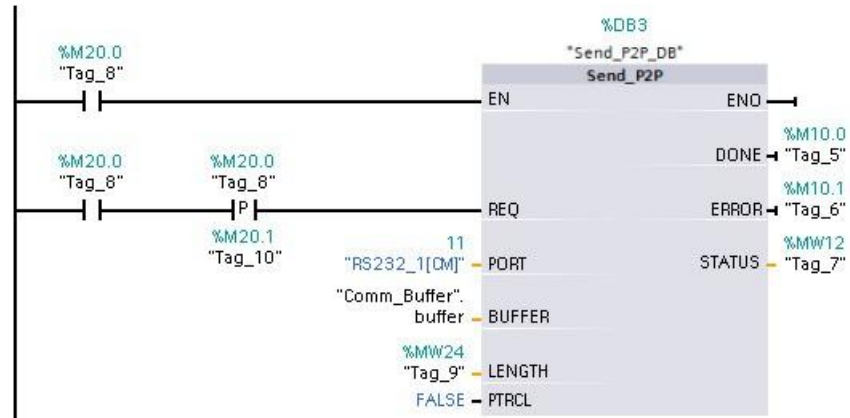
Сегмент 1: Разрешайте инструкцию RCV_PTP каждый раз, когда SEND_PTP неактивен. Tag_8 в MW20.0 указывает на то, что передача завершена в Сегменте 4, и коммуникационный модуль, таким образом, готов принять сообщение.



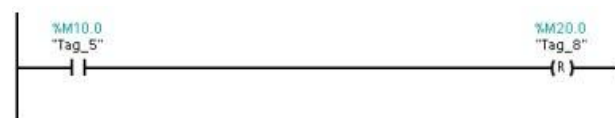
Сегмент 2: Используйте значение NDR (Tag_1 в M0.0), установленный инструкцией RCV_PTP, чтобы сделать копию количества принятых байтов и установить флаг (Tag_8 в M20.0), чтобы активировать инструкцию SEND_PTP.



Сегмент 3: Разрешите инструкцию SEND_PTP, когда флаг M20.0 будет установлен. Также используйте этот флаг, чтобы установить вход REQ в TRUE на один скан. Вход REQ сообщает инструкции SEND_PTP, что должен быть передан новый запрос. Вход REQ должен быть установлен в TRUE только для одного выполнения SEND_PTP. Инструкция SEND_PTP выполняется на каждом скане, пока передача не завершится. Передача завершена, когда последний байт сообщения был передан из CM 1241. Когда передача завершена, выход DONE (Tag_5 в M10.0) установлен в TRUE для одного выполнения SEND_PTP.



Сегмент 4: Контролируйте выход DONE в SEND_PTP и сбросьте флаг передачи (Tag_8 в M20.0), когда операция передачи будет завершена. Когда флаг передачи сброшен, инструкция RCV_PTP в Сегменте 1 позволяет принять следующее сообщение.



13.3.5.4. Конфигурирование эмулятора терминала

Вы должны настроить эмулятор терминала для поддержки программы из примера. Вы можете использовать большую часть существующих эмуляторов терминала на своем ПК, таких как HyperTerminal. Удостоверьтесь, что эмулятор терминала находится в отключенном режиме прежде, чем редактировать настройки следующим образом:

1. Настройте эмулятор терминала для использования порта RS232 на ПК (обычно COM1).
2. Сконфигурируйте порт на 9600 бод, 8 битов данных, отсутствие контроля четности (none), 1 стоповый бит и отсутствие управления потоком.
3. Измените настройки эмулятора терминала, чтобы эмулировать терминал ANSI.
4. Сконфигурируйте установку ASCII эмулятора терминала, чтобы посылать перевод строки после каждой строки (после того, как пользователь нажмет клавишу Enter).
5. Отображайте символы локально так, чтобы эмулятор терминала вывел на экран то, что введено.

13.3.5.5. Обработка примера программы

Чтобы использовать пример программы, выполните следующие шаги:

1. Загрузите STEP 7 программу в ЦПУ и убедитесь, что он находится в режиме RUN.
2. Нажмите кнопку "connect" в эмуляторе терминала, чтобы применить изменения конфигурации и открыть терминальный сеанс с СМ 1241.
3. Введите символы в ПК и нажмите Enter.

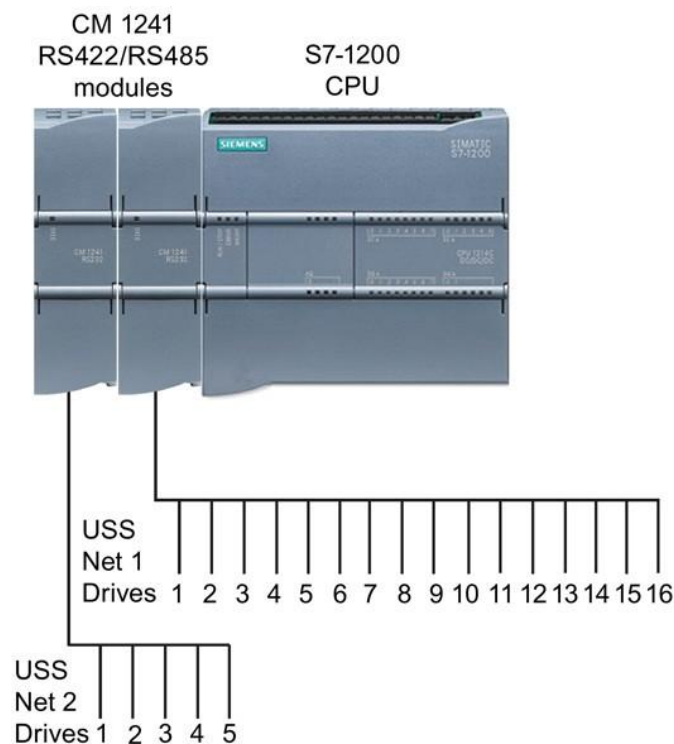
Эмулятор терминала отправляет символы в СМ 1241 и в ЦПУ. Программа ЦПУ затем отправляет символы назад в эмулятор терминала.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

13.4.1. Обзор

USS-инструкции управляют работой электроприводов, которые поддерживают протокол универсального последовательного интерфейса (USS). Вы можете использовать USS-инструкции, чтобы связаться с несколькими приводами через RS485-соединения с коммуникационными модулями CM 1241 RS485 или коммуникационной платой CB 1241 RS485. До трех модулей CM 1241 RS422/RS485 и одна плата CB 1241 RS485 могут быть установлены в ЦПУ S7-1200. Каждый порт RS485 может управлять максимум шестнадцатью приводами.

USS-протокол использует сеть "ведущий-ведомый" для коммуникаций по последовательной шине. Ведущее устройство использует параметр адреса, чтобы отправить сообщение выбранному ведомому устройству. Ведомое устройство никогда не может самостоятельно передавать без предшествующего получения запроса. Прямая передача сообщений между отдельными ведомыми устройствами невозможна. USS-обмен работает в полудуплексном режиме. Следующая иллюстрация показывает сетевую диаграмму для примера приложения с приводами.



USS -коммуникации по PROFIBUS или PROFINET

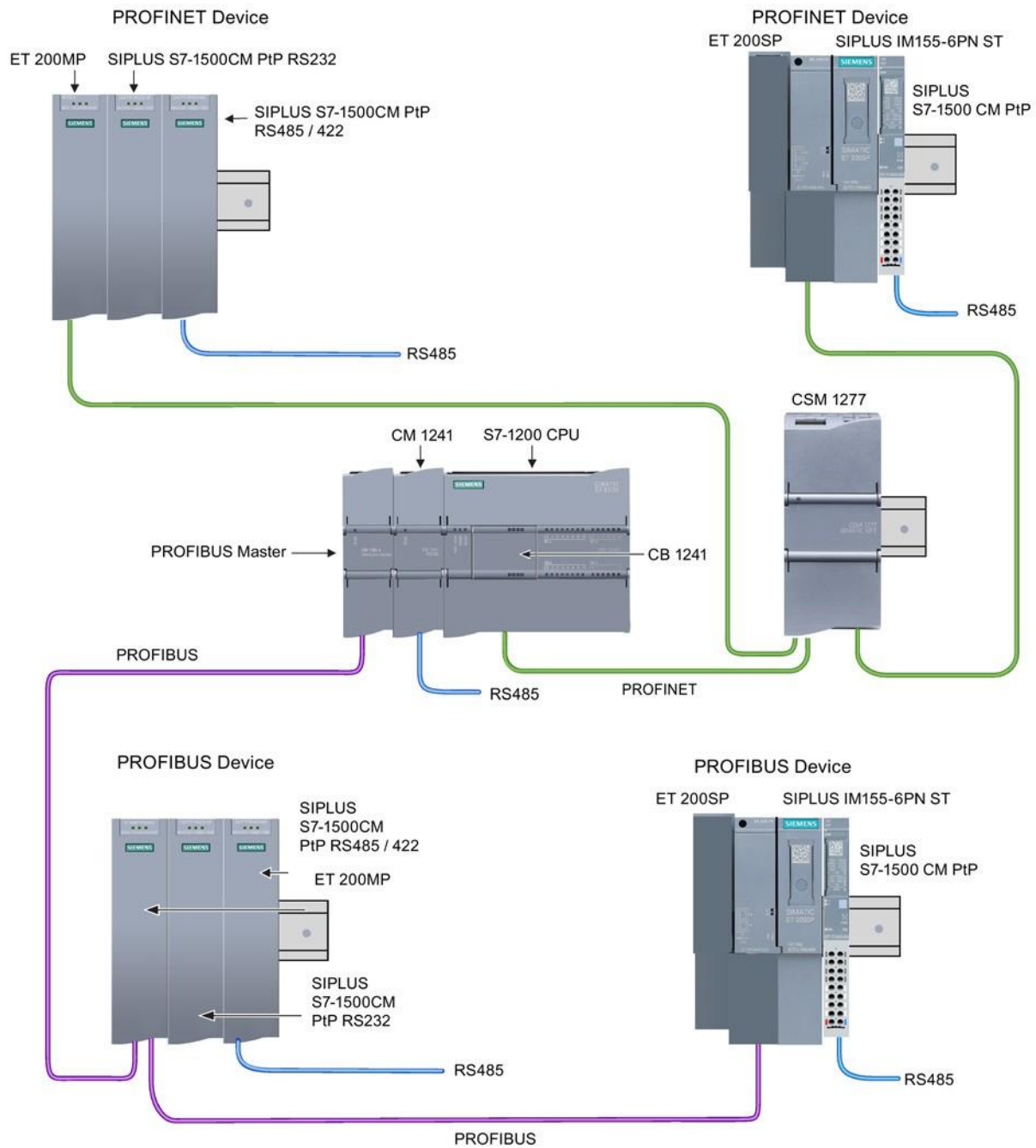
Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможности USS для использования PROFINET или PROFIBUS стойки распределенного ввода-вывода для связи с различными устройствами (RFID-считывателями, GPS-устройствами и другими):

- PROFINET (стр. 616): Вы подключаете Ethernet- интерфейс ЦПУ S7-1200 к PROFINET интерфейсному модулю. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную передачу данных в PtP-устройство.
- PROFIBUS (стр. 752): Вы вставляете коммуникационный модуль PROFIBUS в левую часть стойки с ЦПУ S7-1200. Вы соединяете коммуникационный модуль PROFIBUS со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную передачу данных в PtP-устройство.

По этой причине S7-1200 поддерживает два набора PtP-инструкций:

- Унаследованные USS-инструкции (стр. 997): Эти USS-инструкции существовали до версии V4.0 S7-1200 и работали только с последовательной передачей данных, используя коммуникационный модуль CM 1241 или коммуникационную плату CB 1241.
- USS-инструкции (стр. 918): Эти USS-инструкции обеспечивают всю функциональность унаследованных инструкций плюс возможность подключения к PROFINET и PROFIBUS распределенному вводу-выводу. Инструкции "точка-к-точке" позволяют Вам конфигурировать связи между PtP коммуникационными модулями в распределенной стойке ввода-вывода и PtP-устройствами.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

**Примечание**

С версией V4.1 S7-1200 Вы можете использовать инструкции "точка-к-точке" для всех типов обмена: последовательный, последовательный по PROFINET и последовательный по PROFIBUS. STEP 7 обеспечивает унаследованные инструкции "точка-к-точке" только для поддержки существующих программы. Унаследованные инструкции все еще работают со всеми ЦПУ S7-1200. Вы не должны предварительно преобразовывать программы от одного набора инструкций к другому.

13.4.2. Выбор версии USS-инструкций

Существуют две версии USS-инструкций, доступных в STEP 7:

- Версия 2.0 (унаследованные инструкции) была первоначально доступна в STEP 7 Basic/Professional V13.
- Версия 2.1 доступна в STEP 7 Basic/Professional V13, SP1.

Для совместимости и упрощения миграции, Вы можете выбрать то, какую версию инструкции вставить в Вашу пользовательскую программу.

Вы не можете использовать обе версии инструкций с тем же модулем, но два различных модуля могут использовать различные версии инструкций.



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.

USS communication		V2.1
USS_Port_Scan	Communication via US...	V2.1
USS_Drive_Control	Data exchange with th...	V2.0
USS_Read_Param	Read data from drive	V2.1
USS_Write_Param	Change data in drive	V1.4

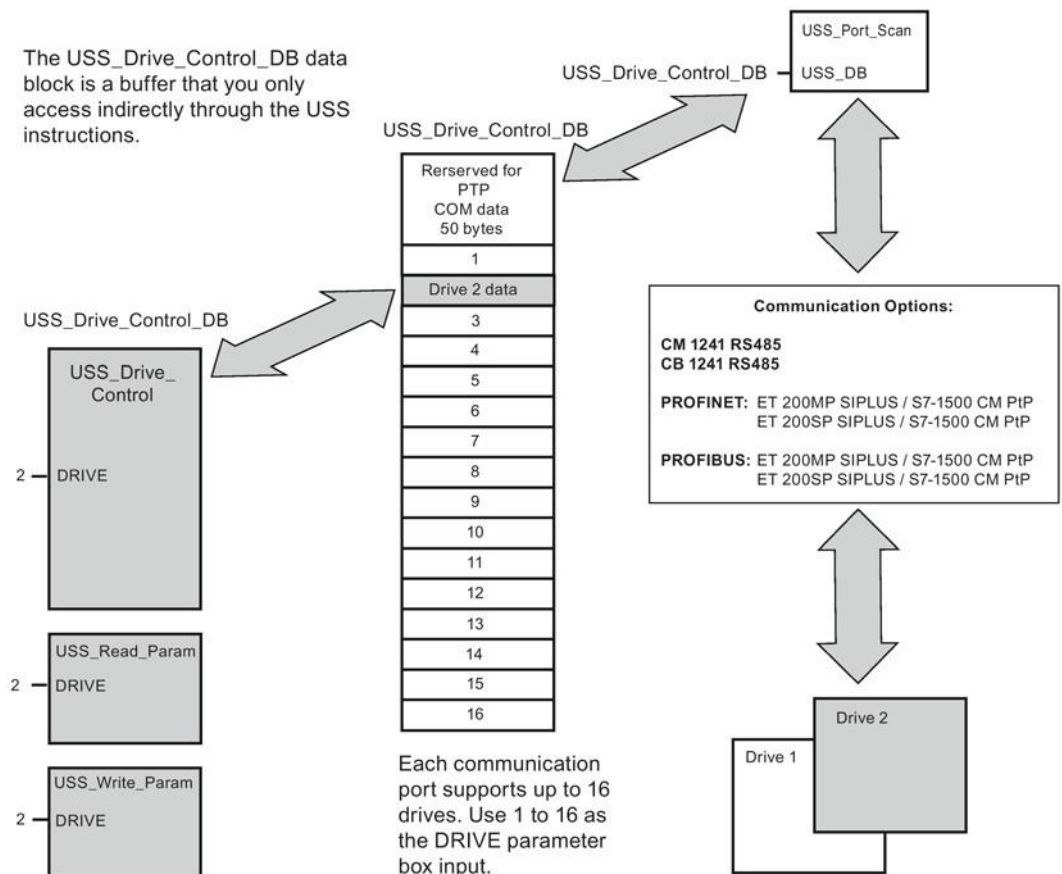
Чтобы изменить версию USS-инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить USS-инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной USS-инструкции, создается в дереве проекта. Вы можете видеть новый экземпляр FB или FC в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию USS-инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите USS-экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии USS-инструкции.

13.4.3. Требования к использованию USS-протокола

Четыре USS-инструкции используют два функциональных блока (FB) и две функции (FC), чтобы поддержать USS-протокол. Один экземплярный блок данных (DB) USS_Port_Scan используется для каждой USS-сети. Экземплярный блок данных USS_Port_Scan содержит временное хранилище и буферы для всех приводов в этой USS-сети. USS-инструкции совместно используют информацию в этом блоке данных.



Все приводы (до 16), подключенные к единственному порту RS485, являются участниками одной и той же USS-сети. Все приводы, подключенные к различным портам RS485, являются участниками различных USS-сетей. Каждой USS-сетью управляют, используя единый блок данных. Все инструкции, связанные с отдельной USS-сетью, должны совместно использовать этот блок данных. Сюда относятся все USS_Drive_Control, USS_Port_Scan, USS_Read_Param и USS_Write_Param инструкции, используемые для управления всеми приводами в отдельной USS-сети.

Инструкция USS_Drive_Control является функциональным блоком (FB). Когда Вы поместите инструкцию USS_Drive_Control в редактор программы, Вам будет предложено диалоговое окно "Call options" для назначения DB этому FB. Если это первая инструкция USS_Drive_Control в этой программе для этой USS-сети, то Вы можете принять назначение DB по умолчанию (или изменить имя, если хотите), и новый DB создается для Вас. Если, однако, это не первая инструкция USS_Drive_Control для этого канала, то Вы должны использовать выпадающий список в диалоговом окне "Call options", чтобы выбрать имя DB, которое было ранее назначено для этой USS-сети.

Инструкция USS_Port_Scan является функциональным блоком (FB) и обрабатывает фактический обмен между ЦПУ и приводами через коммуникационный порт RS485 "точка-к-точке" (PtP). Каждый вызов этого FB обрабатывает одно соединение с одним приводом. Ваша программа должна вызвать этот FB достаточно часто, чтобы предотвратить коммуникационный тайм-аут приводов. Вы можете вызывать этот функциональный FB в OB основного программного цикла или любом OB прерывания.

Инструкции USS_Read_Param и USS_Write_Param обе являются функциями (FC). Никакой DB не назначается, когда Вы помещаете эти FC в редакторе. Вместо этого Вы должны указать надлежащую ссылку на DB на входе "USS_DB" для этих инструкций. Дважды щелкните по полю параметра, а затем по значку помощника параметра, чтобы увидеть доступные имена DB.

Как правило, Вы должны вызывать FB USS_Port_Scan из OB циклического прерывания. Время цикла для OB должно быть установлено приблизительно на половину минимального интервала вызова (Например, обмен на скорости 1200 бод должен использовать время цикла 350 мс или меньше).

FB USS_Drive_Control обеспечивает Вашей программе доступ к указанному приводу в USS-сети. Его входы и выходы являются состояниями и органами управления для привода. Если в сети 16 приводов, то в Вашей программе должно быть по крайней мере 16 вызовов USS_Drive_Control, по одному для каждого привода. Эти блоки нужно вызывать с интервалом, который требуется для управления работой привода.

Вы можете вызвать FB USS_Drive_Control только из OB основного программного цикла.

ПРЕДУПРЕЖДЕНИЕ

Соображения по вызову USS -инструкций из OB

Вызывайте USS_Drive_Control, USS_Read_Param и USS_Write_Param только из OB основного программного цикла. FB USS_Port_Scan можно вызывать из любого OB, обычно из OB циклического прерывания.

Не используйте инструкции USS_Drive_Control, USS_Read_Param и USS_Write_Param в OB с более высоким приоритетом, чем соответствующая инструкция USS_Port_Scan. Например, не помещайте USS_Port_Scan в основной OB, а USS_Read_Param в OB циклического прерывания. Отказ предотвратить прерывание обработки USS_Port_Scan может привести к неожиданным ошибкам, которые могли бы привести к телесным повреждениям.

USS_Read_Param и USS_Write_Param FC читают и записывают рабочие параметры удаленного привода. Эти параметры управляют внутренней работой привода. См. руководство на привод для определения этих параметров. Ваша программа может содержать столько этих функций сколько необходимо, но только один запрос на чтение или запись может быть активным на привод в любой момент времени. Вы можете вызывать USS_Read_Param и USS_Write_Param FC только из OB основного программного цикла.

Расчет времени, требуемого для обмена с приводом

Коммуникации с приводом являются асинхронными к циклу сканирования S7-1200. S7-1200 обычно завершает несколько сканов, прежде чем одна коммуникационная транзакция привода будет завершена.

Интервал USS_Port_Scan представляет время, требуемое для одной транзакции привода. Таблица ниже показывает минимальный интервал USS_Port_Scan для каждой скорости обмена в бодах. Вызов FB USS_Port_Scan более частый, чем интервал USS_Port_Scan не увеличит число транзакций. Интервал тайм-аута привода соответствует времени, которое могло бы занять транзакция, если бы коммуникационные ошибки вызвали 3 попытки завершить транзакцию. По умолчанию библиотека USS-протокола автоматически делает до 2 повторений на каждой транзакции.

Таблица 13- 38 Расчет временных характеристик

Скорость в бодах	Расчетный минимальный интервал вызова для USS_Port_Scan (мс)	Тайм-аут интервала сообщения на один привод (мс)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.4.4. USS-инструкции

13.4.4.1. Инструкция USS_Port_Scan (редактировать обмен, использующий USS-сеть)

Таблица 13- 39 Инструкция USS_Port_Scan

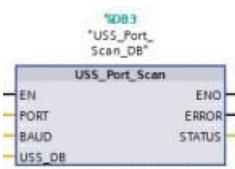
LAD / FBD	SCL	Описание
	<pre>USS_Port_Scan(PORT:=_uint_in_, BAUD:=_dint_in_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	Инструкция USS_Port_Scan обрабатывает обмен по USS-сети.

Таблица 13- 40 Типы данных для параметров

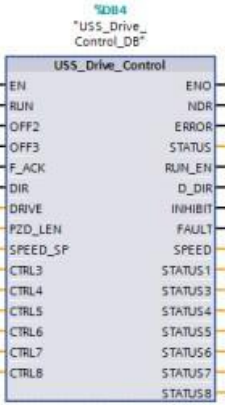
Параметр и тип		Тип данных	Описание
PORT	IN	Port	После того, как Вы установите и сконфигурируете СМ или СВ, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта СМ или СВ является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов.
BAUD	IN	DInt	Скорость в бодах используемая для USS-обмена.
USS_DB	INOUT	USS_BASE	Имя DB экземпляра, который создан и инициализирован, когда инструкция USS_Drive_Control помещена в Вашу программу.
ERROR	OUT	Bool	Когда true, этот вывод указывает на то, что произошла ошибка, и вывод STATUS допустим.
STATUS	OUT	Word	Значение состояния запроса указывает результат сканирования или инициализации. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

Как правило, существует только одна инструкция USS_Port_Scan на коммуникационный PtP-порт в программе, и каждый вызов этого функционального блока (FB) обрабатывает передачу в или из одиночного привода. Все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом должны использовать тот же экземплярный DB.

Ваша программа должна выполнять инструкцию USS_Port_Scan достаточно часто, чтобы предотвратить тайм-ауты приводов. USS_Port_Scan обычно вызывают из OB циклического прерывания, чтобы предотвратить тайм-ауты приводов и сохранить последние обновления USS-данных, доступными для вызовов USS_Drive_Control.

13.4.4.2. Инструкция USS_Drive_Control (обмениваться данными с приводом)

Таблица 13- 41 Инструкция USS_Drive_Control

LAD / FBD	SCL	Описание
	<pre> "USS_Drive_Control_DB"(RUN:=_bool_in_, OFF2:=_bool_in_, OFF3:=_bool_in_, F_ACK:=_bool_in_, DIR:=_bool_in_, DRIVE:=_usint_in_, PZD_LEN:=_usint_in_, SPEED_SP:=_real_in_, CTRL3:=_word_in_, CTRL4:=_word_in_, CTRL5:=_word_in_, CTRL6:=_word_in_, CTRL7:=_word_in_, CTRL8:=_word_in_, NDR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, RUN_EN=>_bool_out_, D_DIR=>_bool_out_, INHIBIT=>_bool_out_, FAULT=>_bool_out_, SPEED=>_real_out_, STATUS1=>_word_out_, STATUS3=>_word_out_, STATUS4=>_word_out_, STATUS5=>_word_out_, STATUS6=>_word_out_, STATUS7=>_word_out_, STATUS8=>_word_out_); </pre>	Инструкция USS_Drive_Control обменивается данными с приводом, создавая сообщения запроса, и интерпретируя ответные сообщения привода. Отдельный функциональный блок должен использоваться для каждого привода, но все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом, должны использовать тот же экземплярный блок данных. Вы должны создать имя DB, когда Вы помещаете первую инструкцию USS_Drive_Control, а затем обращаться к DB, который создавался при первом использовании инструкции. STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

- ¹ LAD и FBD: Разверните прямоугольник, чтобы показать все параметры, щелкнув по его нижней части. Контакты параметров, которые выделены серым, являются дополнительными, и присвоение параметра не требуется.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Таблица 13- 42 Типы данных для параметров

Параметр и тип		Тип данных	Описание
RUN	IN	Bool	Бит запуска привода: Когда true, этот вход разрешает приводу вращаться с заданной скоростью. Когда RUN переходит в false, пока привод работает, двигатель будет замедляться до остановки. Это поведение отличается от сброса мощности (OFF2) или торможения двигателя (OFF3).
OFF2	IN	Bool	Бит электрической остановки: Когда false, этот бит заставляет привод останавливаться по инерции без торможения.
OFF3	IN	Bool	Бит быстрой остановки: Когда false, этот бит вызывает быструю остановку торможением привода вместо того, чтобы просто позволить приводу остановиться по инерции.
F_ACK	IN	Bool	Бит квитирования отказа: Этот бит устанавливается для сброса отказа привода. Бит устанавливается после того, как отказ устранен, чтобы указать приводу на то, что не нужно больше сообщать о предыдущем отказе.
DIR	IN	Bool	Управление направлением привода: Этот бит устанавливается для указания направления вперед (для положительного SPEED_SP).
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PZD_LEN	IN	USInt	Длина в словах: Это количество слов в PZD-данных. Допустимые значения составляют 2, 4, 6, или 8 слов. Значение по умолчанию - 2.
SPEED_SP	IN	Real	Уставка скорости: Это скорость привода как процент заданной частоты. Положительное значение определяет прямое направление (когда DIR в true). Допустимый диапазон от 200.00 до -200.00.
CTRL3	IN	Word	Слово управления 3: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL4	IN	Word	Слово управления 4: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL5	IN	Word	Слово управления 5: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL6	IN	Word	Слово управления 6: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL7	IN	Word	Слово управления 7: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL8	IN	Word	Слово управления 8: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
NDR	OUT	Bool	Новые данные готовы: Когда true, бит указывает на то, что выходы содержат данные из нового коммуникационного запроса.
ERROR	OUT	Bool	Ошибка произошла: Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	Значение состояния запроса указывает на результат сканирования. Это не слово состояния, возвращенное приводом.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Параметр и тип		Тип данных	Описание
RUN_EN	OUT	Bool	Работа разрешена: Этот бит указывает на то, работает ли привод.
D_DIR	OUT	Bool	Направление привода: Этот бит указывает на то, вращается ли привод вперед.
INHIBIT	OUT	Bool	Привод запрещен: Этот бит указывает на состояние бита запрета привода.
FAULT	OUT	Bool	Отказ привода: Этот бит указывает на то, что привод зарегистрировал отказ. Вы должны устранить проблему и затем установить бит F_ACK, чтобы сбросить этот бит, когда он установлен.
SPEED	OUT	Real	Текущая скорость привода (масштабированное значение слова состояния привода Word 2): значение скорости привода, как процент заданной скорости.
STATUS1	OUT	Word	Слово состояния привода Word 1: Это значение содержит фиксированные биты состояния привода.
STATUS3	OUT	Word	Слово состояния привода Word 3: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS4	OUT	Word	Слово состояния привода Word 4: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS5	OUT	Word	Слово состояния привода Word 5: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS6	OUT	Word	Слово состояния привода Word 6: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS7	OUT	Word	Слово состояния привода Word 7: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS8	OUT	Word	Слово состояния привода Word 8: Это значение содержит конфигурируемое пользователем слово состояния привода.

Когда происходит начальное выполнение USS_Drive_Control, привод, обозначенный USS-адресом (параметр DRIVE), инициализируется в экземплярном DB. После этой инициализации последующие выполнения USS_Port_Scan могут начать обмен с приводом по указанному номеру.

Изменение номера привода требует перехода ЦПУ из STOP в RUN, который инициализирует экземплярный DB. Входные параметры конфигурируются в буфер сообщения USS TX, и выходы читаются из "предыдущего" допустимого буфера ответа, если он существует. Во время выполнения USS_Drive_Control нет никакой передачи данных. Приводы выполняют коммуникации, когда выполняется USS_Port_Scan.

USS_Drive_Control только конфигурирует сообщения, которые необходимо передать и интерпретирует данные, которые, возможно, были получены от предыдущего запроса.

Вы можете управлять направлением привода, используя либо вход DIR (Bool), либо знак (положительный, или отрицательный) со входом SPEED_SP (Real). Следующая таблица указывает на то, как эти входы работают вместе, чтобы определить направление привода, предполагая, что двигатель подключен для вращения в прямом направлении.

Таблица 13- 43 Взаимодействие параметров SPEED_SP и DIR

SPEED_SP	DIR	Направление вращения привода
Значение > 0	0	Обратное
Значение > 0	1	Прямое
Значение < 0	0	Прямое
Значение < 0	1	Обратное

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

13.4.4.3. Инструкция USS_Read_Param (считывание параметров из привода)

Таблица 13- 44 Инструкция USS_Read_Param

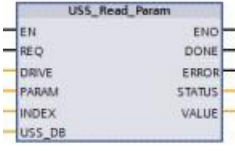
LAD / FBD	SCL	Описание
	<pre>USS_Read_Param(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, IN- DEX:=_uint_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, VAL- UE=>_variant_out_, USS_DB:=_fbtref_inout_);</pre>	Инструкция USS_Read_Param читает параметр из привода. Все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом, должны использовать тот же блок данных. USS_Read_Param нужно вызвать из ОБ основного программного цикла.

Таблица 13- 45 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Отправить запрос: Когда true, REQ указывает на то, что необходим новый запрос на чтение. Он игнорируется, если запрос на этот параметр уже находится на рассмотрении.
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PARAM	IN	UInt	Номер параметра: PARAM определяет, какой параметр привода записывается. Диапазон этого параметра от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Смотри руководство на свой привод для получения дополнительной информации о том, как получить доступ к расширенному диапазону.
INDEX	IN	UInt	Индекс параметра: INDEX определяет, какой индекс параметра привода должен быть записан. 16-разрядное значение, где младший значащий Байт представляет фактическое значение индекса в диапазоне (от 0 до 255). Старший значащий Байт может также использоваться приводом и специфичен для привода. Смотри руководство на свой привод для получения детальной информации.
USS_DB	INOUT	USS_BASE	Имя экземплярного DB, который создается и инициализируется, когда инструкция USS_Drive_Control помещена в Вашу программу.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Это значение параметра, который был считан, допустимо только, когда DONE находится в true.
DONE ¹	OUT	Bool	Когда true, указывает на то, что выход VALUE содержит ранее запрошенное считанное значение параметра. Этот бит устанавливается, когда USS_Drive_Control видит прочитанные ответные данные от привода. Этот бит сбрасывается, когда либо Вы запрашиваете ответные данные, используя другой опрос USS_Read_Param, либо при втором из следующих двух вызовов USS_Drive_Control.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Параметр и тип		Тип данных	Описание
ERROR	OUT	Bool	Ошибка произошла: Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	STATUS указывает на результат запроса на чтение. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

- ¹ Бит DONE указывает на то, что допустимые данные были считаны из адресованного электропривода, и доставлены ЦПУ. Это не говорит о том, что USS-библиотека способна к незамедлительному чтению другого параметра. Пустой PKW запрос должен быть отправлен в электропривод и должен также быть подтвержден инструкцией, прежде чем канал параметра для определенного привода станет доступным для использования. Незамедлительный вызов USS_Read_Param или USS_Write_Param FC для указанного электропривода приведет к ошибке "0x818A".

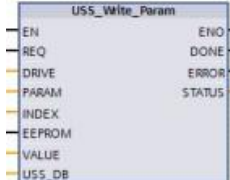
13.4.4.4. Инструкция USS_Write_Param (изменить параметры в приводе)

Примечание

Операции записи EEPROM (для EEPROM в USS-приводе)

Не злоупотребляйте постоянной операцией записи EEPROM. Минимизируйте число операций записи EEPROM, чтобы продлить срок службы EEPROM.

Таблица 13- 46 Инструкция USS_Write_Param

LAD / FBD	SCL	Описание
	<pre> USS_Write_Param(REQ:=_bool_in_ —, DRIVE:=_usint_in_, PARAM:=_uint_in_, IN- DEX:=_uint_in_, EEPROM:=_bool_in_, VAL- UE:=_variant_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, USS_DB:=_fbtref_inout_); </pre>	Инструкция USS_Write_Param изменяет параметр в приводе. Все USS-функции, связанные с одной USS-сетью и PTP коммуникационным портом, должны использовать тот же блок данных. USS_Write_Param нужно вызвать из ОБ основного программного цикла.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Таблица 13- 47 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Отправить запрос: Когда true, REQ указывает на то, что необходим новый запрос на запись. Он игнорируется, если запрос на этот параметр уже находится на рассмотрении.
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PARAM	IN	UInt	Номер параметра: PARAM определяет, какой параметр привода записывается. Диапазон этого параметра от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Смотри руководство на свой привод для получения дополнительной информации о том, как получить доступ к расширенному диапазону.
INDEX	IN	UInt	Индекс параметра: INDEX определяет, какой индекс параметра привода должен быть записан. 16-разрядное значение, где младший значащий Байт представляет фактическое значение индекса в диапазоне (от 0 до 255). Старший значащий Байт может также использоваться приводом и специфичен для привода. Смотри руководство на свой привод для получения детальной информации.
EEPROM	IN	Bool	Сохранить в EEPROM привода: Когда true, транзакция записи параметра привода будет сохранена в EEPROM привода. Если false, запись является временной и не будет сохранена, при выключении привода.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Значение параметра, который должен быть записан. Оно должно быть допустимо при переходе REQ.
USS_DB	INOUT	USS_BASE	Имя экземплярного DB, который создается и инициализируется, когда инструкция USS_Drive_Control помещена в Вашу программу.
DONE ¹	OUT	Bool	Когда true, DONE, указывает на то, что значение на входе VALUE было записано в привод. Этот бит устанавливается, когда USS_Drive_Control видит ответные данные записи от привода. Этот бит сбрасывается, когда либо Вы запрашиваете ответные данные, используя другой опрос USS_Drive_Control, либо при втором из следующих двух вызовов USS_Drive_Control.
ERROR	OUT	Bool	Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	STATUS указывает на результат запроса на запись. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

¹ Бит DONE указывает на то, что допустимые данные были считаны из адресованного электропривода, и представлены ЦПУ. Это не говорит о том, что USS-библиотека способна к незамедлительному чтению другого параметра. Пустой PKW запрос должен быть отправлен в электропривод и должен также быть подтвержден инструкцией, прежде чем канал параметра для определенного привода станет доступным для использования. Незамедлительный вызов USS_Read_Param или USS_Write_Param FC для указанного электропривода приведет к ошибке "0x818A".

13.4.5. Коды состояния USS

Коды состояния USS-инструкций возвращаются на выходе STATUS для функций.

Таблица 13- 48 Коды на выходе STATUS ¹

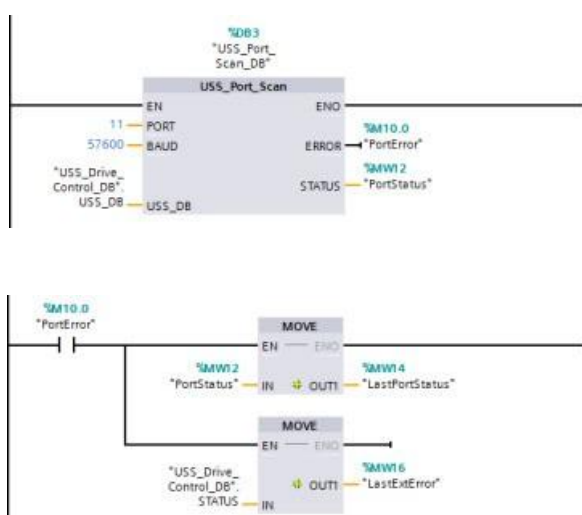
STATUS (W#16#....)	Описание
0000	Без ошибок
8180	Длина ответа привода не соответствовала символам, принятым от привода. Номер привода, где произошла ошибка, возвращается в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8181	Параметр VALUE не имел тип Word, Real или DWord.
8182	Пользователь предоставил Word для значения параметра и получил DWord или Real от привода в ответе.
8183	Пользователь предоставил DWord или Real для значения параметра и получил Word от привода в ответе.
8184	У ответной телеграммы от привода была неверная контрольная сумма. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8185	Недопустимый адрес привода (допустимый диапазон адреса : от 1 до16)
8186	Уставка скорости вне допустимого диапазона (допустимый диапазон SP скорости: от -200% до 200%).
8187	Привод с неправильным номером ответил на отправленный запрос. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8188	Определена недопустимая длина PZD в словах (допустимые значения = 2, 4, 6 или 8 слов)
8189	Была определена недопустимая скорость в бодах.
818A	Канал запроса параметра используется другим запросом к этому приводу.
818B	Привод не ответил на запросы и повторения. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
818C	Привод возвратил расширенную ошибку на операцию запроса параметра. Смотрите описание расширенной ошибки ниже данной таблицы.
818D	Привод возвратил ошибку несанкционированного доступа на операцию запроса параметра. Смотрите руководство на свой привод для получения информации относительно того, почему доступ к параметру может быть ограничен.
818E	Привод не был инициализирован. Этот код ошибки возвращен USS_Read_Param или USS_Write_Param, когда USS_Drive_Control, для этого привода, не был вызван, по крайней мере, один раз. Это предохраняет инициализацию на первом сканировании USS_Drive_Control от перезаписи незаконченного запроса на чтение или запись параметра, так как это инициализирует привод как новую запись. Чтобы устранить эту ошибку, вызовите USS_Drive_Control для этого номера привода.
80Ax-80Fx	Определенные ошибки возвращены из FB для PtP-коммуникаций, вызванных USS-библиотекой. - Эти значения кодов ошибки не изменяются USS-библиотекой и определены в описаниях PtP-инструкций.

¹ В дополнение к упомянутым выше ошибкам USS- инструкции ошибки могут быть возвращены из основных коммуникационных PtP-инструкций (стр. 877).

Для нескольких кодов параметра STATUS дополнительная информация предоставлена в экземплярном DB для USS_Drive_Control в переменной "USS_Extended_Error". Для шестнадцатеричных кодов параметра STATUS 8180, 8184, 8187 и 818B, USS_Extended_Error содержит номер привода, где произошла ошибка связи. Для шестнадцатеричного кода параметра STATUS 818C, USS_Extended_Error содержит код ошибки привода, возвращенный из привода при использовании инструкции USS_Read_Param или USS_Write_Param.

Пример: Оповещение о коммуникационных ошибках

О коммуникационных ошибках (STATUS = 16#818B) сообщает только инструкция USS_Port_Scan, но не инструкция USS_Drive_Control. Например, если сеть должным образом не завершена, то для привода возможно перейти в RUN, но инструкция USS_Drive_Control покажет все "0" для выходных параметров. В этом случае Вы можете обнаружить коммуникационную ошибку только в инструкции USS_Port_Scan. Так как эта ошибка видима только на один скан, Вы должны будете добавить некоторую логику захвата, как проиллюстрировано в следующем примере. В этом примере, когда бит ошибки инструкции USS_Port_Scan находится в TRUE, тогда значения STATUS и USS_Extended_Error сохраняются в M области памяти. Номер привода помещается в переменную USS_Extended_Error, когда значение кода в параметре STATUS равно шестнадцатеричным 8180, 8184, 8187 или 818B.



Сегмент 1: Значения состояния порта "PortStatus" и кода расширенной ошибки "USS_Drive_Control_DB". USS_Extended_Error допустимы только для одного скана программы. Значения должны быть захвачены для дальнейшей обработки.

Сегмент 2: Контакт "PortError" активирует сохранение значения "PortStatus" в "LastPortStatus" и "USS_Drive_Control_DB". USS_Extended_Error в "LastExtError".

Доступ для чтения и записи к внутренним параметрам привода

USS-приводы поддерживают доступ к внутренним параметрам для чтения и записи. Эта функция позволяет дистанционное управление и конфигурирование привода. Операции доступа к параметрам привода могут перестать работать из-за ошибок, таких как выход значения из диапазона или недопустимых запросов на текущий режим привода. Привод генерирует значение кода ошибки, которое возвращается в переменной "USS_Extended_Error". Это значение кода ошибки допустимо только для последнего выполнения инструкции USS_Read_Param или USS_Write_Param. Код ошибки привода помещен в переменную USS_Extended_Error, когда значением кода параметра STATUS является шестнадцатеричное 818C. Значение кода ошибки USS_Extended_Error зависит от модели привода. Смотри руководство на привод для получения описания кодов расширенной ошибки для операций чтения и записи параметров.

13.4.6. Общие требования USS по настройке привода

Общие требования USS по настройке привода состоят из следующих пунктов:

- Приводы должны быть настроены для использования 4 PKW слов.
- Приводы могут быть сконфигурированы для 2, 4, 6 или 8 PZD слов.
- Количество PZD слов в приводе должно соответствовать входу PZD_LEN в инструкции USS_Drive_Control для данного привода.
- Скорость обмена во всех приводах должна соответствовать входу BAUD в инструкции USS_Port_Scan.
- Привод должен быть настроен для дистанционного управления.
- Привод должен быть настроен для установки частоты по USS на COM канале.
- Адрес привода должен быть установлен в диапазоне 1 - 16 и соответствовать входу DRIVE в блоке USS_Drive_Control для данного привода.
- Управление направлением привода должно быть настроено для использования полярности установки привода.
- Сеть RS485 должна быть завершена должным образом.

13.4.7. Пример: Общее USS-подключение и настройка привода

Подключение привода MicroMaster

Эта информация о приводах MicroMaster от Сименс предоставлена в качестве примера. Для других приводов обратитесь к руководству на привод для инструкций по установке.

Чтобы выполнить соединение с приводом MicroMaster Серии 4 (MM4), вставьте концы кабеля RS485 в два зажимных, безвинтовых терминала предусмотренных для работы по USS. Стандартный кабель PROFIBUS и соединители могут использоваться для подключения S7-1200.

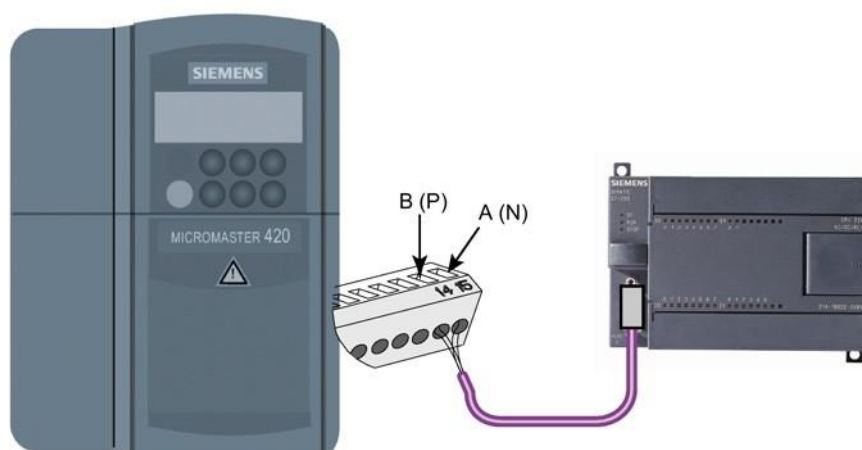


ПРЕДУПРЕЖДЕНИЕ

Взаимосвязанное оборудование с различными опорными потенциалами может стать причиной протекания нежелательных токов через соединительный кабель

Эти нежелательные токи могут вызвать коммуникационные ошибки или повредить оборудование. Убедитесь, что все оборудование, которое Вы собираетесь соединить с коммуникационным кабелем или совместно использует общий опорный потенциал или изолировано, чтобы предотвратить нежелательные электрические токи. Экран должен быть присоединен к корпусной земле или контакту 1 на 9-контактном соединителе. Рекомендуется, чтобы Вы подключили клемму 2 - 0 В на приводе MicroMaster к корпусной земле.

Два провода на противоположном конце кабеля RS485 должны быть вставлены в клеммные колодки привода MM4. Чтобы выполнить кабельное соединение на приводе MM4, снимите крышку(-и) привода, чтобы получить доступ к клеммным колодкам. Смотрите руководство пользователя на MM4 для получения дополнительной информации о том, как снять крышку(-и) Вашего определенного привода.



Соединения клеммной колодки имеют цифровую маркировку. Используя соединитель PROFIBUS на стороне S7-1200, соедините терминал A кабеля с терминалом 15 привода (для MM420) или терминалом 30 привода (MM440). Соедините терминал кабельного соединителя B (P) A (N) с терминалом 14 привода (MM420) или терминалом 29 привода (MM440).

Если S7-1200 является окончечным узлом в сети, или для соединения "точка-к-точке" необходимо использовать терминалы A1 и B1 (не A2 и B2) соединителя, так как они позволяют выполнить завершение (например, с использованием DP-соединителя типа 6ES7 972 - 0BA40 - 0X40).

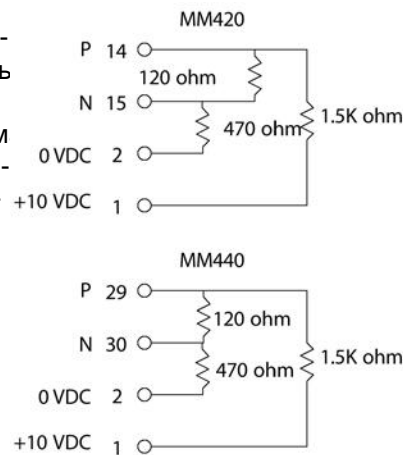
⚠ ПРЕДУПРЕЖДЕНИЕ

Установите крышки привода должным образом перед подачей питания

Удостоверьтесь, что крышки привода установлены должным образом перед подачей питания на блок.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Если привод сконфигурирован как оконечный узел в сети, то нагрузочные резисторы и резисторы смещения должны также быть присоединены к надлежащим клеммным соединениям. Эта схема показывает примеры подключений привода MM4, необходимых для завершения и смещения.



Настройка привода MM4

Прежде чем Вы подключите привод к S7-1200, Вы должны гарантировать, что у привода есть следующие системные параметры. Используйте клавиатуру на приводе для установки параметров:

1. Сбросьте привод на заводские настройки (по желанию).	P0010=30 P0970=1
Если Вы пропускаете шаг 1, то убедитесь, что эти параметры установлены на указанные значения.	USS PZD length = P2012 Index 0=(2, 4, 6 или 8) USS PKW length = P2013 Index 0=4
2. Разрешите доступ для чтения-записи всех параметров (Expert mode).	P0003=3
3. Проверьте параметры мотора для Вашего привода. Параметры настройки изменяются согласно используемому двигателю (-ям). Чтобы установить параметры P304, P305, P307, P310 и P311, Вы должны сначала установить параметр P010 на 1 (быстрый режим ввода в эксплуатацию). Когда Вы завершили настройку параметров, установите параметр P010 на 0. Параметры P304, P305, P307, P310 и P311 могут быть изменены только в режиме быстрого ввода в эксплуатацию.	P0304 = Номин.напряжение двигателя (В) P0305 = Номин. ток двигателя (А) P0307 = Номин. мощность двигателя (Вт) P0310 = Номин. частота двигателя (Гц) P0311 = Номин. обороты двигателя
4. Установите локальный / дистанционный режим управления.	P0700 Index 0=5
5. Выберите уставку частоты по USS на COM канале.	P1000 Index 0=5
6. Время набора оборотов (по желанию) Это время в секундах, которые необходимо двигателю, чтобы ускориться до максимальной частоты.	P1120=(от 0 до 650.00)
7. Время снижения оборотов (по желанию) Это время в секундах, которые необходимо двигателю, чтобы замедлиться до полной остановки.	P1121=(от 0 до 650.00)
8. Установите опорную частоту последовательного канала:	P2000=(от 1 до 650 Гц)
9. Установите нормализацию USS:	P2009 Index 0=0

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

10. Установите скорость последовательного интерфейса RS485:	P2010 Index 0= 4 (2400 бод) 5 (4800 бод) 6 (9600 бод) 7 (19200 бод) 8 (38400 бод) 9 (57600 бод) 12 (115200 бод)
11. Введите адрес ведомого устройства. Каждый привод (максимум 31) может управляться по шине.	P2011 Index 0=(от 0 до 31)
12. Установите тайм-аут последовательного канала. Это максимально допустимый период между двумя входящими телеграммами данных. Эта функция используется, чтобы выключить инвертор в случае коммуникационного отказа. Синхронизация запускается после того, как была принята допустимая телеграмма данных. Если последующая телеграмма данных не будет получена за указанный период времени, то инвертор отключится и выведет на экран код ошибки F0070. Обнуление значения выключает управление.	P2014 Index 0=(от 0 до 65,535 мс) 0=таймаут запрещен
13. Передайте данные из RAM в EEPROM:	P0971=1 (стандартная передача) Сохранить изменения настроек параметров в EEPROM.

13.5 Коммуникации по Modbus

13.5.1 Обзор Modbus RTU и TCP коммуникационных инструкций V13

Коды функций Modbus

- ЦПУ, действующий в качестве ведущего устройства Modbus RTU (или клиента Modbus TCP), может читать/записывать как данные, так и состояния ввода-вывода в удаленном ведомом устройстве Modbus RTU (или сервере Modbus TCP). Удаленные данные могут быть считаны и затем обработаны в Вашей программной логике.
- ЦПУ, действующий в качестве ведомого устройства Modbus RTU (или сервера Modbus TCP), позволяет контрольному устройству читать/записывать как данные, так и состояния ввода-вывода в памяти ЦПУ. Ведущее RTU устройство (или клиент Modbus TCP) может записывать новые значения в память ЦПУ ведомого устройства/сервера, которая доступна для Вашей программной логики.

ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник может физически получить доступ к Вашим сетям, то он может считать и записать данные.

TIA Portal, ЦПУ и устройства HMI (кроме HMI использующих GET/PUT), используют безопасный обмен, который защищает от атак повторением и "человек посередине". Как только коммуникация разрешена, обмен заверенными сообщениями осуществляется открытым текстом, который позволяет злоумышленнику считывать данные, но защищает от несанкционированной записи данных. TIA Portal, не коммуникационный процесс, шифрует данные ноу-хау защищенных блоков.

У всех других способов коммуникаций (обмен ввода-вывода через PROFIBUS, PROFINET, ASI или другую шину ввода-вывода, GET/PUT, Т-блок и коммуникационные модули (CM)) нет средств защиты. Вы должны защитить эти формы коммуникаций, ограничив физический доступ. Если злоумышленник может физически получить доступ к Вашим сетям, использующим эти формы коммуникаций, то он сможет считать и записать данные.

Для получения информации о безопасности и рекомендаций, смотри "Рабочие инструкции для промышленной безопасности"

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте обслуживания поддержки от西门子公司.

Таблица 13- 49 Функции чтения данных: Прочитать удаленный ввод-вывод и программные данные

Коды функций Modbus	Функции чтения ведомого устройства (сервера) - стандартная адресация
01	Читать выходные биты: от 1 до 2000 битов на запрос
02	Читать входные биты: от 1 до 2000 битов на запрос
03	Читать регистры хранения: от 1 до 125 слов на запрос
04	Читать входные слова: от 1 до 125 слов на запрос

Таблица 13- 50 Функции записи данных: Записать удаленный ввод-вывод и изменить программные данные

Коды функций Modbus	Функции записи ведомого устройства (сервера) - стандартная адресация
05	Запись одного выходного бита: 1 бит на запрос
06	Запись одного регистра хранения: 1 слово на запрос
15	Запись одного или более выходных битов: от 1 до 1968 битов на запрос
16	Запись одного или более регистров хранения: от 1 до 123 слов на запрос

- Коды Modbus функций 08 и 11 предоставляют диагностическую информацию о коммуникациях ведомого устройства.
- Коды Modbus функции 0 рассылает сообщение всем ведомым устройствам (без ответа ведомого). Функция широковещательной передачи недоступна для Modbus TCP, так как коммуникация основана на установлении соединения.

Таблица 13- 51 Адреса сетевых станций Modbus

Станция	Адрес	
RTU станция	Стандартный адрес станции	от 1 до 247
	Расширенный адрес станции	от 1 до 65535
TCP станция	Адрес станции	IP-адрес и номер порта

Адреса памяти Modbus

Фактическое количество доступных адресов памяти Modbus зависит от модели ЦПУ, объема рабочей памяти и процента использования памяти ЦПУ другими программными данными. Таблица ниже приводит номинальное значение диапазона адресов.

Таблица 13- 52 Адреса памяти Modbus

Станция	Диапазон адреса	
RTU станция	Стандартный адрес памяти	10K
	Расширенный адрес памяти	64K
TCP станция	Стандартный адрес памяти	10K

Коммуникации Modbus RTU

Modbus RTU (Удаленный терминал) является стандартным протоколом сетевых коммуникаций, который использует RS232 или RS485 электрическое подключение для последовательной передачи данных между сетевыми Modbus устройствами. Вы можете добавить PtP сетевые порты (точка-к-точке) в ЦПУ с помощью RS232 или RS485 CM или RS485 CB.

Modbus RTU использует сеть ведущий/ведомый, где все коммуникации инициируются единственным ведущим устройством, а ведомые устройства могут только ответить на запрос ведущего устройства. Ведущее устройство отправляет запрос одному ведомому адресу и только этот ведомый адрес отвечает на команду.

Коммуникации Modbus TCP

Modbus TCP (протокол управления передачей) является стандартным протоколом сетевых коммуникаций, который использует соединитель PROFINET в ЦПУ для TCP/IP коммуникаций. Никакой дополнительный коммуникационный аппаратный модуль не требуется.

Modbus TCP использует соединения Open User Communications (OUC) в качестве канала связи Modbus. Несколько соединений клиент-сервер могут существовать, в дополнение к соединению между STEP 7 и ЦПУ. Смешанные соединения клиента и сервера поддерживаются до максимального количества соединений, допустимых для данной модели ЦПУ (стр. 613).

Каждое соединение MB_SERVER должно использовать уникальный экземплярный DB и номер IP порта. Поддерживается только 1 соединение на IP порт. Каждый MB_SERVER (с его уникальным экземплярным DB и IP портом) должен быть выполнен индивидуально для каждого соединения.

Клиент Modbus TCP (ведущее устройство) должен управлять соединением клиент-сервер с помощью параметра DISCONNECT. Базовые действия Modbus клиента показаны ниже.

1. Инициировать соединение с определенным сервером (ведомое устройство) по IP-адресу и номеру IP-порта
2. Инициировать клиентскую передачу Modbus сообщений и получить ответы сервера
3. При необходимости инициировать разъединение клиента и сервера, чтобы позволить соединение с другим сервером.

Modbus RTU инструкции в Вашей программе

- **Modbus_Comm_Load:** Одно выполнение Modbus_Comm_Load используется, чтобы установить такие параметры PtP порта как скорость обмена, четность и управление потоком. После того, как порт CPU сконфигурирован для протокола Modbus RTU, он может использоваться только инструкциями Modbus_Master или Modbus_Slave .
- **Modbus_Master:** инструкция Modbus_Master разрешает ЦПУ действовать как ведущее устройство Modbus RTU и выполнять обмен с одним или более ведомыми устройствами Modbus.
- **Modbus_Slave:** инструкция Modbus_Slave позволяет ЦПУ действовать как ведомое устройство Modbus RTU и выполнять обмен с ведущим устройством Modbus.

Modbus TCP инструкции в Вашей программе

- **MB_CLIENT:** Выполняет TCP соединение клиент-сервер, отправляет сообщение с командой, принимает ответ и управляет разъединением от сервера
- **MB_SERVER:** Соединяет с Modbus TCP клиентом по запросу, получает Modbus сообщение и отправляет ответ

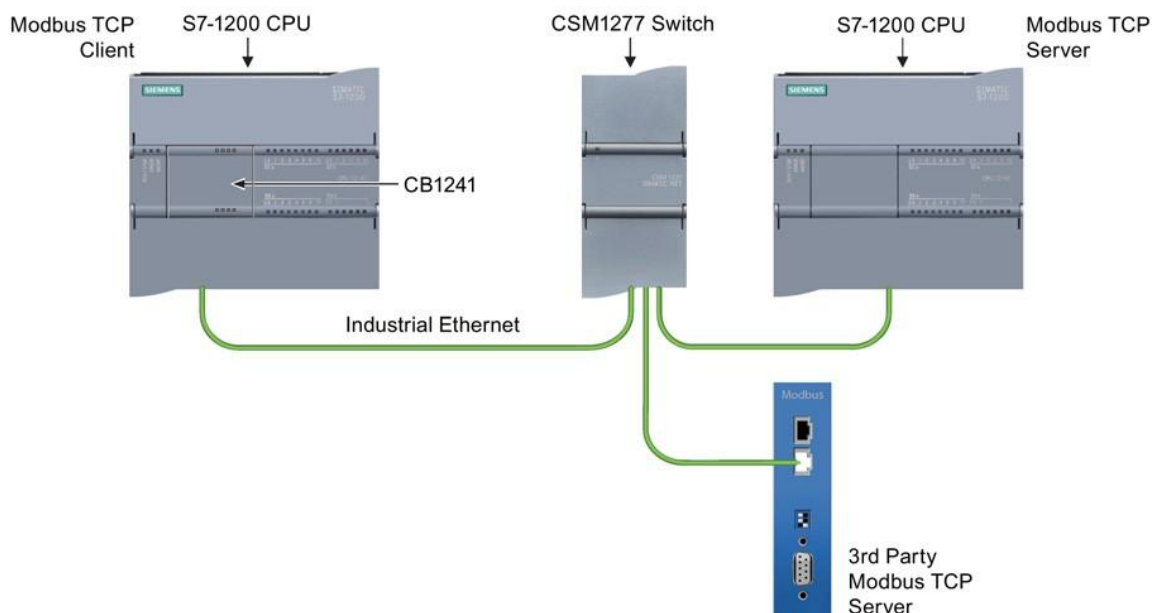
13.5.2 Modbus TCP

13.5.2.1. Обзор

Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможности Modbus TCP для использования T-block инструкций.

По этой причине S7-1200 поддерживает два набора PtP инструкций:

- Унаследованные Modbus TCP инструкции (стр. 1009): Эти Modbus TCP инструкции существовали до версии V4.0 для S7-1200.
- Modbus TCP инструкции (стр. 935): Эти Modbus TCP инструкции обеспечивают полную функциональность унаследованных инструкций.



13.5.2.2. Выбор версии Modbus TCP инструкций

Существует две версии Modbus TCP инструкций доступных в STEP 7:

- Версия 3.0 была изначально доступна в STEP 7 Basic/Professional V13.
- Версия 3.1 доступна в STEP 7 Basic/Professional V13, SP1.

Для совместимости и простоты миграции, Вы можете выбирать какую версию инструкции вставлять в Вашу пользовательскую программу.

Не используйте обе версии инструкции 3.0 и 3.1 в одной программе ЦПУ. Ваши программные Modbus TCP инструкции должны иметь один и тот же старший номер версии (1.x, 2.y или V.z). Отдельные инструкции в рамках группы старшей версии могут иметь различные младшие версии (1.x).



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.



Чтобы изменить версию Modbus TCP инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить Modbus TCP инструкцию в Вашу программу, новый экземпляр FB создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию Modbus TCP инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите Modbus TCP экземпляр FB в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии Modbus TCP инструкции.

13.5.2.3. Modbus TCP инструкции

Инструкция MB_CLIENT (Выполнять обмен, используя PROFINET, как Modbus TCP клиент)

Таблица 13- 53 Инструкция MB_CLIENT

LAD / FBD	SCL	Описание
	<pre>"MB_CLIENT_DB" (REQ:=_bool_in_, DISCON- NECT:=_bool_in_, MB_MODE:=_usint_in_, MB_DATA_ADDR:=_uint_in_, MB_DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_DATA_PTR:=_variant_inout_, CONNECT:=_variant_inout_);</pre>	MB_CLIENT выполняет обмен как Modbus TCP клиент через порт PROFINET на ЦПУ S7-1200. Никакой дополнительный коммуникационный аппаратный модуль не требуется. MB_CLIENT может выполнить соединение клиент-сервер, отправить запрос функции Modbus, получить ответ и управлять разъединением от сервера Modbus TCP.

Таблица 13- 54 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	In	Bool	FALSE = Нет коммуникационного запроса Modbus TRUE = Запрос на обмен с сервером Modbus TCP
DISCONNECT	IN	Bool	Параметр DISCONNECT позволяет Вашей программе управлять подключением и отключением к серверу Modbus. Если DISCONNECT = 0 и соединение не существует, то MB_CLIENT пытается выполнить соединение с назначенным IP-адресом и номером порта. Если DISCONNECT = 1 и соединение существует, то предпринимается попытка отключения. Пока этот вход активирован, никакая другая операция не будет предпринята.
MB_MODE	IN	USInt	Выбор режима: Назначает тип запроса (чтение, запись или диагностика). Смотрите, расположенную ниже таблицу функций Modbus для выяснения деталей.
MB_DATA_ADDR	IN	UDInt	Начальный адрес Modbus: Назначает начальный адрес данных, к которым получит доступ MB_CLIENT. Смотрите последующую таблицу функций Modbus для допустимых адресов.
MB_DATA_LEN	IN	UInt	Длина Modbus данных : Назначает число битов или слов, к которым осуществляет доступ в этом запросе. Смотрите следующую таблицу функций Modbus для допустимых длин.
MB_DATA_PTR	IN_OUT	Variant	Указатель на регистр данных Modbus: регистр помещает в буфер данные, идущие в или приходящие от сервера Modbus. Указатель должен назначать стандартный глобальный DB или адрес в M памяти.
CONNECT	IN_OUT	Variant	Ссылка на структуру блока данных, которая содержит параметры соединения в системном типе данных "TCON_IP_v4".
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки.
BUSY	OUT	Bool	0 - Никакой операции MB_CLIENT в работе 1 - выполняется операция MB_CLIENT
ERROR	OUT	Bool	ERROR устанавливается в TRUE на один скан после того, как выполнение MB_CLIENT завершилось с ошибкой. Код ошибки в параметре STATUS допустим только в течение одного цикла, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Примечание**Требования к версии встроенного ПО для ЦПУ**

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют встроенного ПО версии V4.1 или более поздней.

Параметр REQ

FALSE = Нет коммуникационного запроса Modbus

TRUE = Запрос на обмен с сервером Modbus TCP

Если никакой из экземпляров MB_CLIENT не активен и параметр DISCONNECT=0, когда REQ=1, то запускается новый запрос Modbus. Если соединение еще не установлено, то выполняется новое соединение.

Если тот же экземпляр MB_CLIENT будет выполняться снова с DISCONNECT=0 и REQ=1 перед завершением текущего запроса, то последующая передача Modbus не будет выполнена. Однако, как только текущий запрос завершен, новый запрос может быть обработан, если MB_CLIENT выполняется с REQ=1.

Когда текущий коммуникационный запрос MB_CLIENT завершен, бит DONE устанавливается в TRUE на один цикл. Бит DONE может использоваться в качестве временной селектор для упорядочения множественных запросов MB_CLIENT.

Примечание

Согласованность входных данных во время обработки MB_CLIENT

Как только Modbus клиент инициирует Modbus операцию, все входные состояния сохраняются внутри и затем сравниваются при каждом последующем вызове. Сравнение используется, чтобы определить, был ли этот определенный вызов инициатором активного клиентского запроса. Больше чем один вызов MB_CLIENT может быть выполнен, используя общий экземплярный DB.

Важно, чтобы входы не были изменены в течение промежутка времени, когда операция MB_CLIENT активно обрабатывается. Если это правило не соблюдается, то MB_CLIENT не может определить активный экземпляр..

Параметры MB_MODE и MB_DATA_ADDR выбирают коммуникационные функции Modbus

Инструкция MB_CLIENT использует вход MB_MODE, а не функциональный код. MB_DATA_ADDR назначает стартовый адрес удаленных данных Modbus.

Комбинация MB_MODE и MB_DATA_ADDR определяет функциональный код, который используется в фактическом сообщении Modbus. Следующая таблица показывает соответствие между параметром MB_MODE, MB_DATA_ADDR и функцией Modbus.

Таблица 13- 55 Modbus функции

MB_MODE	MB_DATA_ADDR	Длина данных	Активируемый код функции Modbus	Операция и данные
0	От 1 до 9999	От 1 до 2000	01	Читать выходные биты: от 1 до 2000 битов за запрос
0	От 10001 до 19999	От 1 до 2000	02	Читать входные биты: от 1 до 2000 битов за запрос
0	От 40001 до 49999 или от 400001 до 465535	От 1 до 125	03	Читать регистры хранения: от 1 до 125 слов за запрос
0	От 30001 до 39999	От 1 до 125	04	Читать входные слова: от 1 до 125 слов за запрос
1	От 1 до 9999	1	05	Записать один выходной бит: один бит на запрос
1	От 40001 до 49999 или от 400001 до 465535	1	06	Записать один регистр хранения: 1 слово за запрос
1	От 1 до 9999	От 2 до 1968	15	Записать несколько выходных битов: от 2 до 1968 битов за запрос
1	От 40001 до 49999 или от 400001 до 465535	От 2 до 123	16	Записать несколько регистров хранения: от 2 до 123 слов за запрос
2	От 1 до 9999	От 1 до 1968	15	Записать один или более выходных битов: от 1 до 1968 битов за запрос
2	От 40001 до 49999 или от 400001 до 465535	От 1 до 123	16	Записать один или более регистров хранения: от 1 до 123 слов за запрос
11		0	11	Читать коммуникационное слово состояния сервера и счетчик событий. Слово состояния указывает занятость (0 = не занято, 0xFFFF = занято). Счетчик событий постепенно увеличивается для каждого успешного завершения сообщения. Как MB_DATA_ADDR, так и MB_DATA_LEN параметры MB_CLIENT игнорируются для этой функции.
80		1	08	Проверить состояние сервера с помощью диагностического кода 0x0000 (Петлевой тест, сервер отвечает эхом) 1 слово за запрос

MB_MODE	MB_DATA_ADDR	Длина данных	Активируемый код функции Modbus	Операция и данные
81		1	08	Сбросить счетчик событий сервера с помощью диагностического кода 0x000A 1 слово за запрос
от 3 до 10, от 12 до 79, от 82 до 255				Зарезервировано

Примечание

MB_DATA_PTR назначает буфер для хранения данных чтения/записи в Modbus TCP сервер

Буфер данных может быть расположен в стандартном глобальном DB или M ячейке памяти. Для буфера в M памяти используйте формат указателя Any. Это формат R# "Адрес бита" "Тип данных" "Длина", примером будет R#M1000.0 WORD 500.

Параметр MB_DATA_PTR назначает коммуникационный буфер

- Коммуникационные функции MB_CLIENT:
 - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 00001 до 09999)
 - Читать 1-битовые данные из адресов сервера Modbus (от 10001 до 19999)
 - Читать 16-разрядные данные слов из адресов сервера Modbus (от 30001 до 39999) и (от 40001 до 49999)
 - Записывать 16-разрядные данные слов в адреса сервера Modbus (от 40001 до 49999)
- Данные размерностью слово или бит передаются в/из DB или M буфера памяти, назначенного MB_DATA_PTR.
- Если DB назначен, как буфер MB_DATA_PTR, то Вы должны присвоить типы данных всем элементам данных DB.
 - 1-битовый тип данных Bool представляет один Modbus адрес бита
 - 16-битовые типы данных отдельного слова такие как WORD, UInt и Int представляют один Modbus адрес слова
 - 32-разрядные типы данных двойного слова такие как DWORD, DInt и Real представляют два Modbus адреса слова

- MB_DATA_PTR могут быть назначены сложные элементы DB, такие как
 - Массивы
 - Именованные структуры, где каждый элемент уникален.
 - Именованные сложные структуры, где каждый элемент имеет уникальное имя и 16- или 32-битный типа данных.
- Не требуется, чтобы области данных MB_DATA_PTR были в том же глобальном блоке данных (или M области памяти). Вы можете назначить один блок данных для чтения по Modbus, другой блок данных для записи по Modbus или один блок данных для каждого MB_CLIENT.

Параметр CONNECT назначает данные, используемые для установления PROFINET соединения

Вы должны использовать глобальный блок данных и сохранить требуемые данные соединения, прежде чем Вы сможете сослаться на этот DB в параметре CONNECT.

1. Создайте новый глобальный DB или используйте существующий глобальный DB, чтобы сохранить данные CONNECT. Вы можете использовать один DB, чтобы сохранить несколько структур данных TCON_IP_v4. Каждое подключение клиента или сервера Modbus TCP использует структуру данных TCON_IP_v4. Вы ссылаетесь на данные соединения в параметре CONNECT.
2. Называйте DB и статическую переменную смысловым именем. Например, назовите блок данных "Modbus connections", а статическую переменную "TCPActive_1" (для Modbus TCP подключения клиента 1).
3. В редакторе DB назначьте системный тип данных "TCON_IP_v4" в столбце Data Type, например, для статической переменной "TCPActive_1".
4. Разверните структуру TCON_IP_v4, таким образом Вы можете изменить параметры соединения, как показано на следующем изображении.
5. Измените данные в структуре TCON_IP_v4 для соединения MB_CLIENT.
6. Введите ссылку на структуру из DB для параметра CONNECT в MB_CLIENT. К примеру, это могло бы выглядеть как Modbus connections".TCPActive_1.

Modbus connections				
	Name	Data type	Start value	Comment
1	Static			
2	TCPActive_1	TCON_IP_v4		
3	InterfaceId	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connection: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	True	active/passive connection establishment
7	RemoteAddress	IP_V4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	502	remote UDP/TCP port number
14	LocalPort	UInt	0	local UDP/TCP port number

Измените TCP_IP_v4 DB данные для каждого соединения MB_CLIENT

- **InterfaceID:** Используя окно Device configuration, щелкните по изображению PROFINET-порта для ЦПУ. Затем щелкните по вкладке свойств General и примените Аппаратный идентификатор, который Вы видите там.
- **ID:** Введите ID соединения от 1 до 4095. Modbus TCP выполняется, используя лежащие в основе TCON, TDISCON, TSEND и TRCV инструкции для OUC (Открытые Пользовательские Коммуникации).
- **ConnectionType:** Для TCP/IP, используйте по умолчанию 16#0B (десятичное 11).
- **ActiveEstablished:** Это значение, должно равняться 1 или TRUE. В этом случае соединение активно и MB_CLIENT иницирует обмен по Modbus.
- **RemoteAddress:** Введите IP-адрес целевого сервера Modbus TCP в четыре элемента массива ADDR. Например, введите 192.168.2.241, как на предыдущем изображении.
- **RemotePort:** Значение по умолчанию 502. Это число представляет номер IP порта сервера Modbus, к которому MB_CLIENT пытается подключиться и выполнить обмен. Некоторые Modbus серверы сторонних производителей требуют, чтобы Вы использовали другой номер порта.
- **LocalPort:** Это значение должно равняться 0, для MB_CLIENT соединения.

Подключение нескольких клиентов

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для Открытых Пользовательских Коммуникаций, разрешенных в ПЛК. Общее количество соединений для ПЛК, включая клиенты и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений для Открытых Пользовательских Коммуникаций (стр. 613).

Отдельные параллельные клиентские соединения должны следовать данным правилам:

- Каждое соединение MB_CLIENT должно использовать уникальный экземплярный DB
- Каждое соединение MB_CLIENT должно назначать уникальный IP-адрес сервера
- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения
- Уникальные номера портов IP могут или не могут потребоваться в зависимости от конфигурации сервера

13.5 Коммуникации по Modbus

Различные ID соединения должны использоваться с каждым экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13- 56 Экземплярный блок данных MB_CLIENT: Пользовательские доступные статические переменные

Переменная	Тип данных	По умолчанию	Описание
Blocked_Proc_Timeout	Real	3.0	Временной интервал (в секундах) ожидания заблокированного экземпляра клиента Modbus, прежде, чем удалить этот экземпляр, как являющийся ACTIVE. Это может произойти, например, когда был выставлен клиентский запрос и впоследствии приложение прекращает выполнять клиентскую функцию до полного завершения запроса. Максимальный предел для S7-1200 составляет 55 секунд.
MB_Unit_ID	Word	255	Идентификатор модуля Modbus: Сервер Modbus TCP адресуется с помощью IP-адреса. В результате параметр MB_UNIT_ID не используется для адресации Modbus TCP. Параметр MB_UNIT_ID соответствует адресу ведомого устройства в протоколе Modbus RTU. Если сервер Modbus TCP используется в качестве шлюза к протоколу Modbus RTU, MB_UNIT_ID может использоваться для идентификации ведомого устройства, подключенного к последовательной сети. MB_UNIT_ID мог бы использоваться для передачи запроса корректному адресу ведомого устройства Modbus RTU. Некоторые устройства Modbus TCP могут потребовать, чтобы параметр MB_UNIT_ID был в ограниченном диапазоне.
RCV_TIMEOUT	Real	2.0	Время в секундах, в течение которого MB_CLIENT ожидает сервер, чтобы ответить на запрос.
Connected	Bool	0	Указывает на установление соединения с назначенным сервером: 1=подключено, 0=отключено

Таблица 13- 57 Ошибки протокола MB_CLIENT

STATUS (W#16#)	Код отклика клиенту Modbus (B#16#)	Ошибки протокола Modbus
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или доступ вне границ адресного пространства MB_HOLD_REG
8384	03	Ошибка значения данных
8385	03	Диагностический код данных не поддерживается (код функции 08)

Таблица 13- 58 Коды состояния выполнения MB_CLIENT ¹

STATUS (W#16#)	Ошибки параметров MB_CLIENT
7001	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код возвращается только для первого выполнения операции подключения или отключения.
7002	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код будет возвращен для любого последующего выполнения при ожидании завершения операции подключения или отключения
7003	Операция отключения успешно завершена (допустимо только для одного скана ПЛК).

STATUS (W#16#)	Ошибки параметров MB_CLIENT
80C8	Сервер не ответил в назначенное время. MB_CLIENT должен принять ответ, используя ID транзакции, который был первоначально передан в течение назначенного времени, или будет возвращена данная ошибка. Проверьте соединение с серверным устройством Modbus. Эта ошибка возвращается только после того, как были предприняты повторные попытки (если применимо).
8188	Неправильный режим
8189	Неправильный адрес данных
818A	Неправильная длина данных
818B	Недопустимый указатель на область DATA_PTR. Это может быть комбинацией MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Указатель DATA_PTR указывает на область оптимизированного DB (должна быть область стандартного DB или область M памяти)
8200	Порт занят обработкой существующего запроса Modbus.
8380	Принятый кадр Modbus неправильный, или было получено слишком мало байтов.
8387	Назначенный параметр Connection ID отличается от ID, использованного для предыдущих запросов. Может быть только один Connection ID, используемый в каждом экземпляре DB для MB_CLIENT. Этот код также возвращается как внутренняя ошибка, если ID протокола Modbus TCP, полученный от сервера, не 0.
8388	Сервер Modbus возвратил количество данных, которое отличается от затребованного. Этот код применяется только к функциям Modbus 15 или 16.

- ¹ В дополнение к упомянутым выше ошибкам MB_CLIENT ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV).

Инструкция MB_SERVER (выполнять обмен, используя PROFINET, как Modbus TCP сервер)

Таблица 13- 59 Инструкция MB_SERVER

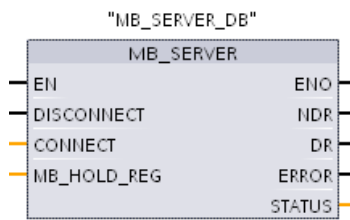
LAD / FBD	SCL	Описание
	<pre>"MB_SERVER_DB"(DISCONNECT:=_bool_in_, CONNECT:=_variant_in_, NDR=>_bool_out_, DR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	MB_SERVER выполняет обмен как сервер Modbus TCP через PROFINET порт ЦПУ S7-1200. Никакой дополнительный аппаратный модуль не требуется. MB_SERVER может утвердить запрос на соединение с клиентом Modbus TCP, принять запрос функции Modbus и отправить ответное сообщение.

Таблица 13- 60 Типы данных для параметров

Параметр и тип		Тип данных	Описание
DISCONNECT	IN	Bool	MB_SERVER пытается выполнить "пассивное" соединение с устройством партнера. Это означает, что сервер пассивно прислушивается к запросу на установление TCP соединения от любого запрашивающего IP-адреса. Если DISCONNECT = 0 и соединение не существует, то пассивное соединение может быть инициировано. Если DISCONNECT = 1 и соединение существует, то инициируется операция отключения. Этот параметр позволяет Вашей программе управлять, когда подтверждать соединение. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT	IN	Variant	Ссылка на структуру блока данных, которая содержит параметры соединения в системном типе данных "TCON_IP_v4".
MB_HOLD_REG	IN_OUT	Variant	Указатель на MB_SERVER Modbus регистр хранения: регистр хранения должен быть либо стандартным глобальным DB, либо адресом М памяти. Эта область памяти используется, чтобы хранить данные, к которым Modbus клиент разрешают доступ, используя регистровые функции: 3 (чтение), 6 (запись) и 16 (запись).
NDR	OUT	Bool	Новые данные готовы: 0 = Нет новых данных, 1 Указывает на то, что новые данные были записаны клиентом Modbus.
DR	OUT	Bool	Чтение данных: 0 = Нет чтения данных, 1 = Указывает на то, что данные были считаны клиентом Modbus.
ERROR	OUT	Bool	ERROR устанавливается в TRUE на один скан после того, как выполнение MB_SERVER завершилось с ошибкой. Код ошибки в параметре STATUS допустим только в течение одного цикла, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Примечание**Требования к версии встроенного ПО для ЦПУ**

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют встроенного ПО версии V4.1 или более поздней.

Параметр CONNECT назначает данные, используемые для установления PROFINET-соединения

Вы должны использовать глобальный блок данных и сохранить требуемые данные соединения, прежде чем Вы сможете сослаться на этот DB в параметре CONNECT.

1. Создайте новый глобальный DB или используйте существующий глобальный DB, чтобы сохранить данные CONNECT. Вы можете использовать один DB, чтобы сохранить несколько структур данных TCON_IP_v4. Каждое подключение клиента или сервера Modbus TCP использует структуру данных TCON_IP_v4. Вы ссылаетесь на данные соединения в параметре CONNECT.
2. Называйте DB и статическую переменную смысловым именем. Например, назовите блок данных "Modbus connections", а статическую переменную "TCPpassive_1" (для Modbus TCP подключения сервера 1).
3. В редакторе DB назначьте системный тип данных "TCON_IP_v4" в столбце Data Type, например, для статической переменной "TCPpassive_1".
4. Разверните структуру TCON_IP_v4, таким образом Вы можете изменить параметры соединения, как показано на следующем изображении.
5. Измените данные в структуре TCON_IP_v4 для соединения MB_SERVER.
6. Введите ссылку на структуру из DB для параметра CONNECT в MB_SERVER. К примеру, это могло бы выглядеть как "Modbus connections". TCPpassive_1.

Modbus connections				
	Name	Data type	Start value	Comment
1	Static			
2	TCFpassive_1	TCON_IP_v4		
3	Interfaceld	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connction: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	False	active/passive connection establishment
7	RemoteAddress	IP_V4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	0	remote UDP/TCP port number
14	LocalPort	UInt	502	local UDP/TCP port number

Измените TCP_IP_v4 DB данные для каждого соединения MB_SERVER

- **InterfacelD:** Используя окно Device configuration, щелкните по изображению PROFINET-порта для ЦПУ. Затем щелкните по вкладке свойств General и примените Аппаратный идентификатор, который Вы видите там.
- **ID:** Введите ID соединения от 1 до 4095. Modbus TCP выполняется, используя лежащие в основе TCON, TDISCON, TSEND и TRCV инструкции для OUC (Открытые Пользовательские Коммуникации). Допустимы до восьми одновременных OUC-соединений.
- **ConnectionType:** Для TCP/IP, используйте по умолчанию 16#0B (десятичное 11).
- **ActiveEstablished:** Это значение, должно равняться 0 или FALSE. В этом случае соединение пассивно и MB_SERVER ожидает коммуникационного запроса от клиента Modbus.
- **RemoteAddress:** Существует две опции.
 - Используйте 0.0.0.0 и MB_CLIENT ответит на запрос по Modbus от любого TCP клиента
 - Введите IP-адрес целевого клиента Modbus TCP, и MB_CLIENT отвечает на запрос, приходящий только от IP-адреса этого клиента. Например, введите 192.168.2.241, как в предыдущем изображении.
- **RemotePort:** Это значение должно равняться 0 для подключения MB_SERVER.
- **LocalPort:** Значение по умолчанию 502. Это число представляет собой номер IP порта клиента Modbus, к которому MB_SERVER пытается подключиться и выполнить обмен. Некоторые клиенты Modbus сторонних производителей требуют другого номера порта.

Modbus и адреса образа процесса

MB_SERVER позволяет входящим кодам функций Modbus (1, 2, 4, 5 и 15) читать/записывать биты/слова непосредственно в образе процесса ввода/вывода. Для кодов функций передачи данных (3, 6 и 16), параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт. Следующая таблица показывает проекцию Modbus адресов в образ процесса в ЦПУ.

Таблица 13- 61 Проекция Modbus адресов в образ процесса

Modbus функции						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес в ЦПУ
01	Считать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Считать биты	Ввод	От 10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Считать слова	Ввод	От 30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Входящие коды функций Modbus сообщений (3, 6, и 16) читают/записывают слова в регистре хранения Modbus, который может располагаться в М памяти или блоке данных. Тип регистра хранения определяется параметром MB_HOLD_REG.

Примечание

Назначение параметра MB_HOLD_REG

Регистр хранения Modbus может находиться в стандартном глобальном DB или ячейке М памяти.

Для регистра хранения Modbus в М памяти используйте формат указателя Any. Это формат R# "Адрес бита" "Тип данных" "Длина", примером будет R#M1000.0 WORD 500.

Следующая таблица показывает примеры проекций Modbus адресов в регистры хранения, используемые для кодов Modbus функций 03 (считать слова), 06 (записать слово) и 16 (записать слова). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом М памяти для каждой из моделей ЦПУ.

Таблица 13- 62 Примеры проекций Modbus адресов в адреса памяти ЦПУ

Modbus адрес	Примеры параметра MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBX0.0 Word 5	"Recipe".ingredient
40001	MW100	DB10.DBW0	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	"Recipe".ingredient[5]

Подключение нескольких серверов

Могут быть созданы несколько подключений серверов. Одиночный ПЛК может установить параллельные соединения с несколькими клиентами Modbus TCP.

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для Открытых Пользовательских Коммуникаций, разрешенных в ПЛК. Общее количество соединений для ПЛК, включая клиенты и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений для Открытых Пользовательских Коммуникаций (стр. 613). Соединения TCP Modbus могут совместно использоваться подключениям типа клиента и сервера.

Отдельные параллельные серверные соединения должны следовать данным правилам:

- Каждое соединение MB_SERVER должно использовать уникальный экземплярный DB
- Каждое соединение MB_SERVER должно назначать уникальный номер IP-порта сервера. Поддерживается только 1 соединение на порт
- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения.
- MB_SERVER нужно вызвать индивидуально для каждого соединения (с его соответствующим экземплярным DB).

13.5 Коммуникации по Modbus

ID соединения должен быть уникальным для каждого отдельного соединения. Отдельный ID соединения должен использоваться с каждым отдельным экземплярным DB. Экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13- 63 Коды диагностических функций Modbus

Коды диагностических функций Modbus для MB_SERVER		
Коды	Подфункция	Описание
08	0x0000	Эхо-тест возврата данных запроса: MB_SERVER отразит клиенту Modbus слово данных, которое принято.
08	0x000A	Сбросить счетчик коммуникаций: MB_SERVER сбросит счетчик коммуникаций, который используется для Modbus функции 11.
11		Получить счетчик коммуникаций: MB_SERVER использует внутренней счетчик коммуникаций для записи числа успешных запросов на чтение и запись по Modbus, отправленных серверу Modbus. Счетчик не увеличивается на единицу по любому запросу на Функцию 8, Функцию 11 или любому запросу, который приводит к ошибке связи. Широковещательная функция недоступна для Modbus TCP, так как только одно клиент-серверное соединение существует в любой момент времени.

Переменные MB_SERVER

Эта таблица показывает общедоступные статические переменные, которые хранятся в экземплярном блоке данных MB_SERVER и могут использоваться в Вашей программе.

Таблица 13- 64 Общедоступные статические переменные MB_SERVER

Переменная	Тип данных	По умолчанию	Описание
HR_Start_Offset	Word	0	Назначает начальный адрес регистра хранения Modbus
Request_Count	Word	0	Количество всех запросов полученных этим сервером.
Server_Message_Count	Word	0	Количество запросов полученных для этого определенного сервера.
Xmt_Rcv_Count	Word	0	Количество передач или приемов, в которых произошли ошибки. Кроме того, увеличивается на единицу, если принято сообщение, которое является недопустимым сообщением Modbus.
Exception_Count	Word	0	Специфические ошибки Modbus, которые требуют возвращенного исключения
Success_Count	Word	0	Количество запросов, полученных для этого определенного сервера, в которых нет ошибок протокола.
Connected	Bool	0	Указывает на состояние соединения с назначенным клиентом: 1=подключено, 0=отключено

Ваша программа может записать данные в HR_Start_Offset и управлять операциями сервера Modbus. Другие переменные могут быть считаны, чтобы контролировать состояние Modbus.

HR_Start_Offset

Адреса регистров хранения Modbus начинаются с 40001. Эти адреса соответствуют началу адресов памяти ПЛК для регистров хранения. Однако Вы можете использовать переменную "HR_Start_Offset", чтобы определить начало адресов регистров хранения Modbus иное, чем 40001.

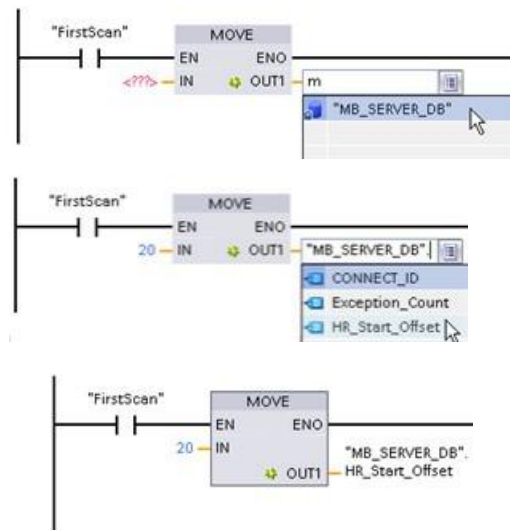
Например, если регистр хранения начинается в MW100 и содержит 100 слов. Смещение 20 определяет начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше чем 40021 или больше, чем 40119 вызовет ошибку адресации.

Таблица 13- 65 Примеры адресации регистров хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Modbus адрес (Слово)	40001	40099
	S7-1200 адрес	MW100	MW298
20	Modbus адрес (Слово)	40021	40119
	S7-1200 адрес	MW100	MW298

HR_Start_Offset представляет собой данные типа word в экземплярном блоке данных MB_SERVER, которые назначают начальный адрес регистра хранения Modbus. Вы можете установить эту общедоступную статическую переменную при помощи выпадающего списка помощника параметра, после того, как MB_SERVER будет помещен в Вашу программу.

Например, после того, как Вы помещаете MB_SERVER в LAD-сегмент, Вы можете перейти к предыдущему сегменту и назначить HR_Start_Offset. Начальный адрес должен быть назначен до выполнения MB_SERVER.



Ввод переменной сервера Modbus, используя имя DB по умолчанию:

1. Установите курсор в поле параметра и введите символ m.
2. Выберите "MB_SERVER_DB" из выпадающего списка имен DB.
3. Выберите "MB_SERVER_DB.HR_Start_Offset" из выпадающего списка переменных DB.

Таблица 13- 66 Коды состояния выполнения MB_SERVER ¹

STATUS (W#16#)	Код ответа серверу Modbus	Ошибки протокола Modbus
7001		MB_SERVER ожидает клиента Modbus, чтобы соединиться с назначенным TCP портом. Этот код возвращается при первом выполнении операции подключения или отключения.
7002		MB_SERVER ожидает клиента Modbus, чтобы соединиться с назначенным TCP портом. Этот код возвращается для любого последующего выполнения при ожидании завершения работы подключения или отключения.
7003		Операция отключения успешно завершилась (Допустимо только для одного скана ПЛК).
8187		Недопустимый указатель на MB_HOLD_REG: область слишком маленькая
818C		Указатель MB_HOLD_REG ссылается на область в оптимизированном DB (область должна быть в стандартном глобальном DB или M памяти) или тайм-аут заблокированного процесса превышает предел в 55 секунд. (Специфический для S7-1200)
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или обращение вне границ адресного пространства MB_HOLD_REG
8384	03	Ошибка значения данных
8385	03	Код диагностики данных не поддерживается (код функции 08)

¹ В дополнение к упомянутым выше ошибкам MB_SERVER ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV).

13.5.2.4. Примеры Modbus TCP

Пример: Несколько TCP соединений для MB_SERVER

У Вас может быть несколько соединений сервера Modbus TCP. Чтобы сделать это, MB_SERVER должен выполняться независимо для каждого соединения. Каждое соединение должно использовать независимые экземпляры DB, ID соединения и IP-порт. S7-1200 позволяет только одно соединение на IP-порт.

Для лучшей производительности MB_SERVER должен выполняться каждый программный цикл для каждого соединения.

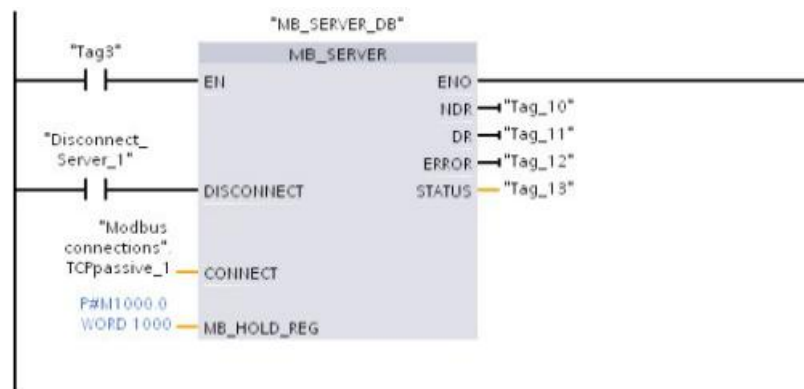
Параметр CONNECT использует системный тип данных TCP_IP_v4. Для примера эти структуры данных находятся в DB с именем "Modbus connections". DB содержит две структуры TCP_IP_v4 "TCPpassive_1" (для соединения 1), и "TCP_passive_2" (для соединения 2). Свойства соединения ID и LocalPort, описанные в комментариях к сегменту, являются элементами данных, сохраненными в структуре данных CONNECT.

Данные TCP_IP_v4 CONNECT также содержат IP-адрес в массиве RemoteAddress ADDR. Назначения IP-адреса в TCPpassive_1 и TCP_passive_2 не влияют на установление соединений TCP сервера, но определяют, каким клиентам Modbus TCP разрешено связаться с каждым MB_SERVER. MB_SERVER пассивно прислушивается к Modbus сообщению клиента и сравнивает IP-адрес входящего сообщения с IP-адресом, сохраненным в соответствующем массиве RemoteAddress ADDR.

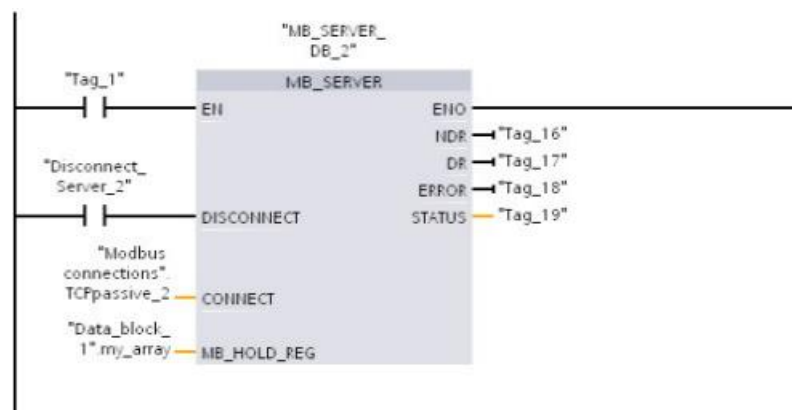
Три разновидности IP-адреса возможны для двух инструкций MB_SERVER:

- **IP address = 0.0.0.0**
Каждый MB_SERVER ответит всем клиентам Modbus TCP, использующим любой IP-адрес.
- **IP address = Один и тот же IP-адрес в TCPpassive_1 и TCPpassive_2**
Оба соединения MB_SERVER отвечают только клиентам Modbus, обращающимся из этого IP-адреса.
- **IP address = Различные IP-адреса в TCP_passive_1 и TCP_passive_2**
Каждый MB_SERVER отвечает только клиентам Modbus, которые обращаются из IP-адреса, сохраненного в их TCP_IP_v4-данных.

Сегмент 1: Подключение #1, Экземплярный DB = "MB_SERVER_DB", в "Modbus connections".TCPpassive_1 (ID = 1 и LocalPort = 502)



Сегмент2: Подключение #2, Экземплярный DB = "MB_SERVER_DB_1", в "Modbus connections".TCPpassive_2 (ID = 2 и LocalPort = 503)



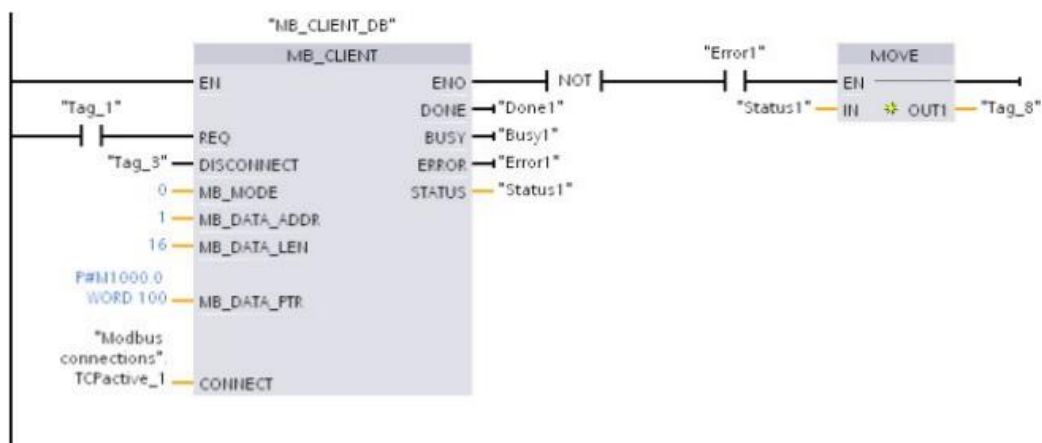
Пример: MB_CLIENT 1: Несколько запросов с общим TCP соединением

Несколько клиентских запросов Modbus могут быть отправлены по тому же соединению. Чтобы сделать это, используйте тот же экземплярный DB , ID соединения и номер порта.

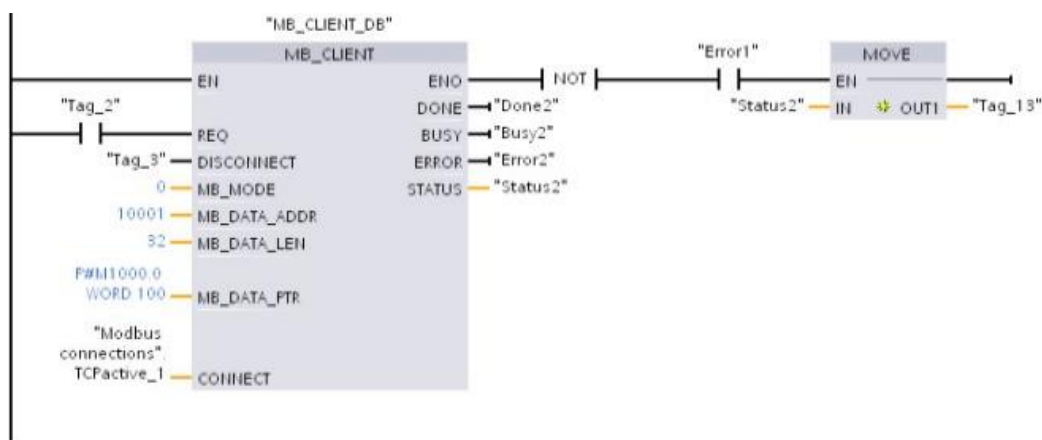
Поскольку оба блока MB_CLIENT используют ту же структуру данных TCON_IP_v4 параметра CONNECT ("Modbus_connections".TCPActive_1), ID соединения, номер порта и IP-адрес идентичны. Данные IP-адреса CONNECT назначают IP-адрес целевого сервера Modbus TCP .

Только один MB_CLIENT может быть активным в любой момент времени. Как только клиент завершает выполнение, следующий клиент может начать выполнение. Ваша программная логика ответственна за последовательность выполнения. Пример демонстрирует обоих клиентов, считывающих удаленные данные из единственного клиента Modbus и передающих данные ЦПУ клиента Modbus (М память, начиная с M1000.0). Возвращенная ошибка захватывается, что является необязательным.

Сегмент 1: Modbus функция 1 - считать 16 выходных битов из сервера Modbus TCP с IP-адресом, назначенным в "Modbus connections".TCPActive_1.



Сегмент 2: Modbus функция 2 - Считать 32 входных бита из сервера Modbus TCP с IP-адресом, назначенным в "Modbus connections".TCPActive_1.



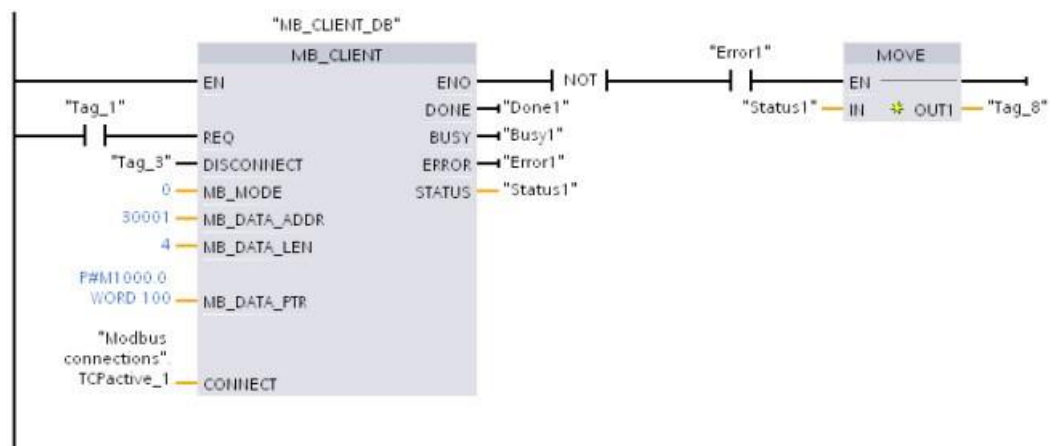
Пример: MB_CLIENT 2: Несколько запросов с различными TCP соединениями

Клиентские запросы Modbus TCP могут быть отправлены по различным соединениям. Чтобы сделать это, должны использоваться различные экземплярные DB и ID соединения.

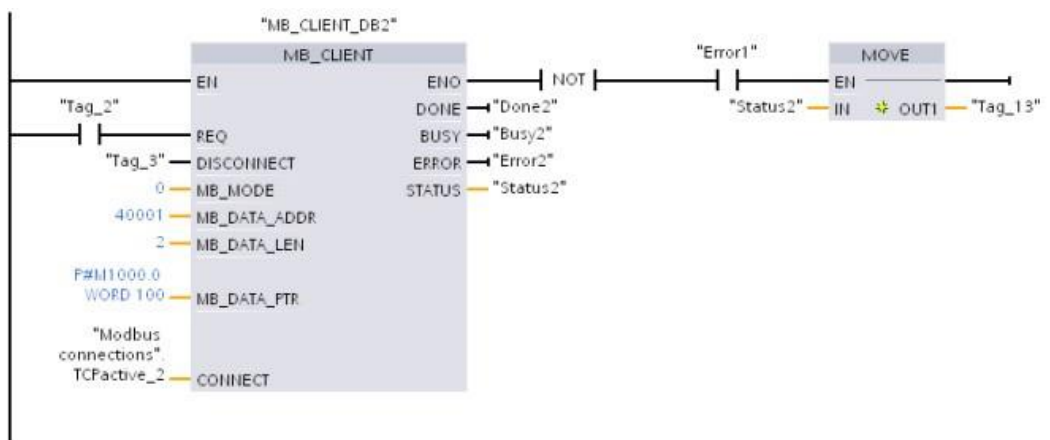
RemotePort (IP-порт) номер должен отличаться, если соединения устанавливаются с тем же сервером Modbus. Если соединения находятся на различных серверах, нет никаких ограничений на номер IP-порта.

Пример демонстрирует двух клиентов Modbus TCP, передающих удаленные данные от двух различных серверов Modbus TCP в ту же локальную область памяти ЦПУ, начиная с адреса M1000.0. Кроме того, возвращенная ошибка захватывается, что является необязательным.

Сегмент 1: Modbus функция 4 - Считать слова образа процесса по вводу из параметра Modbus TCP сервера CONNECT = "Modbus connections".TCPActive_1: ID соединения = 1, RemoteAddress = 192.168.2.241, RemotePort = 502



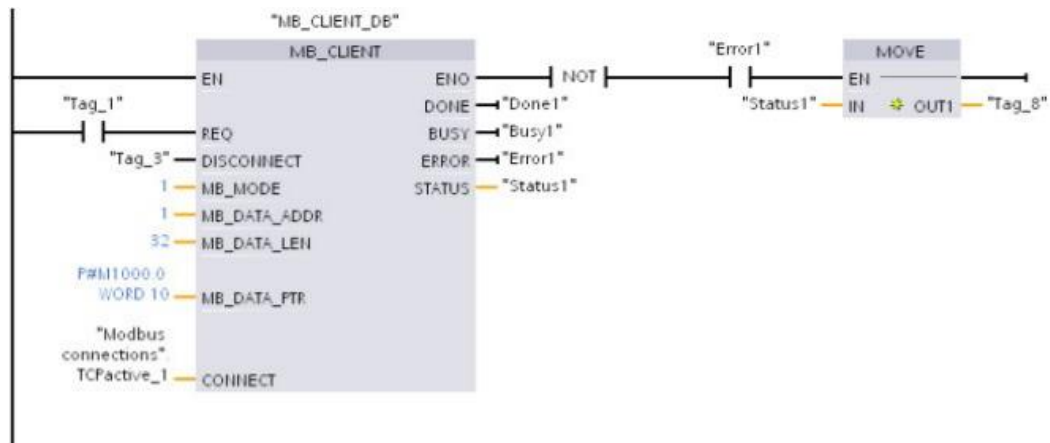
Network 2: Modbus функция 3 – Считать слова регистра хранения из параметра Modbus TCP сервера CONNECT = "Modbus connections".TCPActive_2: ID соединения = 2, RemoteAddress = 192.168.2.242, RemotePort = 502



Пример: MB_CLIENT 3: Запрос записи образа по выводу

Этот пример показывает запрос клиента Modbus, который передает битовые данные из локальной памяти ЦПУ (начиная с M1000.0) в удаленный сервер Modbus TCP.

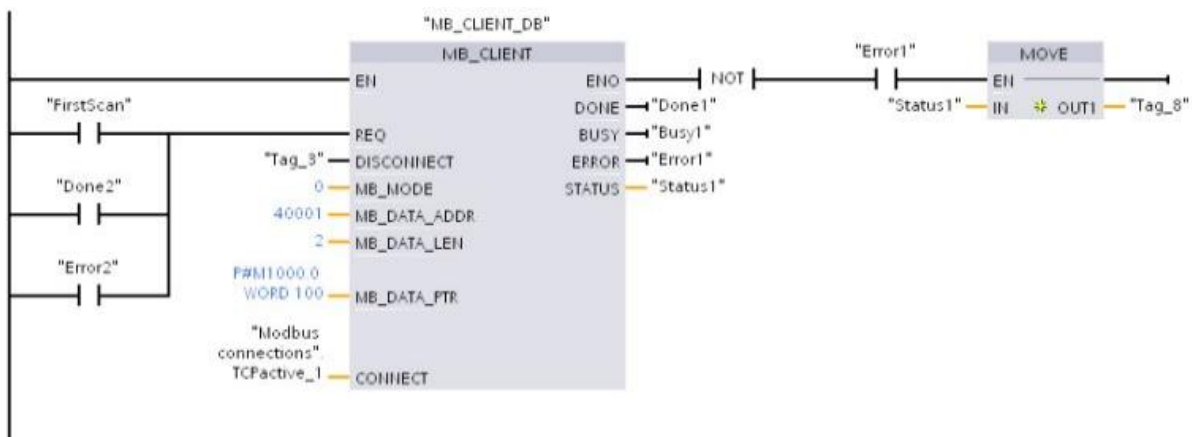
Сегмент 1: Modbus функция 15 – Записать выходные биты в сервер Modbus.

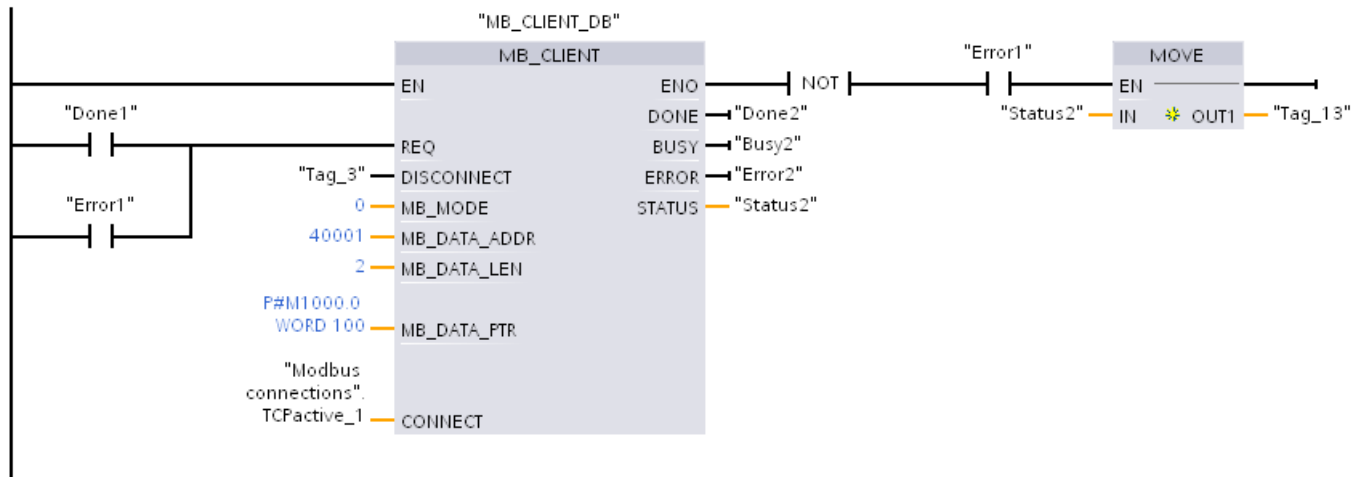
**Пример: MB_CLIENT 4: Координация нескольких запросов**

Вы должны обеспечить, чтобы каждый отдельный Modbus TCP запрос завершал выполнение. Последовательностью выполнения должна управлять Ваша программная логика. Приведенный ниже пример демонстрирует, как выходы первого и второго клиентских запросов могут управлять последовательностью выполнения.

Пример показывает обоих клиентов, использующих те же данные CONNECT (используемые в различные моменты времени). Клиенты передают данные регистра хранения из того же удаленного сервера Modbus TCP в ту же локальную ячейку в М памяти ЦПУ. Кроме того, возвращенная ошибка захватывается, что является необязательным.

Сегмент 1: Modbus функция 3 – Считать слова регистра хранения в сервере Modbus TCP



Сегмент 2: Modbus функция 3 - Считать слова регистра хранения в сервере Modbus TCP

13.5.3 Modbus RTU

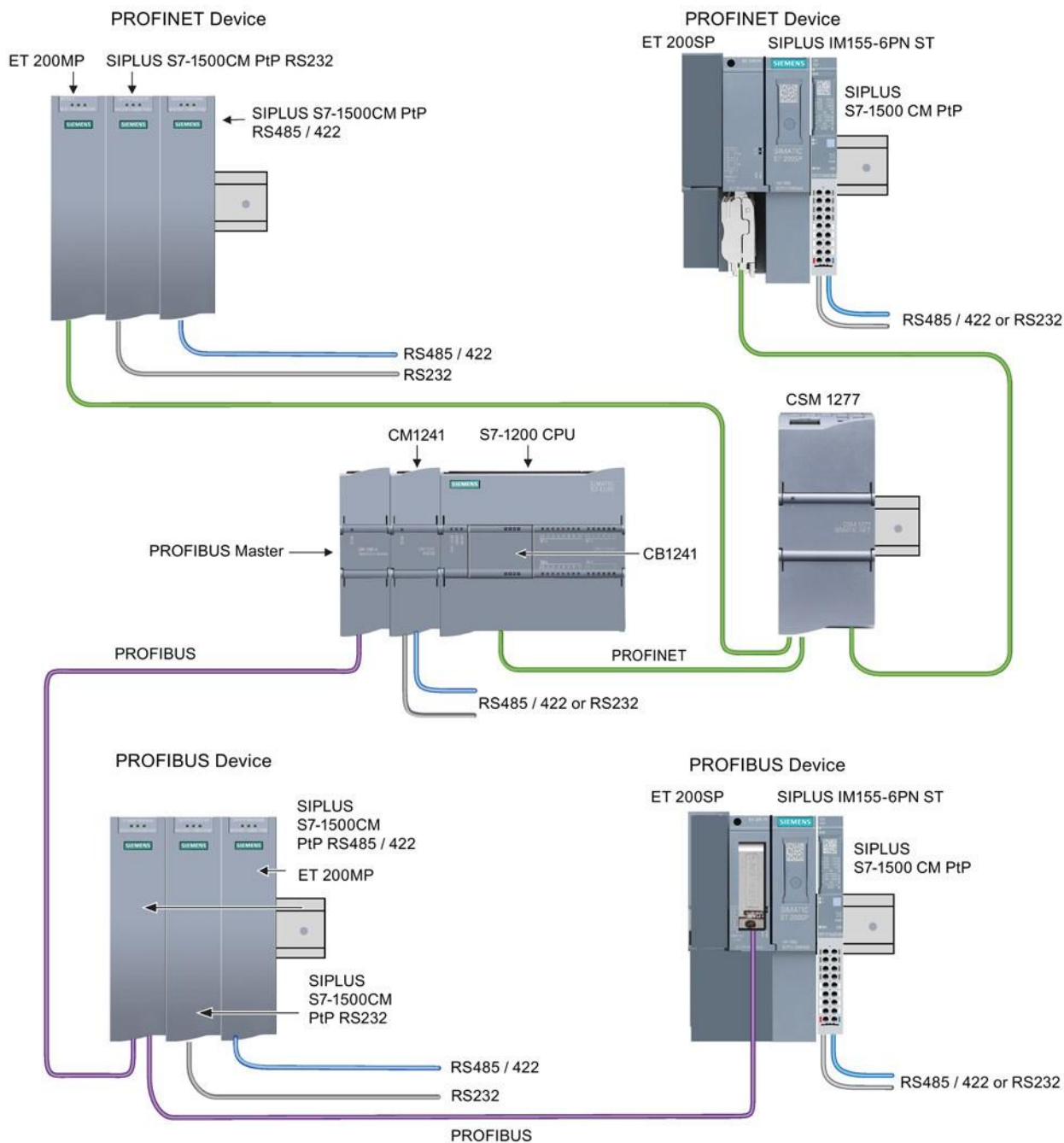
13.5.3.1. Обзор

Версия V4.1 ЦПУ S7-1200 вместе со STEP 7 V13 SP1 расширяет возможности Modbus RTU по использованию PROFINET или PROFIBUS распределенных стоек ввода-вывода для связи с различными устройствами (RFID-считыватели, GPS-устройства и другие):

- PROFINET (стр. 616): Вы соединяете Ethernet-интерфейс ЦПУ S7-1200 с интерфейсным модулем PROFINET. Коммуникационные PtP модули в стойке с интерфейсным модулем могут затем обеспечить последовательную передачу данных для PtP-устройств.
- PROFIBUS (стр. 752): Вы вставляете коммуникационный модуль PROFIBUS в левую часть стойки с ЦПУ S7-1200. Вы соединяете коммуникационный модуль PROFIBUS со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные PtP модули в стойке с интерфейсным модулем могут затем обеспечить последовательную передачу данных для PtP-устройств.

По этой причине S7-1200 поддерживает два набора PtP инструкций:

- Унаследованные Modbus RTU инструкции (стр. 1027): Эти Modbus RTU инструкции существовали до версии V4.0 S7-1200, и работают только с последовательной передачей данных, используя коммуникационный модуль CM 1241 или коммуникационную плату CB 1241.
- Modbus RTU инструкции (стр. 958): Эти Modbus RTU инструкции обеспечивают всю функциональность унаследованных инструкций плюс возможность подключения к распределенному вводу-выводу PROFINET и PROFIBUS. Инструкции "точка-точка" позволяют Вам конфигурировать обмен между коммуникационными PtP-модулями в распределенной стойке ввода-вывода и PtP-устройствами.



Примечание

С версией V4.1 S7-1200 Вы можете использовать инструкции "точка-к-точке" для всех типов обмена "точка-к-точке": последовательный, последовательный по PROFINET и последовательный по PROFIBUS. STEP 7 обеспечивает унаследованные инструкции "точка-к-точке" только для поддержки существующих программ. Унаследованные инструкции все еще функционируют с ЦПУ V4.1, а также V4.0 и более ранними ЦПУ. Вы не должны преобразовывать ранее созданные программы от одного набора инструкций к другому.

13.5.3.2. Выбор версии инструкций Modbus RTU

Существует две версии Modbus RTU инструкций доступных в STEP 7:

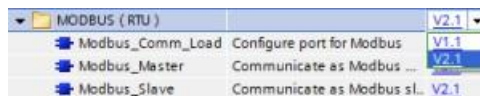
- Версия 1 была изначально доступна в STEP 7 Basic V10.5.
- Версия 2 доступна со STEP 7 Basic/Professional V11. Реализация версии 2 добавляет параметры REQ и DONE в MB_COMM_LOAD. Кроме того, параметр MB_ADDR для MB_MASTER и MB_SLAVE теперь позволяет использование UInt для расширенной адресации.

Для совместимости и простоты миграции, Вы можете выбирать какую версию инструкции вставлять в Вашу пользовательскую программу.

Вы не можете использовать обе версии инструкций с тем же модулем, но два различных модуля могут использовать различные версии инструкций. Не используйте обе версии инструкции 1.x и 2.y в одной программе ЦПУ. Ваши программные Modbus RTU инструкции должны иметь один и тот же старший номер версии (1.x, 2.y или V.z). Отдельные инструкции в рамках группы старшей версии могут иметь различные младшие версии (1.x).



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.



Чтобы изменить версию Modbus RTU инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить Modbus RTU инструкцию в Вашу программу, новый экземпляр FB создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию Modbus RTU инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите Modbus TCP экземпляр FB в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии Modbus RTU инструкции.

13.5.3.3. Modbus RTU инструкции

Инструкция Modbus_Comm_Load (Сконфигурировать ввод-вывод SIPLUS или порт на PtP модуле для Modbus RTU)

Таблица 13- 67 Инструкция Modbus_Comm_Load

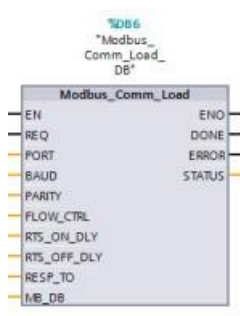
LAD / FBD	SCL	Описание
	<pre>"Modbus_Comm_Load_DB"(REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_uint_in_, PARI- TY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	Инструкция Modbus_Comm_Load конфигурирует ввод-вывод SIPLUS или PtP порт для обмена по протоколу Modbus RTU. Аппаратные опции порта Modbus RTU: Установка до трех CM (RS485 или RS232) плюс одна CB (R4845). Опции Modbus RTU ввода-вывода SIPLUS: Установка ET 200MP S7-1500CM PtP (RS485 / 422 или RS232) или ET 200SP S7-1500 CM PtP (RS485 / 422 или RS232) Экземплярный блок данных назначается автоматически, когда Вы помещаете инструкцию Modbus_Comm_Load в Вашу программу.

Таблица 13- 68 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Изменение сигнала от низкого уровня к высокому (положительный фронт) запускает работу. (Только версия 2.0)
PORT	IN	Port	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов.
BAUD	IN	UDInt	Выбор скорости обмена: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Все другие значения недействительны.
PARITY	IN	UInt	Выбор четности: 0 – Без контроля четности 1 – С контролем нечетности 2 – С контролем четности
FLOW_CTRL ¹	IN	UInt	Выбор управления потоком: 0 – Без управления потоком (по умолчанию) 1 – Аппаратное управление потоком с RTS всегда в ON (не применимо к RS485 портам) 2 – Аппаратное управление потоком с переключением RTS

Параметр и тип		Тип данных	Описание
RTS_ON_DLY ¹	IN	UInt	Выбор задержки RTS ON: 0 – Никакой задержки от активного RTS до передачи первого символа сообщения (по умолчанию) - От 1 до 65535 – Задержка в миллисекундах от активного RTS до передачи первого символа сообщения (не применимо к RS485 портам). Задержки RTS должны быть применены независимо от выбора FLOW_CTRL.
RTS_OFF_DLY ¹	IN	UInt	Выбор задержки RTS OFF: 0 – Никакой задержки от последнего переданного символа до перехода RTS в неактивное состояние (по умолчанию) - От 1 до 65535 – Задержка в миллисекундах от последнего переданного символа до перехода RTS в неактивное состояние (не применимо к RS485 портам). Задержки RTS должны быть применены независимо от выбора FLOW_CTRL.
RESP_TO ¹	IN	UInt	Тайм-аут ответа: Время в миллисекундах, отводимое Modbus_Master для ответа ведомому устройству. Если ведомое устройство не ответит в течение этого периода времени, то Modbus_Master повторит запрос или завершит запрос с ошибкой, когда было отправлено определенное количество повторений. От 5 мс до 65535 мс (значение по умолчанию = 1000 мс).
MB_DB	IN	Variant	Ссылка на экземплярный блок данных, используемый инструкциями Modbus_Master или Modbus_Slave. После того, как Modbus_Master или Modbus_Slave помещены в Вашу программу, идентификатор DB появляется в выпадающем списке помощника параметра, доступном в подключении MB_DB.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки. (Только версия 2.0)
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

¹ Дополнительные параметры для Modbus_Comm_Load (Версии 2.x или более поздней). Щелкните по стрелке у основания прямоугольника LAD/FBD, чтобы развернуть его и отобразить эти параметры.

Modbus_Comm_Load выполняется, чтобы сконфигурировать порт для протокола Modbus RTU. Как только порт сконфигурирован для протокола Modbus RTU, он может использоваться только инструкциями Modbus_Master или Modbus_Slave.

Одно выполнение Modbus_Comm_Load должно использоваться, чтобы сконфигурировать каждый коммуникационный порт, который используется для Modbus коммуникаций. Назначьте Modbus_Comm_Load уникальный экземплярный DB для каждого порта, который Вы используете. Вы можете установить до трех коммуникационных модулей (RS232 или RS485) и одну коммуникационную плату (RS485) в ЦПУ. Вызовите Modbus_Comm_Load из ОВ запуска и выполните ее однократно или используйте системный флаг первого скана (стр. 103), чтобы инициировать однократный вызов. Выполните Modbus_Comm_Load снова, только если необходимо изменить коммуникационные параметры, такие как скорость обмена или четность.

Если Вы пользуетесь библиотекой Modbus с модулем в распределенной стойке, инструкция Modbus_Comm_Load должна быть выполнена в процедуре циклического прерывания (например, один раз в секунду или один раз в 10 секунд). В случае пропадания питания распределенной стойки или снятия модуля после восстановления его работы, только набор параметра HWConfig отправляется в PtP модуль. Все запросы, инициируемые Modbus_Master оканчиваются тайм-аутом и Modbus_Slave переходит в режим молчания (отсутствие ответа на любое сообщение). Циклическое выполнение инструкции Modbus_Comm_Load решает эти вопросы.

Экземплярный блок данных назначается для Modbus_Master или Modbus_Slave, когда Вы помещаете эти инструкции в свою программу. На этот экземплярный блок данных ссылаются, когда Вы определяете параметр MB_DB для инструкции Modbus_Comm_Load.

Переменные блока данных Modbus_Comm_Load

Следующая таблица показывает общедоступные статические переменные, сохраненные в экземплярном DB для Modbus_Comm_Load, которые могут использоваться в Вашей программе.

Таблица 13- 69 Статические переменные в экземплярном DB

Переменная	Тип данных	Описание
ICHAR_GAP	UInt	Задержка разрыва между символами. Этот параметр определен в миллисекундах и используется, чтобы увеличить ожидаемое количество времени между полученными символами. Соответствующее число битовых интервалов для этого параметра добавляется к значению по умолчанию для Modbus равному 35 битовым интервалам (3.5 x время символа).
RETRIES	UInt	Число повторений, которые ведущее устройство будет пытаться сделать прежде, чем возвратит код ошибки отсутствия ответа "0x80C8".
STOP_BITS	USInt	Число стоповых битов, используемых в синхронизации каждого символа. Допустимыми значениями являются 1 и 2.

Таблица 13- 70 Коды состояния выполнения Modbus_Comm_Load ¹

STATUS (W#16#)	Описание
0000	Без ошибок
8180	Недопустимое значение ID порта (неправильный идентификатор порта/аппаратных
8181	Недопустимое значение скорости обмена
8182	Недопустимое значение четности
8183	Недопустимое значение управления потоком
8184	Недопустимое значение тайм-аута ответа (тайм-аут ответа меньше, чем 5 мс)
8185	Параметр MB_DB не является экземплярным блоком данных инструкции Modbus_Master или Modbus_Slave.

¹ В дополнение к упомянутым выше ошибкам Modbus_Comm_Load ошибки могут быть возвращены из базовых инструкций PtP коммуникаций.

Инструкция Modbus_Master (Выполнять обмен, используя ввод-вывод SIPLUS или PtP порт в качестве ведущего устройства Modbus RTU)

Таблица 13- 71 Инструкция Modbus_Master

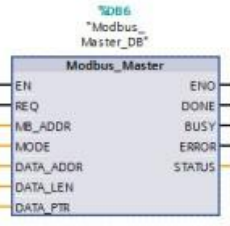
LAD / FBD	SCL	Описание
	<pre>"Modbus_Master_DB"(REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DA- TA_ADDR:=_udint_in_, DA- TA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	Инструкция Modbus_Master выполняет обмен, как ведущее устройство Modbus, используя порт, который был сконфигурирован предыдущим выполнением инструкции Modbus_Comm_Load. Экземплярный блок данных назначается автоматически, когда Вы помещаете инструкцию Modbus_Master в свою программу. Этот экземплярный блок данных Modbus_Master используется, когда Вы определяете параметр MB_DB для инструкции Modbus_Comm_Load.

Таблица 13- 72 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	0= Нет запроса 1= Запрос на передачу данных ведомому устройству Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции Modbus RTU: Стандартный диапазон адресов (от 1 до 247) Расширенный диапазон адресов (от 1 до 65535) Значение 0 зарезервировано для ширококестельной передачи сообщений всем ведомым устройствам Modbus. Только коды функций Modbus 05, 06, 15 и 16 поддерживаются ширококестельной передачей.
MODE	IN	USInt	Выбор режима: Определяет тип запроса (чтение, запись или диагностика). Смотрите таблицу функций Modbus, расположенную ниже для получения подробностей.
DATA_ADDR	IN	UDInt	Начальный адрес в ведомом устройстве: Определяет начальный адрес данных в ведомом устройстве Modbus, к которым осуществляется доступ. Смотрите таблицу функций Modbus, расположенную ниже для получения допустимых адресов.
DATA_LEN	IN	UInt	Длина данных: Определяет число битов или слов, к которым осуществляется доступ в этом запросе. Смотрите таблицу функций Modbus, расположенную ниже для получения допустимых длин.
DATA_PTR	IN	Variant	Указатель данных: Указывает на M или DB адрес (DB со стандартным доступом) для записываемых или считываемых данных.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки.
BUSY	OUT	Bool	0 – Нет выполняемых операций Modbus_Master 1 – Modbus_Master операция выполняется
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Правила коммуникаций Modbus_Master

- MB_COMM_LOAD должен быть выполнен, чтобы сконфигурировать порт, прежде чем инструкция Modbus_Master сможет связаться с этим портом.
- Если порт должен использоваться, чтобы инициировать запросы ведущего устройства Modbus, то порт не должен использоваться MB_SLAVE. Один или более экземпляров выполнения Modbus_Master могут использоваться с этим портом, но любое выполнение Modbus_Master должно использовать один и тот же экземплярный DB Modbus_Master для этого порта.
- Инструкции Modbus не используют события прерывания коммуникаций, чтобы управлять коммуникационным процессом. Ваша программа должна опросить инструкцию Modbus_Master на предмет завершения передачи и приема.
- Рекомендуется, чтобы Вы вызывали любое выполнение Modbus_Master для данного порта из ОВ программного цикла. Инструкции Modbus_Master могут выполняться только в одном из программных циклов или циклических/с временной задержкой уровнях выполнения. Они не должны выполняться на обоих приоритетных уровнях выполнения. Вытеснение инструкции Modbus_Master другой инструкцией Modbus_Master на более высоком уровне приоритета выполнения приведет к нештатной работе. Инструкции Modbus_Master не должны выполняться в уровнях выполнения, связанных с запуском, диагностикой или ошибкой времени выполнения.
- Как только инструкция Modbus_Master иницирует передачу, этот экземпляр должен постоянно выполняться с активным входом EN, до возврата состояния DONE=1 или ERROR=1. Определенный экземпляр Modbus_Master считают активным, пока не наступает одно из этих двух событий. В то время как исходный экземпляр активен, любой вызов другого экземпляра с активным входом REQ приведет к ошибке. Если непрерывное выполнение исходного экземпляра останавливается, состояние запроса остается активным на время, определенное статической переменной "Blocked_Proc_Timeout". Как только это время истекает, следующая Modbus_Master инструкция, вызванная с активным входом REQ, станет активным экземпляром. Это препятствует тому, чтобы единственный экземпляр Modbus_Master монополизировал или заблокировал доступ к порту. Если исходный активный экземпляр не будет активирован в течение времени, определенного статической переменной "Blocked_Proc_Timeout", то следующее выполнение этого экземпляра (с установленным REQ) сбросит активное состояние. Если REQ установлен, то это выполнение иницирует новый запрос Modbus_Master, как будто никакой другой экземпляр не был активен.

Параметр REQ

0 = Нет запроса; 1 = Запрос на передачу данных ведомому устройству Modbus

Вы можете управлять этим входом либо с помощью уровня либо активируемого фронт-том контакта. Каждый раз, когда этот вход активирован, конечный автомат запускается, чтобы гарантировать, что никакой другой Modbus_Master, использующий тот же экземплярный DB, не будет разрешено выставить запрос, пока текущий запрос не завершен. Состояния всех других входов захватываются и сохраняются внутри для текущего запроса, пока не получен ответ или не обнаружена ошибка.

Если тот же экземпляр Modbus_Master выполняется снова с входом REQ = 1 до завершения текущего запроса, то никакие последующие передачи не выполняются. Однако, когда запрос завершен, новый запрос выставляется каждый раз, когда Modbus_Master выполняется вновь с входом REQ = 1.

Параметры DATA_ADDR и MODE выбирают тип функции Modbus

DATA_ADDR (начальный Modbus адрес в ведомом устройстве): Определяет начальный адрес данных в ведомом устройстве Modbus, к которым осуществляют доступ .

Инструкция Modbus_Master использует вход MODE, а не вход Function Code. Комбинация MODE и адреса Modbus определяет код функции, который используется в фактическом сообщении Modbus. Следующая таблица показывает соответствие между параметром MODE, кодом функции Modbus и диапазоном адресов Modbus.

Таблица 13- 73 Modbus функции

MODE	Modbus функция	Длина данных	Операция и данные	Modbus адрес
0	01	От 1 до 2000 От 1 до 1992 ¹	Читать выходные биты: От 1 до (1992 или 2000) битов за запрос	От 1 до 9999
0	02	От 1 до 2000 1 до 1992 ¹	Читать входные биты: От 1 до (1992 или 2000) битов за запрос	От 10001 до 19999
0	03	От 1 до 125 От 1 до 124 ¹	Читать регистры хранения: От 1 до (124 или 125) слов за запрос	От 40001 до 49999 или От 400001 до 465535
0	04	От 1 до 125 От 1 до 124 ¹	Читать входные слова: От 1 до (124 или 125) слов за запрос	От 30001 до 39999
104	04	От 1 до 125 От 1 до 124 ¹	Читать входные слова: От 1 до (124 или 125) слов за запрос	От 00000 до 65535
1	05	1	Записать один выходной бит: один бит на запрос	От 1 до 9999
1	06	1	Записать один регистр хранения: 1 слово за запрос	От 40001 до 49999 или От 400001 до 465535
1	15	От 2 до 1968 От 2 до 1960 ¹	Записать несколько выходных битов: От 2 до (1960 или 1968) битов за запрос	От 1 до 9999
1	16	От 2 до 123 От 2 до 122 ¹	Записать несколько регистров хранения: От 2 до (122 или 123) слов за запрос	От 40001 до 49999 или От 400001 до 465535
2	15	От 1 до 1968 От 2 до 1960 ¹	Записать один или более выходных битов: От 1 до (1960 или 1968) битов за запрос	От 1 до 9999
2	16	От 1 до 123 От 1 до 122 ¹	Записать один или более регистров хранения: От 1 до (122 или 123) слов за запрос	От 40001 до 49999 или От 400001 до 465535
11	11	0	Читать коммуникационное слово состояния сервера и счетчик событий. Слово состояния указывает занятость (0 = не занято, 0xFFFF = занято). Счетчик событий постепенно увеличивается для каждого успешного завершения сообщения. Как MB_DATA_ADDR, так и MB_DATA_LEN параметры Modbus_Master игнорируются для этой функции.	
80	08	1	Проверить состояние ведомого устройства с помощью диагностического кода 0x0000 (Петлевой тест, сервер отвечает эхом) 1 слово за запрос	

MODE	Modbus функция	Длина данных	Операция и данные	Modbus адрес
81	08	1	Сбросить счетчик событий сервера с помощью диагностического кода 0x000A 1 слово за запрос	
3 ... 10, 12 ... 79, 82 ... 255			Зарезервировано	

¹ Для режима "расширенной адресации" максимальные длины данных уменьшены на 1 байт или 1 слово в зависимости от типа данных, используемого функцией.

Параметр DATA_PTR

Параметр DATA_PTR указывает на адрес в DB или M области, который записывается или считывается. Если Вы используете блок данных, то Вы должны создать глобальный блок данных, который обеспечивает хранение данных для чтения и записи в ведомые устройства Modbus.

Примечание

Тип блока данных DATA_PTR должен позволять прямую адресацию

Блок данных должен позволять как прямую (абсолютную), так и символьную адресацию. Когда Вы создаете блок данных, должен быть выбран атрибут доступа "Standard".

Структуры блока данных для параметра DATA_PTR

- Следующие типы данных допустимы для **чтений слов** из Modbus адресов с 30001 по 39999, с 40001 по 49999 и с 400001 по 465536, а также для **записи слов** в Modbus адреса с 40001 по 49999 и с 400001 по 465536.
 - Стандартный массив типов данных WORD, UINT или INT.
 - Именованная структура WORD, UINT или INT, где каждый элемент имеет уникальное имя и 16-битный тип данных.
 - Именованная сложная структура, где каждый элемент имеет уникальное имя и 16- или 32-битный тип данных.
- Для **чтения** и записи **битов** из адресов Modbus с 00001 по 09999 и чтения битов с 10001 по 19999.
 - Стандартный массив булевых типов данных.
 - Именованная булева структура уникально именованных булевых переменных.
- Несмотря на отсутствие требования, рекомендуется, чтобы у каждой инструкции Modbus_Master была своя собственная отдельная область памяти. Причина этой рекомендации кроется в том, что есть большая вероятность повреждения данных, если несколько инструкций Modbus_Master читают и пишут в ту же самую область памяти.
- Не обязательно, чтобы области данных DATA_PTR были в том же глобальном блоке данных. Вы можете создать один блок данных с несколькими областями для чтений Modbus, один блок данных для Modbus записей или один блок данных для каждой ведомой станции.

Переменные блока данных Modbus master

Следующая таблица показывает общедоступные статические теги в экземплярном DB для Modbus_Master, которые Вы можете использовать в своей программе.

Таблица 13- 74 Статические теги в экземплярном DB

Тег	Тип данных	Стандарт	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах) ожидания заблокированного экземпляра Modbus_Master прежде, чем удалить этот экземпляр, как ACTIVE. Это может произойти, например, если выставляется запрос Modbus_Master, а затем программа останавливается, чтобы вызвать функцию Modbus_Master до того, как она полностью не завершит запрос. Значение времени должно быть больше 0 и меньше 55 секунд или возникает ошибка.
Extended_Addressing	Bool	FALSE	Конфигурирует одно- или двухбайтовую адресацию ведомой станции: - FALSE = однобайтовый адрес, от 0 до 247 - TRUE = двухбайтовый адрес (соответствует расширенной адресации), от 0 до 65535
MD_DB	MB_BAS E	-	Параметр MD_DB инструкции Modbus_Comm_Load должен быть подключен к параметру MD_DB инструкции Modbus_Master.

Ваша программа может записать значения в переменные Blocked_Proc_Timeout и Extended_Addressing, чтобы управлять операциями Modbus_Master. Смотри описание HR_Start_Offset и Extended_Addressing в разделе MB_SLAVE для примера того, как использовать эти переменные в программном редакторе и деталях расширенной адресации Modbus (стр.967).

Коды состояния

Таблица 13- 75 Коды состояния выполнения Modbus_Master (ошибки коммуникаций и конфигурации) ¹

STATUS (W#16#)	Описание
0000	Без ошибок
80C8	Тайм-аут ведомого устройства. Определенное ведомое устройство не ответило в течение определенного времени. Проверьте скорость обмена, четность и проводные соединения ведомого устройства. Об этой ошибке сообщается только после того, как были предприняты все сконфигурированные повторы.
80C9	Инструкция Modbus_Master привела к таймауту по одной из следующих причин: - Инструкция ожидает ответа от модуля, который используется для коммуникаций. - Установлено слишком маленькое значение Blocked_Proc_Timeout. Об этой ошибке сообщается, если PROFIBUS- или PROFINET-устройство распределенного ввода-вывода восстановилось после одного из следующих событий: - Прерывание питания или коммуникаций - События снятия/установки коммуникационного модуля В этих случаях перезагружается аппаратная конфигурация из ПЛК, и Modbus_Comm_Load должна быть выполнена снова, чтобы должным образом сконфигурировать коммуникационный модуль.
80D1	Получатель выставил запрос на управление потоком, чтобы приостановить активную передачу и повторно не активировал передачу в течение указанного времени ожидания. Эта ошибка также генерируется во время аппаратного управления потоком, когда получатель не утверждает CTS в течение указанного времени ожидания.

13.5 Коммуникации по Modbus

STATUS (W#16#)	Описание
80D2	Запрос на передачу был прерван, так как никакой сигнал DSR не получен от DCE.
80E0	Сообщение было прервано, так как приемный буфер заполнен.
80E1	Сообщение было прервано в результате ошибки четности.
80E2	Сообщение было прервано в результате ошибки кадровой синхронизации.
80E3	Сообщение было прервано в результате ошибки переполнения.
80E4	Сообщение было прервано в результате указания длины, превышающей общий размер буфера.
8180	Недопустимое значение ID порта или ошибка в инструкции Modbus_Comm_Load
8186	Недопустимый адрес станции Modbus
8188	Недопустимый режим определен для широковещательного запроса
8189	Недопустимое значение адреса данных
818A	Недопустимое значение длины данных
818B	Недопустимый указатель на локальный источник/ назначение данных: Размер неправильный
818C	Недопустимый указатель для DATA_PTR или недопустимый Blocked_Proc_Timeout. Область данных должна быть одной из следующих: • Классический DB • Массив элементарных типов данных в символьном или сохраняемом DB • М память
8200	Порт занят обработкой запрос передачи.
8280	Отрицательное подтверждение при чтении модуля. Проверьте ввод в параметре PORT. Это может быть вызвано потерей PROFIBUS, или PROFINET модуля распределенного ввода-вывода, потерей питания станции или обмена, либо снятием модуля.
8281	Отрицательное подтверждение при записи в модуль. Проверьте ввод в параметре PORT. Это может быть вызвано потерей PROFIBUS, или PROFINET модуля распределенного ввода-вывода, потерей питания станции или обмена, либо снятием модуля.

Таблица 13- 76 Коды состояния выполнения Modbus_Master (ошибки Modbus протокола) ¹

STATUS (W#16#)	Ответ ведомого устройства	Ошибки Modbus протокола
8380	-	Ошибка CRC error
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или адрес вне допустимого диапазона области DATA_PTR
8384	Более чем 03	Ошибка значения данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)
8386	-	Код функции в ответе не соответствует коду в запросе.
8387	-	Ответило неправильное ведомое устройство
8388	-	Ответ ведомого устройства на запрос записи неправильный. Запрос записи, возвращенный ведомым устройством, не соответствует тому, что фактически отправило ведущее устройство.

¹ В дополнение к упомянутым выше ошибкам Modbus_Master ошибки могут быть возвращены из базовых коммуникационных PtP инструкций.

Инструкция Modbus_Slave (Выполнять обмен, используя ввод-вывод SIPLUS или PtP порт в качестве ведомого устройства Modbus RTU)

Таблица 13- 77 Инструкция Modbus_Slave

LAD / FBD	SCL	Описание
	<pre>"Modbus_Slave_DB"(MB_ADDR:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	Инструкция Modbus_Slave позволяет Вашей программе выполнять обмен одним из двух способов: - В качестве ведомого устройства Modbus RTU через PtP порт на CM (RS485 или RS232) и CB (RS485) - В качестве ведомого устройства Modbus RTU через опции Modbus RTU SIPLUS I/O: - Установите ET 200MP S7-1500CM PtP (RS485 / 422 или RS232). - Установите ET 200SP S7-1500 CM PtP (RS485 / 422 или RS232). Когда удаленное ведущее устройство Modbus RTU выставляет запрос, Ваша пользовательская программа отвечает на запрос выполнением Modbus_Slave. STEP 7 автоматически создает экземплярный DB, когда Вы вставляете инструкцию. Используйте это Modbus_Slave_DB имя, когда Вы задаете параметр MB_DB для инструкции Modbus_Comm_Load.

Таблица 13- 78 Типы данных для параметров

Параметр и тип		Тип данных	Описание
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес ведомой станции Modbus RTU: Стандартный диапазон адресов (от 1 до 247) Расширенный диапазон адресов (от 1 до 65535)
MB_HOLD_REG	IN	Variant	Указатель на DB регистра хранения Modbus: регистр хранения Modbus может быть M памятью или блоком данных.
NDR	OUT	Bool	Новые данные готовы: 0 – Нет новых данных 1 – Указывает на то, что новые данные были записаны ведущим устройством Modbus
DR	OUT	Bool	Чтение данных: 0 – Нет чтения данных 1 – Указывает на то, что данные были считаны ведущим устройством Modbus
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

13.5 Коммуникации по Modbus

Коды функций Modbus (1, 2, 4, 5 и 15) могут читать и записывать биты и слова непосредственно в образ процесса по вводу и по выводу. Для этих кодов функций параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт.

Следующая таблица показывает проекцию Modbus адресов в образ процесса в ЦПУ.

Таблица 13- 79 Проекция Modbus адресов в образ процесса

Modbus функции						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес в ЦПУ
01	Считать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Считать биты	Ввод	От 10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Считать слова	Ввод	От 30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Коды коммуникационных функций Modbus (3, 6, и 16) используют регистр хранения Modbus, который может располагаться в М памяти или блоке данных. Тип регистра хранения определяется параметром MB_HOLD_REG инструкции Modbus_Slave.

Примечание**Тип блока данных MB_HOLD_REG**

Блок данных регистра хранения Modbus должен позволять как прямую (абсолютную), так и символьную адресацию. Когда Вы создаете блок данных, должен быть выбран атрибут доступа "Standard".

Следующая таблица показывает примеры проекций Modbus адресов в регистры хранения, используемые для кодов Modbus функций 03 (считать слова), 06 (записать слово) и 16 (записать слова). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом М памяти для каждой из моделей ЦПУ.

Таблица 13- 80 Проекция Modbus адресов в адреса памяти ЦПУ

Адрес ведущего устройства Modbus	Примеры параметра MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recipe".ingredient[5]

Таблица 13- 81 Диагностические функции

Диагностические функции Modbus_Slave для S7-1200		
Коды	Подфункция	Описание
08	0x0000	Эхо-тест возврата данных запроса: Modbus_Slave отразит ведущему устройству Modbus слово данных, которое принято.
08	0x000A	Сбросить счетчик коммуникаций: Modbus_Slave сбросит счетчик коммуникаций, который используется для Modbus функции 11.
11		Получить счетчик коммуникаций: Modbus_Slave использует внутренней счетчик коммуникаций для записи числа успешных запросов на чтение и запись по Modbus, отправленных Modbus_Slave. Счетчик не увеличивается на единицу по любому запросу на Функцию 8, Функцию 11 или любому запросу, который приводит к ошибке связи (например, ошибки четности или CRC).

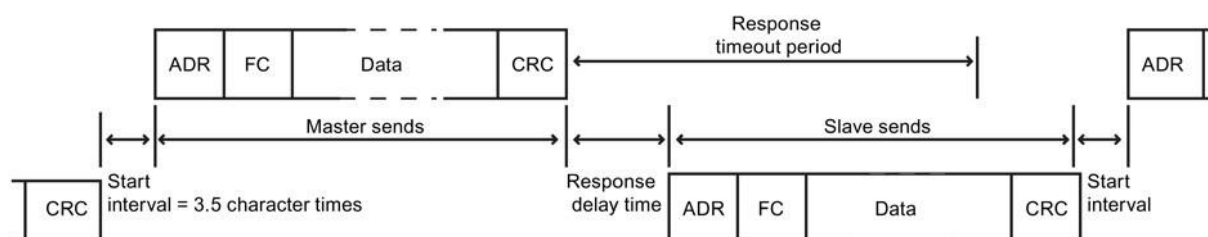
Инструкция Modbus_Slave поддерживает широковещательные запросы записи от любого ведущего устройства Modbus, если запрос использует допустимые адреса. Modbus_Slave выставляет код ошибки "0x8188" для кодов функций, не поддерживаемых в широковещательной передаче.

Правила коммуникаций Modbus_Slave

- Modbus_Comm_Load должен быть выполнен, чтобы сконфигурировать порт, прежде чем инструкция Modbus_Slave сможет выполнять обмен через этот порт.
- Если порт должен ответить Modbus_Master как ведомое устройство, то не программируйте этот порт инструкцией Modbus_Master.
- Только один экземпляр Modbus_Slave может использоваться с данным портом, иначе возможно неустойчивое поведение.
- Инструкции Modbus не используют коммуникационные события прерывания, чтобы управлять коммуникационным процессом. Ваша программа должна управлять коммуникационным процессом, опрашивая инструкцию Modbus_Slave на предмет состояния завершения передачи и приема.
- Инструкция Modbus_Slave должна периодически выполняться на уровне, который позволяет ей своевременно отвечать на входящие запросы от Modbus_Master. Рекомендуется, чтобы Вы выполняли Modbus_Slave каждый скан ОБ программного цикла. Выполнение Modbus_Slave из ОБ циклического прерывания возможно, но не рекомендуется из-за вероятности чрезмерных задержек других процедур прерывания.

Временная диаграмма Modbus сигнала

Modbus_Slave должен выполняться периодически, чтобы принимать каждый запрос от Modbus_Master, а затем ответить при необходимости. Периодичность выполнения для Modbus_Slave зависит от периода тайм-аута ответа Modbus_Master. Это проиллюстрировано на следующей диаграмме.



Период тайм-аута ответа RESP_TO представляет время, в течение которого Modbus_Master ожидает начала ответа от Modbus_Slave. Этот период времени не определен протоколом Modbus, но является параметром каждого Modbus_Master. Периодичность выполнения (время между текущим и последующим выполнениями) Modbus_Slave должна базироваться на определенных параметрах Вашего Modbus_Master. Как минимум Вы должны выполнять Modbus_Slave дважды за период тайм-аута ответа Modbus_Master.

Переменные Modbus_Slave

Настоящая таблица показывает общедоступные статические переменные, сохраненные в экземплярном блоке данных Modbus_Slave, которые могут использоваться в Вашей программе.

Таблица 13- 82 Переменные Modbus_Slave

Переменная	Тип данных	Описание
Request_Count	Word	Количество всех запросов, принятых этим ведомым устройством
Slave_Message_Count	Word	Количество принятых этим определенным ведомым устройством запросов
Bad_CRC_Count	Word	Количество принятых запросов, в которых есть ошибка CRC
Broadcast_Count	Word	Количество принятых широковещательных запросов
Exception_Count	Word	Определенные ошибки Modbus, которые требуют возвращенного исключения
Success_Count	Word	Количество принятых этим определенным ведомым устройством запросов, в которых нет ошибок протокола
HR_Start_Offset	Word	Определяет начальный адрес регистра хранения Modbus (значение по умолчанию = 0)
Extended_Addressing	Bool	Конфигурирует одно- или двухбайтную адресацию ведомого устройства (0=однобайтовый адрес, 1=двухбайтовый адрес, значение по умолчанию = 0)

Ваша программа может записать значения в переменные HR_Start_Offset и Extended_Addressing и управлять операциями ведомого устройства Modbus. Другие переменные могут быть считаны для контроля состояния Modbus.

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001 или 400001. Эти адреса соответствуют началу адресов регистра хранения в ПЛК. Однако, Вы можете сконфигурировать переменную "HR_Start_Offset", чтобы установить начало адресов регистра хранения Modbus на другое значение вместо 40001 или 400001.

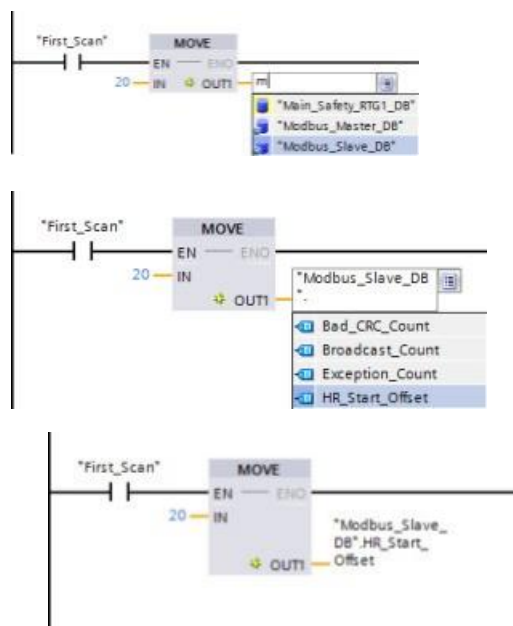
Например, если регистр хранения сконфигурирован, чтобы начинаться в MW100 и имеет длину 100 слов. Смещение 20 определяет начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 400119 приведет к ошибке адресации.

Таблица 13- 83 Пример адресации регистра хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Modbus адрес (слово)	40001	40099
	S7-1200 адрес	MW100	MW298
20	Modbus адрес (слово)	40021	40119
	S7-1200 адрес	MW100	MW298

HR_Start_Offset - значение типа word, которое определяет начальный адрес регистра хранения Modbus и сохраняется в экземплярном блоке данных Modbus_Slave. Вы можете установить это общедоступное значение статической переменной при помощи выпадающего списка помощника параметра, после того, как Modbus_Slave будет помещен в Вашу программу.

Например, после того, как Modbus_Slave помещен в сегмент LAD, Вы можете перейти к предыдущему сегменту и присвоить значение HR_Start_Offset. Значение должно быть присвоено до выполнения Modbus_Slave.



Ввод переменной ведомого устройства Modbus, используя имя DB по умолчанию:

1. Установите курсор в поле параметра и введите символ m.
2. Выберите "Modbus_Slave_DB" в выпадающем списке.
3. Установите курсор справа от имени DB (после символа кавычки) и введите символ точки.
4. Выберите "Modbus_Slave_DB.HR_Start_Offset" в выпадающем списке.

Extended Addressing

К переменной Extended Addressing обращаются похожим образом как к HR_Start_Offset, что обсуждено выше, за исключением того, что переменная Extended Addressing является булевой переменной. Булева переменная должно быть записана с помощью выходной катушки, а не оператором move.

Адресация ведомого устройства Modbus может быть сконфигурирована либо как од-нобайтовая (что является стандартом Modbus), либо, как двухбайтовая. Расширенная адресация используется, чтобы адресовать больше чем 247 устройств в отдельной сети. Выбор расширенной адресации позволяет Вам адресовать максимум 64000 адресов. Кадр Modbus функции 1 представлен ниже, в качестве примера.

Таблица 13- 84 Однобайтовый адрес ведомого устройства (байт 0)

Функция 1	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	
Запрос	Адрес ведом.	Код функции	Начальный адрес		Количество катушек		
Действит.ответ	Адрес ведом.	Код функции	Длина	Данные катушек			
Ответ с ошибкой	Адрес ведом.	0x81	Код ошибки				

Таблица 13- 85 Двухбайтовый адрес ведомого устройства (байты 0 и 1)

	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6
Запрос	Адрес ведомого устройства		Код функции	Начальный адрес		Количество катушек	
Действит.ответ	Адрес ведомого устройства		Код функции	Длина	Данные катушек		
Ответ с ошибкой	Адрес ведомого устройства		0x81	Код ошибки			

Коды состояния

Таблица 13- 86 Коды состояния выполнения Modbus_Slave (ошибки обмена и конфигурации) ¹

STATUS (W#16#)	Описание
80D1	Получатель выставил запрос на управление потоком, чтобы приостановить активную передачу и повторно не включал передачу в течение указанного времени ожидания. Эта ошибка также генерируется в процессе аппаратного управления потоком, когда получатель не утверждает CTS в течение указанного времени ожидания.
80D2	Запрос на передачу был прерван, так как не был получен сигнал DSR от DCE.
80E0	Сообщение было прервано, так как приемный буфер заполнен.
80E1	Сообщение было прервано в результате ошибки четности.
80E2	Сообщение было прервано в результате ошибки кадровой синхронизации.
80E3	Сообщение было прервано в результате ошибки переполнения.
80E4	Сообщение было прервано в результате задания длины, превышающей общий размер буфера.
8180	Недопустимое значение ID порта или ошибка с инструкцией Modbus_Comm_Load
8186	Неверный адрес Modbus станции

STATUS (W#16#)	Описание
8187	Недопустимый указатель на DB MB_HOLD_REG: область слишком маленькая
818C	Недопустимый указатель MB_HOLD_REG. Данные должны располагаться в одной из следующих областей: • Классический DB • Массив элементарных типов данных в символьном или сохраняемом DB • М память

Таблица 13- 87 Коды состояния выполнения Modbus_Slave (ошибки Modbus протокола) ¹

STATUS (W#16#)	Ответ ведомого устройства	Ошибки Modbus протокола
8380	Нет ответа	Ошибка CRC
8381	01	Код функции не поддерживается или не поддерживается в широковещательной рассылке
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или адрес вне допустимого диапазона области DATA_PTR
8384	03	Ошибка значения данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)

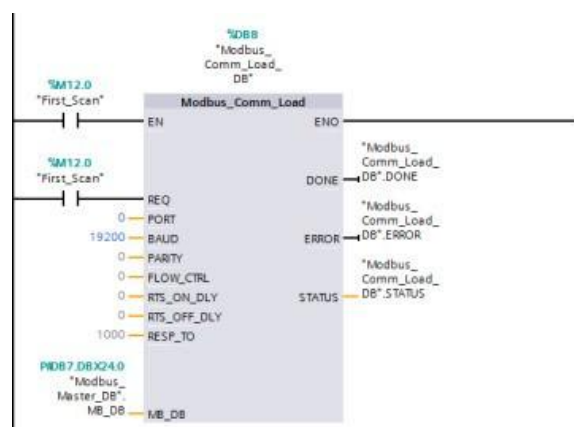
¹ В дополнение к упомянутым выше ошибкам Modbus_Slave ошибки могут быть возвращены из базовых коммуникационных PtP инструкций.

13.5.3.4. Примеры Modbus RTU

Пример: Программа ведущего устройства Modbus RTU

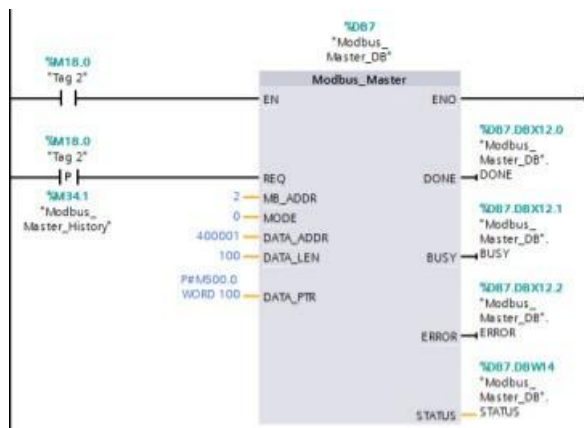
Modbus_Comm_Load инициализируется во время запуска при помощи флага первого скана. Обработка Modbus_Comm_Load подобным образом должна выполняться только тогда, когда конфигурация последовательного порта не изменяется во время работы.

Сегмент 1: Инициализировать параметры RS485 модуля только один раз во время первого скана.

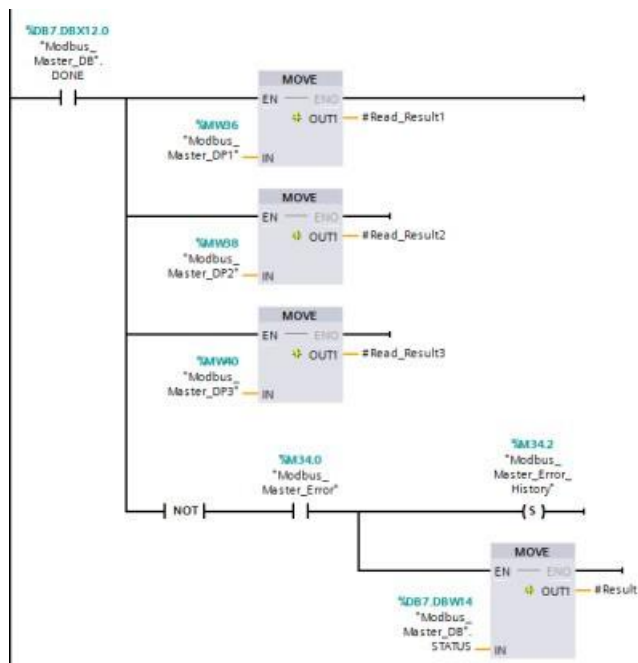


Одна инструкция Modbus_Master используется в ОВ программного цикла, чтобы выполнить обмен с одиночным ведомым устройством. Дополнительные инструкции Modbus_Master могут использоваться в ОВ программного цикла, чтобы выполнить обмен с другими ведомыми устройствами, или один FB Modbus_Master мог бы быть повторно использован, чтобы выполнить обмен с дополнительными ведомыми устройствами.

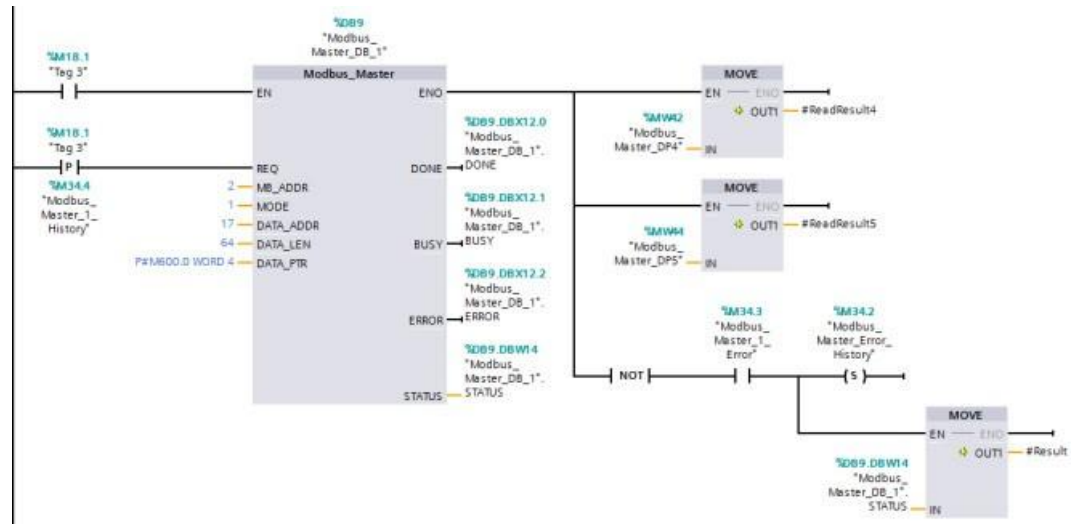
Сегмент 2: Считать 100 слов из регистра хранения ведомого устройства.



Сегмент 3: Это дополнительный сегмент, который просто показывает значения первых 3 слов, как только операция чтения выполнена.



Сегмент 4: Записать 64 бита в выходной регистр образа по выводу, начиная с адреса Q2.0.

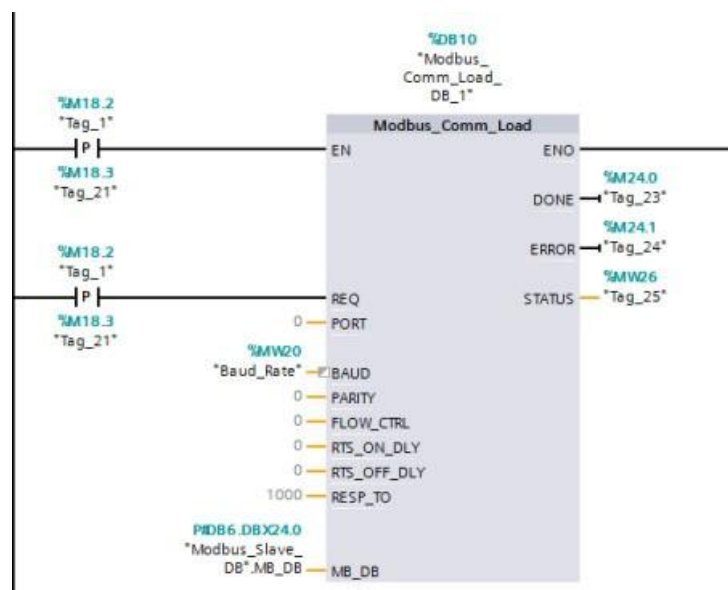


Пример: Программа ведомого устройства Modbus RTU

MB_COMM_LOAD, показанный ниже, инициализируется, каждый раз при активации тега "Tag_1".

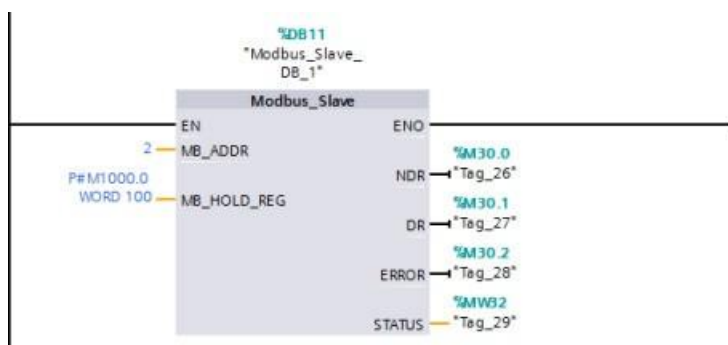
Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта изменяется во время работы, в результате конфигурирования устройства HMI.

Сегмент 1: Инициализировать параметры модуля RS485 каждый раз, когда они изменяются устройством HMI.



MB_SLAVE, показанный ниже, помещен в циклический OB, который выполняется каждые 10 мс. Несмотря на то, что это не обеспечивает абсолютно быстрый ответ ведомого устройства, но действительно обеспечивает хорошую производительность при 9600 бодах для коротких сообщений (20 байтов или меньше в запросе).

Сегмент 2: Проверять запросы от ведущего устройства Modbus на каждом скане. Регистр хранения Modbus сконфигурирован для 100 слов, начиная с MW1000.



13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

До выпуска STEP 7 V13 SP1 и ЦПУ S7-1200 V4.1, коммуникационные инструкции "точка-к-точке" существовали с различными именами, и в некоторых случаях, с немного отличающимися интерфейсами. Общие принципы коммуникаций "точка-к-точке" (стр. 864), конфигурирования порта (стр. 865) и задания параметров (стр. 877) применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным инструкциям "точка-к-точке" для получения сведений о программировании.

Таблица 13- 88 Классы общих ошибок

Описание класса	Классы ошибок	Описание
Конфигурация порта	80Ax	Используется для определения общих ошибок конфигурации порта
Конфигурация передачи	80Bx	Используется для определения общих ошибок конфигурации передачи
Конфигурация приема	80Cx	Используется для определения общих ошибок конфигурации приема
При выполнении передачи	80Dx	Используется для определения общих ошибок выполнения передачи
При выполнении приема	80Ex	Используется для определения общих ошибок выполнения приема
Обработка сигнала	80Fx	Используется для определения общих ошибок, связанных с любой обработкой сигналов

13.6.1. Унаследованные инструкции "точка-к-точке"

13.6.1.1. Инструкция PORT_CFG (Конфигурировать коммуникационные параметры динамически)

Таблица 13- 89 Инструкция PORT_CFG (конфигурирование порта)

LAD / FBD	SCL	Описание
	<pre> "PORT_CFG_DB"(REQ:=_bool_in_, PORT:=_uint_in_, PROTOCOL:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, DATABITS:=_uint_in_, STOPBITS:=_uint_in_, FLOWCTRL:=_uint_in_, XONCHAR:=_char_in_, XOFFCHAR:=_char_in_, WAITTIME:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	PORT_CFG позволяет Вам изменять параметры порта, такие как скорость обмена из Вашей программы. Вы можете установить начальную статическую конфигурацию порта в свойствах конфигурации устройства, или просто использовать значения по умолчанию. Вы можете выполнить инструкцию PORT_CFG в своей программе, чтобы изменить конфигурацию.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

Изменения конфигурации с помощью PORT_CFG не сохраняются в ЦПУ. Параметры, заданные в конфигурации устройства, восстанавливаются, когда ЦПУ переходит из режима RUN в режим STOP и после цикла включения питания. Смотрите Конфигурирование коммуникационных портов (стр. 865) и Организация управления потоком (стр. 867) для получения дополнительной информации.

Таблица 13- 90 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
PROTOCOL	IN	UInt	0 –Коммуникационный протокол Point-to-Point (Значение по умолчанию) 1..n - будущее определение для специфических протоколов
BAUD	IN	UInt	Скорость обмена порта (Значение по умолчанию: 6): 1 = 300 бод, 2 = 600 бод, 3 = 1200 бод, 4 = 2400 бод, 5 = 4800 бод, 6 = 9600 бод, 7 = 19200 бод, 8 = 38400 бод, 9 = 57600 бод, 10 = 76800 бод, 11 = 115200 бод
PARITY	IN	UInt	Контроль четности порта (Значение по умолчанию: 1): 1 = Без контроля, 2 = На четность, 3 = На нечетность, 4 = По единичному биту, 5 = По нулевому биту
DATABITS	IN	UInt	Битов на символ (Значение по умолчанию: 1): 1 = 8 битов данных, 2 = 7 битов данных.
STOPBITS	IN	UInt	Стоповые биты (Значение по умолчанию: 1): 1 = 1 стоповый бит, 2 = 2 стоповых бита
FLOWCTRL	IN	UInt	Управление потоком (Значение по умолчанию: 1): 1 = Без управления, 2 = XON/XOFF, 3 = Аппаратное: RTS всегда ON, 4 = Аппаратное: с переключением RTS
XONCHAR	IN	Char	Определяет символ, который используется в качестве символа XON. Это обычно символ DC1 (16#11). Этот параметр оценивается только, если включено управление потоком. (Значение по умолчанию: 16#11)
XOFFCHAR	IN	Char	Определяет символ, который используется в качестве символа XOFF. Это обычно символ DC3 (16#13). Этот параметр оценивается только, если управление потоком включено. (Значение по умолчанию: 16#13)
XWAITIME	IN	UInt	Определяет время ожидания символа XON после получения символа XOFF, или время ожидания сигнала CTS после активации RTS (от 0 до 65535 мс). Этот параметр оценивается только, если управление потоком включено. (Значение по умолчанию: 2000)
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Таблица 13- 91 Коды состояния

STATUS (W#16#....)	Описание
80A0	Специфический протокол не существует.
80A1	Специфическая скорость обмена не существует.
80A2	Специфическая опция четности не существует.
80A3	Специфическое число битов данных не существует.
80A4	Специфическое число стоповых битов не существует.
80A5	Специфический тип управления потоком не существует.
80A6	Время ожидания - 0, и управление потоком разрешено
80A7	XON и XOFF - недопустимые значения (например, одно то же значение)

13.6.1.2. Инструкция SEND_CFG (динамическое конфигурирование параметров последовательной передачи)

Таблица 13- 92 Инструкция SEND_CFG (конфигурация передачи)

LAD / FBD	SCL	Описание
	<pre>"SEND_CFG_DB"(REQ:=_bool_in_, PORT:=_uint_in_, RTSONDLY:=_uint_in_, RTSOFFDLY:=_uint_in_, BREAK:=_uint_in_, IDLELINE:=_uint_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STATUS=>_word_out_);</pre>	SEND_CFG позволяет динамическую конфигурацию параметров последовательного передатчика для PtP коммуникационного порта. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется SEND_CFG.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Вы можете установить начальную статическую конфигурацию порта в свойствах конфигурации устройства, или просто использовать значения по умолчанию. Вы можете выполнить инструкцию SEND_CFG в своей программе, чтобы изменить конфигурацию.

ЦПУ не хранит постоянно значения, которые Вы устанавливаете инструкцией SEND_CFG. ЦПУ восстанавливает параметры, заданные в конфигурации устройства, когда ЦПУ переходит из RUN в STOP, а также после цикла включения питания. Смотрите раздел Конфигурирование параметров передачи (отправки) (стр. 868).

13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

Таблица 13- 93 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
RTSONDLY	IN	UInt	Число миллисекунд для ожидания после активации RTS прежде чем произойдет какая-либо передача Tx данных. Этот параметр допустим только, если разрешено аппаратное управление потоком. Допустимый диапазон 0 - 65535 мс. Значение 0 отключает опцию. (Значение по умолчанию: 0).
RTSOFFDLY	IN	UInt	Число миллисекунд для ожидания после того, как произойдет какая-либо передача Tx данных, прежде чем RTS будет запрещен. Этот параметр допустим только, если разрешено аппаратное управление потоком. Допустимый диапазон 0 - 65535 мс. Значение 0 отключает опцию. (Значение по умолчанию: 0).
BREAK	IN	UInt	Этот параметр определяет, что разрыв будет отправлен при запуске каждого сообщения для конкретного количества битовых интервалов. Максимум составляет 65535 битовых интервалов, то есть 8 секунд максимум. Значение 0 отключает опцию. (Значение по умолчанию: 12)
IDLELINE	IN	UInt	Этот параметр определяет, что линия останется свободной в течение указанного числа битовых интервалов перед запуском каждого сообщения. Максимум составляет 65535 битовых интервалов, то есть 8 секунд максимум. Значение 0 отключает опцию. (Значение по умолчанию: 12)
DONE	OUT	Bool	TRUE на один скан после последнего запроса завершенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса завершенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Таблица 13- 94 Коды состояния

STATUS (W#16#....)	Описание
81B0	Конфигурация прерывания передачи недопустима.
81B1	Разрыв больше, чем максимально допустимое значение.
81B2	Время простоя больше, чем максимальное допустимое значение.

13.6.1.3. Инструкция RCV_CFG (динамическое конфигурирование параметров последовательного приема)

Таблица 13- 95 Инструкция RCV_CFG (конфигурация приема)

LAD / FBD	SCL	Описание
	<pre>"RCV_CFG_DB" (REQ:=_bool_in_, PORT:=_uint_in_, CONDI- TIONS:=_struct_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STATUS=>_word_out_);</pre>	RCV_CFG выполняет динамическую конфигурацию параметров последовательного приемника для PtP коммуникационного порта. Эта инструкция конфигурирует условия, которые сигнализируют о начале и конце принятого сообщения. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется RCV_CFG.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Вы можете установить начальную статическую конфигурацию порта в свойствах конфигурации устройства, или просто использовать значения по умолчанию. Вы можете выполнить инструкцию RCV_CFG в своей программе, чтобы изменить конфигурацию.

ЦПУ не хранит постоянно значения, которые Вы устанавливаете инструкцией RCV_CFG. ЦПУ восстанавливает параметры, заданные в конфигурации устройства, когда ЦПУ переходит из RUN в STOP, а также после цикла включения питания. Смотрите раздел Конфигурирование параметров приема (стр. 869) для получения более подробной информации.

Таблица 13- 96 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация изменения конфигурации по нарастающему фронту на этом входе. (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
CONDITIONS	IN	CONDITIONS	Структура данных Conditions определяет условия начала и окончания сообщения, как описано ниже.
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

Условия начала для инструкции RCV_P2P

Инструкция Receive_P2P использует конфигурацию, определенную инструкцией RCV_CFG, чтобы определить начало и окончание коммуникационных сообщений "точка-к-точке". Начало сообщения определяется условиями начала. Начало сообщения может быть определено одним или комбинацией условий. Если определено больше чем одно условие начала, то все условия должны быть удовлетворены, прежде чем сообщение будет начато.

Смотрите раздел Конфигурирование параметров приема (стр. 869) для получения описания условий начала сообщения.

Структура данных параметра CONDITIONS, Часть 1 (условия начала)

Таблица 13- 97 Структура CONDITIONS для условий начала

Параметр и тип	Тип данных	Описание
STARTCOND	IN	UInt Определяет условие начала (Значение по умолчанию: 1) • 01H – Стартовый символ • 02H – Любой символ • 04H – Разрыв линии • 08H – Свободная линия • 10H - Последовательность 1 • 20H - Последовательность 2 • 40H - Последовательность 3 • 80H - Последовательность 4
IDLETIME	IN	UInt Число битовых интервалов, требуемых для тайм-аута свободной линии. (Значение по умолчанию: 40). Используется только с условием свободной линии. От 0 до 65535
STARTCHAR	IN	Byte Начальный символ, используемый с условием начального символа. (Значение по умолчанию: B#16#2)
SEQ[1].CTL	IN	Byte Управление игнорированием/сравнением для каждого символа последовательности 1: (Значение по умолчанию: B#16#0) Это биты разрешения для каждого символа в последовательности начала • 01H - Символ 1 • 02H - Символ 2 • 04H - Символ 3 • 08H - Символ 4 • 10H - Символ 5 Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
SEQ[1].STR	IN	Char[5] Стартовые символы последовательности 1 (5 символов). Значение по умолчанию: 0
SEQ[2].CTL	IN	Byte Управление игнорированием/сравнением для каждого символа последовательности 2: (Значение по умолчанию: B#16#0)
SEQ[2].STR	IN	Char[5] Стартовые символы последовательности 2 (5 символов). Значение по умолчанию: 0
SEQ[3].CTL	IN	Byte Управление игнорированием/сравнением для каждого символа последовательности 3: (Значение по умолчанию: B#16#0)
SEQ[3].STR	IN	Char[5] Стартовые символы последовательности 3 (5 символов). Значение по умолчанию: 0

Параметр и тип		Тип данных	Описание
SEQ[4].CTL	IN	Byte	Управление игнорированием/сравнением для каждого символа последовательности 3: (Значение по умолчанию: B#16#0)
SEQ[4].STR	IN	Char[5]	Стартовые символы последовательности 3 (5 символов). Значение по умолчанию: 0

Пример

Рассмотрите следующее принятое шестнадцатеричное кодированное сообщение: "68 10 aa 68 bb 10 aa 16" и сконфигурированные последовательности начала, показанные в таблице ниже. Последовательности начинают оцениваться, когда первый 68H символ успешно получен. После успешного получения четвертого символа (второе 68H) условие 1 удовлетворено. Как только условия начала удовлетворены, начинается оценка условий окончания.

Обработка последовательности начала может быть завершена из-за различных ошибок четности, синхронизации кадров или символов. Эти ошибки исключают какое-либо принятое сообщение, так как условие начала не было удовлетворено.

Таблица 13- 98 Условия начала

Условия начала	Первый символ	Первый символ +1	Первый символ +2	Первый символ +3	Первый символ +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Условия завершения для инструкции RCV_PTP

Конец сообщения определяется спецификацией условий завершения. Конец сообщения определяется первым появлением одного или нескольких сконфигурированных условий завершения. Раздел "Условия завершения сообщения" в теме "Конфигурирование параметров приема (стр. 869)", описывает условия завершения, которые Вы можете сконфигурировать в инструкции RCV_CFG.

Вы можете сконфигурировать условия завершения в свойствах коммуникационного интерфейса в конфигурации устройства или с помощью инструкции RCV_CFG. Каждый раз, когда ЦПУ переходит из STOP в RUN, параметры приема (и условия начала и завершения) возвращаются к параметрам конфигурации устройства. Если пользовательская STEP 7 программа обрабатывает RCV_CFG, то настройки изменяются на условия, заданные в RCV_CFG.

Структура данных параметра CONDITIONS, Часть 2 (условия завершения)

Таблица 13- 99 Структура CONDITIONS для условий завершения

Параметр	Тип параметра	Тип данных	Описание
ENDCOND	IN	UInt 0	Этот параметр определяет условие завершения сообщения: • 01H – Время отклика • 02H – Время сообщения • 04H – Межсимвольный интервал • 08H – Максимальная длина • 10H - N + LEN + M • 20H - Последовательность
MAXLEN	IN	UInt 1	Максимальная длина сообщения: используется только, когда выбрано условие завершения по максимальной длине. От 1 до 1024 байтов.
N	IN	UInt 0	Позиция байта в сообщении поля длины. Используется только с условием завершения N + LEN + M. От 1 до 1022 байтов.
LENGTHSIZE	IN	UInt 0	Размер поля длины (1, 2 или 4 байта). Используется только с условием завершения N + LEN + M.
LENGTHM	IN	UInt 0	Определяет число символов после поля длины, которые не включены в значение поля длины. Используется только с условием завершения N + LEN + M. От 0 до 255 байтов.
RCVTIME	IN	UInt 200	Определяет время ожидания первого, подлежащего приему символа. Операция приема будет завершена с ошибкой, если символ не будет успешно получен за установленное время. Используется только с условием времени отклика. (От 0 до 65535 битовых интервалов, то есть до 8 секунд максимум). Этот параметр не является условием завершения сообщения, так как оценка прерывается, когда первый символ отклика получен. Это является условием завершения только в том смысле, что приводит к завершению работы приемника при отсутствии какого-либо ответа, когда ответ ожидается. Вы должны выбрать отдельное условие завершения.
MSGTIME	IN	UInt 200	Определяет время ожидания всего сообщения, которое будет полностью получено, с момента получения первого символа. Этот параметр используется только, когда выбрано условие тайм-аута сообщения. (От 0 до 65535 миллисекунд).
CHARGAP	IN	UInt 12	Определяет число битовых интервалов между символами. Если число битовых интервалов между символами превысит указанное значение, то условие завершения будет удовлетворено. Используется только с условием межсимвольного интервала. (От 0 до 65535 битовых интервалов, то есть до 8 секунд максимум).

Параметр	Тип параметра	Тип данных	Описание
SEQ.CTL	IN	Byte B#16#0	Управление игнорированием/сравнением для каждого символа последовательности 1: Это биты активации для каждого символа в последовательности завершения. Символ 1 является битом 0, символ 2 является битом 1, ..., символ 5 является битом 4. Деактивация бита, связанного с символом, означает, что любой символ будет соответствовать в этой позиции последовательности.
SEQ.STR	IN	Char[5] 0	Начальные символы последовательности 1 (5 символов)

Таблица 13- 100 Коды состояния

STATUS (W#16#....)	Описание
81C0	Выбрано недопустимое условие начала
81C1	Выбрано недопустимое условие завершения, не выбрано условие завершения
81C2	Разрешено прерывание приема, и это недопустимо.
81C3	Разрешено условие завершения по максимальной длине, и максимальная длина = 0 или > 1024.
81C4	Разрешена расчетная длина, и N >= 1023.
81C5	Разрешена расчетная длина, и длина не равна 1, 2 или 4.
81C6	Разрешена расчетная длина, и значение M > 255.
81C7	Разрешена расчетная длина, и она > 1024.
81C8	Разрешен тайм-аут отклика, и он равен нулю.
81C9	Разрешен тайм-аут межсимвольного интервала, и он равен нулю.
81CA	Разрешен тайм-аут свободной линии, и он равен нулю.
81CB	Последовательность завершения разрешена, но все символы, "игнорируются".
81CC	Последовательность начала (любая из 4) разрешена, но все символы, "игнорируются".

13.6.1.4. Инструкция SEND_PTP (отправить данные из приемного буфера)

Таблица 13- 101 Инструкция SEND_PTP (отправить данные точка-к-точке)

LAD / FBD	SCL	Описание
	<pre>"SEND_PTP_DB"(REQ:=_bool_in_, PORT:=_uint_in_, BUFFER:=_variant_in_, LENGTH:=_uint_in_, PTRCL:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	SEND_PTP инициирует передачу данных и передает назначенный буфер в коммуникационный интерфейс. Программа ЦПУ продолжается, в то время как CM или CB отправляют данные с заданной скоростью передачи. Только одна операция отправки может обрабатываться в данный момент времени. CM или CB возвращают ошибку, если вторая SEND_PTP обрабатывается, в то время как CM или CB уже передают сообщение.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

Таблица 13- 102 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активация требуемой передачи по нарастающему фронту на этом входе. Иницирует передачу содержимого буфера в коммуникационный интерфейс "точка-к-точке". (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальное положение буфера передачи. (Значение по умолчанию: 0) Примечание: Булевы данные или булевы массивы не поддерживаются.
LENGTH ¹	IN	UInt	Длина переданного кадра в байтах (Значение по умолчанию: 0) При передаче сложной структуры всегда используйте длину 0.
PTRCL	IN	Bool	Зарезервировано для будущего использования
DONE	OUT	Bool	TRUE на один скан после последнего запроса заверщенного без ошибки
ERROR	OUT	Bool	TRUE на один скан после последнего запроса заверщенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)

- ¹ Дополнительный параметр: Щелкните по стрелке у основания прямоугольника LAD/FBD, чтобы развернуть его и отобразить этот параметр.

Во время передачи, выходы DONE и ERROR находятся в состоянии FALSE. Когда передача будет завершена, или DONE или ERROR выходы будут установлены в TRUE, чтобы показать состояние передачи. Когда выходы DONE или ERROR находятся в TRUE, выход STATUS допустим.

Инструкция возвращает состояние 16#7001, если коммуникационный интерфейс принимает переданные данные. Последующее выполнение SEND_PTP возвращает 16#7002, если CM или CB все еще заняты передачей. Когда передача завершена, CM или CB возвращают состояние процесса передачи как 16#0000 (при отсутствии ошибок). Последующее выполнение SEND_PTP с REQ в состоянии низкого уровня возвращает состояние 16#7000 (незанято).

Следующие диаграммы показывают отношение выходных значений к REQ. Это предполагает, что инструкция вызывается периодически, чтобы проверить состояние процесса передачи. На диаграмме ниже, предполагается, что инструкция вызывается на каждом скане (представлено значениями STATUS).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

Следующая диаграмма показывает, что параметры DONE и STATUS допустимы только для одного скана, если вход REQ управляется импульсно (для одного скана), чтобы инициировать процесс передачи.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Следующая диаграмма показывает отношение параметров DONE, ERROR и STATUS, когда есть ошибка.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

Значения DONE, ERROR и STATUS допустимы только, пока SEND_PTP выполняется снова с тем же экземплярным DB.

Таблица 13- 103 Коды состояния

STATUS (W#16#....)	Описание
81D0	Новый запрос, в то время как передатчик активен
81D1	Передача прервана из-за отсутствия CTS во время ожидания
81D2	Передача прервана из-за отсутствия DSR от DCE-устройства
81D3	Передача прервана из-за переполнения очереди (передача более 1024 байтов)
81D5	Сигнал обратного смещения (условие обрыва провода)
833A	DB для параметра BUFFER не существует.

13.6.1.5. Инструкция RCV_PTP (разрешить прием сообщений)

Таблица 13- 104 Инструкция RCV_PTP (принять данные точка-к-точке)

LAD / FBD	SCL	Описание
	<pre>"RCV_PTP_DB" (EN_R:=_bool_in_, PORT:=_uint_in_, BUFFER:=_variant_in_, NDR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, LENGTH=>_uint_out_);</pre>	RCV_PTP выполняет проверку на сообщения, которые были получены в CM или CB. Если сообщение будет доступно, то оно будет передано из CM или CB в ЦПУ. Ошибка возвращает соответствующее значение в параметре STATUS.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 105 Типы данных для параметров

Параметр и тип		Тип данных	Описание
EN_R	IN	Bool	Когда на этом входе TRUE, и сообщение доступно, оно передается в CM или CB в BUFFER. Когда на входе EN_R сигнал FALSE, CM или CB проверяется на сообщения и выходы NDR, ERROR и STATUS обновляются, но сообщение не передается в BUFFER. (Значение по умолчанию: 0)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
BUFFER	IN	Variant	Этот параметр указывает на начальное расположение приемного буфера. Этот буфер должен быть достаточно большим, чтобы получить сообщение максимальной длины. Булевы данные или булевы массивы не поддерживаются. (Значение по умолчанию: 0)
NDR	OUT	Bool	TRUE на один скан, когда новые данные готовы и операция завершена без ошибок.
ERROR	OUT	Bool	TRUE на один скан после последнего запроса завершенного с ошибкой
STATUS	OUT	Word	Код состояния обработки (Значение по умолчанию: 0)
LENGTH	OUT	UInt	Длина возвращенного сообщения в байтах (Значение по умолчанию: 0)

Имейте ввиду следующую связь между входом EN_R и буфером сообщения инструкции RCV_PTP:

Вход EN_R управляет копированием полученного сообщения в BUFFER.

Когда на входе EN_R сигнал TRUE, и сообщение доступно, ЦПУ передает сообщение из CM или CB в BUFFER и обновляет выходы NDR, ERROR, STATUS и LENGTH.

Когда на входе EN_R сигнал FALSE, ЦПУ проверяет CM или CB на сообщения и обновляет выходы NDR, ERROR и STATUS, но не передает сообщение в BUFFER. (Обратите внимание на то, что значением по умолчанию для EN_R является FALSE.)

На практике рекомендуется установить EN_R в TRUE и контролировать выполнение инструкции RCV_PTP с помощью входа EN.

Значение STATUS допустимо, когда либо NDR, либо ERROR установлены в состояние TRUE. Значение STATUS описывает причину завершения операции приема в CM или CB. Обычно это положительное значение, указывающее на то, что операция приема была успешна и что процесс приема завершен обычным образом. Если STATUS имеет отрицательное значение (старший значащий бит шестнадцатеричного значения установлен), операция приема была завершена в состоянии ошибки, таких как ошибки четности, синхронизации кадров или переполнения.

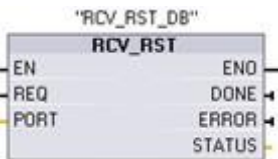
Каждый PtP коммуникационный интерфейс может поместить в буфер максимум 1024 байта. Это могло быть одним большим сообщением или несколькими меньшими сообщениями. Если больше чем одно сообщение доступно в CM или CB, инструкция RCV_PTP возвращает самое старое доступное сообщение. Последующее выполнение инструкции RCV_PTP возвращает следующее самое старое доступное сообщение.

Таблица 13- 106 Коды состояния

STATUS (W#16#...)	Описание
0000	Не существует буфера
0094	Сообщение прервано из-за максимальной длины в символах
0095	Сообщение прервано из-за тайм-аута сообщения
0096	Сообщение прервано из-за межсимвольного тайм-аута
0097	Сообщение прервано из-за тайм-аута ответа
0098	Сообщение прервано, так как было удовлетворено условие длины "N+LEN+M"
0099	Сообщение прервано, так как была распознана последовательность завершения
80E0	Сообщение прервано, так как приемный буфер полон
80E1	Сообщение прервано из-за ошибки четности
80E2	Сообщение прервано из-за ошибки кадровой синхронизации
80E3	Сообщение прервано из-за ошибки переполнения
80E4	Сообщение прервано, так как расчетная длина превышает размер буфера
80E5	Сигнал обратного смещения (условие обрыва провода)
833A	DB для параметра BUFFER не существует.

13.6.1.6. Инструкция RCV_RST (Удалить приемный буфер)

Таблица 13- 107 Инструкция RCV_RST (сбросить приемник)

LAD / FBD	SCL	Описание
	<pre>"RCV_RST_DB" (REQ:=_bool_in_, PORT:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	RCV_RST очищает приемный буфер в CM или CB.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 108 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Активирует сброс приемника по нарастающему фронту этого разрешающего входа (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
DONE	OUT	Bool	При установке в TRUE на один скан извещает о том, что последний запрос был завершен без ошибок.

13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

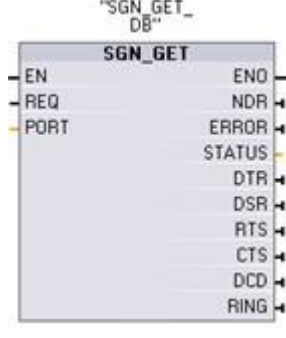
Параметр и тип		Тип данных	Описание
ERROR	OUT	Bool	Установка в TRUE извещает о том, что последний запрос был завершен с ошибками. Таким образом, когда этот выход установлен в TRUE, выход STATUS будет содержать соответствующую информацию об ошибке.
STATUS	OUT	Word	Код ошибки (Значение по умолчанию: 0). Смотрите Общие параметры для инструкций "точка-к-точке" (стр. 877) для получения кодов состояния коммуникаций.

Примечание

Вы могли бы захотеть использовать инструкцию RCV_RST, чтобы быть уверенными, что буферы сообщения стерты вслед за коммуникационной ошибкой, или после изменения коммуникационного параметра, такого как скорость обмена. Выполнение RCV_RST заставляет модуль очищать все внутренние буферы сообщения. После очистки буферов сообщения Вы можете быть уверены, что, когда Ваша программа выполняет последующую инструкцию приема, то сообщения, которые она возвращает, являются новыми, а не старыми сообщениями, принятыми до вызова RCV_RST.

13.6.1.7. Инструкция SGN_GET (Запросить RS-232 сигналы)

Таблица 13- 109 Инструкция SGN_GET (получить RS232 сигналы)

LAD / FBD	SCL	Описание
	<pre>"SGN_GET_DB" (REQ:=_bool_in_, PORT:=_uint_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DTR=>_bool_out_, DSR=>_bool_out_, RTS=>_bool_out_, CTS=>_bool_out_, DCD=>_bool_out_, RING=>_bool_out_);</pre>	SGN_GET читает текущие состояния RS232 коммуникационных сигналов. Эта функция допустима только для RS232 CM.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

Таблица 13- 110 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Получение состояний RS232 состояний по нарастающему фронту этого входа (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
NDR	OUT	Bool	TRUE на один скан, когда новые данные готовы, и операция завершена без ошибок
ERROR	OUT	Bool	TRUE на один скан после того, как операция была завершена с ошибками
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)
DTR	OUT	Bool	Терминал данных готов, модуль готов (выход). Значение по умолчанию: False
DSR	OUT	Bool	Источник данных готов, коммуникационный партнер готов (вход). Значение по умолчанию: False
RTS	OUT	Bool	Запрос на передачу, модуль, готов к передаче (выход). Значение по умолчанию: False
CTS	OUT	Bool	Разрешение на передачу, коммуникационный партнер может получить данные (вход). Значение по умолчанию: False
DCD	OUT	Bool	Обнаружен носитель данных, уровень приемного сигнала (всегда False, не поддерживается)
RING	OUT	Bool	Кольцевой индикатор, индикация входящего вызова (всегда False, не поддерживается)

Таблица 13- 111 Коды состояния

STATUS (W#16#...)	Описание
80F0	CM или CB в режиме RS485, и никакие сигналы не доступны

13.6.1.8. Инструкция SGN_SET (Установить RS-232 сигналы)

Таблица 13- 112 Инструкция SGN_SET (установить RS232 сигналы)

LAD / FBD	SCL	Описание
	<pre> "SGN_SET_DB"(REQ:=_bool_in_, PORT:=_uint_in_, SIGNAL:=_byte_in_, RTS:=_bool_in_, DTR:=_bool_in_, DSR:=_bool_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	SGN_SET устанавливает состояния RS232 коммуникационных сигналов. Эта функция допустима только для RS232 CM.

¹ STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

13.6 Унаследованные PtP коммуникации (только CM/CB 1241)

Таблица 13- 113 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Запускает операцию установки RS232 сигналов по нарастающему фронту на этом входе (Значение по умолчанию: False)
PORT	IN	PORT	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов. (Значение по умолчанию: 0).
SIGNAL	IN	Byte	Выбирает какой сигнал установить: (допустимо несколько). Значение по умолчанию: 0 • 01H = Установить RTS • 02H = Установить DTR • 04H = Установить DSR
RTS	IN	Bool	Запрос на передачу, модуль готов отправить значение для установки (true или false), Значение по умолчанию: False
DTR	IN	Bool	Терминал данных готов, модуль готов отправить значение для установки (true или false), Значение по умолчанию: False
DSR	IN	Bool	Сигнал готовности данных (применим только для интерфейсов DCE-типа), не используется.
DONE	OUT	Bool	TRUE на один скан после того, как последний запрос был завершен без ошибок
ERROR	OUT	Bool	TRUE на один скан после того, как последний запрос был завершен с ошибками
STATUS	OUT	Word	Код состояния выполнения (Значение по умолчанию: 0)

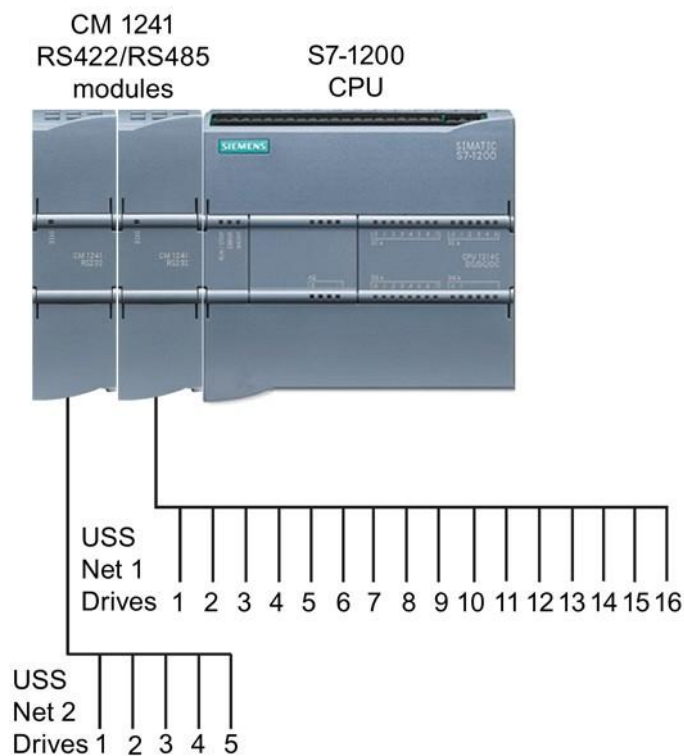
Таблица 13- 114 Коды состояния

STATUS (W#16#...)	Описание
80F0	CM или CB в режиме RS485, и никакие сигналы не могут быть установлены
80F1	Сигналы не могут быть установлены из-за аппаратного управления потоком
80F2	Нельзя установить DSR, так как модуль - DTE
80F3	Нельзя установить DTR, так как модуль - DCE

13.7 Унаследованные USS коммуникации (только CM/CB 1241)

USS-инструкции управляют работой электроприводов, которые поддерживают протокол универсального последовательного интерфейса (USS). Вы можете использовать USS-инструкции, чтобы связаться с несколькими приводами через RS485-соединения с коммуникационными модулями CM 1241 RS485 или коммуникационной платой CB 1241 RS485. До трех модулей CM 1241 RS422/RS485 и одна плата CB 1241 RS485 могут быть установлены в ЦПУ S7-1200. Каждый порт RS485 может управлять максимум шестнадцатью приводами.

USS-протокол использует сеть "ведущий-ведомый" для коммуникаций по последовательной шине. Ведущее устройство использует параметр адреса, чтобы отправить сообщение выбранному ведомому устройству. Ведомое устройство никогда не может самостоятельно передавать без предшествующего получения запроса. Прямая передача сообщений между отдельными ведомыми устройствами невозможна. USS-обмен работает в полудуплексном режиме. Следующая иллюстрация показывает сетевую диаграмму для примера приложения с приводами.



До выпуска STEP 7 V13 SP1 и ЦПУ S7-1200 V4.1 USS коммуникационные инструкции существовали с различными именами, и в некоторых случаях, с немного отличающимися интерфейсами. Общие принципы применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным USS инструкциям для получения сведений о программировании.

13.7.1. Выбор версии USS инструкций

Существуют две версии USS-инструкций, доступных в STEP 7:

- Версия 2.0 (унаследованные инструкции) была первоначально доступна в STEP 7 Basic/Professional V13.
- Версия 2.1 доступна в STEP 7 Basic/Professional V13, SP1.

Для совместимости и упрощения миграции, Вы можете выбрать то, какую версию инструкции вставить в Вашу пользовательскую программу.

Вы не можете использовать обе версии инструкций с тем же модулем, но два различных модуля могут использовать различные версии инструкций



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.

USS communication		V2.1
USS_Port_Scan	Communication via US...	V2.1
USS_Drive_Control	Data exchange with th...	V2.0
USS_Read_Param	Read data from drive	V2.1
USS_Write_Param	Change data in drive	V1.4

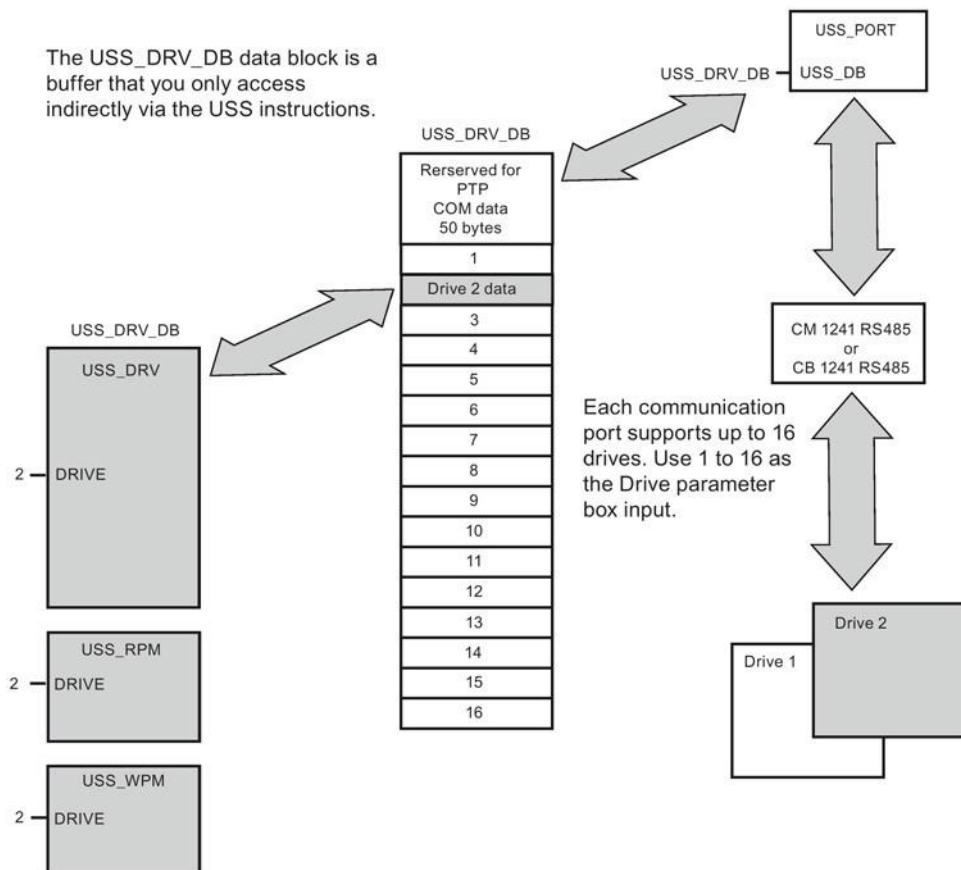
Чтобы изменить версию USS-инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить USS-инструкцию в Вашу программу, новый экземпляр FB или FC, в зависимости от выбранной USS-инструкции, создается в дереве проекта. Вы можете видеть новый экземпляр FB или FC в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию USS-инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите USS-экземпляр FB или FC в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии USS-инструкции.

13.7.2. Требования к использованию USS протокола

Четыре USS-инструкции используют 1 FB и 3 FC, чтобы поддержать USS-протокол. Один экземплярный блок данных (DB) USS_Port_Scan используется для каждой USS-сети. Экземплярный блок данных USS_Port_Scan содержит временное хранилище и буферы для всех приводов в этой USS-сети. USS-инструкции совместно используют информацию в этом блоке данных.



Все приводы (до 16), подключенные к единственному порту RS485, являются участниками одной и той же USS-сети. Все приводы, подключенные к различным портам RS485, являются участниками различных USS-сетей. Каждой USS-сетью управляют, используя одиночный блок данных. Все инструкции, связанные с отдельной USS-сетью, должны совместно использовать этот блок данных. Сюда относятся все USS_DRV, USS_PORT, USS_RPM и USS_WPM инструкции, используемые для управления всеми приводами в отдельной USS-сети.

Инструкция USS_DRV является функциональным блоком (FB). Когда Вы поместите инструкцию USS_DRV в редактор программы, Вам будет предложено диалоговое окно "Call options" для назначения DB этому FB. Если это первая инструкция USS_DRV в этой программе для этой USS-сети, то Вы можете принять назначение DB по умолчанию (или изменить имя, если хотите), и новый DB создается для Вас. Если, однако, это не первая инструкция USS_DRV для этого канала, то Вы должны использовать выпадающий список в диалоговом окне "Call options", чтобы выбрать имя DB, которое было ранее назначено для этой USS-сети.

Инструкции USS_PORT, USS_RPM и USS_WPM являются функциями (FC). Никакой DB не назначается, когда Вы помещаете эти FC в редакторе. Вместо этого Вы должны указать надлежащую ссылку на DB на входе "USS_DB" для этих инструкций. Дважды щелкните по полю параметра, а затем по значку помощника параметра, чтобы увидеть доступные имена DB.

Функция USS_PORT обрабатывает фактические коммуникации между ЦПУ и приводами через коммуникационный порт RS485 "точка-к-точке" (PtP). Каждый вызов этой функции обрабатывает одну коммуникацию с одним приводом. Ваша программа должна вызывать эту функцию достаточно часто, чтобы предотвратить тайм-аут коммуникации с приводом. Вы можете вызывать эту функцию в OB основного программного цикла или любом OB прерывания.

Как правило, Вы должны вызывать функцию USS_PORT из OB циклического прерывания. Время цикла для OB должно быть установлено приблизительно на половину минимального интервала вызова (Например, обмен на скорости 1200 бод должен использовать время цикла 350 мс или меньше).

Функциональный блок USS_DRV обеспечивает Вашей программе доступ к указанному приводу в USS-сети. Его входы и выходы являются состояниями и органами управления для привода. Если в сети 16 приводов, то в Вашей программе должно быть по крайней мере 16 вызовов USS_DRV, по одному для каждого привода.

Эти блоки нужно вызывать с интервалом, который требуется для управления работой привода.

Вы можете вызывать функциональный блок USS_DRV только из OB основного программного цикла.

ПРЕДУПРЕЖДЕНИЕ

Соображения по вызову USS -инструкций из OB

Вызывайте USS_DRV, USS_RPM и USS_WPM только из OB основного программного цикла. Функцию USS_PORT можно вызывать из любого OB, обычно из OB циклического прерывания.

Не используйте инструкции USS_DRV, USS_RPM и USS_WPM в OB с более высоким приоритетом, чем соответствующая инструкция USS_PORT. Например, не помещайте USS_PORT в основной OB, а USS_RPM в OB циклического прерывания. Отказ предотвратить прерывание обработки USS_PORT может привести к неожиданным ошибкам, которые могли бы привести к телесным повреждениям.

Функции USS_RPM и USS_WPM FC читают и записывают рабочие параметры удаленного привода. Эти параметры управляют внутренней работой привода. Смотри руководство на привод для определения этих параметров. Ваша программа может содержать столько этих функций сколько необходимо, но только один запрос на чтение или запись может быть активным на привод в любой момент времени. Вы можете вызывать функции USS_RPM и USS_WPM только из OB основного программного цикла.

Расчет времени, требуемого для обмена с приводом

Коммуникации с приводом являются асинхронными к циклу сканирования S7-1200. S7-1200 обычно завершает несколько сканов, прежде чем одна коммуникационная транзакция привода будет завершена.

Интервал USS_PORT представляет время, требуемое для одной транзакции привода. Таблица ниже показывает минимальный интервал USS_PORT для каждой скорости обмена в бодах. Вызов функции USS_PORT более частый, чем интервал USS_PORT не увеличит число транзакций. Интервал тайм-аута привода соответствует времени, которое могло бы занять транзакция, если бы коммуникационные ошибки вызвали 3 попытки завершить транзакцию. По умолчанию библиотека USS-протокола автоматически делает до 2 повторений на каждой транзакции.

Таблица 13- 115 Расчет временных характеристик

Скорость в бодах	Расчетный минимальный интервал вызова для USS_PORT (мс)	Тайм-аут интервала сообщения на один привод (мс)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.7.3. Унаследованные USS инструкции

13.7.3.1. Инструкция USS_PORT (редактировать обмен, использующий USS-сеть)

Таблица 13- 116 Инструкция USS_PORT

LAD / FBD	SCL	Описание
	<pre>USS_PORT(PORT:=_uint_in_, BAUD:=_dint_in_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	Инструкция USS_PORT обрабатывает обмен по USS-сети.

13.7 Унаследованные USS коммуникации (только CM/CB 1241)

Таблица 13- 117 Типы данных для параметров

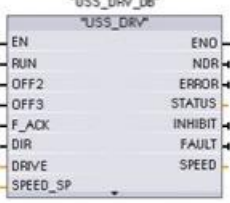
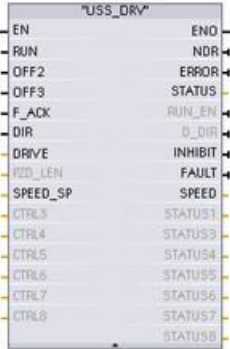
Параметр и тип		Тип данных	Описание
PORT	IN	Port	После того, как Вы установите и сконфигурируете CM или CB, идентификатор порта появляется во вспомогательном выпадающем списке параметра, доступном в соединении поля PORT. Присвоенное значение порта CM или CB является свойством конфигурации устройства "аппаратный идентификатор". Символьное имя порта присваивается на вкладке "System constants" таблицы ПЛК тегов.
BAUD	IN	DInt	Скорость в бодах используемая для USS-обмена.
USS_DB	INOUT	USS_BASE	Имя DB экземпляра, который создан и инициализирован, когда инструкция USS_Drive_Control помещена в Вашу программу.
ERROR	OUT	Bool	Когда true, этот вывод указывает на то, что произошла ошибка, и вывод STATUS допустим.
STATUS	OUT	Word	Значение состояния запроса указывает результат сканирования или инициализации. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

Как правило, существует только одна инструкция USS_PORT на коммуникационный PtP-порт в программе, и каждый вызов этой функции обрабатывает передачу в или из одиночного привода. Все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом должны использовать тот же экземплярный DB.

Ваша программа должна выполнять инструкцию USS_PORT достаточно часто, чтобы предотвратить тайм-ауты приводов. USS_PORT обычно вызывают из OB циклического прерывания, чтобы предотвратить тайм-ауты приводов и сохранить последние обновления USS-данных, доступными для вызовов USS_DRV.

13.7.3.2. Инструкция USS_DRV (обмениваться данными с приводом)

Таблица 13- 118 Инструкция USS_DRV

LAD / FBD	SCL	Описание
Вид по умолчанию  Расширенный вид 	<pre> "USS_DRV_DB" (RUN:=_bool_in_, OFF2:=_bool_in_, OFF3:=_bool_in_, F_ACK:=_bool_in_, DIR:=_bool_in_, DRIVE:=_usint_in_, PZD_LEN:=_usint_in_, SPEED_SP:=_real_in_, CTRL3:=_word_in_, CTRL4:=_word_in_, CTRL5:=_word_in_, CTRL6:=_word_in_, CTRL7:=_word_in_, CTRL8:=_word_in_, NDR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, RUN_EN=>_bool_out_, D_DIR=>_bool_out_, INHIBIT=>_bool_out_, FAULT=>_bool_out_, SPEED=>_real_out_, STATUS1=>_word_out_, STATUS3=>_word_out_, STATUS4=>_word_out_, STATUS5=>_word_out_, STATUS6=>_word_out_, STATUS7=>_word_out_, STATUS8=>_word_out_); </pre>	Инструкция USS_DRV обменивается данными с приводом, создавая сообщения запроса, и интерпретируя ответные сообщения привода. Отдельный функциональный блок должен использоваться для каждого привода, но все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом, должны использовать тот же экземплярный блок данных. Вы должны создать имя DB, когда Вы помещаете первую инструкцию USS_DRV, а затем обращаться к DB, который создавался при первом использовании инструкции. STEP 7 автоматически создает DB, когда Вы вставляете инструкцию.

¹ LAD и FBD: Разверните прямоугольник, чтобы показать все параметры, щелкнув по его нижней части. Контакты параметров, которые выделены серым, являются дополнительными, и присвоение параметра не требуется.

13.7 Унаследованные USS коммуникации (только CM/CB 1241)

Таблица 13- 119 Типы данных для параметров

Параметр и тип		Тип данных	Описание
RUN	IN	Bool	Бит запуска привода: Когда true, этот вход разрешает приводу вращаться с заданной скоростью. Когда RUN переходит в false, пока привод работает, двигатель будет замедляться до остановки. Это поведение отличается от сброса мощности (OFF2) или торможения двигателя (OFF3).
OFF2	IN	Bool	Бит электрической остановки: Когда false, этот бит заставляет привод останавливаться по инерции без торможения.
OFF3	IN	Bool	Бит быстрой остановки: Когда false, этот бит вызывает быструю остановку торможением привода вместо того, чтобы просто позволить приводу остановиться по инерции.
F_ACK	IN	Bool	Бит квитирования отказа: Этот бит устанавливается для сброса отказа привода. Бит устанавливается после того, как отказ устранен, чтобы указать приводу на то, что не нужно больше сообщать о предыдущем отказе.
DIR	IN	Bool	Управление направлением привода: Этот бит устанавливается для указания направления вперед (для положительного SPEED_SP).
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PZD_LEN	IN	USInt	Длина в словах: Это количество слов в PZD-данных. Допустимые значения составляют 2, 4, 6, или 8 слов. Значение по умолчанию - 2.
SPEED_SP	IN	Real	Уставка скорости: Это скорость привода как процент заданной частоты. Положительное значение определяет прямое направление (когда DIR в true). Допустимый диапазон от 200.00 до -200.00.
CTRL3	IN	Word	Слово управления 3: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL4	IN	Word	Слово управления 4: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL5	IN	Word	Слово управления 5: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL6	IN	Word	Слово управления 6: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL7	IN	Word	Слово управления 7: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
CTRL8	IN	Word	Слово управления 8: Значение, записанное в конфигурируемый пользователем параметр в приводе. Вы должны сконфигурировать его в приводе. (дополнительный параметр)
NDR	OUT	Bool	Новые данные готовы: Когда true, бит указывает на то, что выходы содержат данные из нового коммуникационного запроса.
ERROR	OUT	Bool	Ошибка произошла: Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	Значение состояния запроса указывает на результат сканирования. Это не слово состояния, возвращенное приводом.
RUN_EN	OUT	Bool	Работа разрешена: Этот бит указывает на то, работает ли привод.

Параметр и тип		Тип данных	Описание
D_DIR	OUT	Bool	Направление привода: Этот бит указывает на то, вращается ли привод вперед.
INHIBIT	OUT	Bool	Привод запрещен: Этот бит указывает на состояние бита запрета привода.
FAULT	OUT	Bool	Отказ привода: Этот бит указывает на то, что привод зарегистрировал отказ. Вы должны устранить проблему и затем установить бит F_ACK, чтобы сбросить этот бит, когда он установлен.
SPEED	OUT	Real	Текущая скорость привода (масштабированное значение слова состояния привода Word 2): значение скорости привода, как процент заданной скорости.
STATUS1	OUT	Word	Слово состояния привода Word 1: Это значение содержит фиксированные биты состояния привода.
STATUS3	OUT	Word	Слово состояния привода Word 3: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS4	OUT	Word	Слово состояния привода Word 4: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS5	OUT	Word	Слово состояния привода Word 5: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS6	OUT	Word	Слово состояния привода Word 6: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS7	OUT	Word	Слово состояния привода Word 7: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS8	OUT	Word	Слово состояния привода Word 8: Это значение содержит конфигурируемое пользователем слово состояния привода.

Когда происходит начальное выполнение USS_DRV, привод, обозначенный USS-адресом (параметр DRIVE), инициализируется в экземплярном DB. После этой инициализации последующие выполнения USS_DRV могут начать обмен с приводом по указанному номеру.

Изменение номера привода требует перехода ЦПУ из STOP в RUN, который инициализирует экземплярный DB. Входные параметры конфигурируются в буфер сообщения USS TX, и выходы читаются из "предыдущего" допустимого буфера ответа, если он существует. Во время выполнения USS_DRV нет никакой передачи данных. Приводы выполняют коммуникации, когда выполняется USS_PORT. USS_DRV только конфигурирует сообщения, которые необходимо передать и интерпретирует данные, которые, возможно, были получены от предыдущего запроса.

Вы можете управлять направлением привода, используя либо вход DIR (Bool), либо знак (положительный, или отрицательный) со входом SPEED_SP (Real). Следующая таблица указывает на то, как эти входы работают вместе, чтобы определить направление привода, предполагая, что двигатель подключен для вращения в прямом направлении.

Таблица 13- 120 Взаимодействие параметров SPEED_SP и DIR

SPEED_SP	DIR	Направление вращения привода
Значение > 0	0	Обратное
Значение > 0	1	Прямое
Значение < 0	0	Прямое
Значение < 0	1	Обратное

13.7.3.3. Инструкция USS_RPM (Считывание параметров из привода)

Таблица 13- 121 Инструкция USS_RPM

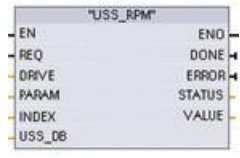
LAD / FBD	SCL	Описание
	<pre>USS_RPM(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, IN- DEX:=_uint_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, VAL- UE=>_variant_out_, USS_DB:=_fbtref_inout_);</pre>	Инструкция USS_RPM читает параметр из привода. Все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом, должны использовать тот же блок данных. USS_RPM нужно вызывать из OB основного программного цикла.

Таблица 13- 122 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Отправить запрос: Когда true, REQ указывает на то, что необходим новый запрос на чтение. Он игнорируется, если запрос на этот параметр уже находится на рассмотрении.
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PARAM	IN	UInt	Номер параметра: PARAM определяет, какой параметр привода записывается. Диапазон этого параметра от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Смотри руководство на свой привод для получения дополнительной информации о том, как получить доступ к расширенному диапазону.
INDEX	IN	UInt	Индекс параметра: INDEX определяет, какой индекс параметра привода должен быть записан. 16-разрядное значение, где младший значащий Байт представляет фактическое значение индекса в диапазоне (от 0 до 255). Старший значащий Байт может также использоваться приводом и специфичен для привода. Смотри руководство на свой привод для получения детальной информации.
USS_DB	INOUT	USS_BASE	Имя экземплярного DB, который создается и инициализируется, когда инструкция USS_DRV помещена в Вашу программу.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Это значение параметра, который был считан, допустимо только, когда DONE находится в true.
DONE ¹	OUT	Bool	Когда true, указывает на то, что выход VALUE содержит ранее запрошенное считанное значение параметра. Этот бит устанавливается, когда USS_DRV видит прочитанные ответные данные от привода. Этот бит сбрасывается, когда либо Вы запрашиваете ответные данные, используя другой опрос USS_RPM, либо при втором из следующих двух вызовов USS_DRV.

Параметр и тип		Тип данных	Описание
ERROR	OUT	Bool	Ошибка произошла: Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	STATUS указывает на результат запроса на чтение. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

- ¹ Бит DONE указывает на то, что допустимые данные были считаны из адресованного электропривода, и доставлены ЦПУ. Это не говорит о том, что USS-библиотека способна к незамедлительному чтению другого параметра. Пустой PKW запрос должен быть отправлен в электропривод и должен также быть подтвержден инструкцией, прежде чем канал параметра для определенного привода станет доступным для использования. Незамедлительный вызов функции USS_RPM USS_WPM для указанного электропривода приведет к ошибке "0x818A".

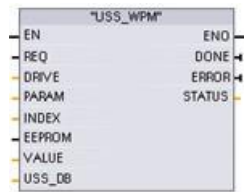
13.7.3.3 Инструкция USS_WPM (Изменить параметры в приводе)

Примечание

Операции записи EEPROM (для EEPROM в USS-приводе)

Не злоупотребляйте постоянной операцией записи EEPROM. Минимизируйте число операций записи EEPROM, чтобы продлить срок службы EEPROM.

Таблица 13- 123 Инструкция USS_WPM

LAD / FBD	SCL	Описание
	<pre>USS_WPM(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, IN- DEX:=_uint_in_, EEPROM:=_bool_in_, VAL- UE:=_variant_in_, DONE=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	Инструкция USS_WPM изменяет параметр в приводе. Все USS-функции, связанные с одной USS-сетью и PtP коммуникационным портом, должны использовать тот же блок данных. USS_WPM нужно вызвать из ОБ основного программного цикла.

13.7 Унаследованные USS коммуникации (только CM/CB 1241)

Таблица 13- 124 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	Отправить запрос: Когда true, REQ указывает на то, что необходим новый запрос на запись. Он игнорируется, если запрос на этот параметр уже находится на рассмотрении.
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом USS-привода. Допустимый диапазон от 1 до 16.
PARAM	IN	UInt	Номер параметра: PARAM определяет, какой параметр привода записывается. Диапазон этого параметра от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Смотри руководство на свой привод для получения дополнительной информации о том, как получить доступ к расширенному диапазону.
INDEX	IN	UInt	Индекс параметра: INDEX определяет, какой индекс параметра привода должен быть записан. 16-разрядное значение, где младший значащий Байт представляет фактическое значение индекса в диапазоне (от 0 до 255). Старший значащий Байт может также использоваться приводом и специфичен для привода. Смотри руководство на свой привод для получения детальной информации.
EEPROM	IN	Bool	Сохранить в EEPROM привода: Когда true, транзакция записи параметра привода будет сохранена в EEPROM привода. Если false, запись является временной и не будет сохранена, при выключении привода.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDIInt, Real	Значение параметра, который должен быть записан. Оно должно быть допустимо при переходе REQ.
USS_DB	INOUT	USS_BASE	Имя экземплярного DB, который создается и инициализируется, когда инструкция USS_DRV помещена в Вашу программу.
DONE ¹	OUT	Bool	Когда true, DONE, указывает на то, что значение на входе VALUE было записано в привод. Этот бит устанавливается, когда USS_DRV видит ответные данные записи от привода. Этот бит сбрасывается, когда либо Вы запрашиваете ответные данные, используя другой опрос USS_WPM, либо при втором из следующих двух вызовов USS_DRV.
ERROR	OUT	Bool	Когда true, указывает на то, что ошибка произошла, и значение на выходе STATUS допустимо. Все другие выходы обнулены при ошибке. Об ошибках обмена сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	STATUS указывает на результат запроса на запись. Дополнительная информация доступна в переменной "USS_Extended_Error" для некоторых кодов состояния.

¹ Бит DONE указывает на то, что допустимые данные были считаны из адресованного электропривода, и доставлены ЦПУ. Это не говорит о том, что USS-библиотека способна к незамедлительному чтению другого параметра. Пустой PKW запрос должен быть отправлен в электропривод и должен также быть подтвержден инструкцией, прежде чем канал параметра для определенного привода станет доступным для использования. Незамедлительный вызов функции USS_RPM или USS_WPM для указанного электропривода приведет к ошибке "0x818A".

13.7.4. Унаследованные коды состояний USS

Коды состояния USS-инструкций возвращаются на выходе STATUS USS-функций.

Таблица 13- 125 Коды на выходе STATUS ¹

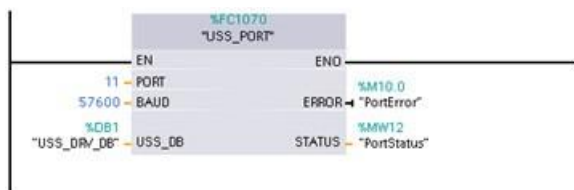
STATUS (W#16#....)	Описание
0000	Без ошибок
8180	Длина ответа привода не соответствовала символам, принятым от привода. Номер привода, где произошла ошибка, возвращается в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8181	Параметр VALUE не имел тип Word, Real или DWord.
8182	Пользователь предоставил Word для значения параметра и получил DWord или Real от привода в ответе.
8183	Пользователь предоставил DWord или Real для значения параметра и получил Word от привода в ответе.
8184	У ответной телеграммы от привода была неверная контрольная сумма. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8185	Недопустимый адрес привода (допустимый диапазон адреса : от 1 до16)
8186	Уставка скорости вне допустимого диапазона (допустимый диапазон SP скорости: от -200% до 200%).
8187	Привод с неправильным номером ответил на отправленный запрос. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
8188	Определена недопустимая длина PZD в словах (допустимые значения = 2, 4, 6 или 8 слов)
8189	Была определена недопустимая скорость в бодах.
818A	Канал запроса параметра используется другим запросом к этому приводу.
818B	Привод не ответил на запросы и повторения. Номер привода, где ошибка произошла, возвращен в переменной "USS_Extended_Error". Смотрите описание расширенной ошибки ниже данной таблицы.
818C	Привод возвратил расширенную ошибку на операцию запроса параметра. Смотрите описание расширенной ошибки ниже данной таблицы.
818D	Привод возвратил ошибку несанкционированного доступа на операцию запроса параметра. Смотрите руководство на свой привод для получения информации относительно того, почему доступ к параметру может быть ограничен.
818E	Привод не был инициализирован. Этот код ошибки возвращен USS_RPM или USS_WPM, когда USS_DRV, для этого привода, не был вызван, по крайней мере, один раз. Это предохраняет инициализацию на первом сканировании USS_DRV от перезаписи незаконченного запроса на чтение или запись параметра, так как это инициализирует привод как новую запись. Чтобы устранить эту ошибку, вызовите USS_DRV для этого номера привода.
80Ax-80Fx	Определенные ошибки возвращены из FB для PtP-коммуникаций, вызванных USS-библиотекой. - Эти значения кодов ошибки не изменяются USS-библиотекой и определены в описаниях PtP-инструкций.

¹ В дополнение к упомянутым выше ошибкам USS- инструкции ошибки могут быть возвращены из основных коммуникационных PtP-инструкций .

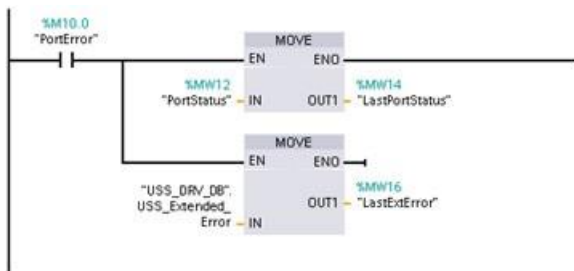
Для нескольких кодов параметра STATUS дополнительная информация предоставлена в экземплярном DB для USS_DRV в переменной "USS_Extended_Error". Для шестнадцатеричных кодов параметра STATUS 8180, 8184, 8187 и 818B, USS_Extended_Error содержит номер привода, где произошла ошибка связи. Для шестнадцатеричного кода параметра STATUS 818C, USS_Extended_Error содержит код ошибки привода, возвращенный из привода при использовании инструкции USS_RPM или USS_WPM.

Пример: Оповещение о коммуникационных ошибках

О коммуникационных ошибках (STATUS = 16#818B) сообщает только инструкция USS_PORT, но не инструкция USS_DRV. Например, если сеть должным образом не завершена, то для привода возможно перейти в RUN, но инструкция USS_DRV покажет все "0" для выходных параметров. В этом случае Вы можете обнаружить коммуникационную ошибку только в инструкции USS_PORT. Так как эта ошибка видима только на один скан, Вы должны будете добавить некоторую логику захвата, как проиллюстрировано в следующем примере. В этом примере, когда бит ошибки инструкции USS_PORT находится в TRUE, тогда значения STATUS и USS_Extended_Error сохраняются в M области памяти. Номер привода помещается в переменную USS_Extended_Error, когда значение кода в параметре STATUS равно шестнадцатеричным 8180, 8184, 8187 или 818B.



Сегмент 1: Значения статуса порта "PortStatus" и расширенного кода ошибки "USS_DRV_DB".USS_Extended_Error действительны только в течение одного скана. Значения должны быть захвачены для дальнейшей обработки.



Сегмент 2: Контакт "PortError" активирует сохранение значения "PortStatus" в "LastPortStatus" и значения "USS_DRV_DB".USS_Extended_Error в "LastExtError".

Доступ для чтения и записи к внутренним параметрам привода

USS-приводы поддерживают доступ к внутренним параметрам для чтения и записи. Эта функция позволяет дистанционное управление и конфигурирование привода. Операции доступа к параметрам привода могут перестать работать из-за ошибок, таких как выход значения из диапазона или недопустимых запросов на текущий режим привода. Привод генерирует значение кода ошибки, которое возвращается в переменной "USS_Extended_Error". Это значение кода ошибки допустимо только для последнего выполнения инструкции USS_RPM или USS_WPM. Код ошибки привода помещен в переменную USS_Extended_Error, когда значением кода параметра STATUS является шестнадцатеричное 818C. Значение кода ошибки USS_Extended_Error зависит от модели привода. Смотри руководство на привод для получения описания кодов расширенной ошибки для операций чтения и записи параметров.

13.7.5. Общие требования унаследованных конфигураций USS по настройке привода

Общие требования унаследованных конфигураций USS по настройке привода состоят из следующих пунктов:

- Приводы должны быть настроены для использования 4 PKW слов.
- Приводы могут быть сконфигурированы для 2, 4, 6 или 8 PZD слов.
- Количество PZD слов в приводе должно соответствовать входу PZD_LEN в инструкции USS_DRV для данного привода.
- Скорость обмена во всех приводах должна соответствовать входу BAUD в инструкции USS_PORT.
- Привод должен быть настроен для дистанционного управления.
- Привод должен быть настроен для установки частоты по USS на COM канале.
- Адрес привода должен быть установлен в диапазоне 1 - 16 и соответствовать входу DRIVE в блоке USS_DRV для данного привода.
- Управление направлением привода должно быть настроено для использования полярности установки привода.
- Сеть RS485 должна быть завершена должным образом.

Общее USS-подключение и настройка привода являются одинаковыми для USS-инструкций (V4.1) и унаследованных USS-инструкций (V4.0 и ранее). Обратитесь к примеру: Общее USS-подключение и настройка привода (стр. 927) для получения дополнительной информации.

13.8 Унаследованные Modbus TCP коммуникации

13.8.1. Обзор

До выпуска STEP 7 V13 SP1 и ЦПУ S7-1200 V4.1 Modbus TCP коммуникационные инструкции существовали с отличными именами, и в некоторых случаях, с немного отличающимися интерфейсами. Общие принципы применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным Modbus TCP инструкциям для получения сведений о программировании.

13.8.2. Выбор версии Modbus TCP инструкций

Существует две версии Modbus TCP инструкций доступных в STEP 7:

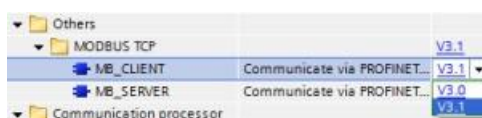
- Версия 3.0 была изначально доступна в STEP 7 Basic/Professional V13.
- Версия 3.1 доступна в STEP 7 Basic/Professional V13, SP1.

Для совместимости и простоты миграции, Вы можете выбирать какую версию инструкции вставлять в Вашу пользовательскую программу.

Не используйте обе версии инструкции 3.0 и 3.1 в одной программе ЦПУ. Ваши программные Modbus TCP инструкции должны иметь один и тот же старший номер версии (1.x, 2.y или V.z). Отдельные инструкции в рамках группы старшей версии могут иметь различные младшие версии (1.x).



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.



Чтобы изменить версию Modbus TCP инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить Modbus TCP инструкцию в Вашу программу, новый экземпляр FB создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию Modbus TCP инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите Modbus TCP экземпляр FB в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии Modbus TCP инструкции.

13.8.3. Унаследованные Modbus TCP инструкции

13.8.3.1. Инструкция MB_CLIENT (Выполнять обмен, используя PROFINET, как Modbus TCP клиент)

Таблица 13- 126 Инструкция MB_CLIENT

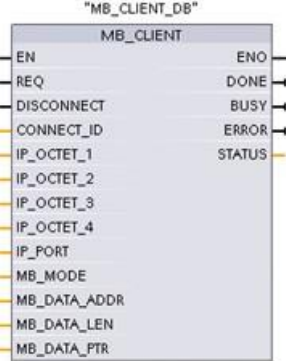
LAD / FBD	SCL	Описание
	<pre>"MB_CLIENT_DB" (REQ:=_bool_in_, DISCON- NECT:=_bool_in_, CON- NECT_ID=_uint_in_, IP_OCTET_1:=_byte_in_, IP_OCTET_2:=_byte_in_, IP_OCTET_3:=_byte_in_, IP_OCTET_4:=_byte_in_, IP_PORT:=_uint_in_, MB_MODE:=_usint_in_, MB_DATA_ADDR:=_udint_in_, MB_DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_DATA_PTR:=_variant_inout_);</pre>	MB_CLIENT выполняет обмен как Modbus TCP клиент через порт PROFINET на ЦПУ S7-1200. Никакой дополнительный коммуникационный аппаратный модуль не требуется. MB_CLIENT может выполнить соединение клиент-сервер, отправить запрос функции Modbus, получить ответ и управлять разъединением от сервера Modbus TCP.

Таблица 13- 127 Типы данных для параметров

Параметр и тип		Тип дан-ных	Описание
REQ	In	Bool	FALSE = Нет коммуникационного запроса Modbus TRUE = Запрос на обмен с Modbus TCP сервером
DISCONNECT	IN	Bool	Параметр DISCONNECT позволяет Вашей программе управлять подключением и отключением к серверу Modbus. Если DISCONNECT = 0 и соединение не существует, то MB_CLIENT попытается выполнить соединение с назначенным IP-адресом и номером порта. Если DISCONNECT = 1 и соединение существует, то предпринимается попытка отключения. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT_ID	IN	UInt	Параметр CONNECT_ID должен однозначно идентифицировать каждое соединение внутри ПЛК. Каждый уникальный экземпляр инструкции MB_CLIENT или MB_SERVER должен содержать уникальный параметр CONNECT_ID.
IP_OCTET_1	IN	USInt	IP-адрес сервера Modbus TCP: Октет 1 8-разрядная часть 32-разрядного IP-адреса IPv4 Modbus TCP сервера, к которому клиент подключится и выполнит обмен по протоколу Modbus TCP.
IP_OCTET_2	IN	USInt	IP-адрес сервера Modbus TCP: Октет 2
IP_OCTET_3	IN	USInt	IP-адрес сервера Modbus TCP: Октет 3
IP_OCTET_4	IN	USInt	IP-адрес сервера Modbus TCP: Октет 4
IP_PORT	IN	UInt	Значение по умолчанию = 502: номер IP порта сервера, к которому клиент попытается подключиться и выполнить обмен, используя TCP/IP.

Параметр и тип		Тип данных	Описание
MB_MODE	IN	USInt	Выбор режима: Назначает тип запроса (чтение, запись или диагностика). Смотрите, таблицу функций Modbus ниже для выяснения деталей.
MB_DATA_ADDR	IN	UDInt	Начальный адрес Modbus: Назначает начальный адрес данных, к которым получит доступ MB_CLIENT. Смотрите последующую таблицу функций Modbus для допустимых адресов.
MB_DATA_LEN	IN	UInt	Длина Modbus данных : Назначает число битов или слов, к которым осуществляет доступ в этом запросе. Смотрите следующую таблицу функций Modbus для допустимых длин.
MB_DATA_PTR	IN_OUT	Variant	Указатель на регистр данных Modbus: регистр помещает в буфер данные, идущие в или приходящие от сервера Modbus. Указатель должен назначать стандартный глобальный DB или адрес в M памяти.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки.
BUSY	OUT	Bool	0 - Никакой операции MB_CLIENT в работе 1 - выполняется операция MB_CLIENT
ERROR	OUT	Bool	ERROR устанавливается в TRUE на один скан после того, как выполнение MB_CLIENT завершилось с ошибкой. Код ошибки в параметре STATUS допустим только в течение одного цикла, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Параметр REQ

FALSE = Нет коммуникационного запроса Modbus

TRUE = Запрос на обмен с сервером Modbus TCP

Если никакой из экземпляров MB_CLIENT не активен и параметр DISCONNECT=0, когда REQ=1, то запускается новый запрос Modbus. Если соединение еще не установлено, то выполняется новое соединение.

Если тот же экземпляр MB_CLIENT будет выполняться снова с DISCONNECT=0 и REQ=1 перед завершением текущего запроса, то последующая передача Modbus не будет выполнена. Однако, как только текущий запрос завершен, новый запрос может быть обработан, если MB_CLIENT выполняется с REQ=1.

Когда текущий коммуникационный запрос MB_CLIENT завершен, бит DONE устанавливается в TRUE на один цикл. Бит DONE может использоваться в качестве временной селектор для упорядочения множественных запросов MB_CLIENT.

Примечание

Согласованность входных данных во время обработки MB_CLIENT

Как только Modbus клиент инициирует Modbus операцию, все входные состояния сохраняются внутри и затем сравниваются при каждом последующем вызове. Сравнение используется, чтобы определить, был ли этот определенный вызов инициатором активного клиентского запроса. Больше чем один вызов MB_CLIENT может быть выполнен, используя общий экземплярный DB.

Важно, чтобы входы не были изменены в течение промежутка времени, когда операция MB_CLIENT активно обрабатывается. Если это правило не соблюдается, то MB_CLIENT не может определить активный экземпляр..

Параметры MB_MODE и MB_DATA_ADDR выбирают коммуникационные функции Modbus

Инструкция MB_CLIENT использует вход MB_MODE, а не функциональный код. MB_DATA_ADDR назначает стартовый адрес удаленных данных Modbus.

Комбинация MB_MODE и MB_DATA_ADDR определяет функциональный код, который используется в фактическом сообщении Modbus. Следующая таблица показывает соответствие между параметром MB_MODE, MB_DATA_ADDR и функцией Modbus.

Таблица 13- 128 Modbus функции

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
0	01	От 1 до 2000	Читать выходные биты: от 1 до 2000 битов за запрос	От 1 до 9999
0	02	От 1 до 2000	Читать входные биты: от 1 до 2000 битов за запрос	От 10001 до 19999
0	03	От 1 до 125	Читать регистры хранения: от 1 до 125 слов за запрос	От 40001 до 49999 или от 400001 до 465535
0	04	От 1 до 125	Читать входные слова: от 1 до 125 слов за запрос	От 30001 до 39999
1	05	1	Записать один выходной бит: один бит на запрос	От 1 до 9999
1	06	1	Записать один регистр хранения: 1 слово за запрос	От 40001 до 49999 или от 400001 до 465535
1	15	От 2 до 1968	Записать несколько выходных битов: от 2 до 1968 битов за запрос	От 1 до 9999
1	16	От 2 до 123	Записать несколько регистров хранения: от 2 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
2	15	От 1 до 1968	Записать один или более выходных битов: от 1 до 1968 битов за запрос	От 1 до 9999
2	16	От 1 до 123	Записать один или более регистров хранения: от 1 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
11	11	0	Читать коммуникационное слово состояния сервера и счетчик событий. Слово состояния указывает занятость (0 = не занято, 0xFFFF = занято). Счетчик событий постепенно увеличивается для каждого успешного завершения сообщения. Как MB_DATA_ADDR, так и MB_DATA_LEN параметры MB_CLIENT игнорируются для этой функции.	
80	08	1	Проверить состояние сервера с помощью диагностического кода 0x0000 (Петлевой тест, сервер отвечает эхом) 1 слово за запрос	

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
81	08	1	Сбросить счетчик событий сервера с помощью диагностического кода 0x000A 1 слово за запрос	
от 3 до 10, от 12 до 79, от 82 до 255			Зарезервировано	

Примечание

MB_DATA_PTR назначает буфер для хранения данных чтения/записи в Modbus TCP сервер

Буфер данных может быть расположен в стандартном глобальном DB или M ячейке памяти.

Для буфера в M памяти используйте формат указателя Any. Это формат R# "Адрес бита" "Тип данных" "Длина", примером будет R#M1000.0 WORD 500.

Параметр MB_DATA_PTR назначает коммуникационный буфер

- Коммуникационные функции MB_CLIENT:
 - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 00001 до 09999)
 - Читать 1-битовые данные из адресов сервера Modbus (от 10001 до 19999)
 - Читать 16-разрядные данные слов из адресов сервера Modbus (от 30001 до 39999) и (от 40001 до 49999)
 - Записывать 16-разрядные данные слов в адреса сервера Modbus (от 40001 до 49999)
- Данные размерностью слово или бит передаются в/из DB или M буфера памяти, назначенного MB_DATA_PTR.
- Если DB назначен, как буфер MB_DATA_PTR, то Вы должны присвоить типы данных всем элементам данных DB.
 - 1-битовый тип данных Bool представляет один Modbus адрес бита
 - 16-битовые типы данных отдельного слова такие как WORD, UInt и Int представляют один Modbus адрес слова
 - 32-разрядные типы данных двойного слова такие как DWORD, DInt и Real представляют два Modbus адреса слова

- MB_DATA_PTR могут быть назначены сложные элементы DB, такие как
 - Стандартные массивы
 - Именованные структуры, где каждый элемент уникален.
 - Именованные сложные структуры, где каждый элемент имеет уникальное имя и 16- или 32-битный типа данных.
- Не требуется, чтобы области данных MB_DATA_PTR были в том же глобальном блоке данных (или М области памяти). Вы можете назначить один блок данных для чтения по Modbus, другой блок данных для записи по Modbus или один блок данных для каждого MB_CLIENT.

Подключение нескольких клиентов

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для Открытых Пользовательских Коммуникаций, разрешенных в ПЛК. Общее количество соединений для ПЛК, включая клиенты и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений для Открытых Пользовательских Коммуникаций (стр. 613). Соединения Modbus TCP могут быть совместно использованы соединениями типа клиент и/или сервер.

Отдельные клиентские соединения должны следовать данным правилам:

- Каждое соединение MB_CLIENT должно использовать определенный экземплярный DB
- Каждое соединение MB_CLIENT должно назначать уникальный IP-адрес сервера
- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения
- Уникальные номера портов IP могут или не могут потребоваться в зависимости от конфигурации сервера

ID Соединения должен быть уникальным для каждого отдельного соединения. Это означает, что отдельный, уникальный ID соединения должен использоваться только с каждым отдельным экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13- 129 Пользовательские доступные статические переменные экземплярного блока данных MB_CLIENT.

Переменная	Тип данных	По умолчанию	Описание
Blocked_Proc_Timeout	Real	3.0	Временной интервал (в секундах) ожидания заблокированного экземпляра клиента Modbus, прежде, чем удалить этот экземпляр, как являющийся ACTIVE. Это может произойти, например, когда был выставлен клиентский запрос и впоследствии приложение прекращает выполнять клиентскую функцию до полного завершения запроса. Максимальный предел для S7-1200 составляет 55 секунд.
MB_Unit_ID	Word	255	Идентификатор модуля Modbus: Сервер Modbus TCP адресуется с помощью IP-адреса. В результате параметр MB_UNIT_ID не используется для адресации Modbus TCP. Параметр MB_UNIT_ID соответствует адресу ведомого устройства в протоколе Modbus RTU. Если сервер Modbus TCP используется в качестве шлюза к протоколу Modbus RTU, MB_UNIT_ID может использоваться для идентификации ведомого устройства, подключенного к последовательной сети. MB_UNIT_ID мог бы использоваться для передачи запроса корректному адресу ведомого устройства Modbus RTU. Некоторые устройства Modbus TCP могут потребовать, чтобы параметр MB_UNIT_ID был в ограниченном диапазоне.
RCV_TIMEOUT	Real	2.0	Время в секундах, в течение которого MB_CLIENT ожидает сервер, чтобы ответить на запрос.
Connected	Bool	0	Указывает на установление соединения с назначенным сервером: 1=подключено, 0=отключено

Таблица 13- 130 Ошибки протокола MB_CLIENT

STATUS (W#16#)	Код отклика клиенту Modbus (B#16#)	Ошибки протокола Modbus
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или доступ вне границ адресного пространства MB_HOLD_REG
8384	03	Ошибка значения данных
8385	03	Диагностический код данных не поддерживается (код функции 08)

Таблица 13- 131 Коды состояния выполнения MB_CLIENT ¹

STATUS (W#16#)	Ошибки параметров MB_CLIENT
7001	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код возвращается только для первого выполнения операции подключения или отключения.
7002	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код будет возвращен для любого последующего выполнения при ожидании завершения операции подключения или отключения
7003	Операция отключения успешно завершена (допустимо только для одного скана ПЛК).

STATUS (W#16#)	Ошибки параметров MB_CLIENT
80C8	Сервер не ответил в назначенное время. MB_CLIENT должен принять ответ, используя ID транзакции, который был первоначально передан в течение назначенного времени, или будет возвращена данная ошибка. Проверьте соединение с серверным устройством Modbus. Эта ошибка возвращается только после того, как были предприняты повторные попытки (если применимо).
8188	Неправильный режим
8189	Неправильный адрес данных
818A	Неправильная длина данных
818B	Недопустимый указатель на область DATA_PTR. Это может быть комбинацией MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Указатель DATA_PTR указывает на область оптимизированного DB (должна быть область стандартного DB или область M памяти)
8200	Порт занят обработкой существующего запроса Modbus.
8380	Принятый кадр Modbus неправильный, или было получено слишком мало байтов.
8387	Назначенный параметр Connection ID отличается от ID, использованного для предыдущих запросов. Может быть только один Connection ID, используемый в каждом экземплярном DB для MB_CLIENT. Этот код также возвращается как внутренняя ошибка, если ID протокола Modbus TCP, полученный от сервера, не 0.
8388	Сервер Modbus возвратил количество данных, которое отличается от затребованного. Этот код применяется только к функциям Modbus 15 или 16.

¹ В дополнение к упомянутым выше ошибкам MB_CLIENT ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV (стр. 664)).

13.8.3.2. Инструкция MB_SERVER (выполнять обмен, используя PROFINET, как Modbus TCP сервер)

Инструкция "MB_SERVER" выполняет обмен как сервер Modbus TCP через порт PROFINET на ЦПУ S7-1200. Инструкция "MB_SERVER" обрабатывает запросы на установление соединения клиента Modbus TCP, получает и обрабатывает запросы Modbus и отправляет ответы.

Чтобы использовать инструкцию, Вам не требуется дополнительный аппаратный модуль.

ЗАМЕТКА
Информация по безопасности Обратите внимание на то, что каждый клиент сети получает доступ на чтение и на запись во входы и выходы образа процесса, в блоки данных или битовую область памяти, определенные регистром хранения Modbus. Доступна опция по ограничению доступа к IP-адресу, чтобы предотвратить несанкционированные операции чтения и записи. Имейте, однако, ввиду, что совместно используемый адрес может также использоваться для несанкционированного доступа.

Таблица 13- 132 Инструкция MB_SERVER

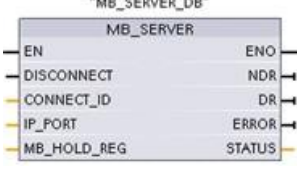
LAD / FBD	SCL	Описание
	<pre>"MB_SERVER_DB" (DISCONNECT:=_bool_in_, CON- NECT_ID:=_uint_in_, IP_PORT:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	MB_SERVER выполняет обмен как сервер Modbus TCP через PROFINET порт ЦПУ S7-1200. Никакой дополнительный аппаратный модуль не требуется. MB_SERVER может утвердить запрос на соединение с клиентом Modbus TCP, принять запрос функции Modbus и отправить ответное сообщение.

Таблица 13- 133 Типы данных для параметров

Параметр и тип		Тип данных	Описание
DISCONNECT	IN	Bool	MB_SERVER пытается выполнить "пассивное" соединение с устройством партнера. Это означает, что сервер пассивно прислушивается к запросу на установление TCP соединения от любого запрашивающего IP-адреса. Если DISCONNECT = 0 и соединение не существует, то пассивное соединение может быть инициировано. Если DISCONNECT = 1 и соединение существует, то инициируется операция отключения. Этот параметр позволяет Вашей программе управлять, когда подтверждать соединение. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT_ID	IN	UInt	CONNECT_ID однозначно определяет каждое соединение в ПЛК. Каждый уникальный экземпляр инструкции MB_CLIENT или MB_SERVER должен содержать уникальный параметр CONNECT_ID.
IP_PORT	IN	UInt	Значение по умолчанию = 502: номер, который идентифицирует IP порт, который будет контролироваться на предмет запроса на установление соединения от клиента Modbus. Следующие номера TCP портов запрещены для пассивного соединения MB_SERVER: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 и 34964.
MB_HOLD_REG	IN_OUT	Variant	Указатель на MB_SERVER Modbus регистр хранения: регистр хранения должен быть либо стандартным глобальным DB, либо адресом М памяти. Эта область памяти используется, чтобы хранить данные, к которым Modbus клиент разрешают доступ, используя регистровые функции: 3 (чтение), 6 (запись) и 16 (запись).
NDR	OUT	Bool	Новые данные готовы: 0 = Нет новых данных, 1 Указывает на то, что новые данные были записаны клиентом Modbus.
DR	OUT	Bool	Чтение данных: 0 = Нет чтения данных, 1 = Указывает на то, что данные были считаны клиентом Modbus.
ERROR	OUT	Bool	ERROR устанавливается в TRUE на один скан после того, как выполнение MB_SERVER завершилось с ошибкой. Код ошибки в параметре STATUS допустим только в течение одного цикла, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

MB_SERVER позволяет входящим кодам функций Modbus (1, 2, 4, 5 и 15) читать / записывать биты/слова непосредственно в образе процесса ввода/вывода. Для кодов функций передачи данных (3, 6 и 16), параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт. Следующая таблица показывает проекцию Modbus адресов в образ процесса в ЦПУ.

Таблица 13- 134 Проекция Modbus адресов в образ процесса

Modbus функции						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес в ЦПУ
01	Считать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Считать биты	Ввод	От 10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Считать слова	Ввод	От 30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Входящие коды функций Modbus сообщений (3, 6, и 16) читают/записывают слова в регистре хранения Modbus, который может располагаться в М памяти или блоке данных. Тип регистра хранения определяется параметром MB_HOLD_REG.

Примечание

Назначение параметра MB_HOLD_REG

Регистр хранения Modbus может находиться в стандартном глобальном DB или ячейке М памяти.

Для регистра хранения Modbus в М памяти используйте формат указателя Any. Это формат R# "Адрес бита" "Тип данных" "Длина", примером будет R#M1000.0 WORD 500.

Следующая таблица показывает примеры проекций Modbus адресов в регистры хранения, используемые для кодов Modbus функций 03 (считать слова), 06 (записать слово) и 16 (записать слова). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом М памяти для каждой из моделей ЦПУ.

Таблица 13- 135 Примеры проекций Modbus адресов в адреса памяти ЦПУ

Modbus адрес	Примеры параметра MB_HOLD_REG		
	R#M100.0 Word 5	R#DB10.DBX0.0 Word 5	"Recipe".ingredient
40001	MW100	DB10.DBW0	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	"Recipe".ingredient[5]

Подключение нескольких серверов

Могут быть созданы несколько подключений серверов. Одиночный ПЛК может установить параллельные соединения с несколькими клиентами Modbus TCP.

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для Открытых Пользовательских Коммуникаций, разрешенных в ПЛК. Общее количество соединений для ПЛК, включая клиенты и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений для Открытых Пользовательских Коммуникаций (стр. 613). Соединения TCP Modbus могут совместно использоваться подключениям типа клиента и сервера.

Отдельные параллельные серверные соединения должны следовать данным правилам:

- Каждое соединение MB_SERVER должно использовать определенный экземплярный DB
- Каждое соединение MB_SERVER должно назначать уникальный номер IP-порта сервера. Поддерживается только 1 соединение на порт
- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения.
- MB_SERVER нужно вызвать индивидуально для каждого соединения (с его соответствующим экземплярным DB).

ID Соединения должен быть уникальным для каждого отдельного соединения. Это означает, что отдельный, уникальный ID соединения должен использоваться только с каждым отдельным экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13- 136 Коды диагностических функций Modbus

Коды диагностических функций Modbus для MB_SERVER		
Коды	Подфункция	Описание
08	0x0000	Эхо-тест возврата данных запроса: MB_SERVER отразит клиенту Modbus слово данных, которое принято.
08	0x000A	Сбросить счетчик коммуникаций: MB_SERVER сбросит счетчик коммуникаций, который используется для Modbus функции 11.
11		Получить счетчик коммуникаций: MB_SERVER использует внутренней счетчик коммуникаций для записи числа успешных запросов на чтение и запись по Modbus, отправленных серверу Modbus. Счетчик не увеличивается на единицу по любому запросу на Функцию 8, Функцию 11 или любому запросу, который приводит к ошибке связи. Широковещательная функция недоступна для Modbus TCP, так как только одно клиент-серверное соединение существует в любой момент времени.

Переменные MB_SERVER

Эта таблица показывает общедоступные статические переменные, которые хранятся в экземплярном блоке данных MB_SERVER и могут использоваться в Вашей программе.

Таблица 13- 137 Общедоступные статические переменные MB_SERVER

Переменная	Тип данных	По умолчанию	Описание
HR_Start_Offset	Word	0	Назначает начальный адрес регистра хранения Modbus
Request_Count	Word	0	Количество всех запросов полученных этим сервером.
Server_Message_Count	Word	0	Количество запросов полученных для этого определенного сервера.
Xmt_Rcv_Count	Word	0	Количество передач или приемов, в которых произошли ошибки. Кроме того, увеличивается на единицу, если принято сообщение, которое является недопустимым сообщением Modbus.
Exception_Count	Word	0	Специфические ошибки Modbus, которые требуют возвращенного исключения
Success_Count	Word	0	Количество запросов, полученных для этого определенного сервера, в которых нет ошибок протокола.
Connected	Bool	0	Указывает на состояние соединения с назначенным клиентом: 1=подключено, 0=отключено

Ваша программа может записать данные в HR_Start_Offset и управлять операциями сервера Modbus. Другие переменные могут быть считаны, чтобы контролировать состояние Modbus.

HR_Start_Offset

Адреса регистров хранения Modbus начинаются с 40001. Эти адреса соответствуют началу адресов памяти ПЛК для регистров хранения. Однако Вы можете использовать переменную "HR_Start_Offset", чтобы определить начало адресов регистров хранения Modbus иное, чем 40001.

Например, если регистр хранения начинается в MW100 и содержит 100 слов. Смещение 20 определяет начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше чем 40021 или больше, чем 40119 вызовет ошибку адресации.

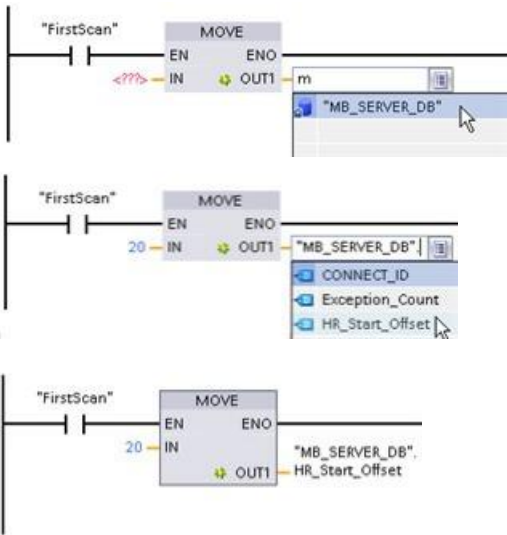
Таблица 13- 138 Примеры адресации регистров хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Modbus адрес (Слово)	40001	40099
	S7-1200 адрес	MW100	MW298
20	Modbus адрес (Слово)	40021	40119
	S7-1200 адрес	MW100	MW298

HR_Start_Offset представляет собой данные типа word в экземплярном блоке данных MB_SERVER, которые назначают начальный адрес регистра хранения Modbus. Вы можете установить эту общедоступную статическую переменную при помощи выпадающего списка помощника параметра, после того, как MB_SERVER будет помещен в Вашу программу.

13.8 Унаследованные Modbus TCP коммуникации

Например, после того, как Вы помещаете MB_SERVER в LAD-сегмент, Вы можете перейти к предыдущему сегменту и назначить HR_Start_Offset. Начальный адрес должен быть назначен до выполнения MB_SERVER.



- Ввод переменной сервера Modbus, используя имя DB по умолчанию:
1. Установите курсор в поле параметра и введите символ m.
 2. Выберите "MB_SERVER_DB" из выпадающего списка имен DB.
 3. Выберите "MB_SERVER_DB.HR_Start_Offset" из выпадающего списка переменных DB.

Таблица 13- 139 Коды состояния выполнения MB_SERVER ¹

STATUS (W#16#)	Код ответа серверу Modbus	Ошибки протокола Modbus
7001		MB_SERVER ожидает клиента Modbus, чтобы соединиться с назначенным TCP портом. Этот код возвращается при первом выполнении операции подключения или отключения.
7002		MB_SERVER ожидает клиента Modbus, чтобы соединиться с назначенным TCP портом. Этот код возвращается для любого последующего выполнения при ожидании завершения работы подключения или отключения.
7003		Операция отключения успешно завершилась (Допустимо только для одного скана ПЛК).
8187		Недопустимый указатель на MB_HOLD_REG: область слишком маленькая
818C		Указатель MB_HOLD_REG ссылается на область в оптимизированном DB (область должна быть в стандартном глобальном DB или M памяти) или тайм-аут блокированного процесса превышает предел в 55 секунд. (Специфический для S7-1200)
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или обращение вне границ адресного пространства MB_HOLD_REG
8384	03	Ошибка значения данных
8385	03	Код диагностики данных не поддерживается (код функции 08)

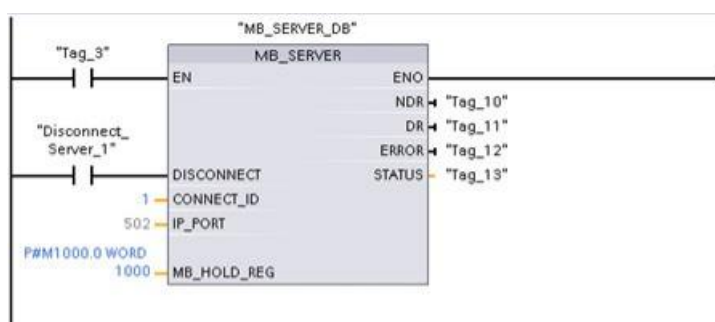
¹ В дополнение к упомянутым выше ошибкам MB_SERVER ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV (стр. 664)).

13.8.4. Пример: Несколько TCP соединений для унаследованных MB_SERVER

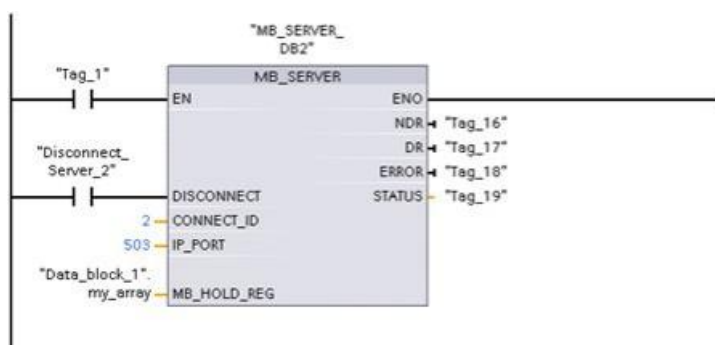
У Вас может быть несколько соединений сервера Modbus TCP. Чтобы сделать это, MB_SERVER должен выполняться независимо для каждого соединения. Каждое соединение должно использовать независимые экземпляры DB, ID соединения и IP-порт. S7-1200 позволяет только одно соединение на IP-порт.

Для лучшей производительности MB_SERVER должен выполняться каждый программный цикл для каждого соединения.

Сегмент 1: Подключение #1 с независимыми IP_PORT, ID соединения и экземпляром DB



Сегмент 2: Подключение #2 с независимыми IP_PORT, ID соединения и экземпляром DB



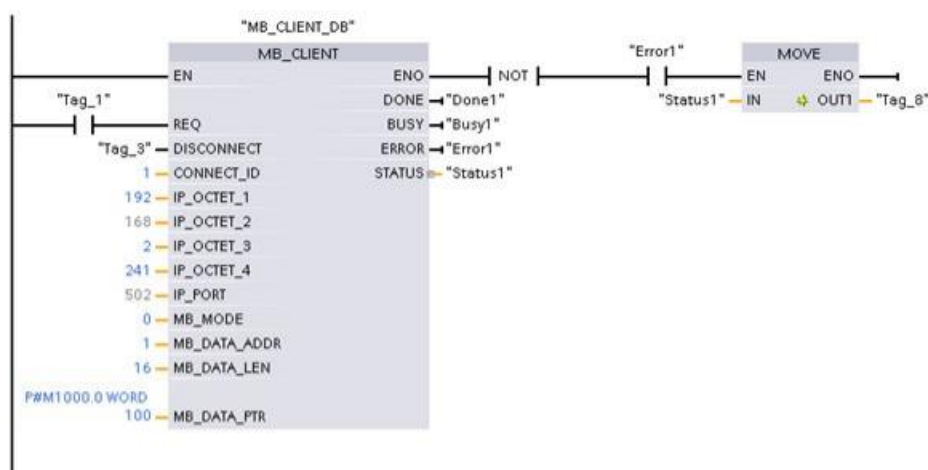
13.8.5. Пример: Унаследованный MB_CLIENT 1: Несколько запросов с общим TCP соединением

Несколько клиентских запросов Modbus могут быть отправлены по тому же соединению. Чтобы сделать это, используйте тот же экземплярный DB, ID соединения и номер порта.

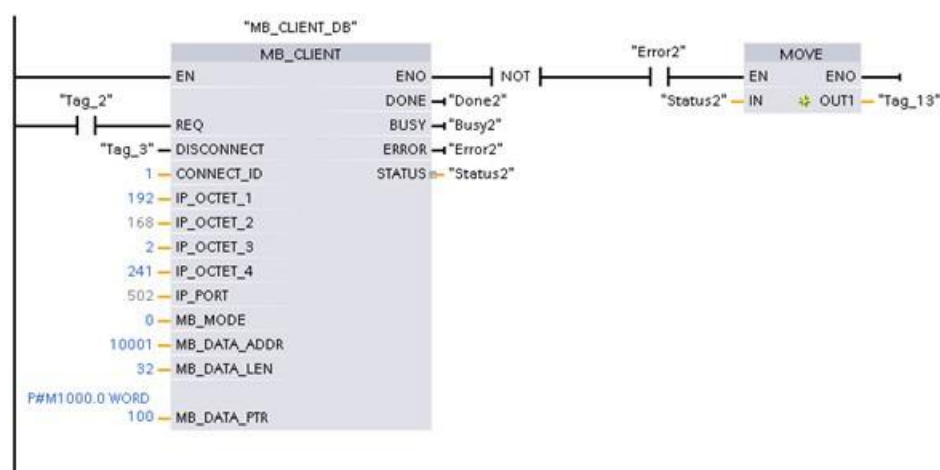
Только 1 клиент может быть активным в любой момент времени. Как только клиент завершает выполнение, следующий клиент может начать выполнение. Ваша программная логика ответственна за последовательность выполнения.

Пример демонстрирует обоих клиентов, выполняющих запись в одну область памяти. Возвращенная ошибка захватывается, что является необязательным.

Сегмент 1: Modbus функция 1- Читать 16 выходных битов образа процесса



Сегмент 2: Modbus функция 2 - Читать 32 входных бита образа процесса



13.8.6. Пример: Унаследованный MB_CLIENT 2: Несколько запросов с различными TCP соединениями

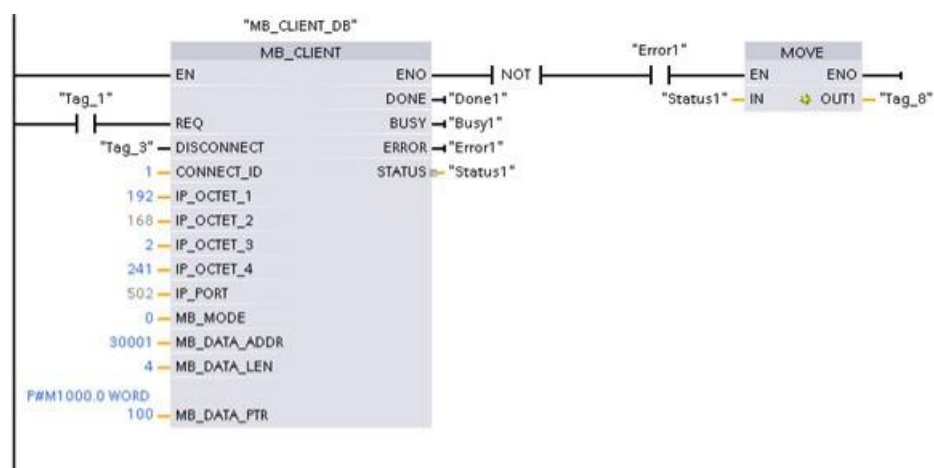
Клиентские запросы Modbus TCP могут быть отправлены по различным соединениям. Чтобы сделать это, должны использоваться различные экземпляры DB, IP-адреса и ID соединения.

Номер порта должен отличаться, если соединения устанавливаются с тем же сервером Modbus. Если соединения находятся на различных серверах, нет никаких ограничений на номер порта.

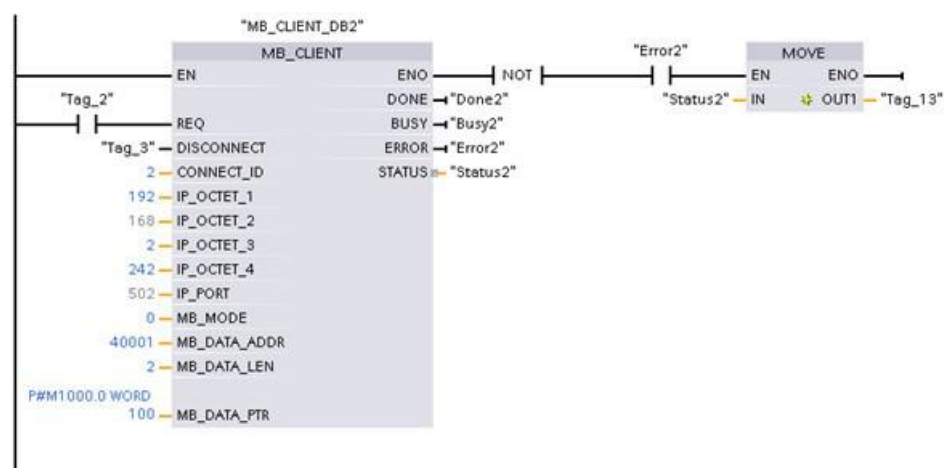
Пример демонстрирует двух клиентов Modbus TCP, выполняющих запись в ту же область памяти. Кроме того, возвращенная ошибка захватывается, что является необычным.

Сегмент 1:

Modbus функция 4 – Читать входные слова (в памяти S7-1200)



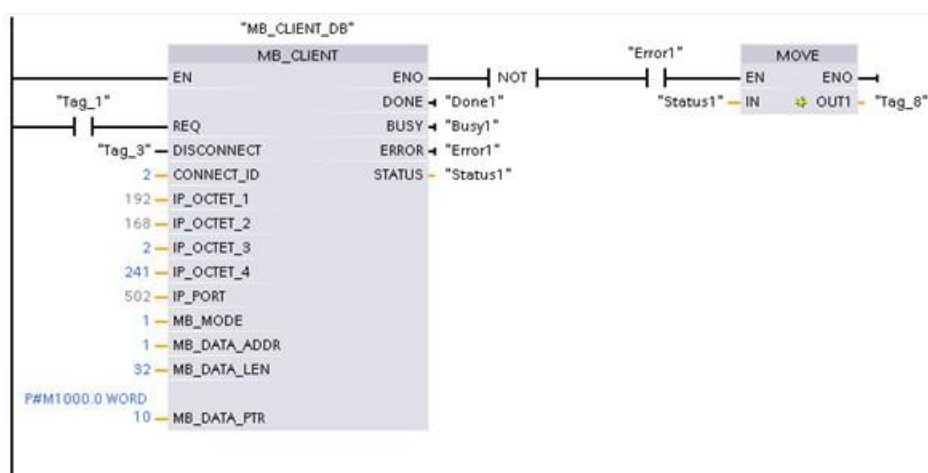
Network 2: Modbus функция 3 – Читать слова регистра хранения из сервера Modbus TCP.



13.8.7. Пример: Унаследованный MB_CLIENT 3: Запрос записи образа по выводу

Этот пример показывает запрос клиента Modbus на запись в образ по выводу для S7-1200.

Сегмент 1: Modbus функция 15 – Записать биты в образ по выводу для S7-1200.

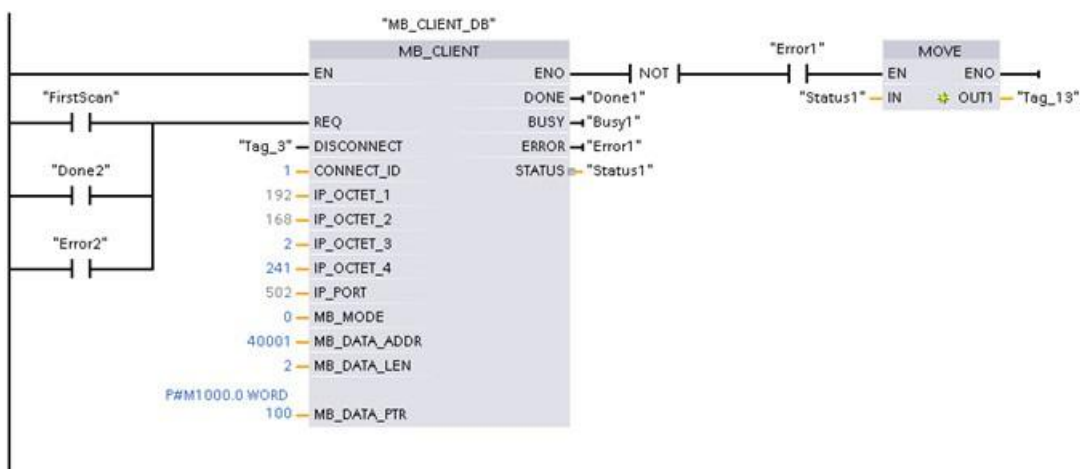


13.8.8. Пример: Унаследованный MB_CLIENT 4: Координация нескольких запросов

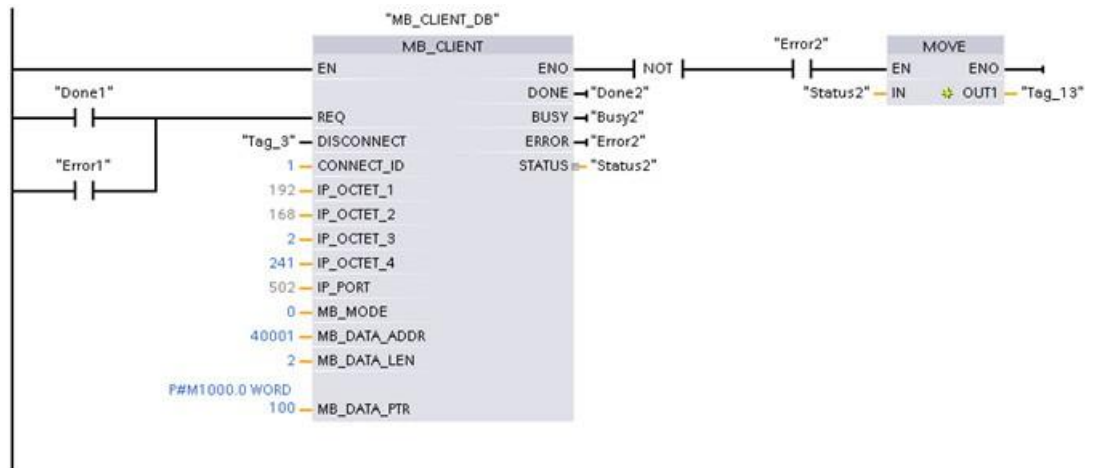
Вы должны обеспечить, чтобы каждый отдельный Modbus TCP запрос завершал выполнение. Эту координацию должна осуществлять Ваша программная логика. Приведенный ниже пример демонстрирует, как выходы первого и второго клиентских запросов могут управлять последовательностью выполнения.

Пример показывает обоих клиентов, выполняющих запись в ту же область памяти. Кроме того, возвращенная ошибка захватывается, что является необязательным.

Сегмент 1: Modbus функция 3 - Считать слова регистра хранения



Сегмент 2: Modbus функция 3 - Считать слова регистра хранения



13.9 Унаследованные Modbus RTU коммуникации (только CM/CB 1241)

13.9.1. Обзор

До выпуска STEP 7 V13 SP1 и ЦПУ S7-1200 V4.1 Modbus RTU коммуникационные инструкции существовали с отличными именами, и в некоторых случаях, с немного отличающимися интерфейсами. Общие принципы применимы к обоим наборам инструкций. Обратитесь к отдельным унаследованным Modbus RTU инструкциям для получения сведений о программировании.

13.9.2. Выбор версии Modbus RTU инструкций

Существует две версии Modbus RTU инструкций доступных в STEP 7:

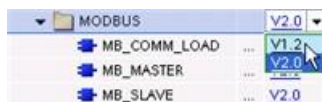
- Версия 1 была изначально доступна в STEP 7 Basic V10.5.
- Версия 2 доступна со STEP 7 Basic/Professional V11. Реализация версии 2 добавляет параметры REQ и DONE в MB_COMM_LOAD. Кроме того, параметр MB_ADDR для MB_MASTER и MB_SLAVE теперь позволяет использование UInt для расширенной адресации.

Для совместимости и простоты миграции, Вы можете выбирать какую версию инструкции вставлять в Вашу пользовательскую программу.

Вы не можете использовать обе версии инструкций с тем же модулем, но два различных модуля могут использовать различные версии инструкций. Не используйте обе версии инструкции 1.x и 2.y в одной программе ЦПУ. Ваши программные Modbus RTU инструкции должны иметь один и тот же старший номер версии (1.x, 2.y или V.z). Отдельные инструкции в рамках группы старшей версии могут иметь различные младшие версии (1.x).



Щелкните по значку в дереве инструкций на карте задач, чтобы включить заголовки и столбцы дерева инструкций.



Чтобы изменить версию Modbus RTU инструкций, выберите версию из выпадающего списка. Вы можете выбрать группу или отдельные инструкции.

Когда Вы используете дерево инструкций, чтобы поместить Modbus RTU инструкцию в Вашу программу, новый экземпляр FB создается в дереве проекта. Вы можете видеть новый экземпляр FB в дереве проекта под PLC_x > Program blocks > System blocks > Program resources.

Чтобы проверить версию Modbus инструкции в программе, Вы должны проверить свойства дерева проекта, а не свойства прямоугольника, выведенного на экран в редакторе программы. Выберите Modbus TCP экземпляр FB в дереве проекта, щелкните правой кнопкой, выберите "Properties" и выберите страницу "Information", чтобы увидеть номер версии Modbus инструкции.

13.9.3. Унаследованные Modbus RTU инструкции

13.9.3.1. Инструкция MB_COMM_LOAD (Сконфигурировать порт на PtP модуле для Modbus RTU)

Таблица 13- 140 Инструкция MB_COMM_LOAD

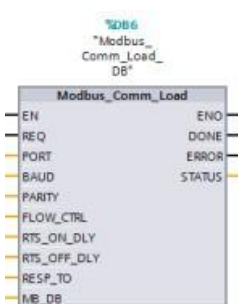
LAD / FBD	SCL	Описание
	<pre>"Modbus_Comm_Load_DB"(REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_uint_in_, PARITY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	Инструкция MB_COMM_LOAD конфигурирует PtP порт для обмена по протоколу Modbus RTU. Аппаратные опции порта Modbus RTU: Установка до трех CM (RS485 или RS232) плюс одна CB (R4845). Экземплярный блок данных назначается автоматически, когда Вы помещаете инструкцию MB_COMM_LOAD в Вашу программу.

Таблица 13- 141 Типы данных для параметров

Параметр и тип	Тип данных	Описание
REQ	IN	Bool
PORT	IN	Port
BAUD	IN	UDInt
PARITY	IN	UInt
FLOW_CTRL ¹	IN	UInt

13.9 Унаследованные Modbus RTU коммуникации (только CM/CB 1241)

Параметр и тип		Тип данных	Описание
RTS_ON_DLY ¹	IN	UInt	Выбор задержки RTS ON: 0 – Никакой задержки от активного RTS до передачи первого символа сообщения (по умолчанию) - От 1 до 65535 – Задержка в миллисекундах от активного RTS до передачи первого символа сообщения (не применимо к RS485 портам). Задержки RTS должны быть применены независимо от выбора FLOW_CTRL.
RTS_OFF_DLY ¹	IN	UInt	Выбор задержки RTS OFF: 0 – Никакой задержки от последнего переданного символа до перехода RTS в неактивное состояние (по умолчанию) - От 1 до 65535 – Задержка в миллисекундах от последнего переданного символа до перехода RTS в неактивное состояние (не применимо к RS485 портам). Задержки RTS должны быть применены независимо от выбора FLOW_CTRL.
RESP_TO ¹	IN	UInt	Тайм-аут ответа: Время в миллисекундах, отводимое MB_MASTER для ответа ведомому устройству. Если ведомое устройство не ответит в течение этого периода времени, то MB_MASTER повторит запрос или завершит запрос с ошибкой, когда было отправлено определенное количество повторений. От 5 мс до 65535 мс (значение по умолчанию = 1000 мс).
MB_DB	IN	Variant	Ссылка на экземплярный блок данных, используемый инструкциями MB_MASTER или MB_SLAVE. После того, как MB_MASTER или MB_SLAVE помещены в Вашу программу, идентификатор DB появляется в выпадающем списке помощника параметра, доступном в подключении MB_DB.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки. (Только версия 2.0)
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

¹ Дополнительные параметры для MB_COMM_LOAD (Версии 2.x или более поздней). Щелкните по стрелке у основания прямоугольника LAD/FBD, чтобы развернуть его и отобразить эти параметры.

MB_COMM_LOAD выполняется, чтобы сконфигурировать порт для протокола Modbus RTU. Как только порт сконфигурирован для протокола Modbus RTU, он может использоваться только инструкциями MB_MASTER или MB_SLAVE.

Одно выполнение MB_COMM_LOAD должно использоваться, чтобы сконфигурировать каждый коммуникационный порт, который используется для Modbus коммуникаций. Назначьте MB_COMM_LOAD уникальный экземплярный DB для каждого порта, который Вы используете. Вы можете установить до трех коммуникационных модулей (RS232 или RS485) и одну коммуникационную плату (RS485) в ЦПУ. Вызовите MB_COMM_LOAD из ОВ запуска и выполните ее однократно или используйте системный флаг первого скана (стр. 103), чтобы инициировать однократный вызов. Выполните MB_COMM_LOAD снова, только если необходимо изменить коммуникационные параметры, такие как скорость обмена или четность.

Экземплярный блок данных назначается для MB_MASTER или MB_SLAVE, когда Вы помещаете эти инструкции в свою программу. На этот экземплярный блок данных ссылаются, когда Вы определяете параметр MB_DB для инструкции MB_COMM_LOAD.

Переменные блока данных MB_COMM_LOAD

Следующая таблица показывает общедоступные статические переменные, сохраненные в экземплярном DB для MB_COMM_LOAD, которые могут использоваться в Вашей программе.

Таблица 13- 142 Статические переменные в экземплярном DB

Переменная	Тип данных	Описание
ICHAR_GAP	UInt	Задержка разрыва между символами. Этот параметр определен в миллисекундах и используется, чтобы увеличить ожидаемое количество времени между принятыми символами. Соответствующее число битовых интервалов для этого параметра добавляется к значению по умолчанию для Modbus равному 35 битовым интервалам (3.5 x время символа).
RETRIES	UInt	Число повторений, которые ведущее устройство будет пытаться сделать прежде, чем вернуть код ошибки отсутствия ответа "0x80C8".
STOP_BITS	USInt	Число стоповых битов, используемых в синхронизации каждого символа. Допустимыми значениями являются 1 и 2.

Таблица 13- 143 Коды состояния выполнения MB_COMM_LOAD ¹

STATUS (W#16#)	Описание
0000	Без ошибок
8180	Недопустимое значение ID порта (неправильный идентификатор порта/аппаратных
8181	Недопустимое значение скорости обмена
8182	Недопустимое значение четности
8183	Недопустимое значение управления потоком
8184	Недопустимое значение тайм-аута ответа (тайм-аут ответа меньше, чем 5 мс)
8185	Параметр MB_DB не является экземплярным блоком данных инструкции MB_MASTER или MB_SLAVE.

¹ В дополнение к упомянутым выше ошибкам MB_COMM_LOAD ошибки могут быть возвращены из базовых инструкций PtP коммуникаций.

13.9 Унаследованные Modbus RTU коммуникации (только CM/CB 1241)

13.9.3.2. Инструкция MB_MASTER (Выполнять обмен, используя PtP порт в качестве ведущего устройства Modbus RTU)

Таблица 13- 144 Инструкция MB_MASTER

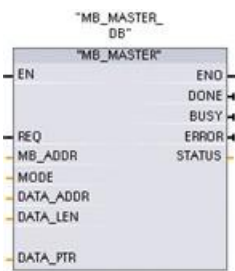
LAD / FBD	SCL	Описание
	<pre>"MB_MASTER_DB" (REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	Инструкция MB_MASTER выполняет обмен, как ведущее устройство Modbus, используя порт, который был сконфигурирован предыдущим выполнением инструкции MB_COMM_LOAD. Экземплярный блок данных назначается автоматически, когда Вы помещаете инструкцию MB_MASTER в свою программу. Этот экземплярный блок данных MB_MASTER используется, когда Вы определяете параметр MB_DB для инструкции MB_COMM_LOAD.

Таблица 13- 145 Типы данных для параметров

Параметр и тип		Тип данных	Описание
REQ	IN	Bool	0= Нет запроса 1= Запрос на передачу данных ведомому устройству Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции Modbus RTU: Стандартный диапазон адресов (от 1 до 247) Расширенный диапазон адресов (от 1 до 65535) Значение 0 зарезервировано для широковещательной передачи сообщений всем ведомым устройствам Modbus. Только коды функций Modbus 05, 06, 15 и 16 поддерживаются широковещательной передачей.
MODE	IN	USInt	Выбор режима: Определяет тип запроса (чтение, запись или диагностика). Смотрите таблицу функций Modbus, расположенную ниже для получения подробностей.
DATA_ADDR	IN	UDInt	Начальный адрес в ведомом устройстве: Определяет начальный адрес данных в ведомом устройстве Modbus, к которым осуществляется доступ. Смотрите таблицу функций Modbus, расположенную ниже для получения допустимых адресов.
DATA_LEN	IN	UInt	Длина данных: Определяет число битов или слов, к которым осуществляется доступ в этом запросе. Смотрите таблицу функций Modbus, расположенную ниже для получения допустимых длин.
DATA_PTR	IN	Variant	Указатель данных: Указывает на M или DB адрес (DB со стандартным доступом) для записываемых или считываемых данных.
DONE	OUT	Bool	Бит DONE устанавливается в TRUE на один скан, после того, как последний запрос был завершен без ошибки.
BUSY	OUT	Bool	0 – Нет выполняемых операций MB_MASTER 1 – MB_MASTER операция выполняется
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Правила коммуникаций ведущего устройства Modbus

- MB_COMM_LOAD должен быть выполнен, чтобы сконфигурировать порт, прежде чем инструкция MB_MASTER сможет связаться с этим портом.
- Если порт должен использоваться, чтобы инициировать запросы ведущего устройства Modbus, то порт не должен использоваться MB_SLAVE. Один или более экземпляров выполнения MB_MASTER могут использоваться с этим портом, но любое выполнение MB_MASTER должно использовать один и тот же экземплярный DB MB_MASTER для этого порта.
- Инструкции Modbus не используют события прерывания коммуникаций, чтобы управлять коммуникационным процессом. Ваша программа должна опросить инструкцию MB_MASTER на предмет завершения передачи и приема.
- Рекомендуется, чтобы Вы вызывали любое выполнение MB_MASTER для данного порта из OB программного цикла. Инструкции MB_MASTER могут выполняться только в одном из программных циклов или циклических/с временной задержкой уровней выполнения. Они не должны выполняться на обоих приоритетных уровнях выполнения. Вытеснение инструкции MB_MASTER другой инструкцией MB_MASTER на более высоком уровне приоритета выполнения приведет к нештатной работе. Инструкции MB_MASTER не должны выполняться в уровнях выполнения, связанных с запуском, диагностикой или ошибкой времени выполнения.
- Как только инструкция ведущего устройства инициирует передачу, этот экземпляр должен постоянно выполняться с активным входом EN, до возврата состояния DONE=1 или ERROR=1. Определенный экземпляр MB_MASTER считают активным, пока не наступает одно из этих двух событий. В то время как исходный экземпляр активен, любой вызов другого экземпляра с активным входом REQ приведет к ошибке. Если непрерывное выполнение исходного экземпляра останавливается, состояние запроса остается активным на время, определенное статической переменной "Blocked_Proc_Timeout". Как только это время истекает, следующая инструкция ведущего устройства, вызванная с активным входом REQ, станет активным экземпляром. Это препятствует тому, чтобы единственный экземпляр ведущего устройства Modbus монополизировал или заблокировал доступ к порту. Если исходный активный экземпляр не будет активирован в течение времени, определенного статической переменной "Blocked_Proc_Timeout", то следующее выполнение этого экземпляра (с не установленным REQ) сбросит активное состояние. Если REQ установлен, то это выполнение инициирует новый запрос ведущего устройства, как будто никакой другой экземпляр не был активен.

Параметр REQ

0 = Нет запроса; 1 = Запрос на передачу данных ведомому устройству Modbus

Вы можете управлять этим входом либо с помощью уровня либо активируемого фронтом контакта. Каждый раз, когда этот вход активирован, конечный автомат запускается, чтобы гарантировать, что никакому другому MB_MASTER, использующему тот же экземплярный DB, не будет разрешено выставить запрос, пока текущий запрос не завершен. Состояния всех других входов захватываются и сохраняются внутри для текущего запроса, пока не получен ответ или не обнаружена ошибка.

Если тот же экземпляр MB_MASTER выполняется снова с входом REQ = 1 до завершения текущего запроса, то никакие последующие передачи не выполняются. Однако, когда запрос завершен, новый запрос выставляется каждый раз, когда MB_MASTER выполняется вновь с входом REQ = 1.

Параметры DATA_ADDR и MODE выбирают тип функции Modbus

DATA_ADDR (начальный Modbus адрес в ведомом устройстве): Определяет начальный адрес данных в ведомом устройстве Modbus, к которым осуществляют доступ.

Инструкция MB_MASTER использует вход MODE, а не вход Function Code. Комбинация MODE и адреса Modbus определяет код функции, который используется в фактическом сообщении Modbus. Следующая таблица показывает соответствие между параметром MODE, кодом функции Modbus и диапазоном адресов Modbus.

Таблица 13- 146 Modbus функции

MODE	Modbus функция	Длина данных	Операция и данные	Modbus адрес
0	01	От 1 до 2000 От 1 до 1992 ¹	Читать выходные биты: От 1 до (1992 или 2000) битов за запрос	От 1 до 9999
0	02	От 1 до 2000 1 до 1992 ¹	Читать входные биты: От 1 до (1992 или 2000) битов за запрос	От 10001 до 19999
0	03	От 1 до 125 От 1 до 124 ¹	Читать регистры хранения: От 1 до (124 или 125) слов за запрос	От 40001 до 49999 или От 400001 до 465535
0	04	От 1 до 125 От 1 до 124 ¹	Читать входные слова: От 1 до (124 или 125) слов за запрос	От 30001 до 39999
1	05	1	Записать один выходной бит: один бит на запрос	От 1 до 9999
1	06	1	Записать один регистр хранения: 1 слово за запрос	От 40001 до 49999 или От 400001 до 465535
1	15	От 2 до 1968 От 2 до 1960 ¹	Записать несколько выходных битов: От 2 до (1960 или 1968) битов за запрос	От 1 до 9999
1	16	От 2 до 123 От 2 до 122 ¹	Записать несколько регистров хранения: От 2 до (122 или 123) слов за запрос	От 40001 до 49999 или От 400001 до 465535
2	15	От 1 до 1968 От 2 до 1960 ¹	Записать один или более выходных битов: От 1 до (1960 или 1968) битов за запрос	От 1 до 9999
2	16	От 1 до 123 От 1 до 122 ¹	Записать один или более регистров хранения: От 1 до (122 или 123) слов за запрос	От 40001 до 49999 или От 400001 до 465535
11	11	0	Читать коммуникационное слово состояния сервера и счетчик событий. Слово состояния указывает занятость (0 = не занято, 0xFFFF = занято). Счетчик событий постепенно увеличивается для каждого успешного завершения сообщения. Как MB_DATA_ADDR, так и MB_DATA_LEN параметры MB_MASTER игнорируются для этой функции.	
80	08	1	Проверить состояние ведомого устройства с помощью диагностического кода 0x0000 (Петлевой тест, сервер отзывается эхом) 1 слово за запрос	

MODE	Modbus функция	Длина данных	Операция и данные	Modbus адрес
81	08	1	Сбросить счетчик событий сервера с помощью диагностического кода 0x000A 1 слово за запрос	
3 ... 10, 12 ... 79, 82 ... 255			Зарезервировано	

¹ Для режима "расширенной адресации" максимальные длины данных уменьшены на 1 байт или 1 слово в зависимости от типа данных, используемого функцией.

Параметр DATA_PTR

Параметр DATA_PTR указывает на адрес в DB или M области, который записывается или считывается. Если Вы используете блок данных, то Вы должны создать глобальный блок данных, который обеспечивает хранение данных для чтения и записи в ведомые устройства Modbus.

Примечание

Тип блока данных DATA_PTR должен позволять прямую адресацию

Блок данных должен позволять как прямую (абсолютную), так и символьную адресацию. Когда Вы создаете блок данных, должен быть выбран атрибут доступа "Standard".

Структуры блока данных для параметра DATA_PTR

- Следующие типы данных допустимы для **чтений слов** из Modbus адресов с 30001 по 39999, с 40001 по 49999 и с 400001 по 465536, а также для **записи слов** в Modbus адреса с 40001 по 49999 и с 400001 по 465536.
 - Стандартный массив типов данных WORD, UINT или INT.
 - Именованная структура WORD, UINT или INT, где каждый элемент имеет уникальное имя и 16-битный тип данных.
 - Именованная сложная структура, где каждый элемент имеет уникальное имя и 16- или 32-битный тип данных.
- Для **чтения** и записи **битов** из адресов Modbus с 00001 по 09999 и чтения битов с 10001 по 19999.
 - Стандартный массив булевых типов данных.
 - Именованная булева структура уникально именованных булевых переменных.
- Несмотря на отсутствие требования, рекомендуется, чтобы у каждой инструкции MB_MASTER была своя собственная отдельная область памяти. Причина этой рекомендации кроется в том, что есть большая вероятность повреждения данных, если несколько инструкций MB_MASTER читают и пишут в ту же самую область памяти.
- Не обязательно, чтобы области данных DATA_PTR были в том же глобальном блоке данных. Вы можете создать один блок данных с несколькими областями для чтений Modbus, один блок данных для Modbus записей или один блок данных для каждой ведомой станции.

Переменные блока данных ведущего устройства Modbus

Следующая таблица показывает общедоступные статические теги в экземплярном DB для MB_MASTER, которые Вы можете использовать в своей программе.

Таблица 13- 147 Статические теги в экземплярном DB

Тег	Тип данных	Стандарт	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах) ожидания заблокированного экземпляра MB_MASTER прежде, чем удалить этот экземпляр, как ACTIVE. Это может произойти, например, если выставляется запрос MB_MASTER, а затем программа останавливается, чтобы вызвать функцию MB_MASTER до того, как она полностью не завершит запрос. Значение времени должно быть больше 0 и меньше 55 секунд или возникает ошибка. Значение по умолчанию составляет 5 секунд.
Extended_Addressing	Bool	FALSE	Конфигурирует одно- или двухбайтовую адресацию ведомой станции. Значение по умолчанию = 0. • (0= однобайтовый адрес, 1= двухбайтовый адрес)

Ваша программа может записать значения в переменные Blocked_Proc_Timeout и Extended_Addressing, чтобы управлять операциями ведущего устройства Modbus. Смотри описание HR_Start_Offset и Extended_Addressing для примера того, как использовать эти переменные в программном редакторе и получения деталей относительно расширенной адресации Modbus (стр.1035).

Коды состояния

Таблица 13- 148 Коды состояния выполнения MB_MASTER (ошибки коммуникаций и конфигурации) ¹

STATUS (W#16#)	Описание
0000	Без ошибок
80C8	Тайм-аут ведомого устройства. Проверьте скорость обмена, четность и проводные соединения ведомого устройства.
80D1	Получатель выставил запрос на управление потоком, чтобы приостановить активную передачу и повторно не активировал передачу в течение указанного времени ожидания. Эта ошибка также генерируется во время аппаратного управления потоком, когда получатель не утверждает CTS в течение указанного времени ожидания.
80D2	Запрос на передачу был прерван, так как никакой сигнал DSR не получен от DCE.
80E0	Сообщение было прервано, так как приемный буфер заполнен.
80E1	Сообщение было прервано в результате ошибки четности.
80E2	Сообщение было прервано в результате ошибки кадровой синхронизации.
80E3	Сообщение было прервано в результате ошибки переполнения.
80E4	Сообщение было прервано в результате указания длины, превышающей общий размер буфера.
8180	Недопустимое значение ID порта или ошибка в инструкции MB_COMM_LOAD
8186	Недопустимый адрес станции Modbus
8188	Недопустимый режим определен для широковещательного запроса
8189	Недопустимое значение адреса данных
818A	Недопустимое значение длины данных
818B	Недопустимый указатель на локальный источник/ назначение данных: Размер неправильный

STATUS (W#16#)	Описание
818C	Недопустимый указатель для DATA_PTR или недопустимый Blocked_Proc_Timeout. Область данных должна быть представлена DB (который позволяет как символьную, так и абсолютную адресацию) или M памятью
8200	Порт занят обработкой запроса передачи.

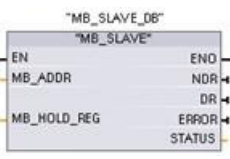
Таблица 13- 149 Коды состояния выполнения MB_MASTER (ошибки Modbus протокола) ¹

STATUS (W#16#)	Ответ ведомого устройства	Ошибки Modbus протокола
8380	-	Ошибка CRC
8381	01	Код функции не поддерживается
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или адрес вне допустимого диапазона области DATA_PTR
8384	Более чем 03	Ошибка значения данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)
8386	-	Код функции в ответе не соответствует коду в запросе.
8387	-	Ответило неправильное ведомое устройство
8388	-	Ответ ведомого устройства на запрос записи неправильный. Запрос записи, возвращенный ведомым устройством, не соответствует тому, что фактически отправило ведущее устройство.

¹ В дополнение к упомянутым выше ошибкам MB_MASTER ошибки могут быть возвращены из базовых коммуникационных PtP инструкций.

13.9.3.3. Инструкция MB_SLAVE (Выполнять обмен, используя PtP порт в качестве ведомого устройства Modbus RTU)

Таблица 13- 150 Инструкция MB_SLAVE

LAD / FBD	SCL	Описание
	<pre>"MB_SLAVE_DB" (MB_ADDR:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ER- ROR=>_bool_out_, STA- TUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	Инструкция MB_SLAVE позволяет Вашей программе выполнять обмен в качестве ведомого устройства Modbus RTU через PtP порт на CM (RS485 или RS232) и CB (RS485). Когда удаленное ведущее устройство Modbus RTU выставляет запрос, Ваша пользовательская программа отвечает на запрос выполнением MB_SLAVE. STEP 7 автоматически создает экземплярный DB, когда Вы вставляете инструкцию. Используйте это MB_SLAVE_DB имя, когда Вы задаете параметр MB_DB для инструкции MB_COMM_LOAD.

13.9 Унаследованные Modbus RTU коммуникации (только CM/CB 1241)

Таблица 13- 151 Типы данных для параметров

Параметр и тип		Тип данных	Описание
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес ведомой станции Modbus: Стандартный диапазон адресов (от 1 до 247) Расширенный диапазон адресов (от 0 до 65535)
MB_HOLD_REG	IN	Variant	Указатель на DB регистра хранения Modbus: регистр хранения Modbus может быть М памятью или блоком данных.
NDR	OUT	Bool	Новые данные готовы: 0 – Нет новых данных 1 – Указывает на то, что новые данные были записаны ведущим устройством Modbus
DR	OUT	Bool	Чтение данных: 0 – Нет чтения данных 1 – Указывает на то, что данные были считаны ведущим устройством Modbus
ERROR	OUT	Bool	Бит ERROR устанавливается в TRUE на один скан, после того, как последний запрос был завершен с ошибкой. Значение кода ошибки в параметре STATUS действительно только во время одного скана, когда ERROR = TRUE.
STATUS	OUT	Word	Код состояния выполнения

Коды функций Modbus (1, 2, 4, 5 и 15) могут читать и записывать биты и слова непосредственно в образе процесса по вводу и по выводу. Для этих кодов функций параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт.

Следующая таблица показывает проекцию Modbus адресов в образ процесса в ЦПУ.

Таблица 13- 152 Проекция Modbus адресов в образ процесса

Modbus функции						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес в ЦПУ
01	Считать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Считать биты	Ввод	От 10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Считать слова	Ввод	От 30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	От 1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Коды коммуникационных функций Modbus (3, 6, и 16) используют регистр хранения Modbus, который может располагаться в М памяти или блоке данных. Тип регистра хранения определяется параметром MB_HOLD_REG инструкции MB_SLAVE.

Примечание**Тип блока данных MB_HOLD_REG**

Блок данных регистра хранения Modbus должен позволять как прямую (абсолютную), так и символическую адресацию. Когда Вы создаете блок данных, должен быть выбран атрибут доступа "Standard".

Следующая таблица показывает примеры проекций Modbus адресов в регистры хранения, используемые для кодов Modbus функций 03 (считать слова), 06 (записать слово) и 16 (записать слова). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом М памяти для каждой из моделей ЦПУ.

Таблица 13- 153 Проекция Modbus адресов в адреса памяти ЦПУ

Адрес ведущего устройства Modbus	Примеры параметра MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recipe".ingredient[5]

Таблица 13- 154 Диагностические функции

Диагностические функции MB_SLAVE для S7-1200		
Коды	Подфункция	Описание
08	0000H	Эхо-тест возврата данных запроса: MB_SLAVE отразит ведущему устройству Modbus слово данных, которое принято.
08	000AH	Сбросить счетчик коммуникаций: MB_SLAVE сбросит счетчик коммуникаций, который используется для Modbus функции 11.
11		Получить счетчик коммуникаций: MB_SLAVE использует внутренней счетчик коммуникаций для записи числа успешных запросов на чтение и запись по Modbus, отправленных ведомому устройству Modbus. Счетчик не увеличивается на единицу по любому запросу на Функцию 8, Функцию 11 или любому запросу, который приводит к ошибке связи (например, ошибки четности или CRC).

Инструкция Modbus_Slave поддерживает широковещательные запросы записи от любого ведущего устройства Modbus, если запрос использует допустимые адреса. MB_SLAVE выставляет код ошибки "0x8188" для кодов функций, не поддерживаемых в широковещательной передаче.

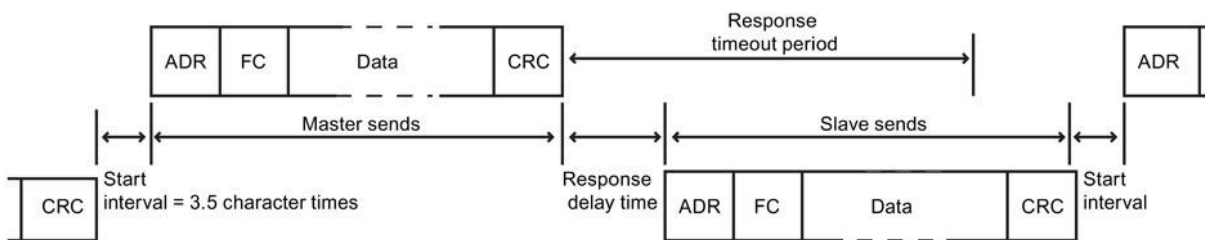
Правила коммуникаций ведомого устройства Modbus

- MB_COMM_LOAD должен быть выполнен, чтобы сконфигурировать порт, прежде чем инструкция MB_SLAVE сможет выполнять обмен через этот порт.
- Если порт должен ответить ведущему устройству Modbus как ведомое устройство, то не программируйте этот порт инструкцией MB_MASTER.
- Только один экземпляр MB_SLAVE может использоваться с данным портом, иначе возможно неустойчивое поведение.
- Инструкции Modbus не используют коммуникационные события прерывания, чтобы управлять коммуникационным процессом. Ваша программа должна управлять коммуникационным процессом, опрашивая инструкцию MB_SLAVE на предмет состояния завершения передачи и приема.

- Инструкция MB_SLAVE должна периодически выполняться на уровне, который позволяет ей своевременно отвечать на входящие запросы от ведущего устройства Modbus. Рекомендуется, чтобы Вы выполняли MB_SLAVE каждый скан OB программного цикла. Выполнение MB_SLAVE из OB циклического прерывания возможно, но не рекомендуется из-за вероятности чрезмерных задержек других процедур прерывания.

Временная диаграмма Modbus сигнала

MB_SLAVE должен выполняться периодически, чтобы принимать каждый запрос от ведущего устройства Modbus, а затем ответить при необходимости. Периодичность выполнения для MB_SLAVE зависит от периода тайм-аута ответа ведущего устройства Modbus. Это проиллюстрировано на следующей диаграмме.



Период тайм-аута ответа RESP_TO представляет время, в течение которого ведущее устройство Modbus ожидает начала ответа от ведомого устройства Modbus. Этот период времени не определен протоколом Modbus, но является параметром каждого ведущего устройства Modbus. Периодичность выполнения (время между текущим и последующим выполнениями) MB_SLAVE должна базироваться на определенных параметрах Вашего ведущего устройства Modbus. Как минимум Вы должны выполнять MB_SLAVE дважды за период тайм-аута ответа ведущего устройства Modbus.

Переменные ведомого устройства Modbus

Настоящая таблица показывает общедоступные статические переменные, сохраненные в экземплярном блоке данных MB_SLAVE, которые могут использоваться в Вашей программе.

Таблица 13- 155 Переменные ведомого устройства Modbus

Переменная	Тип данных	Описание
Request_Count	Word	Количество всех запросов, принятых этим ведомым устройством
Slave_Message_Count	Word	Количество принятых этим определенным ведомым устройством запросов
Bad_CRC_Count	Word	Количество принятых запросов, в которых есть ошибка CRC
Broadcast_Count	Word	Количество принятых широковещательных запросов
Exception_Count	Word	Определенные ошибки Modbus, которые требуют возвращенного исключения
Success_Count	Word	Количество принятых этим определенным ведомым устройством запросов, в которых нет ошибок протокола
HR_Start_Offset	Word	Определяет начальный адрес регистра хранения Modbus (значение по умолчанию = 0)
Extended_Addresssing	Bool	Конфигурирует одно- или двухбайтную адресацию ведомого устройства (0=однобайтовый адрес, 1=двухбайтовый адрес, значение по умолчанию = 0)

Ваша программа может записать значения в переменные HR_Start_Offset и Extended_Addressing и управлять операциями ведомого устройства Modbus. Другие переменные могут быть считаны для контроля состояния Modbus.

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001 или 400001. Эти адреса соответствуют началу адресов регистра хранения в ПЛК. Однако, Вы можете сконфигурировать переменную "HR_Start_Offset", чтобы установить начало адресов регистра хранения Modbus на другое значение вместо 40001 или 400001.

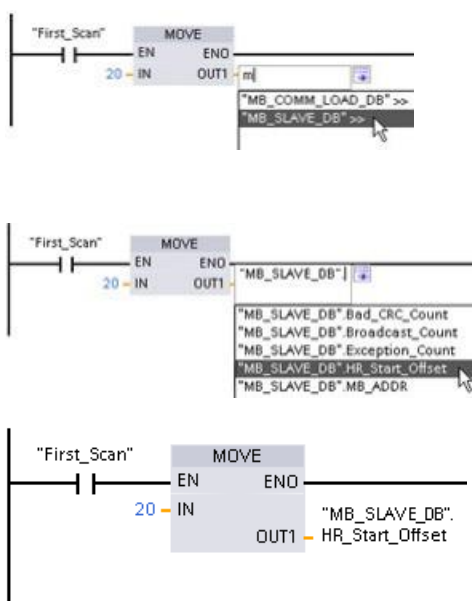
Например, если регистр хранения сконфигурирован, чтобы начинаться в MW100 и имеет длину 100 слов. Смещение 20 определяет начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 400119 приведет к ошибке адресации.

Таблица 13- 156 Пример адресации регистра хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Modbus адрес (слово)	40001	40099
	S7-1200 адрес	MW100	MW298
20	Modbus адрес (слово)	40021	40119
	S7-1200 адрес	MW100	MW298

HR_Start_Offset - значение типа word, которое определяет начальный адрес регистра хранения Modbus и сохраняется в экземплярном блоке данных MB_SLAVE. Вы можете установить это общедоступное значение статической переменной при помощи выпадающего списка помощника параметра, после того, как MB_SLAVE будет помещен в Вашу программу.

Например, после того, как MB_SLAVE помещен в сегмент LAD, Вы можете перейти к предыдущему сегменту и присвоить значение HR_Start_Offset. Значение должно быть присвоено до выполнения MB_SLAVE.



Ввод переменной ведомого устройства Modbus, используя имя DB по умолчанию:

1. Установите курсор в поле параметра и введите символ m.
2. Выберите " MB_SLAVE_DB " в выпадающем списке.
3. Установите курсор справа от имени DB (после символа кавычки) и введите символ точки.
4. Выберите "MB_SLAVE_DB.HR_Start_Offset" в выпадающем списке.

Extended Addressing

К переменной Extended Addressing обращаются похожим образом как к HR_Start_Offset, что обсуждено выше, за исключением того, что переменная Extended Addressing является булевой переменной. Булева переменная должно быть записана с помощью выходной катушки, а не оператором move.

Адресация ведомого устройства Modbus может быть сконфигурирована либо как однокбайтовая (что является стандартом Modbus), либо, как двухбайтовая. Расширенная адресация используется, чтобы адресовать больше чем 247 устройств в отдельной сети. Выбор расширенной адресации позволяет Вам адресовать максимум 64000 адресов. Кадр Modbus функции 1 представлен ниже, в качестве примера.

Таблица 13- 157 Однокбайтовый адрес ведомого устройства (байт 0)

Функция 1	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	
Запрос	Адрес ведом.	Код функции	Начальный адрес		Количество катушек		
Действит.ответ	Адрес ведом.	Код функции	Длина	Данные катушек			
Ответ с ошибкой	Адрес ведом.	0x81	Код ошибки				

Таблица 13- 158 Двухбайтовый адрес ведомого устройства (байты 0 и 1)

	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6
Запрос	Адрес ведомого устройства		Код функции	Начальный адрес		Количество катушек	
Действит.ответ	Адрес ведомого устройства		Код функции	Длина	Данные катушек		
Ответ с ошибкой	Адрес ведомого устройства		0x81	Код ошибки			

Коды состояния

Таблица 13- 159 Коды состояния выполнения MB_SLAVE (ошибки обмена и конфигурации) ¹

STATUS (W#16#)	Описание
80D1	Получатель выставил запрос на управление потоком, чтобы приостановить активную передачу и повторно не включал передачу в течение указанного времени ожидания. Эта ошибка также генерируется в процессе аппаратного управления потоком, когда получатель не утверждает CTS в течение указанного времени ожидания.
80D2	Запрос на передачу был прерван, так как не был получен сигнал DSR от DCE.
80E0	Сообщение было прервано, так как приемный буфер заполнен.
80E1	Сообщение было прервано в результате ошибки четности.
80E2	Сообщение было прервано в результате ошибки кадровой синхронизации.
80E3	Сообщение было прервано в результате ошибки переполнения.
80E4	Сообщение было прервано в результате задания длины, превышающей общий размер буфера.
8180	Недопустимое значение ID порта или ошибка с инструкцией MB_COMM_LOAD
8186	Неверный адрес Modbus станции
8187	Недопустимый указатель на DB MB_HOLD_REG: область слишком маленькая
818C	Недопустимый указатель MB_HOLD_REG на M память или DB (область DB должна допускать, как символьную, так и абсолютную адресацию)

Таблица 13- 160 Коды состояния выполнения MB_SLAVE (ошибки Modbus протокола) ¹

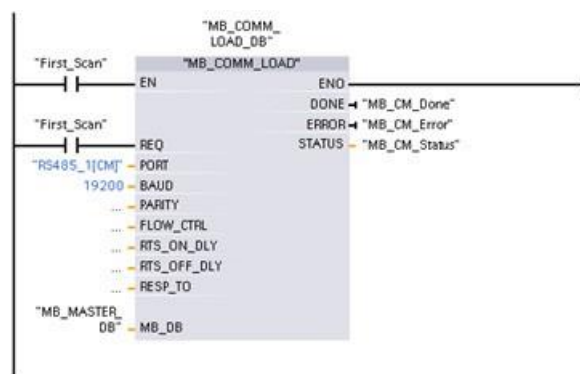
STATUS (W#16#)	Ответ ведомого устройства	Ошибки Modbus протокола
8380	Нет ответа	Ошибка CRC
8381	01	Код функции не поддерживается или не поддерживается в широковещательной рассылке
8382	03	Ошибка длины данных
8383	02	Ошибка адреса данных или адрес вне допустимого диапазона области DATA_PTR
8384	03	Ошибка значения данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)

¹ В дополнение к упомянутым выше ошибкам MB_SLAVE ошибки могут быть возвращены из базовых коммуникационных PTP инструкций.

13.9.3.4. Пример: Программа унаследованного ведущего устройства Modbus RTU

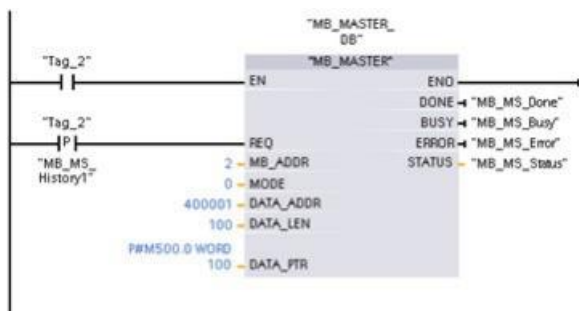
Сегмент 1: Инициализировать параметры RS485 модуля только один раз во время первого скана.

MB_COMM_LOAD инициализируется во время запуска при помощи флага первого скана. Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта не изменяется во время работы.

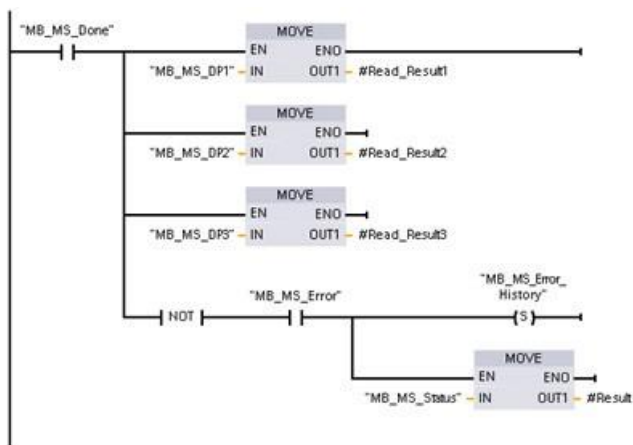


Одна инструкция MB_MASTER используется в ОВ программного цикла, чтобы выполнить обмен с одиночным ведомым устройством. Дополнительные инструкции MB_MASTER могут использоваться в ОВ программного цикла, чтобы выполнить обмен с другими ведомыми устройствами, или один FB MB_MASTER мог бы быть повторно использован, чтобы выполнить обмен с дополнительными ведомыми устройствами.

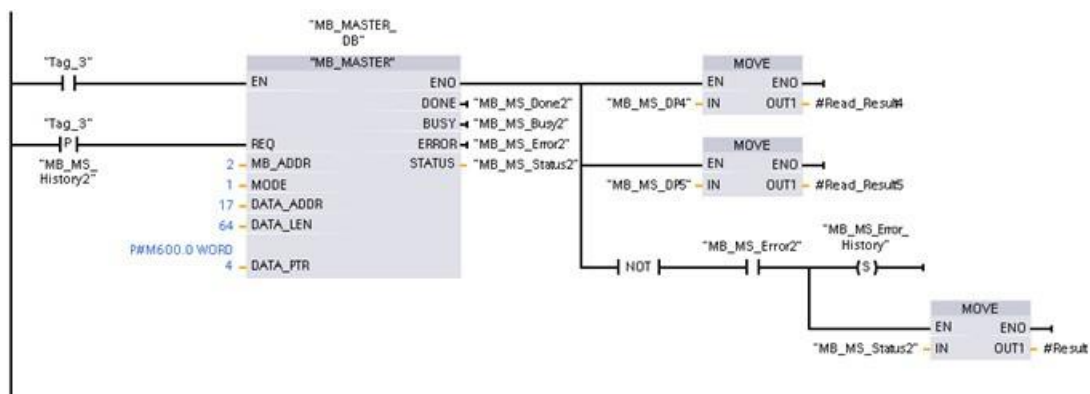
Сегмент 2: Считать 100 слов из регистра хранения ведомого устройства.



Сегмент 3: Это дополнительный сегмент, который просто показывает значения первых 3 слов, как только операция чтения выполнена.



Сегмент 4: Записать 64 бита в выходной регистр образа по выводу, начиная с адреса Q2.0.

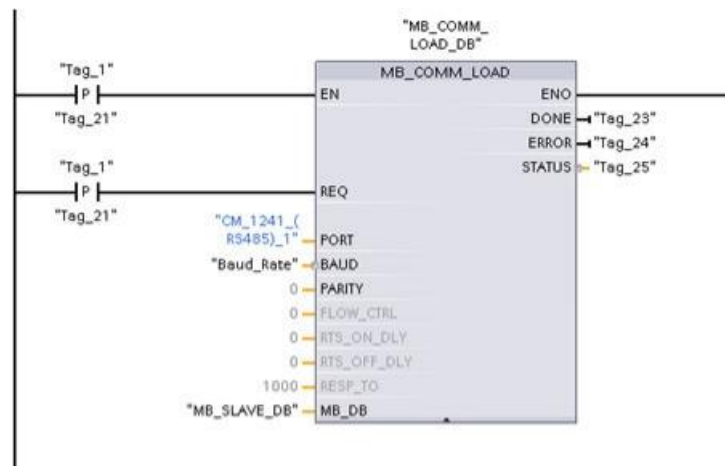


13.9.3.5. Пример: Программа унаследованного ведомого устройства Modbus RTU

MB_COMM_LOAD, показанный ниже, инициализируется, каждый раз при активации тега "Tag_1".

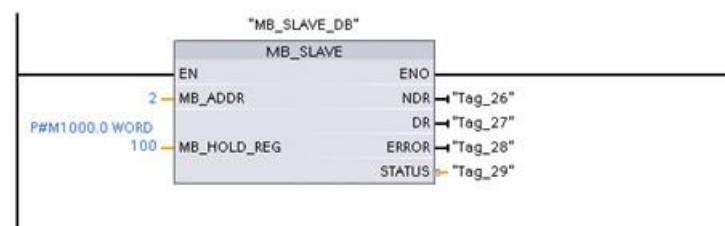
Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта изменяется во время работы в результате конфигурирования устройства HMI.

Сегмент 1: Инициализировать параметры модуля RS485 каждый раз, когда они изменяются устройством HMI.



MB_SLAVE, показанный ниже, помещен в циклический OB, который выполняется каждые 10 мс. Несмотря на то, что это не обеспечивает абсолютно быстрый ответ ведомого устройства, но действительно обеспечивает хорошую производительность при 9600 бодах для коротких сообщений (20 байтов или меньше в запросе).

Сегмент 2: Проверять запросы от ведущего устройства Modbus на каждом скане. Регистр хранения Modbus сконфигурирован для 100 слов, начиная с MW1000.



13.10 Удаленное управление и обслуживание с помощью CP 1242-7

13.10.1. Обзор коммуникационных процессоров для телеуправления

Коммуникационные процессоры для телеуправления S7-1200

Для приложений удаленного управления доступны следующие коммуникационные процессоры:

- **CP 1243-1**

Номер по каталогу 6GK7 243-1BX30-0XE0

Коммуникационный процессор для подключения SIMATIC S7-1200 через общедоступную инфраструктуру (например, DSL) к центру управления с TeleControl Server Basic (TCSB версии V3).

С помощью технологии VPN и брандмауэра, КП позволяет защищенный доступ к S7-1200.

КП может использоваться в качестве дополнительного Ethernet-интерфейса ЦПУ для S7 коммуникаций.

Обмен между КП и ЦПУ осуществляется через конфигурируемые точки данных, которые имеют доступ к ПЛК тегам.

- **CP 1243-1 DNP3**

Номер по каталогу 6GK7 243-1JX30-0XE0

Коммуникационный процессор для подключения SIMATIC S7-1200 к центрам управления, используя DNP3 протокол.

Обмен между КП и ЦПУ осуществляется через конфигурируемые точки данных, которые имеют доступ к ПЛК тегам.

- **CP 1243-1 IEC**

Номер по каталогу 6GK7 243-1PX30-0XE0

Коммуникационный процессор для подключения SIMATIC S7-1200 к центрам управления, используя IEC 60870-5 протокол.

Обмен между КП и ЦПУ осуществляется через конфигурируемые точки данных, которые имеют доступ к ПЛК тегам.

- **CP 1242-7**

Номер по каталогу 6GK7 242-7KX30-0XE0

Коммуникационный процессор для подключения SIMATIC S7-1200 к центру управления с TeleControl Server Basic, используя мобильную беспроводную связь (GPRS) и общедоступную инфраструктуру (DSL)

- **CP 1242-7 GPRS V2**

Номер по каталогу 6GK7 242-7KX31-0XE0

Коммуникационный процессор для подключения SIMATIC S7-1200) к центру управления с TeleControl Server Basic (TCSB версии V3), используя мобильную беспроводную связь (GPRS) и общедоступную инфраструктуру (DSL)

С помощью технологии VPN и брандмауэра, КП позволяет защищенный доступ к S7-1200.

КП может использоваться в качестве дополнительного Ethernet-интерфейса ЦПУ для S7 коммуникаций.

Обмен между КП и ЦПУ осуществляется через конфигурируемые точки данных, которые имеют доступ к ПЛК тега.

- **CP 1243-7 LTE-xx**

Коммуникационный процессор для подключения SIMATIC S7-1200) к центру управления с TeleControl Server Basic (TCSB версии V3), используя мобильную беспроводную связь (GPRS) и общедоступную инфраструктуру (DSL)

Поддержка следующих мобильных беспроводных спецификаций: GSM/GPRS, UMTS (G3), LTE

Для использования в странах с различными мобильными беспроводными спецификациями, КП доступен в двух вариантах:

- CP 1243-7 LTE-US

Североамериканский стандарт

Номер по каталогу 6GK7 243-7SX30-0XE0

- CP 1243-7 LTE-EU

Западноевропейский стандарт

Номер по каталогу 6GK7 243-7KX30-0XE0

С помощью технологии VPN и брандмауэра, КП позволяет защищенный доступ к S7-1200.

КП может использоваться в качестве дополнительного Ethernet-интерфейса ЦПУ для S7 коммуникаций.

Обмен между КП и ЦПУ осуществляется через конфигурируемые точки данных, которые имеют доступ к ПЛК тега.

13.10.2. Подключение к GSM-сети

Основанные на IP WAN-коммуникации через GPRS

Используя коммуникационные процессоры CP 1242-7, S7-1200 может быть подключен к GSM-сетям. CP 1242-7 позволяет WAN-коммуникации из удаленных станций центром управления и межстанционные коммуникации.

Межстанционные коммуникации возможны только через GSM-сеть. Для связи между удаленной станцией и диспетчерской, в центре управления должен быть ПК с доступом к Интернету.

CP 1242-7 поддерживает следующие службы для коммуникаций через GSM-сеть:

- GPRS (General Packet Radio Service)

Пакетно-ориентированная служба для передачи данных "GPRS" осуществляется через GSM-сеть.

- SMS (Short Message Service)

CP 1242-7 может получать и отправлять SMS-сообщения. Коммуникационный партнер может быть мобильным телефоном или S7-1200.

CP 1242-7 подходит для использования в любых отраслях производства и поддерживает следующие диапазоны частот:

- 850 МГц
- 900 МГц
- 1,800 МГц
- 1,900 МГц

Требования

Оборудование, используемое в станциях или центре управления, зависит от конкретного приложения.

- Для коммуникаций с центральной диспетчерской или через нее, центру управления требуется ПК с доступом к Интернету.
- Кроме оборудования станции, удаленная станция S7-1200 с CP 1242-7 должна удовлетворять следующим требованиям, чтобы выполнять коммуникации через GSM-сеть:

- Контракт с подходящим поставщиком услуг GSM-сети

Если используется GPRS, контракт должен позволять использование службы GPRS.

Если должны выполняться непосредственные коммуникации между станциями только через GSM-сеть, поставщик услуг GSM-сети должен присвоить фиксированный IP-адрес коммуникационным процессорам. В этом случае обмен между станциями выполняется не через центр управления.

- SIM-карта, прописанная в контракте. SIM-карта, вставляется в CP 1242-7.
- Локальная доступность GSM-сети в месте расположения станции

13.10.3. Приложения для CP 1242-7

CP 1242-7 может использоваться в следующих приложениях:

Приложения телеуправления

- Отправка сообщений с помощью SMS

Через CP 1242-7, ЦПУ удаленной станции S7-1200 может получать SMS-сообщения от GSM-сети или отправлять SMS-сообщения на сконфигурированный мобильный телефон или S7-1200.

- Коммуникации с центром управления

Удаленные станции S7-1200 выполняют обмен через GSM-сеть и Интернет с сервером телеуправления в ведущей станции. Для передачи данных с использованием GPRS, приложение "TELECONTROL SERVER BASIC" устанавливается на сервере телеуправления в ведущей станции. Сервер телеуправления связывается с центральной системой управления более высокого уровня, используя интегрированную функцию OPC-сервера.

- Коммуникации между станциями S7-1200 через GSM-сеть

Коммуникации между удаленными станциями с CP 1242-7 могут осуществляться двумя различными способами:

- Межстанционная коммуникация через основную станцию

В этой конфигурации постоянное безопасное соединение между станциями S7-1200, которые связываются друг с другом и сервером телеуправления, устанавливается в ведущей станции. Коммуникации между станциями осуществляются через сервер телеуправления. CP 1242-7 работает в режиме "Telecontrol".

- Непосредственные коммуникации между станциями

Для непосредственного обмена между станциями без обхода через ведущую станцию используются SIM-карты с фиксированным IP-адресом, которые позволяют станциям адресовать друг друга непосредственно. Возможные коммуникационные услуги и функции безопасности (например, VPN) зависят от того, что предлагается поставщиком сетевых услуг. CP 1242-7 работает в режиме "GPRS direct".

TeleService через GPRS

Соединение TeleService может быть установлено между инженерной станцией со STEP 7 и удаленной станцией S7-1200 с помощью CP 1242-7 через GSM-сеть и Интернет. Соединение работает с инженерной станцией через сервер телеуправления или шлюз TeleService, который выступает в качестве посредника, передавая кадры и устанавливая авторизацию. Эти ПК используют функции приложения "TELECONTROL SERVER BASIC".

Вы можете использовать соединение TeleService в следующих целях:

- Загрузка конфигурации или данных программы из проекта STEP 7 в станцию
- Запросы данных диагностики на станции

13.10.4. Другие функции CP-1242-7

Другие службы и функции CP 1242-7

- Синхронизация времени суток КП через Интернет. Вы можете установить время на КП следующим образом:
 - В режиме "Telecontrol" время суток передается сервером телеуправления. КП использует его, чтобы установить свое время.

- В режиме "GPRS direct" КП может запросить время, используя SNTP.

Чтобы синхронизировать время ЦПУ, Вы можете считать текущее время из КП, используя блок.

- Временная буферизация сообщений, которые следует отправить, в случае проблем подключения
- Увеличенная доступность благодаря опции подключения к резервному серверу телеуправления
- Оптимизированный объем данных (временное подключение)

В качестве альтернативы постоянному соединению с сервером телеуправления, КП может быть сконфигурирован в STEP 7 с временным подключением к серверу телеуправления. В этом случае соединение с сервером телеуправления устанавливается только при необходимости.

- Регистрация объема данных
Объемы переданных данных регистрируются и могут оцениваться с определенной целью.

13.10.5. Конфигурация и электрические соединения

Конфигурация и замена модуля

Чтобы сконфигурировать модуль, требуется следующий программный пакет: STEP 7 версии V11.0 SP1 или выше

Для STEP 7 V11.0 SP1 Вам также потребуется пакет поддержки "CP 1242-7" (HSP0003001).

Для передачи данных процесса, использующей GPRS, применяйте в пользовательской программе станции коммуникационные инструкции телеуправления.

Данные конфигурации CP 1242-7 сохраняются в локальном ЦПУ. Это позволяет простую замену КП при необходимости.

Вы можете вставить до трех модулей типа CP 1242-7 в один S7-1200. Это, к примеру, позволяет организовать дублирующие каналы обмена.

Электрические соединения

- Источник питания CP 1242-7
У КП есть отдельное подключение для внешнего источника питания на 24 В постоянного тока.
- Беспроводной интерфейс для GSM-сети
Дополнительная антенна требуется для GSM-коммуникаций. Она подключается через SMA-гнездо на КП.

13.10.6. Дополнительная информация

Дополнительная информация

Руководство для CP 1242-7 содержит подробную информацию о коммуникационном процессоре. Вы найдете его в Интернете на страницах Siemens Industrial Automation Customer Support под следующим идентификатором записи:

45605894 (<http://support.automation.siemens.com/WW/view/en/45605894>)

13.10.7. Принадлежности

Антенна ANT794-4MR GSM/GPRS

Следующие антенны доступны для использования в сетях GSM/GPRS и могут быть установлены как в закрытом помещении, так и на открытом воздухе:

- Четырехдиапазонная антенна ANT794-4MR



Краткое название	Заказной номер	Пояснение
ANT794-4MR	6NH9 860-1AA00	Четырехдиапазонная антенна (900, 1800/1900 МГц, UMTS); защищенная от атмосферных воздействий, для использования внутри и снаружи помещений; соединительный кабель длиной 5 м, постоянно подключенный к антенне; соединитель SMA, включая монтажный кронштейн, саморезы, дюбели

- Плоская антенна ANT794-3M



Краткое название	Заказной номер	Пояснение
ANT794-3M	6NH9 870-1AA00	Плоская антенна (900, 1800/1900 МГц); защищенная от атмосферных воздействий, для использования внутри и снаружи помещений; кабель длиной 1,2 м, постоянно подключенный к антенне; соединитель SMA, включая клейкую накладку, возможен монтаж с помощью саморезов

Антенны должны заказываться отдельно.

13.10.8. Ссылка на руководство для GSM-антенны

Дополнительная информация

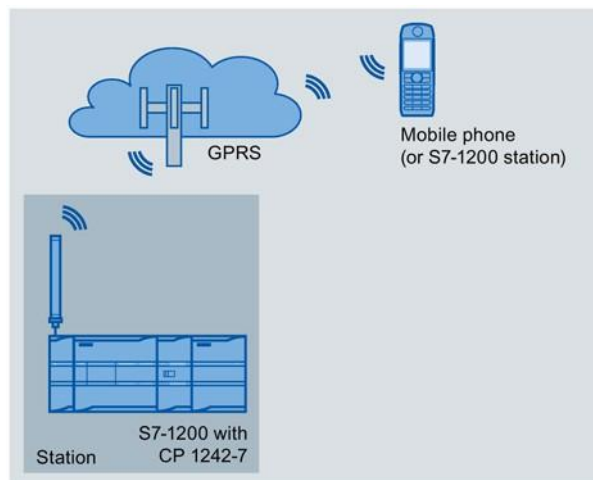
Дополнительная информация содержится в руководстве на изделие. Вы найдете его в Интернете на страницах Siemens Industrial Automation Customer Support под следующим идентификатором записи:

23119005 (<http://support.automation.siemens.com/WW/view/en/23119005>)

13.10.9. Примеры конфигурации для телеуправления

Ниже, Вы найдете несколько примеров конфигурации для станций с CP 1242-7.

Отправка сообщений с помощью SMS



SIMATIC S7-1200 с CP 1242-7 может отправлять SMS-сообщения на мобильный телефон или сконфигурированную станцию S7-1200.

Телеуправление через центр управления

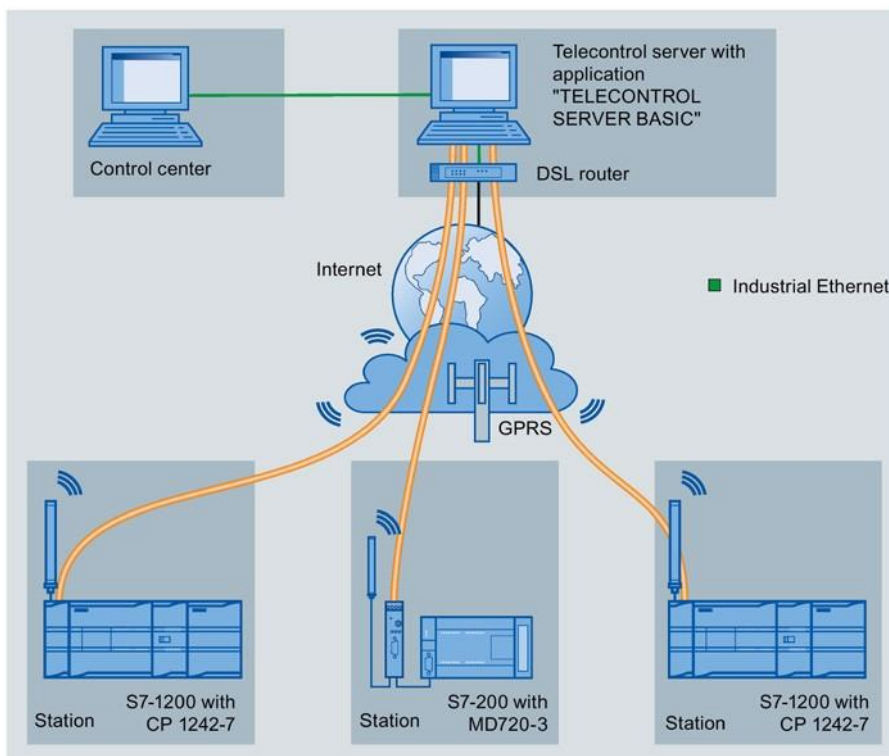


Рисунок 13-1 Коммуникации между станциями S7-1200 и центром управления

В приложениях телеуправления станции SIMATIC S7-1200 с CP 1242-7 выполняют обмен с центром управления через GSM-сеть и Интернет. Приложение "TELECONTROL SERVER BASIC" (TCSB), устанавливается на сервере телеуправления в ведущей станции. Это дает следующие варианты применения:

- Коммуникации телеуправления между станцией и центром управления
В этом варианте использования данные от поля отправляются станциями в сервер телеуправления в ведущей станции через GSM-сеть и Интернет. Сервер телеуправления используется, чтобы контролировать удаленные станции.
- Коммуникации между станцией и диспетчерской с OPC-клиентом
Как в первом случае, станции выполняют обмен с сервером телеуправления. Используя интегрированный OPC-сервер, сервер телеуправления обменивается данными с OPC-клиентом диспетчерской.
OPC-клиент и сервер телеуправления могут быть расположены на единственном компьютере, к примеру, когда TCSB устанавливается на компьютере центра управления с WinCC.
- Межстанционные коммуникации через центр управления
Межстанционная коммуникация возможна для S7-станций, оборудованных CP 1242-7.
Чтобы реализовать межстанционные коммуникации, сервер телеуправления передает сообщения передающей станции в станцию назначения.

Прямые коммуникации между станциями

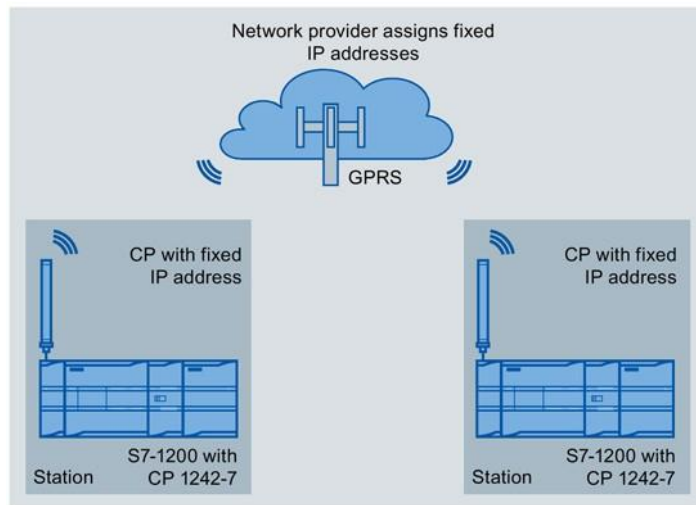


Рисунок 13-2 Прямые коммуникации между двумя S7-1200 станциями

В этой конфигурации две станции SIMATIC S7-1200 выполняют непосредственный обмен друг с другом, используя CP 1242-7, через GSM-сеть. У каждого CP 1242-7 есть фиксированный IP-адрес. Соответствующая служба поставщика сетевых услуг GSM должна позволить это.

TeleService через GPRS

В TeleService через GPRS инженерная станция, на которой установлен STEP 7, связывается через GSM-сеть и Интернет с CP 1242-7 в S7-1200.

Так как брандмауэр нормально закрыт для запросов на установление соединения извне, то необходима станция, переключающаяся между удаленной станцией и инженерной станцией. Эта переключающаяся станция может быть сервером телеуправления или, если в конфигурации нет никакого сервера телеуправления, шлюзом TeleService.

TeleService с сервером телеуправления

Соединение работает через сервер телеуправления.

- Инженерная станция и сервер телеуправления соединены через Интранет (LAN) или Интернет.
- Сервер телеуправления и удаленная станция соединены через Интернет и GSM-сеть.

Инженерная станция и сервер телеуправления могут также быть одним и тем же компьютером; другими словами, STEP 7 и TCSB устанавливаются на одном и том же компьютере.

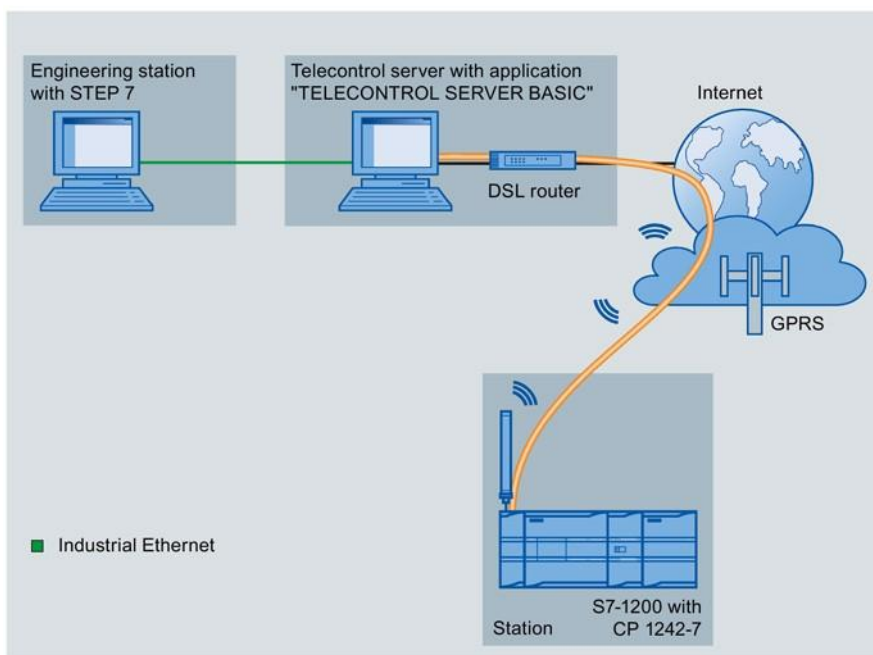


Рисунок 13-3 TeleService через GPRS в конфигурации с сервером телеуправления

TeleService без сервера телеуправления

Соединение работает через шлюз TeleService.

Соединение между инженерной станцией и шлюзом TeleService может быть локальным через LAN или через Интернет.

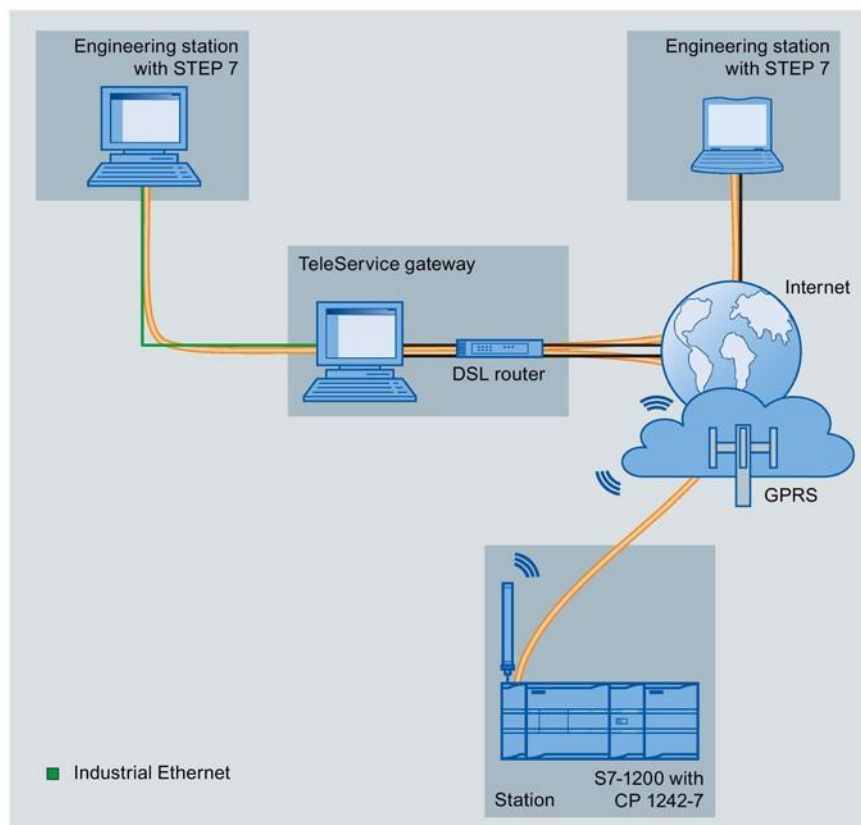
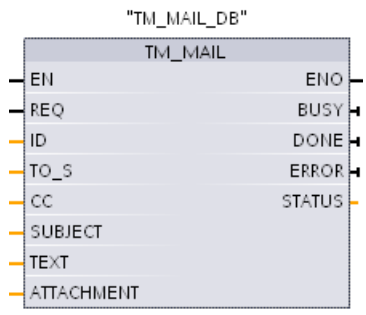


Рисунок 13-4 TeleService через GPRS в конфигурации со шлюзом TeleService

TeleService коммуникации (SMTP email)

14.1 Инструкция TM_Mail (послать электронное сообщение)

Таблица 14- 1 Инструкция TM_MAIL

LAD / FBD	SCL	Описание
 The diagram shows a function block call for 'TM_MAIL' within a network labeled 'TM_MAIL_DB'. The inputs are EN, REQ, ID, TO_S, CC, SUBJECT, TEXT, and ATTACHMENT. The outputs are ENO, BUSY, DONE, ERROR, and STATUS.	<pre> "TM_MAIL_DB" (REQ:=_bool_in_, ID:=_int_in_, TO_S:=_string_in_, CC:=_string_in_, SUBJECT:=_string_in_, TEXT:=_string_in_, ATTACHMENT:=_variant_in_, BUSY=>_bool_out_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_,); </pre>	Инструкция TM_MAIL отправляет электронное письмо, используя SMTP (Simple Mail Transfer Protocol) по TCP/IP через Industrial Ethernet соединение ЦПУ. Там, где основанное на Ethernet подключение к Интернету недоступно, может использоваться дополнительный Teleservice адаптер для соединения с телефонными наземными линиями связи. TM_MAIL выполняется асинхронно, и задание распределяется по нескольким вызовам TM_MAIL. Когда Вы вызываете TM_MAIL, Вы должны назначить DB экземпляр. В DB экземпляре не должен быть установлен атрибут retentive. Это гарантирует, что DB экземпляр инициализируется при переходе ЦПУ из STOP в RUN и, что может быть запущена новая операция TM_MAIL.

¹ STEP 7 автоматически создает экземплярный DB, когда Вы вставляете эту инструкцию.

Вы запускаете посылку электронного письма по положительному фронту входного параметра REQ. Следующая таблица показывает соотношение между BUSY, DONE и ERROR. Вы можете контролировать прогресс выполнения TM_MAIL и обнаружить завершение, оценив эти параметры посредством последовательных вызовов.

Выходные параметры, DONE, ERROR, STATUS, и SFC_STATUS действительны только для одного цикла, когда состояние выходного параметра BUSY изменяется из 1 в 0. Ваша программная логика должна сохранить временные значения состояния выхода, таким образом, Вы можете обнаружить изменения состояния в последовательных циклах выполнения программы.

Таблица 14- 2 Взаимодействие параметров Done, Busy и Error

DONE	BUSY	ERROR	Описание
Неважно	1	Неважно	Задание выполняется.
1	0	0	Задание было завершено успешно.
0	0	1	Задание было завершено с ошибкой. Для выяснения причины ошибки обратитесь к параметру STATUS.
0	0	0	Задание не выполняется

14.1 Инструкция TM_Mail (послать электронное сообщение)

Если ЦПУ переходит в режим STOP, в то время как TM_MAIL активна, коммуникационное соединение с почтовым сервером прерывается. Коммуникационное соединение с почтовым сервером также теряется, если возникают проблемы в коммуникациях ЦПУ на шине Industrial Ethernet. В этих случаях процесс отправки приостановлен, и электронная почта не достигает получателя.

ЗАМЕТКА**Изменение пользовательских программ**

Удаление и замена программных блоков, вызовов TM_MAIL или вызовов DB экземпляров TM_MAIL могут нарушить сцепление программных блоков. Если Вам не удастся сохранить сцепленные программные блоки, то коммуникационные функции TCP/IP могут войти в неопределенное состояние, возможно приводящее к материальному ущербу. После передачи измененного программного блока Вы должны были бы выполнить рестарт ЦПУ (теплый) или холодный запуск.

Чтобы избежать повреждения сцепки программных блоков, только изменяйте части своей пользовательской программы, которые непосредственно влияют на вызовы TM_MAIL только в следующих случаях:

- ЦПУ находится в режиме STOP
- Не посылается электронное сообщение (REQ и BUSY = 0)

Согласованность данных

Входной параметр ADDR_MAIL_SERVER читается, когда операция запущена. Новое значение не вступает в силу, пока текущая операция не завершена, и инициирована новая операция TM_MAIL.

Напротив, параметры WATCH_DOG_TIME, TO_S, CC, OT, SUBJECT, TEXT, ATTACHMENT, USERNAME И PASSWORD читаются во время выполнения TM_MAIL и могут быть изменены только, когда задание закончено (BUSY = 0).

Коммутируемое соединение: конфигурирование IE параметров TS адаптера

Вы должны сконфигурировать IE параметры Teleservice адаптера для исходящих вызовов, чтобы соединиться с коммутируемым сервером Вашего Интернет-провайдера. Если Вы устанавливаете атрибут вызов, "по требованию", то соединение устанавливается только тогда, когда электронное письмо будет послано. При использовании аналогового модема требуется больше времени для процесса соединения (приблизительно на минуту дольше). Вы должны учесть дополнительное время в значении WATCH_DOG_TIME.

Таблица 14- 3 Типы данных для параметров

Параметр и тип		Типы данных	Описание
REQ	IN	Bool	Положительный фронт сигнала запускает операцию.
ID	IN	Int	Идентификатор соединения: Смотри ID параметр инструкций TCON, TDISCON, TSEND и TRCV. Должно использоваться число, которое не используется для других дополнительных экземпляров этой инструкции в пользовательской программе.
TO_S	IN	String	Адреса получателей: данные STRING с максимальной длиной в 240 символов
CC	IN	String	CC копирует по (дополнительным) адресам получателей: данные STRING с максимальной длиной в 240 символов

14.1 Инструкция TM_Mail (послать электронное сообщение)

Параметр и тип		Типы данных	Описание
SUBJECT	IN	String	Название темы сообщения: данные STRING с максимальной длиной в 240 символов.
TEXT	IN	String	Текстовое сообщение (опция) электронной почты: данные STRING с максимальной длиной в 240 символов. Если этот параметр будет пустой строкой, то электронное письмо будет послано без текста сообщения.
ATTACHMENT	IN	Variant	Указатель на присоединяемые данные: Byte, Word или DWord данные с максимальной длиной в 65534 байтов. Если значение не присвоено, то электронное письмо, посылается без вложения.
DONE	OUT	Bool	0 - Задание еще не запускалось или еще выполняется. 1 - Задание было выполнено без ошибок.
BUSY	OUT	Bool	0 – Операция выполняется 1 – Операция не выполняется
ERROR	OUT	Bool	Бит ERROR=1 на одно сканирование, после того, как последний запрос был прерван с ошибкой. Значение кода ошибки на выходе STATUS действительно только в течение одиночного сканирования когда ERROR=1.
STATUS	OUT	Word	Возвращаемое значение или информация об ошибке инструкции TM_MAIL.
ADDR_MAIL_SERVER	¹ Static	DWord	IP-адрес почтового сервера: Вы должны назначить каждый фрагмент IP-адреса как октет двух 4-разрядных шестнадцатеричных символов. Если фрагмент IP-адреса = десятичное значение 10, который равняется шестнадцатеричному значению A, то Вы должны ввести "0A" для этого октета. Например: IP-адрес= 192.168.0.10 ADDR_MAIL_SERVER = DW#16#C0A8000A, где: ☐ 192 = 16#C0, ☐ 168 = 16#A8 ☐ 0 = 16#00 ☐ 10 = 16#0A
WATCH_DOG_TIME	¹ Static	Time	Максимально допустимое время для TM_MAIL, чтобы завершить весь SMTP процесс от инициирования соединения с SMTP до конца SMTP передачи. Если это время превышено, то исполнение TM_MAIL завершается с ошибкой. Фактическая задержка до завершения TM_MAIL может превысить WATCH_DOG_TIME, из-за дополнительного времени, требуемого для разъединения. Сначала Вам следует назначить время 2 минуты. Это время может быть намного меньшим для телефонного ISDN соединения.
USERNAME	¹ Static	String	Имя пользователя почтовой учетной записи: данные STRING с максимальной длиной в 180 символов.
PASSWORD	¹ Static	String	Пароль почтового сервера: данные STRING с максимальной длиной в 180 символов.

14.1 Инструкция TM_Mail (послать электронное сообщение)

Параметр и тип		Типы данных	Описание
FROM	¹ Static	String	Адрес отправителя: данные STRING с максимальной длиной в 240 символов.
SFC_STATUS	¹ Static	Word	Код условия выполнения вызванных коммуникационных блоков

¹ Значения этих параметров не изменяются при каждом вызове TM_MAIL. Значения присвоены в блоке данных экземпляре TM_MAIL и к ним происходит обращение только один раз при первом вызове TM_MAIL.

SMTP проверка подлинности

TM_MAIL поддерживает метод проверки подлинности SMTP AUTH LOGIN. Для получения информации об этом методе, смотри руководство для почтового сервера или веб-сайт Вашего интернет-провайдера.

Метод проверки подлинности AUTH LOGIN использует TM_MAIL параметры USERNAME и PASSWORD, чтобы соединиться с почтовым сервером. Имя пользователя и пароль должны быть заранее настроены в почтовом ящике на почтовом сервере.

Если никакое значение не присвоено параметру USERNAME, то метод проверки подлинности AUTH LOGIN не используется, и электронное письмо посылается без проверки подлинности.

Параметры TO_S:, CC: и FROM:

Параметры TO_S:, CC: и FROM: являются строками, как показано в следующих примерах: TO: <wenna@mydomain.com>, <ruby@mydomain.com>,

CC: <admin@mydomain.com>, <judy@mydomain.com>, FROM: <admin@mydomain.com>

Следующие правила должны использоваться при вводе этих символьных строк:

- Символы "TO:" "CC:" и "FROM:" должны быть введены, включая символ двоеточия.
- Символы пробел и открывающей угловой скобки "<" должны предшествовать каждому адресу. Например, должен быть пробел между "TO:" и <адресом электронной почты>.
- Закрывающая угловая скобка ">" должна быть введена после каждого адреса.
- Символ запятой ",", должен быть введен после каждого адреса электронной почты для TO_S: и CC: адресов. Например, запятая после единственного адреса электронной почты требуется в "TO: <адрес электронной почты>,".
- Только один адрес электронной почты может использоваться для записи FROM: , без запятой в конце.

Из-за нахождения в режиме выполнения и использования памяти, проверка синтаксиса данных TO_S:, CC: и FROM: для TM_MAIL не выполняется. Если правила формата, приведенные выше, не соблюдаются точно, транзакция с SMTP почтовым сервером терпит неудачу.

Параметры STATUS и SFC_STATUS

Коды условия выполнения, возвращаемые TM_MAIL, могут быть классифицированы следующим образом:

- W#16#0000: Операция TM_MAIL была завершена успешно
- W#16#7xxx: Состояние операции TM_MAIL
- W#16#8xxx: Ошибка внутреннего вызова коммуникационного устройства или почтового сервера

Следующая таблица показывает коды условия выполнения TM_MAIL за исключением кодов ошибки из-за внутренне вызванных коммуникационных модулей.

Примечание

Требования к почтовому серверу

TM_MAIL может соединяться с почтовым сервером, используя SMTP только через порт 25. Назначенный номер порта не может быть изменен.

Большинство отделов ИТ и внешние почтовые серверы сейчас блокируют порт 25, чтобы предотвратить заражение ПК вирусом превращения в неконтролируемый генератор электронных сообщений.

Вы можете соединиться с внутренним почтовым сервером через SMTP и позволить внутреннему серверу управлять текущими средствами повышения безопасности, которые требуются, чтобы передать электронную почту через Интернет внешнему почтовому серверу.

Пример: конфигурация сервера внутренней почты

Если Вы используете Microsoft Exchange в качестве сервера внутренней почты, то Вы можете сконфигурировать сервер, чтобы предоставить SMTP доступ от IP-адреса назначенного S7-1200 ПЛК. Сконфигурируйте консоль управления обменом: Server configuration > Hub transport > Receive connectors > IP relay. На вкладке Network есть поле, с именем "Receive mail from remote servers that have these IP addresses". Здесь Вы вводите IP-адрес ПЛК, который исполняет инструкцию TM_MAIL. Никакое подтверждение подлинности не требуется для этого типа соединения с внутренним сервером Microsoft Exchange.

Конфигурация почтового сервера

TM_MAIL может использовать только почтовый сервер, который позволяет использовать коммуникации через порт 25, SMTP и (дополнительно) подтверждение подлинности AUTH LOGIN.

Сконфигурируйте совместимую учетную запись почтового сервера, чтобы допустить удаленный SMTP вход в систему. Затем отредактируйте DB экземпляр для TM_MAIL, чтобы вставить символьные строки USERNAME и PASSWORD, которые используются для подтверждения подлинности соединения с Вашим почтовым ящиком.

14.1 Инструкция TM_Mail (послать электронное сообщение)

Таблица 14- 4 Коды состояния

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Описание
0000	-	Операция TM_MAIL завершилась без ошибок. Этот нулевой код STATUS не гарантирует, что электронное письмо было фактически послано (Смотри первый элемент в примечании после таблицы).
7001	-	TM_MAIL активна (BUSY = 1).
7002	7002	TM_MAIL активна (BUSY = 1).
8xxx	xxxx	Операция TM_MAIL была завершена с ошибкой в вызове инструкции внутренних коммуникаций. Для получения дополнительной информации о параметре SFC_STATUS, смотри описания параметра STATUS положенных в основу PROFINET открытых пользовательских коммуникационных инструкций.
8010	xxxx	Не удалось соединиться: Для получения дополнительной информации о параметре SFC_STATUS, смотри параметр STATUS инструкции TCON.
8011	xxxx	Ошибка, отправки данных: Для получения дополнительной информации о параметре SFC_STATUS, смотри параметр STATUS инструкции TSEND.
8012	xxxx	Ошибка при получении данных: Для получения дополнительной информации о параметре SFC_STATUS, смотри описание параметра STATUS инструкции TRCV.
8013	xxxx	Не удалось соединиться: Для получения дополнительной информации для оценки параметра SFC_STATUS, смотри описание параметра STATUS инструкций TCON и TDISCON.
8014	-	Не удалось соединиться: Вы, возможно, ввели неправильный IP-адрес почтового сервера (ADDR_MAIL_SERVER) или слишком малое время (WATCH_DOG_TIME) для соединения. Также возможно, что у ЦПУ нет соединения с сетью, или конфигурация ЦПУ неправильная.
8015	-	Недопустимый указатель для параметра ATTACHMENT: Используйте варианты указатели с присвоением длины и типа данных. Например, "P#DB.DBX0.0" - это неправильно, а "P#DB.DBX0.0 byte 256" - правильно.
82xx, 84xx, 85xx	-	Сообщение об ошибке приходит из почтового сервера и соответствует коду ошибки "8" SMTP протокола. Смотрите второй элемент в примечании после этой таблицы.
8450	-	Операция не работает: почтовый ящик недоступен; попробуйте еще раз позже.
8451	-	Операция прервана: локальная ошибка в обработке, попробуйте снова позже.
8500	-	Синтаксическая ошибка команды: причина может состоять в том, что почтовый сервер не поддерживает процесс проверки подлинности LOGIN. Проверьте параметры TM_MAIL. Попробуйте послать электронное письмо без проверки подлинности. Попробуйте заменить параметр USERNAME пустой строкой.
8501	-	Синтаксическая ошибка: Неправильный параметр или аргумент; Вы, возможно, ввели неправильный адрес в параметрах TO_S или CC.
8502	-	Команда неизвестна или не реализована: Проверьте введенные значения, особенно параметр FROM. Возможно, он неполон, и Вы опустили символы "@" или ".".
8535	-	Проверка подлинности SMTP неполная. Вы, возможно, ввели неправильное имя пользователя или пароль.
8550	-	Нельзя получить доступ к почтовому серверу, или у Вас нет прав доступа. Вы, возможно, ввели неправильное имя пользователя или пароль, или Ваш почтовый сервер не поддерживает доступ входа в систему. Другой причиной этой ошибки могла бы быть ошибочная запись доменного имени после символа "@" в параметрах TO_S или CC.

8552	-	Операция прервана: Превышен объем выделенной памяти; попробуйте еще раз позже.
8554	-	Передача не удалась: попробуйте еще раз позже.

Примечание**Возможные не представленные ошибки передачи электронных сообщений**

- Неправильная запись адреса получателя не генерирует ошибку STATUS для TM_MAIL. В этом случае нет никакой гарантии, что дополнительные получатели (с корректными адресами электронной почты), получат электронное письмо.
 - Больше информации о кодах ошибки SMTP может быть найдено в Интернете или в документации по ошибкам почтового сервера. Вы можете также считать последнее сообщение об ошибке из почтового сервера. Сообщение об ошибке сохранено в параметре buffer1 DB экземпляра для TM_MAIL.
-

15.1 Индикаторы состояния

ЦПУ и модули ввода/вывода используют светодиоды, чтобы предоставить информацию либо о рабочем состоянии модуля либо о вводе/выводе.

Светодиоды состояния на ЦПУ

ЦПУ имеет следующие светодиоды состояния:

- STOP/RUN
 - Постоянный желтый указывает на режим STOP
 - Постоянный зеленый указывает на режим RUN
 - Мигающий (чередование зеленого и желтого) указывает на то, что ЦПУ находится в режиме STARTUP
- ERROR
 - Мигающий красный указывает на ошибку, такую как внутренняя ошибка ЦПУ, ошибка карты памяти или ошибка конфигурации (неподходящие модули)
 - Неисправное состояние:
 - Постоянный красный указывает на аппаратную неисправность
 - Все светодиоды мигают при обнаружении дефекта программного обеспечения
- MAINT (Обслуживание) вспыхивает каждый раз, когда Вы вставляете карту памяти. ЦПУ в этом случае переходит в режим STOP. После того, как ЦПУ перешел в режим STOP, выполните одну из следующих функций, чтобы начать анализ карты памяти:
 - Переведите ЦПУ в режим RUN
 - Выполните сброс памяти (MRES)
 - Выполните цикл подачи питания ЦПУ

Вы можете также использовать соответствующую инструкцию (стр. 385), чтобы определить статус светодиодов.

Таблица 15- 1 Светодиоды состояния для ЦПУ

Описание	STOP/RUN Желтый / Зеленый	ERROR Красный	MAINT Желтый
Питание выключено	Выкл.	Выкл.	Выкл.
Запуск, самотестирование или обновление встроенного ПО	Мигает (поочередно желтый и зеленый)	-	Выкл.
Режим Stop	Вкл. (желтый)	-	-
Режим Run	Вкл. (зеленый)	-	-
Удалите карту памяти	Вкл. (желтый)	-	Мигает
Ошибка	Вкл. (желтый либо зеленый)	Мигает	-

15.1 Индикаторы состояния

Описание	STOP/RUN Желтый / Зеленый	ERROR Красный	MAINT Желтый
Требуется обслуживание - Форсированный ввод/вывод - Требуется замена батареи (если установлена батарейная плата)	Вкл. (желтый либо зеленый)	-	Вкл.
Аппаратная неисправность	Вкл. (желтый)	Вкл.	Выкл.
Тест светодиодов или дефект встроенного ПО в ЦПУ	Мигает (поочередно желтый и зеленый)	Мигает	Мигает
Неизвестная или несовместимая версия конфигурации ЦПУ	Вкл. (желтый)	Мигает	Мигает

Примечание**Ошибка "неизвестная или несовместимая версия конфигурации ЦПУ"**

Попытка загрузить программу S7-1200 V3.0 в ЦПУ S7-1200 V4.0 вызывает ошибку ЦПУ, и ЦПУ отображает соответствующее сообщение об ошибке в диагностическом буфере. Если Вы перешли в это состояние при использовании карты передачи с программой недопустимой версии (стр. 135), то удалите карту, выполните переход из STOP в RUN, сброс памяти (MRES) или цикл питания. Если Вы переходите в это состояние недопустимой загрузкой программы, выполните сброс ЦПУ к заводским настройкам (стр. 1073). После того, как Вы восстановите ЦПУ из состояния ошибки, Вы можете загрузить допустимую ЦПУ программу V4.0.

ЦПУ также имеет два светодиода, которые отображают состояние коммуникаций PROFINET. Откройте нижнюю крышку клеммной колодки, чтобы увидеть светодиоды PROFINET.

- Link (зеленый) светится, указывая на успешное соединение
- Rx/Tx (желтый) светится, указывая на активность передачи

ЦПУ и каждый сигнальный цифровой модуль (SM) имеют светодиод для каждого из цифровых входных и выходных каналов. Светодиод канала ввода-вывода (зеленый) загорается или гаснет, указывая на состояние отдельного входа или выхода.

Поведение S7-1200, следующее за критической ошибкой

Если встроенное ПО ЦПУ обнаруживает критическую ошибку, оно предпринимает попытку перезапуска, и, в случае успеха, сигнализирует о режиме неисправности непрерывным миганием светодиодов STOP/RUN, ERROR и MAINT. Пользовательская программа и аппаратная конфигурация не загружаются после перезапуска в режиме неисправности.

Если ЦПУ успешно завершает перезапуск в режиме неисправности, то выходы ЦПУ и сигнальных плат устанавливаются в 0, а выходы сигнальных модулей центральной стойки и распределенного ввода-вывода устанавливаются согласно сконфигурированному свойству "Reaction to CPU STOP".

Если перезапуск в режиме неисправности не удастся, (например, из-за аппаратного отказа), то светодиоды STOP и ERROR зажжены, а светодиод MAINT погашен.



ПРЕДУПРЕЖДЕНИЕ

Нельзя гарантировать безопасную работу в неисправном состоянии

Управляющие устройства могут перестать работать в небезопасном состоянии, что может привести к непредсказуемой работе оборудования. Такого рода непредсказуемая работа оборудования могла бы привести к смерти или серьезным травмам персонала, и/или повреждению оборудования.

Используйте функцию аварийного останова, электромеханических блокировок или другого защитного оборудования, которое работает независимо от ПЛК.

Светодиоды состояния на SM

Дополнительно, каждый цифровой SM имеет светодиод DIAG, который указывает на состояние модуля:

- Зеленый цвет сообщает о работоспособном состоянии модуля
- Красный цвет, сообщает, что модуль неисправный или неработоспособный

Каждый аналоговый SM имеет светодиод I/O Channel для каждого аналогового входа или выхода.

- Зеленый цвет сообщает, что канал был сконфигурирован и является активным
- Красный цвет, сообщает о состоянии ошибки отдельного аналогового входа или выхода

Дополнительно, каждый аналоговый SM имеет светодиод, который указывает на состояние модуля:

- Зеленый цвет сообщает о работоспособном состоянии модуля
- Красный цвет, сообщает, что модуль неисправный или неработоспособный

SM обнаруживает наличие или отсутствие питания модуля (питание поля, если требуется).

15.1 Индикаторы состояния

Таблица 15- 2 Светодиоды состояния для сигнального модуля (SM)

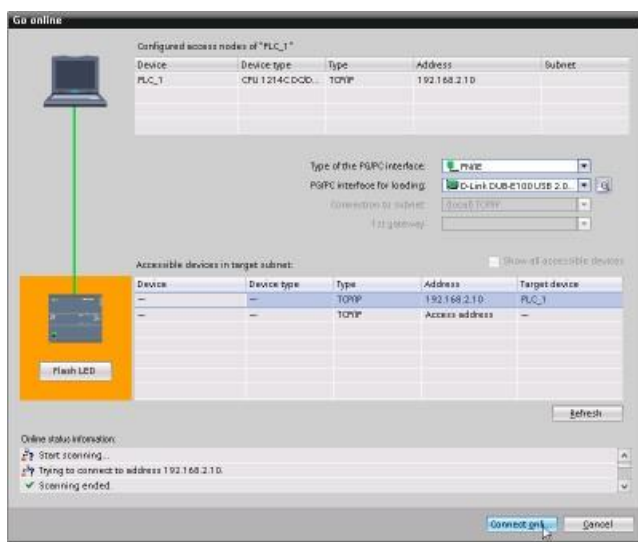
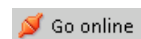
Описание	DIAG (Красный / зеленый)	I/O Channel (Красный / зеленый)
Питание поля выключено	Мигающий красный	Мигающий красный
Не сконфигурирован или выполняется обновление	Мигающий зеленый	Выкл.
Модуль сконфигурирован без ошибок	Вкл. (зеленый)	Вкл. (зеленый)
Состояние ошибки	Мигающий красный	-
Ошибка ввода/вывода (с разрешенной диагностикой)	-	Мигающий красный
I/O error (with diagnostics disabled)	-	Вкл. (зеленый)

15.2 Переход в режим он-лайн и подключение к ЦПУ

Вы должны установить он-лайн соединение между устройством программирования и ЦПУ для загрузки программ и технических данных проекта, а также для таких действий, как:

- Тестирование пользовательских программ
- Отображение и изменение рабочего режима ЦПУ (стр. 1075)
- Отображение и установка даты и времени дня в ЦПУ (стр. 1072)
- Отображение информации о модуле
- Сравнение и синхронизация (стр. 1077) офф-лайн и он-лайн программных блоков
- Выгрузка и загрузка программных блоков
- Отображение диагностики и диагностического буфера (стр. 1076)
- Использование таблицы наблюдения (стр. 1081) для тестирования пользовательской программы посредством контроля и изменения значений
- Использование таблицы форсирования для принудительной установки значений в ЦПУ (стр. 1084)

Чтобы установить он-лайн соединение со сконфигурированным ЦПУ, щелкните по ЦПУ в дереве проекта и нажмите кнопку "Go online" в представлении Project View:



Если это, первый переход в он-лайн для этого ЦПУ, Вы должны выбрать тип PG/PC интерфейса и определенный PG/PC интерфейс в диалоговом окне Go Online до попытки установления он-лайн соединения с ЦПУ, обнаруженным на этом интерфейсе.

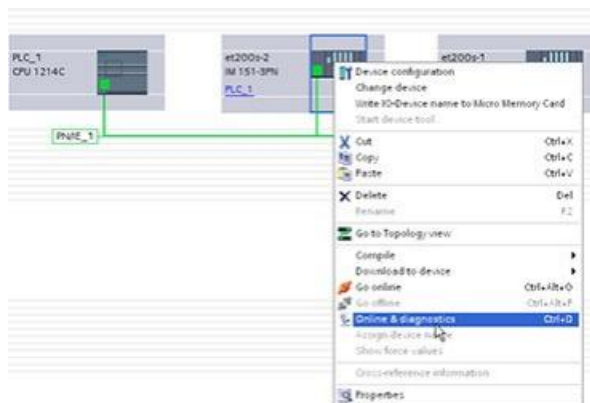
Вы теперь подключили свое устройство программирования к ЦПУ. Оранжевые рамки указывают на он-лайн соединение. Вы можете теперь использовать Online & diagnostics инструменты из дерева проекта и карту задач Online tools.

15.3 Присвоение имени PROFINET IO устройству онлайн

Устройства в Вашей сети PROFINET должны иметь назначенные имена, прежде чем Вы сможете подключить их к ЦПУ. Используйте редактор "Devices & networks", чтобы назначить имена Вашим устройствам PROFINET, если им ранее не были присвоены имена или, если имя устройства должно быть изменено.

Каждому PROFINET IO устройству, Вы должны назначить одно и то же имя как в проекте STEP 7, так и, используя инструмент "Online & diagnostics", в памяти конфигурации PROFINET IO устройства (например, памяти конфигурации интерфейсного модуля ET200-S). Если имя отсутствует или не будет соответствовать в обоих расположениях, то режим обмена PROFINET IO данными не будет работать.

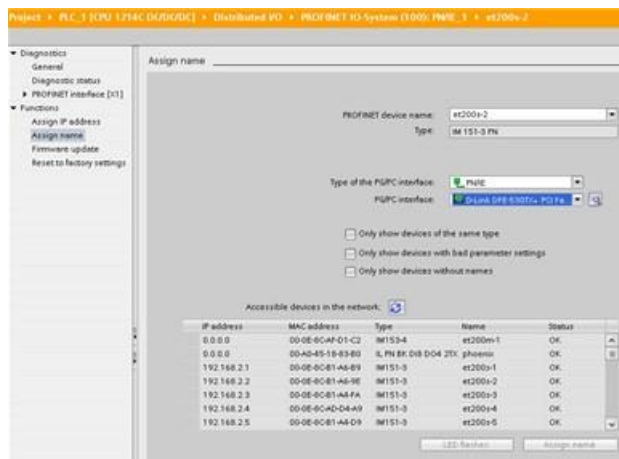
1. В редакторе "Devices & networks", щелкните правой кнопкой по требуемому PROFINET IO устройству и выберите "Online & diagnostics".



2. В диалоговом окне "Online & diagnostics", выберите следующие пункты меню:

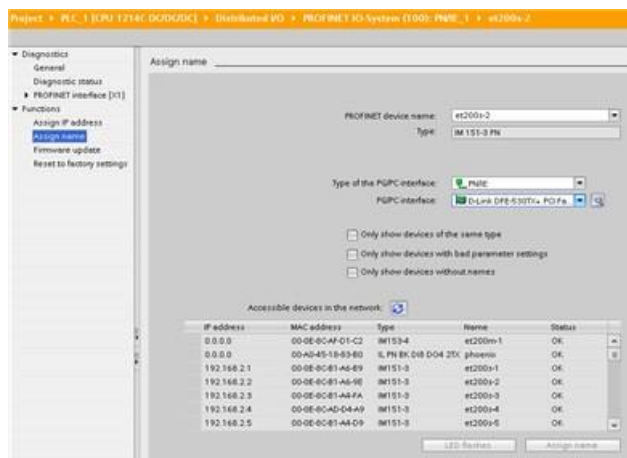
- "Functions"
- "Assign name"

Щелкните по значку "Accessible devices in the network", чтобы отобразить все PROFINET IO устройства в сети.



15.3 Присвоение имени PROFINET IO устройству онлайн

3. В показанном списке щелкните по требуемому PROFINET IO устройству и нажмите кнопку "Assign name", чтобы записать имя в память конфигурации PROFINET IO устройства.



15.4 Установка IP адреса и времени суток

Вы можете установить IP-адрес (стр. 625) и время суток в он-лайн ЦПУ. После обращения через "Online & diagnostics" из дерева проекта к он-лайн ЦПУ, Вы можете показать или изменить IP-адрес. Вы можете также отобразить или установить параметры времени и даты в он-лайн ЦПУ.



Примечание

Эта функция доступна только для ЦПУ, у которого есть только MAC-адрес (еще не был назначен IP-адрес), или, который был сброшен к заводским настройкам.

15.5 Сброс к заводским настройкам

Вы можете выполнить сброс S7-1200 к оригинальным заводским настройкам при следующих условиях:

- У ЦПУ есть он-лайн соединение.
- ЦПУ находится в режиме STOP.

Примечание

Если ЦПУ находится в режиме RUN, и Вы активируете процедуру сброса, то Вы можете перевести его в режим STOP после квитирования запроса подтверждения.

Процедура

Чтобы выполнить сброс ЦПУ к заводским настройкам, выполните следующие действия:

1. Откройте представление Online and Diagnostics для ЦПУ.
2. Выберите "Reset to factory settings" в разделе "Functions".
3. Установите флажок "Retain IP address", если Вы хотите сохранить IP-адрес или флажок "Delete IP address", если Вы хотите удалить IP-адрес.
4. Нажмите кнопку "Reset".
5. Квитируйте запрос подтверждения с помощью "ОК".

Результат

Модуль переключается в режим STOP при необходимости, и он выполняет сброс к заводским настройкам. ЦПУ выполняет следующие действия:

При наличии карты памяти в ЦПУ	При отсутствии карты памяти в ЦПУ
• Очищает диагностический буфер • Сбрасывает время суток • Восстанавливает рабочую память с карты памяти • Устанавливает все области операндов на сконфигурированные начальные значения • Устанавливает все параметры на их сконфигурированные значения • Сохраняет или удаляет IP-адрес согласно Вашему выбору. (MAC-адрес фиксирован и никогда не изменяется.) ¹ • Удаляет запись управляющих данных (стр. 151), если она существует	• Очищает диагностический буфер • Сбрасывает время суток • Очищает рабочую память и внутреннюю загрузочную память • Устанавливает все области операндов на сконфигурированные начальные значения • Устанавливает все параметры на их сконфигурированные значения • Сохраняет или удаляет IP-адрес согласно Вашему выбору. (MAC-адрес фиксирован и никогда не изменяется.) ¹ • Удаляет запись управляющих, если она существует

¹ Если Вы выбрали "Retain IP address", то ЦПУ устанавливает IP-адрес, маску подсети и адрес маршрутизатора (если используется) к параметрам настройки из Вашей аппаратной конфигурации, если Вы не изменили их из пользовательской программы или другого инструмента, когда ЦПУ восстанавливает измененные значения.

15.6 Обновление встроенного ПО

Вы можете обновить встроенное ПО подключенного ЦПУ из STEP 7 инструментов он-лайн и диагностики.

Чтобы выполнить обновление, выполните следующие действия:

1. Откройте представление Online and Diagnostics подключенного ЦПУ.
2. Выберите "Firmware update" из раздела "Functions".
3. Щелкните по кнопке просмотра и перейдите к расположению, которое содержит файл обновления встроенного ПО. Это мог бы быть каталог на Вашем жестком диске, на который Вы загрузили S7-1200 (<http://support.automation.siemens.com/WW/view/en/34612486/133100>) файл обновления встроенного ПО с веб-сайта сервиса и поддержки (<http://www.siemens.com/automation/>).
4. Выберите файл, который является совместимым с Вашим модулем. Для выбранного файла таблица показывает совместимые модули.
5. Нажмите кнопку "Run update". Следуйте при необходимости указаниям в диалоговых окнах, чтобы изменить рабочий режим Вашего ЦПУ.

STEP 7 показывает окна хода выполнения загрузки обновления встроенного ПО. Когда загрузка заканчивается, Вам предлагается запустить модуль с новым встроенным ПО.

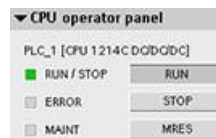
Примечание

Если Вы не принимаете решение запустить модуль с новым встроенным ПО, предыдущее встроенное ПО остается активным, пока Вы не выполните сброс модуля, например, выполняя цикл подачи питания. Новое встроенное ПО становится активным только после того, как Вы выполните сброс модуля.

Вы можете также выполнить обновление встроенного ПО одним из следующих дополнительных методов:

- Используя карту памяти (стр. 141)
- Используя стандартную страницу веб-сервера "Module Information" (стр. 804)

15.7 Операторская панель он-лайн ЦПУ



"Операторская панель ЦПУ" отображает рабочий режим (STOP или RUN) он-лайн ЦПУ. Панель показывает также, есть ли ошибка ЦПУ или форсированные значения.

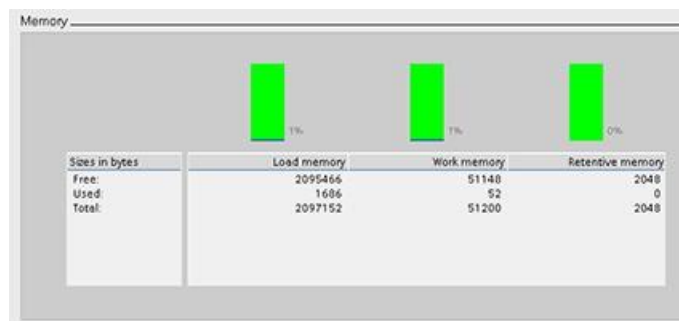
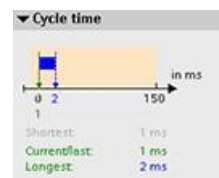
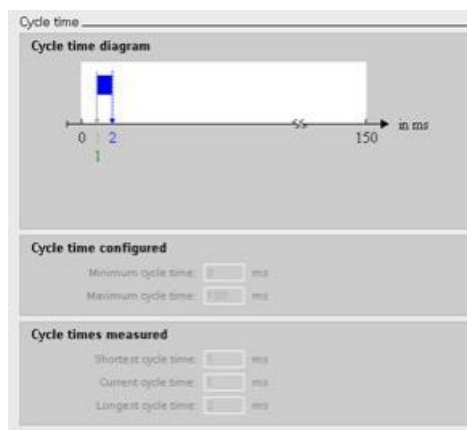
Используйте операторскую панель ЦПУ на карте задач Online Tools, чтобы изменить рабочий режим он-лайн ЦПУ. Карта задач Online Tools доступна каждый раз, когда ЦПУ доступен он-лайн.

15.8 Контроль времени цикла и использование памяти

Вы можете контролировать время цикла и использование памяти он-лайн ЦПУ.

После соединения с он-лайн ЦПУ откройте карту задач Online tools, чтобы наблюдать следующие измерения:

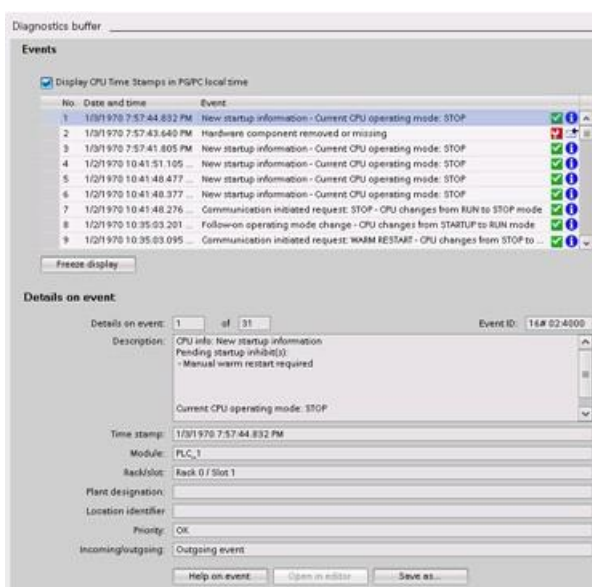
- Время цикла
- Использование памяти



15.9 Отображение диагностических событий в ЦПУ

Используйте диагностический буфер, чтобы просмотреть последние действия в ЦПУ. Диагностический буфер доступен из "Online & Diagnostics" для он-лайн ЦПУ в дереве проекта. Он содержит следующие записи:

- Диагностические события
- Изменения рабочего режима ЦПУ (переходы в режимы STOP или RUN)



Первая запись содержит последнее событие. Каждая запись в диагностическом буфере содержит дату и время, когда событие было зарегистрировано, и описание события.

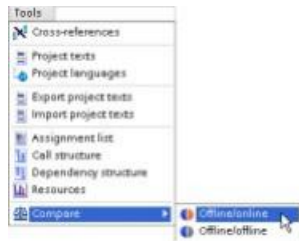
Максимальное количество записей зависит от ЦПУ. Поддерживается максимум 50 записей.

Только 10 последних событий в диагностическом буфере сохраняются долговременно. Сброс ЦПУ к заводским установкам очищает диагностический буфер, удаляя записи.

Вы можете также использовать инструкцию GET_DIAG (стр. 398), чтобы собрать диагностическую информацию.

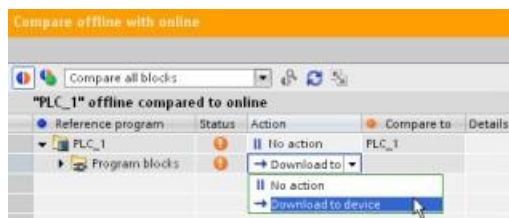
15.10 Сравнение офф-лайн и он-лайн ЦПУ

Вы можете сравнить кодовые блоки в он-лайн ЦПУ с кодовыми блоками в Вашем проекте. Если кодовые блоки Вашего проекта не соответствуют кодовым блокам он-лайн ЦПУ, то редактор "Compare" позволяет Вам синхронизировать свой проект с он-лайн ЦПУ, загружая кодовые блоки из Вашего проекта в ЦПУ или удаляя из проекта блоки, которые не существуют в он-лайн ЦПУ.



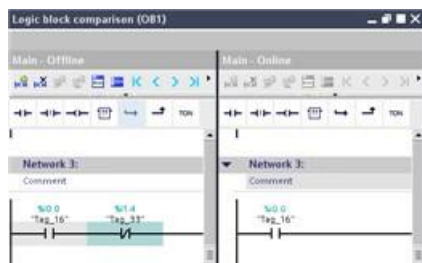
Выберите ЦПУ в своем проекте.

Используйте команду "Compare Offline/online", чтобы открыть редактор "Compare". (Вызовите команду либо из меню "Tools" либо при щелчке правой кнопкой по ЦПУ в проекте.)



Щелкните в столбце "Action" для объекта, чтобы выбрать, удалить ли объект, не предпринимать никаких действий или загрузить объект в устройство.

Нажмите кнопку "Synchronize", чтобы загрузить кодовые блоки.



Щелкните правой кнопкой по объекту в столбце "Compare to" и нажмите кнопку "Start detailed comparison", чтобы отобразить кодовые рядом друг с другом.

Подробное сравнение выделяет различия между кодовыми блоками он-лайн ЦПУ и кодовыми блоками ЦПУ в Вашем проекте.

15.11 Контроль и изменение значений в ЦПУ

STEP 7 предлагает он-лайн инструменты для контроля ЦПУ:

- Вы можете отобразить или наблюдать текущие значения тегов. Функция наблюдения не изменяет последовательность программы. Она предоставляет Вам информацию о последовательности программы и данных программы в ЦПУ.
- Вы можете также использовать другие функции, чтобы управлять последовательностью и данными пользовательской программы:
 - Вы можете изменить значение для тегов в он-лайн ЦПУ, чтобы увидеть реакцию пользовательской программы.
 - Вы можете принудительно устанавливать периферийный вывод (к примеру, Q0.1:P или "Start":P) в определенное состояние.
 - Вы можете разрешить выходы в режиме STOP.

Примечание

Всегда проявляйте осторожность при использовании функций управления. Эти функции могут серьезно повлиять на выполнение пользовательской / системной программы.

Таблица 15- 3 Он-лайн возможности редакторов STEP 7

Редактор	Наблюдение	Изменение	Форсирование
Таблица наблюдения	Да	Да	Нет
Таблица форсирования	Да	Нет	Да
Программный редактор	Да	Да	Нет
Таблица тегов	Да	Нет	Нет
Редактор DB	Да	Нет	Нет

15.11.1. Переход в он-лайн, чтобы наблюдать значения в ЦПУ

Чтобы наблюдать теги у Вас должно быть он-лайн соединение с ЦПУ. Просто нажмите кнопку "Go online" на панели инструментов.



Когда Вы выполнили соединение с ЦПУ, STEP 7 выделяет заголовки рабочих областей оранжевым цветом.

Дерево проекта отображает сравнение офф-лайн проекта и он-лайн ЦПУ. Зеленый кружок означает, что ЦПУ и проект синхронизированы, подразумевая, что они имеют ту же конфигурацию и пользовательскую программу.

Таблица тегов показывает теги. Таблицы наблюдения могут показывать теги, а также абсолютные адреса.

	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I0.0	Bool		
2	"Off"	%I0.1	Bool		
3	"Run"	%Q0.0	Bool		

Чтобы наблюдать выполнение пользовательской программы и отобразить значения тегов, нажмите кнопку "Monitor all" на панели инструментов.

	Name	Address	Display format	Monitor value	Modify value
1	"On"	%I0.0	Bool	FALSE	
2	"Off"	%I0.1	Bool	FALSE	
3	"Run"	%Q0.0	Bool	FALSE	

Поле "Monitor value" отображает значение каждого из тегов.

15.11.2. Отображение состояния в программном редакторе

Вы можете наблюдать состояние до 50 тегов в LAD и FBD программных редакторах. Используйте панель редактора, чтобы вывести на экран редактор LAD. Панель редактора позволяет Вам изменять представление между открытыми редакторами без необходимости открывать или закрывать редакторы.

На панели инструментов редактора программы нажмите кнопку "Monitoring on/off", чтобы отобразить состояние Вашей пользовательской программы.



Сегмент в программном редакторе отображает ход выполнения зеленым цветом.

Вы можете также щелкнуть правой кнопкой по инструкции или параметру, чтобы изменить значение для инструкции.

15.11.3. Захват он-лайн значений DB, чтобы переустановить стартовые значения

Вы можете захватить текущие значения, наблюдаемые в он-лайн ЦПУ, чтобы сделать их стартовыми значениями для глобального DB.

- У Вас должно быть он-лайн соединение с ЦПУ.
- ЦПУ должен находиться в режиме RUN.
- Вы должны открыть DB в STEP 7.



Используйте кнопку "Show a snapshot of the monitored values", чтобы захватить текущие значения выбранных тегов в DB. Вы можете затем скопировать эти значения в столбец DB "Start value".

1. В редакторе DB нажмите кнопку "Monitor all tags". Столбец "Monitor value" отображает текущие значения данных.
2. Нажмите кнопку "Show a snapshot of the monitored values", чтобы показать текущие значения в столбце "Snapshot".
3. Нажмите кнопку "Monitor all", чтобы прекратить наблюдать данные в ЦПУ.
4. Скопируйте значение в столбце "Snapshot" для тега.
 - Выберите копируемое значение.
 - Щелкните правой кнопкой по выбранному значению, чтобы отобразить контекстное меню.
 - Выберите команду "Copy".
5. Вставьте скопированное значение в соответствующий столбец "Start value" для тега. (Щелкните правой кнопкой по ячейке и выберите "Paste" из контекстного меню.)

6. Сохраните проект, чтобы сконфигурировать скопированные значения в качестве новых стартовых значений для DB.
7. Скомпилируйте и загрузите DB в ЦПУ. DB использует новые стартовые значения после того, как ЦПУ перейдет в режим RUN.

Примечание

Значения, которые показаны в столбце "Monitor value", всегда копируются из ЦПУ. STEP 7 не проверяет, поступают ли все значения из одного и того же цикла сканирования ЦПУ.

15.11.4. Использование таблицы наблюдения, чтобы контролировать и изменять значения в ЦПУ

Таблица наблюдения позволяет Вам выполнять функции контроля и управления в точках данных, при обработке ЦПУ Вашей программы. Эти точки данных могут быть образом процесса (I или Q), M, DB или физическими входами (I:P), в зависимости от функции контроля или управления. Вы не можете строго контролировать физические выходы (Q:P), потому что функция контроля может отобразить только последнее значение, записанное из Q памяти, и не читает фактическое значение из физических выходов.

Функция контроля не изменяет последовательность программы. Она предоставляет Вам информацию о последовательности программы и данных программы в ЦПУ.

Функции управления позволяют пользователю управлять последовательностью и данными программы. Вы должны проявить осторожность при использовании функций управления. Эти функции могут серьезно повлиять на выполнение пользовательской / системной программы. Этими тремя функциями управления, являются Modify, Force и Enable Outputs in STOP.

С помощью таблицы наблюдения Вы можете выполнять следующие он-лайн функции:

- Наблюдать состояние тегов
- Изменять значения отдельных тегов

Вы выбираете, когда наблюдать или изменять тег:

- Beginning of scan cycle: Читает или записывает значение в начале цикла сканирования
- End of scan cycle: Читает или записывает значение в конце цикла сканирования
- Switch to stop



Для создания таблицы наблюдения:

1. Дважды щелкните по "Add new watch table", чтобы открыть новую таблицу наблюдения.
2. Введите имя тега, чтобы добавить тег в таблицу наблюдения.

Следующие опции доступны для наблюдения тегов:

- Monitor all: Эта команда запускает контроль видимых тегов в активной таблице наблюдения.
- Monitor now: Эта команда запускает контроль видимых тегов в активной таблице наблюдения. Таблица наблюдения контролирует теги сразу и однократно.

Следующие опции доступны для изменения тегов:

- "Modify to 0" устанавливает значение выбранного адреса на "0".
- "Modify to 1" устанавливает значение выбранного адреса на "1".
- "Modify now" незамедлительно изменяет значение для выбранных адресов на один цикл сканирования.
- "Modify with trigger" изменяет значения для выбранных адресов.
Эта функция не обеспечивает обратную связь, чтобы отобразить фактические изменения выбранных адресов. Если требуется обратная связь по изменениям, используйте функцию "Modify now".
- "Enable peripheral outputs" отменяет запрет на управление выходами и доступна только, когда ЦПУ находится в режиме STOP.

Чтобы наблюдать теги, у Вас должно быть он-лайн соединение с ЦПУ.

	Name	Address	Display format	Monitor value	Monitor with trigger	Modify with trigger	Modify value
1	"Start"	%I0.0	Bool		Permanent	Permanent	
2	"Stop"	%I0.1	Bool		Permanent	Permanent	
3	"Running"	%M0.0	Bool		Permanent	Permanent	

Вы используете кнопки вверху таблицы наблюдения, чтобы выбрать различные функции.

Введите имя тега для наблюдения и выберите формат отображения из выпадающего списка. При наличии он-лайн соединения с ЦПУ нажмите кнопку "Monitor", чтобы отобразить фактическое значение точки данных в поле "Monitor value".

15.11.4.1. Использование триггера при наблюдении или изменении ПЛК тегов

Запуск по триггеру определяет то, в какой точке цикла сканирования будет контролироваться или изменяться выбранный адрес.

Таблица 15- 4 Типы триггеров

Триггер	Описание
Permanent	Постоянно собирает данные
At scan cycle start	Permanent: Постоянно собирает данные в начале цикла сканирования, после того, как ЦПУ считывает входы
	Opse: Собирает данные в начале цикла сканирования, после того, как ЦПУ считывает входы
At scan cycle end	Permanent: Постоянно собирает данные в конце цикла сканирования, прежде чем ЦПУ запишет выходы
	Opse: Собирает данные один раз в конце цикла сканирования, прежде чем ЦП запишет выходы
At transition to STOP	Permanent: Постоянно собирает данные когда ЦПУ переходит в STOP
	Opse: Собирает данные один раз после переходов ЦПУ в STOP

Для изменения ПЛК тега с заданным триггером выберите начало или конец цикла.

- Изменение выхода: Наилучшим моментом для изменения выхода является конец цикла сканирования, непосредственно перед записью ЦПУ в выходы.

Наблюдайте значение выходов в начале цикла сканирования, чтобы определить какое значение записывается в физические выходы. Кроме того, наблюдайте выходы, прежде чем ЦПУ запишет значения в физические выходы, чтобы проверить логику программы и сравнить с поведением фактического ввода-вывода.

- Изменение входа: наилучшим моментом для изменения входа является начало цикла, сразу после того, как ЦПУ читает входы и прежде, чем пользовательская программа будет использовать входные значения.

Если Вы предполагаете, что значения изменяются во время сканирования, Вы возможно захотели бы наблюдать значение входов в конце цикла сканирования, чтобы гарантировать, что значение входа в конце цикл сканирования не изменилось по отношению к началу цикла сканирования. Если есть различие в значениях, то Ваша пользовательская программа, возможно, ошибочно записывает входы.

Чтобы диагностировать, возможную причину перехода ЦПУ в STOP, используйте триггер "Transition to STOP", чтобы захватить последние значения процесса.

15.11.4.2. Разрешение выходов в режиме STOP

Таблица наблюдения позволяет Вам выполнять запись в выходы, когда ЦПУ находится в режиме STOP. Эта функциональность позволяет Вам проверить проводные соединения выходов и убедиться, что провод, соединенный с выходным контактом, инициирует сигнал высокого или низкого уровня на клемме устройства, с которым он соединен.



ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с записью физических выходов в режиме STOP

Даже при условии, что ЦПУ находится в режиме STOP, разрешение физических выходов может активировать канал процесса, к которому этот выход подключен, возможно, вызвав непредсказуемую работу оборудования. Непредсказуемая работа оборудования может привести к смерти или тяжелым телесным повреждениям.

Прежде, чем выполнить запись выхода из таблицы наблюдения, убедитесь, что изменение состояния физического выхода не сможет привести к непредсказуемой работе оборудования. Всегда соблюдайте технику безопасности для своего технологического оборудования.

Вы можете изменить состояние выходов в режиме STOP, когда выходы разрешены. Если выходы запрещены, Вы не можете изменять выходы в режиме STOP. Чтобы разрешить изменение выходов в режиме STOP из таблицы наблюдения, выполните следующие шаги:

1. Выберите команду "Expanded mode" из меню "Online".
2. Выберите опцию "Enable peripheral outputs" команды "Modify" из меню "Online" или из контекстного меню после щелчка правой кнопкой по строке таблицы наблюдения. Вы не можете разрешить выходы в режиме STOP, если Вы сконфигурировали распределенный ввод-вывод. При попытке сделать это возвращается ошибка.

Перевод ЦПУ в режим RUN отменяет опцию "Enable peripheral outputs".

Если какие-либо входы или выходы форсированы, ЦПУ не позволяет разрешить выходы в режиме STOP. Вначале необходимо отменить функцию форсирования.

15.11.5. Форсирование значений в ЦПУ

15.11.5.1. Использование таблицы форсирования

Таблица форсирования обеспечивает функцию "принудительной установки", которая перезаписывает значение для входного или выходного канала на заданное для периферийного входного или выходного адреса значение. ЦПУ применяет это принудительное значение к входному образу процесса до выполнения пользовательской программы и к выходному образу процесса, прежде чем выходы будут записаны в модули.

Примечание

Форсированные значения сохраняются в ЦПУ а не в таблице форсирования.

Вы не можете форсировать вход ("I" адрес) или выход ("Q" адрес). Однако Вы можете форсировать периферийный вход или периферийный выход. Таблица форсирования автоматически добавляет ":P" к адресу (например: "On":P или "Run":P).

	Name	Address	Display format	Monitor value	Force value	
1	"On":P	%I0.0 P	Bool		TRUE	☒
2	"Off":P	%I0.1 P	Bool			☐
3	"Run":P	%Q0.1 P	Bool			☐

В ячейке "Force value" введите значение для входа или выхода, который будет форсирован. Вы можете затем использовать флажок в столбце "Force", чтобы разрешить форсирование входа или выхода.



Используйте кнопку "Start or replace forcing", чтобы форсировать значение тегов в таблице форсирования. Нажмите кнопку "Stop forcing", чтобы сбросить значение тегов.

В таблице форсирования Вы можете наблюдать состояние принудительного значения для входа. Однако Вы не можете наблюдать принудительное значение выхода.

Вы можете также просмотреть состояние принудительного значения в программном редакторе.



Примечание

Когда вход или выход принудительно пишется в таблице форсирования, форсирующие воздействия становятся частью конфигурации проекта. Если Вы закрываете STEP 7, форсированные элементы остаются активными в программе ЦПУ, пока они не будут сброшены. Чтобы сбросить эти принудительные элементы, Вы должны использовать STEP 7, чтобы подключиться к он-лайн ЦПУ, а затем использовать таблицу форсирования, чтобы выключить или остановить функцию форсирования для этих элементов.

15.11.5.2. Действие функции Force

ЦПУ позволяет Вам принудительно устанавливать канал(-ы) ввода и вывода, определяя физический адрес ввода или вывода (I_:P или Q_:P) в таблице форсирования и запуская функцию форсирования.

В программе считанное значение физического входа перезаписывается принудительным значением. Программа использует принудительное значение в обработке. Когда программа пишет физический выход, выходное значение перезаписывается форсированным значением. Принудительное значение появляется на физическом выходе и используется процессом.

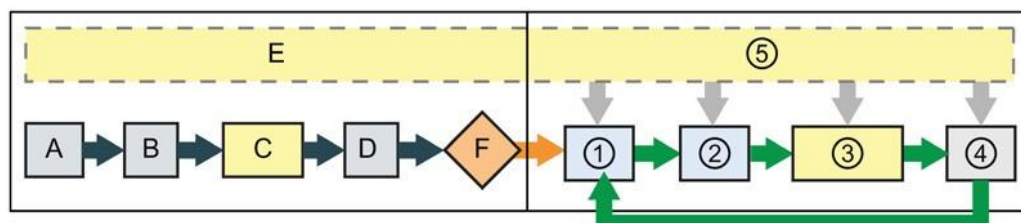
Когда вход или выход вызван в таблице форсирования, форсирующие воздействия становятся частью пользовательской программы. Даже при условии, что среда программирования была закрыта, форсированные значения остаются активными в работающей программе ЦПУ, пока они не будут сброшены при переходе он-лайн с помощью среды программирования, путем остановки функции форсирования. Программы с форсированными каналами, загруженными в другой ЦПУ из карты памяти, будут продолжать форсировать каналы, выбранные в программе.

Если ЦПУ выполняет пользовательскую программу с карты памяти защищенной от записи, Вы не можете инициировать или изменить форсирование ввода-вывода из таблицы наблюдения, потому что Вы не можете переопределить значения в пользовательской программе защищенной от записи. Любая попытка вызвать значения защищенные от записи генерирует ошибку. Если Вы будете использовать карту памяти, чтобы передать пользовательскую программу, то любые форсированные элементы на этой карте памяти будут переданы в ЦПУ.

Примечание

Цифровые каналы ввода-вывода, назначенные устройствам высокоскоростного счета, ШИМ и генерации последовательности импульсов, не могут быть форсированы

Цифровые каналы ввода-вывода, используемые устройствами высокоскоростного счета (HSC), широтно-импульсной модуляции (PWM) и генерации последовательности импульсов (PTO), назначаются во время конфигурации устройства. Когда адреса каналов цифрового ввода-вывода назначаются этим устройствам, значения назначенных адресов каналов ввода-вывода не могут быть изменены функцией принудительной установки в таблицы форсирования.



STARTUP

- A Очистка I области памяти не затрагивается функцией Force.
- B Инициализация выходных значений не затрагивается функцией Force.
- C В процессе выполнения ОВ запуска ЦПУ применяет форсированные значения, когда программа пользователя обращается к физическому входу.
- D Помещение событий прерывания в очередь не затрагивается.
- E Разрешение записи в выходы не затрагивается.

RUN

- ① При записи Q памяти в физические выходы ЦПУ применяет форсированные значения, поскольку выходы обновляются.
- ② При чтении физических входов ЦПУ применяет форсированные значения как раз перед копированием входов в I память.
- ③ Во время выполнения пользовательской программы (ОВ программного цикла) ЦПУ применяет форсированные значения, когда пользовательская программа обращается к физическому входу или пишет физический выход.
- ④ Обработка коммуникационных запросов и самотестирование не затрагиваются функцией Force.
- ⑤ Обработка прерываний во время любого этапа цикла сканирования не затрагивается.

15.12 Загрузка в режиме RUN

ЦПУ поддерживает "Загрузку в режиме RUN". Эта функция предназначена, чтобы дать Вам возможность вносить небольшие изменения в пользовательскую программу с минимальным воздействием на процесс, управляемый программой. Однако использование этой возможности также позволяет крупные изменения в программе, что могло бы стать дезорганизующим или даже опасным воздействием.



ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с загрузкой в режиме RUN

Когда Вы загружаете изменения в ЦПУ в режиме RUN, то изменения сразу влияют на работу процесса. Изменение программы в режиме RUN может привести к непредсказуемой работе системы, что могло бы привести к смерти или серьезным травмам персонала и/или повреждению оборудования.

Только уполномоченные лица, которые осознают эффект от внесения изменений в режиме RUN на работу системы, должны выполнить загрузку в режиме RUN.

Функция "Загрузка в режиме RUN" позволяет Вам вносить изменения в программу и загружать их в Ваш ЦПУ, не переключаясь в режим STOP:

- Вы можете внести незначительные изменения в свой работающий процесс, не останавливая его (например, изменить значение параметра).
- С этой функцией Вы можете отладить программу более быстро (например, инвертировать логику для нормально разомкнутого или нормально замкнутого переключателя).

Вы можете выполнить следующие изменения и загрузить их в режиме RUN:

- Создание, перезапись и удаление функции (FC), функционального блока (FB) и таблицы тегов.
- Создание, удаление и перезапись блоков данных (DB) и экземплярных блоков данных для функциональных блоков (FB). Вы можете добавить элементы в структуры DB и загрузить их в режиме RUN. ЦПУ может сохранить значения существующих тегов блока и инициализировать новые теги блока данных согласно начальным значениям, или ЦПУ может установить все теги блока данных в начальные значения, в зависимости от Ваших настроек конфигурации (стр. 1092). Вы не можете загрузить DB веб-сервера (управление или фрагмент) в режиме RUN.
- Перезапись организационных блоков (OB); однако, нельзя создать или удалить OB.

Вы можете загрузить максимум двадцать блоков в режиме RUN за один раз. Если Вы должны загрузить больше двадцати блоков, Вам следует перевести ЦПУ в режим STOP.

Если Вы загружаете изменения в реальный процесс (в противоположность моделируемому процессу, что Вы могли бы делать в ходе отладки программы), крайне важно продумать с точки зрения безопасности возможные последствия для машин и их операторов, прежде чем Вы выполните загрузку.

Примечание

Если ЦПУ находится в режиме RUN, и в программу были внесены изменения, STEP 7 всегда пытается вначале загрузить в RUN. Если Вы не хотите, чтобы это произошло, Вам следует перевести ЦПУ в STOP.

Если внесенные изменения не поддерживаются в "Загрузкой в RUN", STEP 7 предлагает пользователю перевести ЦПУ в STOP.

15.12.1. Предпосылки для "загрузки в режиме RUN"

Возможность загрузить Ваши программные изменения в ЦПУ, который находится в режиме RUN, предполагает наличие следующих предпосылок:

- Ваш ЦПУ должен иметь версию V3.0 или выше

Примечание

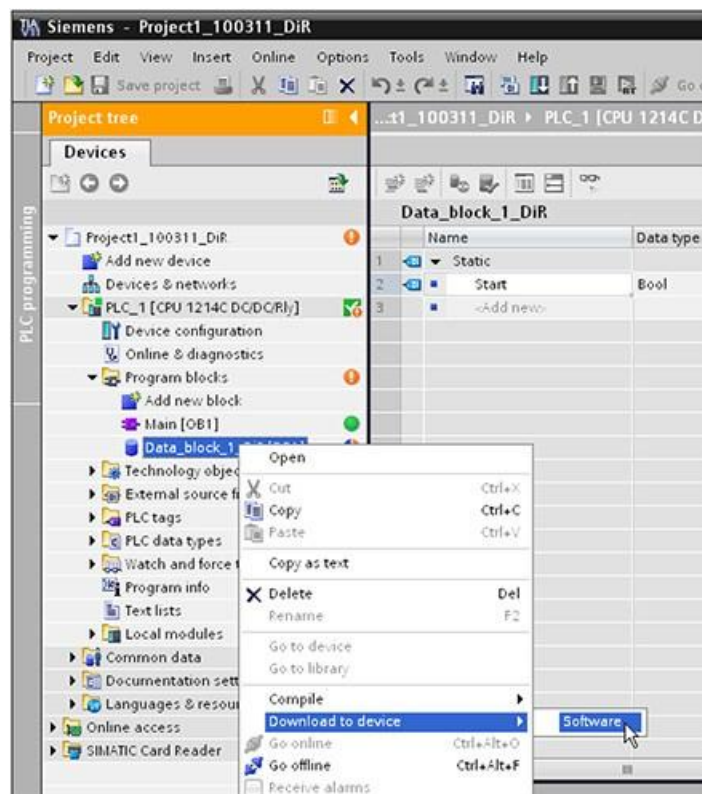
Версия Вашего ЦПУ должна быть не ниже V4.0, чтобы изменить существующие блоки и загрузить расширенный интерфейс блока в режиме RUN (стр. 1092).

- Ваша программа должна быть успешно скомпилирована.
- Вы, должно иметь, успешно установленное соединение между устройством программирования, где работает STEP 7 и ЦПУ.

15.12.2. Изменение Вашей программы в режиме RUN

Чтобы изменить программу в режиме RUN, Вы должны вначале убедиться, чтобы ЦПУ и программа удовлетворяют необходимым предпосылкам (стр.1088), и затем выполнить следующие действия:

1. Чтобы загрузить Вашу программу в режиме RUN, действуйте согласно одному из следующих методов:
 - Выберите команду "Download to device" из меню "Online".
 - Нажмите кнопку "Download to device" на панели инструментов.
 - В "дереве проекта", щелкните правой кнопкой по "Program blocks" и выберите команду "Download to device > Software".



Если программа компилируется успешно, STEP 7 начинает загружать программу в ЦПУ.

2. Когда STEP 7 предложит Вам загрузить Вашу программу или отменить операцию, нажмите "Load", чтобы загрузить программу в ЦПУ.

15.12.3. Загрузка выбранных блоков

Из папки Program blocks Вы можете выбрать отдельный блок или несколько блоков для загрузки.

Если Вы выбираете отдельный блок для загрузки, то единственной опцией в столбце "Action" является "Consistent download".

Вы можете развернуть строку категории, чтобы убедиться, какие блоки должны быть загружены. В данном примере были выполнены незначительные изменения в оффлайн блоке, и никакие другие блоки не должны быть загружены.



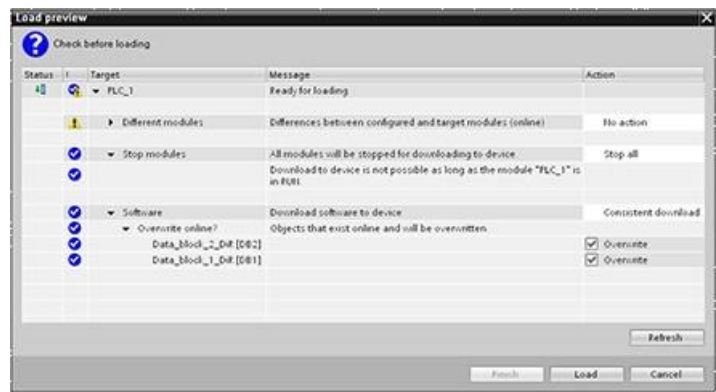
В данном примере необходима загрузка больше чем одного блока.



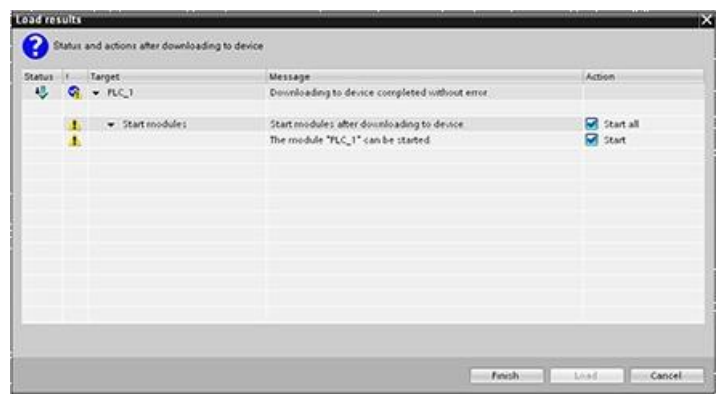
Примечание

Вы можете загрузить максимум двадцать блоков в режиме RUN за один раз. Если Вы должны загрузить больше двадцати блоков, Вам следует перевести ЦПУ в режим STOP.

Если Вы пытаетесь выполнить загрузку в RUN, но система обнаруживает, что это невозможно до фактической загрузки, в диалоговом окне появляется строка категории Stop modules.



Нажмите кнопку "Load" и появляется диалог "Load results". Нажмите кнопку "Finish", чтобы завершить загрузку.



15.12.4. Загрузка одиночного выбранного блока с ошибкой компиляции в другом блоке

Если Вы пытаетесь выполнить согласованную загрузку с ошибкой компиляции в другом блоке, то диалоговое окно указывает на ошибку, и кнопка загрузки неактивна.



15.12 Загрузка в режиме RUN

Вы должны исправить ошибку компиляции в другом блоке. После этого кнопка "Load" становится активной.

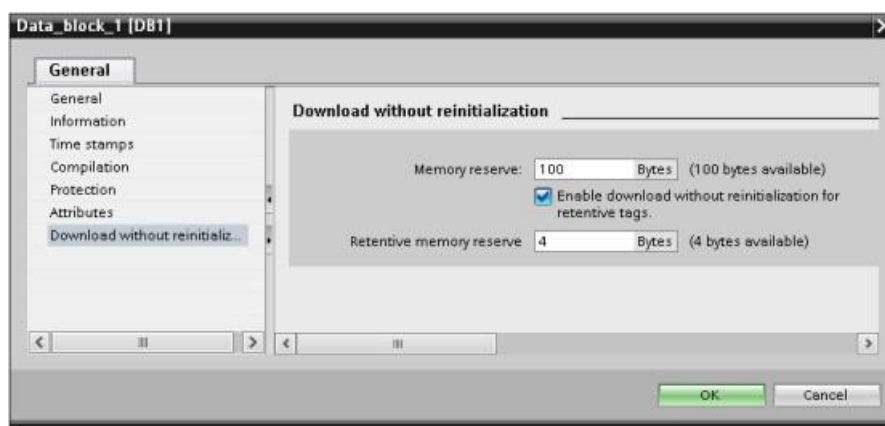


15.12.5. Изменение и загрузка существующих блоков в режиме RUN

Функция "Загрузка в Run" позволяет Вам добавлять и изменять теги в блоках данных и функциональных блоках и затем загружать измененный блок в ЦПУ в режиме RUN.

Загрузка без реинициализации

У каждого DB и FB есть объем зарезервированной памяти, которую Вы можете использовать для добавления тегов в блок, который Вы можете впоследствии загрузить в режиме RUN. По умолчанию начальный размер резерва памяти составляет 100 байтов. Вы можете добавлять дополнительные теги к Вашим данным размером не более, чем резерв памяти и загружать расширенный блок в ЦПУ в режиме RUN. Вы можете также увеличить резерв, если Вам требуется больше памяти для дополнительных тегов в Вашем блоке. Если Вы добавляете больше тегов, чем выделенный объем памяти, Вы не можете загрузить расширенный блок в ЦПУ в режиме RUN.



Функция "Download without reinitialization" позволяет Вам расширять блок данных, добавляя больше тегов блока данных и загружать расширенный блок данных в режиме RUN. В этом случае Вы можете добавить теги в блок данных и загрузить его, не выполняя повторную инициализацию Вашей программы. ЦПУ сохраняет значения существующих тегов блока данных и инициализирует вновь добавленные теги согласно их стартовым значениям.

Чтобы разрешить эту функцию для он-лайн проекта с ЦПУ в режиме RUN, выполните следующие шаги:

1. Из папки Program blocks в дереве STEP 7 проекта откройте блок.
2. Щелкните по выключателю "Download without reinitialization" в блочном редакторе, чтобы разрешить функцию. (Значок выделен рамкой, когда Вы разрешили ее: )
3. Нажмите "ОК" в диалоговом окне, чтобы подтвердить Ваш выбор.
4. Добавьте теги в интерфейс блока и загрузите блок в режиме RUN. Вы можете добавить и загрузить столько новых тегов, сколько позволяет Ваш резерв памяти.

Если Вы добавили больше байтов к своему блоку, чем Вы сконфигурировали для резерва памяти, STEP 7 показывает ошибку, когда Вы пытаетесь загрузить блок в режиме RUN. Вы должны отредактировать свойства блока, чтобы увеличить резерв. Вы не можете удалить существующие записи или изменить "Резерв памяти" блока, в то время как включена функция "Загрузка без реинициализации". Чтобы отключить "Загрузку без реинициализации" функция, выполните следующие шаги:

1. Щелкните по выключателю "Download without reinitialization" в редакторе блока, чтобы отключить функцию. (У значка нет рамки вокруг, когда Вы отключили ее: )
2. Нажмите "ОК" в диалоговом окне, чтобы подтвердить Ваш выбор.
3. Загрузите блок. В диалоговом окне загрузки Вы должны выбрать "reinitialize", чтобы загрузить расширенный блок.

Загрузка в этом случае повторно инициализирует все существующие и новые теги блока согласно их стартовым значениям.

Загрузка сохраняемых тегов блока

Загрузка сохраняемых тегов блока в режиме RUN требует выделения резерва сохраняемой памяти. Чтобы сконфигурировать этот резерв сохраняемой памяти, выполните следующие действия:

1. Из папки Program blocks в дереве STEP 7 проекта щелкните правой кнопкой по блоку и выберите "Properties" из контекстного меню.
2. Выберите свойство "Download without reinitialization".
3. Установите флажок для, "Enable download without reinitialization for retentive tags".
4. Сконфигурируйте число байтов, доступных для резерва сохраняемой памяти.
5. Нажмите "ОК", чтобы сохранить Ваши изменения.
6. Добавьте сохраняемые теги в блок данных и загрузите блок данных в режиме RUN. Вы можете добавить и загрузить столько новых сохраняемых тегов блока данных, сколько позволяет Ваш резерв сохраняемой памяти.

Если Вы добавили больше сохраняемых байтов в свой блок данных, чем Вы сконфигурировали для резерва сохраняемой памяти, STEP 7 показывает ошибку, когда Вы пытаетесь загрузить блок в режиме RUN. Вы можете добавить сохраняемые теги блока объемом не более резерва сохраняемой памяти, чтобы позволить их загрузку в режиме RUN.

Когда Вы загружаете дополнительные сохраняемые теги блока, теги содержат свои текущие значения.

Конфигурирование объема зарезервированной памяти для новых блоков

Объем резерва памяти по умолчанию для новых блоков данных составляет 100 байтов. Когда Вы создаете новый блок, он имеет в наличии 100 резервных байтов. Если Вы хотите, чтобы размер резерва памяти для новых блоков был другим, Вы можете изменить настройки "PLC programming":

1. Из STEP 7 выберите команду меню **Options > Settings**.
2. В диалоговом окне Settings разверните "PLC programming" и выберите "General".
3. В разделе "Download without reinitialization" введите число байтов для резерва памяти.

Когда Вы создаете новые блоки, STEP 7 использует сконфигурированный резерв памяти, который Вы ввели для новых блоков.

Ограничения

Следующие ограничения применяются к редактированию и загрузке блоков в режиме RUN:

- Расширение интерфейса блока путем добавления новых тегов и загрузка в режиме RUN доступны только для оптимизированных блоков (стр. 180).
- Вы не можете изменить структуру блока и загрузить измененный блок в режиме RUN без реинициализации. Добавление новых элементов в тег с типом Struct (стр. 125), изменение имен тегов, размеров массивов, типов данных или состояния сохраняемости требуют, чтобы Вы повторно инициализировали блок, если Вы загружаете его в режиме RUN. Единственными изменениями существующих тегов блока, которые Вы можете выполнить, и загрузить блок в режиме RUN без реинициализации, являются изменения стартовых значений (блоки данных), значения по умолчанию (функциональные блоки) или комментарии.

Примечание

Возможность изменять блоки и загружать их в режиме RUN является новой функцией, начиная с ЦПУ S7-1200 V4.0. До V4.0 Вы могли загрузить измененные блоки только в режиме STOP.

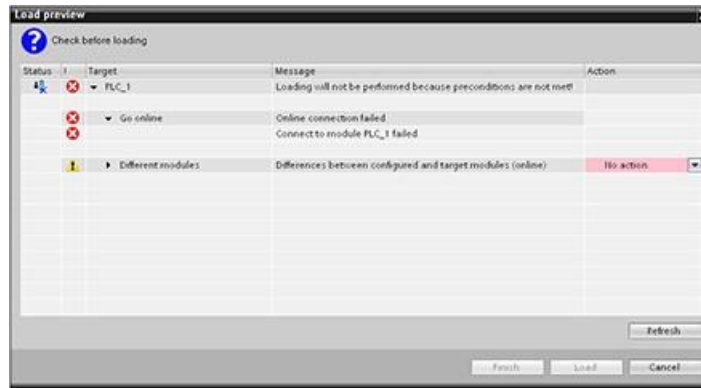
- Вы не можете загрузить больше новых тегов блока в режиме RUN, чем позволяет разместить резерв памяти.
- Вы не можете загрузить больше новых сохраняемых тегов блока в режиме RUN, чем позволяет разместить резерв сохраняемой памяти.

Смотри также

Замена устройств и совместимость запасных частей (стр.1287).

15.12.6. Реакция системы, если процесс загрузки окончился неудачей

Во время начальной загрузки в режиме RUN, если происходит сбой сетевого соединения, STEP 7 выводит на экран следующее "Load preview" диалоговое окно:



15.12.7. Факторы для рассмотрения при загрузке в режиме RUN

Прежде, чем загрузить программу в режиме RUN, продумайте влияние изменений в режиме RUN на работу ЦПУ для следующих ситуаций:

- Если Вы удалили управляющую логику для выхода, ЦПУ сохраняет последнее состояние выхода до следующего цикла включения и выключения питания или перехода в режим STOP.
- Если Вы удалили функции высокоскоростного счета или импульсного выхода, которые работали, то высокоскоростной счетчик или импульсный выход продолжают работать до следующего цикла включения и выключения питания или перехода в режим STOP.
- Любая логика, которая зависит от бита первого сканирования, не будет выполняться до следующего цикла включения и выключения питания или перехода из STOP в RUN. Бит первого сканирования, устанавливается только при переходе в режим RUN и загрузка в режиме RUN не оказывает на него никакого воздействия.

- Текущие значения блоков данных (DB) и/или тегов могут быть перезаписаны.

Примечание

Прежде чем Вы сможете загрузить свою программу в режиме выполнения, ЦПУ должен поддерживать изменения в режиме RUN программа должна скомпилироваться без ошибок, и соединение между STEP 7 и ЦПУ должно быть устойчивым. Вы можете сделать следующие изменения в программных блоках и тегах и загрузить их в режиме RUN:

- Создать, перезаписать и удалить функции (FC), функциональные блоки (FB) и таблицы тегов.
- Создать и удалить блоки данных (DB); однако, Изменения структуры DB не могут быть перезаписаны. Могут быть перезаписаны начальные значения DB. Вы не можете загрузить DB веб-сервера (управление или фрагмент) в режиме RUN.
- Перезаписать организационные блоки (OB); однако, нельзя создать или удалить OB.

Вы можете загрузить максимум двадцать блоков в режиме RUN за один раз. Если Вы должны загрузить больше двадцати блоков, Вам следует перевести ЦПУ в режим STOP.

Как только Вы инициируете загрузку, Вы не можете выполнять другие задачи в STEP 7, пока загрузка не завершится.

Инструкции, которые могли бы перестать работать из-за "загрузки в режиме RUN"

Следующие инструкции могли бы столкнуться с нерегулярной ошибкой, когда в ЦПУ активируются изменения после загрузки в режиме RUN. Ошибка происходит, когда инструкция вызывается, в то время как ЦПУ готовится активировать загруженные изменения. В течение этого времени ЦПУ приостанавливает активацию доступа пользовательской программы к загрузочной памяти, в то время как он завершает активный доступ пользовательской программы к загрузочной памяти. Это делается для того, чтобы загруженные изменения могли быть активированы согласованно.

Инструкция	Отклик при ожидании активации
DataLogCreate	STATUS = W#16#80C0, ERROR = TRUE
DataLogOpen	STATUS = W#16#80C0, ERROR = TRUE
DataLogWrite	STATUS = W#16#80C0, ERROR = TRUE
DataLogClose	STATUS = W#16#80C0, ERROR = TRUE
DataLogNewFile	STATUS = W#16#80C0, ERROR = TRUE
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
RTM	RET_VAL = 0x80C0

Во всех случаях RLO выход инструкции будет иметь значение ЛОЖЬ, когда происходит ошибка. Ошибка является временной. Если это происходит, инструкция должна быть повторена позже.

Примечание

Вы не должны повторять операцию в текущем исполнении OB.

15.13 Трассировка и запись данных ЦПУ по условиям запуска

STEP 7 обеспечивает функции трассировки и анализатора логики, с которыми Вы можете сконфигурировать переменные, которые ПЛК будет отслеживать и записывать. Вы можете затем загрузить зарегистрированные данные измерений в свое устройство программирования и использовать инструменты STEP 7, чтобы анализировать, обрабатывать и представлять Ваши данные в виде графиков. Вы используете папку Traces в дереве STEP 7 проекта, чтобы создавать и обрабатывать трассировки.

Следующий рисунок показывает различные этапы функции трассировки:



- ① Сконфигурируйте трассировку в редакторе трассировки STEP 7. Вы можете сконфигурировать значения данных для записи, продолжительность записи, интервал записи и условие запуска.
- ② Передайте конфигурацию трассировки из STEP 7 в ПЛК.
- ③ ПЛК выполняет программу, и когда выполняется условие запуска, начинает записывать данные трассировки.
- ④ Передайте зарегистрированные значения из ПЛК в STEP 7.
- ⑤ Используйте инструменты в STEP 7, чтобы проанализировать данные, представить их графически и сохранить.

Максимальный объем данных составляет 512 кбайт на трассировку.

Доступ к примерам

Обратитесь к информационной системе STEP 7 для получения дополнительной информации о том, как программировать трассировку, как загрузить конфигурацию, загрузить данные трассировки и отобразить данные в анализаторе логики. Вы можете найти там подробные примеры в главе "Использование он-лайн и диагностических функций > Использование функции трассировки и анализатора логики".

Кроме того превосходной ссылкой является он-лайн руководство "Промышленная Автоматизация SINAMICS/SIMATIC. "Использование функции трассировки и анализатора логики" (<http://support.automation.siemens.com/WW/view/en/64897128>).

Технические данные

A.1. Общие технические данные

Соответствие стандартам

Система автоматизации S7-1200 удовлетворяет следующим стандартам и тестовым техническим заданиям. Критерии тестирования для системы автоматизации S7-1200 основаны на этих стандартах и тестовых технических заданиях.

Обратите внимание на то, что не все модели S7-1200 могут быть сертифицированы согласно этим стандартам, и состояние сертификации может измениться без уведомления. В сферу Вашей ответственности входит определение применимых сертификаций посредством маркировок, нанесенных на продукт. Проконсультируйтесь с местным представителем Сименс, если Вам нужна дополнительная информация, связанная с последним перечнем конкретных допусков по заказному номеру.

Допуск к эксплуатации CE



Система автоматизации S7-1200 удовлетворяет требованиям и целям обеспечения безопасности перечисленных ниже директив ЕС и соответствует согласованным Европейским стандартам (EN) для программируемых контроллеров, опубликованным в официальных бюллетенях Европейского сообщества.

- Директива ЕС 2006/95/ЕС (Директива по низкому напряжению) "Электрическое оборудование, спроектированное для использования в определенных границах напряжения"
 - EN 61131-2:2007 Программируемые контроллеры - Требования к оборудованию и испытания
- Директива ЕС 2004/108/ЕС (Директива по ЭМС) "Электромагнитная совместимость"
 - Стандарт излучения
EN 61000-6-4:2007+A1:2011: Промышленная среда
 - Стандарт помехозащищенности
EN 61000-6-2:2005: Промышленная среда
- EC Directive 94/9/EC (ATEX) "Оборудование и системы защиты, предназначенные для использования в потенциально взрывоопасной атмосфере"
 - –EN 60079-15:2010: Тип защиты 'n'

Декларация соответствия требованиям ЕС хранится для предоставления всем компетентным органам власти по адресу:

Siemens AG
Sector Industry
IA AS FA DH AMB
Postfach 1963
D-92209 Amberg Germany

Допуск к эксплуатации cULus



Underwriters Laboratories Inc. [Компания по стандартизации и сертификации в области техники безопасности, США (Прим.перев.)], отвечая требованиям:

- Underwriters Laboratories, Inc.: UL 508 перечислено (Промышленные устройства управления)
- Canadian Standards Association [Канадская ассоциация стандартов (Прим. перев.)]: CSA C22.2 номер 142 (Оборудование для управления процессами)

Примечание

Серия SIMATIC S7-1200 удовлетворяет стандарту CSA.

Логотип cULus указывает на то, что S7-1200 был проверен и сертифицирован Underwriters Laboratories (UL) в соответствии со стандартами UL 508 и CSA 22.2 No. 142.

Допуск к эксплуатации FM



Factory Mutual Research (FM)

Стандартный класс допуска к эксплуатации 3600 и 3611

Допущено для использования в:

Класс I, раздел 2, газовая группа A, B, C, D, температурный класс T3C Ta = 60 °C

Класс I, зона 2, IIC, температурный класс T3 Ta = 60 °C

Канадский класс I, зона 2 монтаж по CEC 18-150

ВАЖНОЕ ИСКЛЮЧЕНИЕ: См. Технические данные для количества входов или выходов, активированных одновременно. Для некоторых моделей норма снижается для Ta = 60 °C.

ПРЕДУПРЕЖДЕНИЕ

Замена компонентов может негативно повлиять на пригодность для Класса I, раздела 2 и зоны 2.

Восстановление модулей должно выполняться только авторизованным сервисным центром Сименс.

Допуск к эксплуатации IECEx

EN 60079-0: Взрывоопасные среды – общие требования

EN60079-15: Электрический прибор для потенциально взрывоопасных сред;

Тип защиты 'nA'

IECEx FMG14.0012X

Ex nA IIC Tx Gc

IECEx маркировка может появиться на продукте вместе с информацией FM Опасная зона.

Утверждены только изделия, имеющие маркировку IECEx. Проконсультируйтесь с местным представителем Сименс, если Вам нужна дополнительная информация, связанная с последним перечнем конкретных допусков по заказному номеру.

Релейные модели не включены в IECEx допуски. Обратитесь к специфической маркировке продукта для оценки температурного класса.

Устанавливайте модули в подходящем корпусе, обеспечивающем минимальную степень защиты IP54 согласно IEC 60079-15.

Допуск к эксплуатации АТЕХ



ATEX допуск применим только к моделям пост.т. АТЕХ допуск не применим к релейным моделям и моделям пер.т.

EN 60079-0:2009: Взрывоопасные газовые среды – Общие требования

EN 60079-15:2010: Электрический прибор для потенциально взрывоопасных сред; Тип защиты 'nA'

II 3 G Ex nA IIC T4 or T3 Gc

Устанавливайте модули в подходящем корпусе, обеспечивающем минимальную степень защиты IP54 согласно EN 60529 в месте, обеспечивающем эквивалентную степень защиты.

Присоединенные кабели и проводники должны быть классифицированы для фактической температуры, измеренной в расчетных условиях.

Установка должна гарантировать, что переходные процессы ограничены менее чем 119 В. Устойчивость к динамическим изменениям напряжения питания в настоящем разделе.

ВАЖНОЕ ИСКЛЮЧЕНИЕ: См. Технические данные для количества входов или выходов, активированных одновременно. Для некоторых моделей норма снижается для $T_a = 60\text{ }^{\circ}\text{C}$.

Допуск к эксплуатации C-Tick



Система автоматизации S7-1200 удовлетворяет требованиям стандартов AS/NZS CISPR16 (класс A).

Сертификация в Корее



Система автоматизации S7-1200 удовлетворяет требования корейской сертификации (KC ярлык). Она определена как оборудование класса А и предназначена для промышленного применения и не рассматривается для бытового применения.

Допуск Евразийского Таможенного Союза (Белоруссия, Казахстан, Российская Федерация)



ЕАС (Евразийское Соответствие): Заявление о соответствии техническим правилам Таможенного союза (CU TR).

Морской допуск к эксплуатации

Изделия S7-1200 периодически представляются специальным агентствам для одобрений, связанных с определенными рынками и приложениями. Проконсультируйтесь с местным представителем Сименс, если Вам нужна дополнительная информация, связанная с последним перечнем конкретных допусков по заказному номеру.

Сертификационные сообщества:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)
- Korean Register of Shipping

Промышленные среды

Система автоматизации S7-1200 разработана для использования в промышленной среде.

Таблица A- 1 Промышленные среды

Область применения	Требования к излучению	Требования к невосприимчивости	Требования к помехоустойчивости
Промышленность	EN 61000-6-4:2007+A1:2011	EN 61000-6-2:2005	EN 61000-6-2:2005

Примечание

Система автоматизации S7-1200 предназначена для использования в промышленных зонах; использование в жилых районах может оказать влияние на телевизионный или радио прием. Если Вы используете S7-1200 в жилых районах, Вы должны гарантировать, что уровень излучаемых системой радиопомех соответствует предельному значению Класса В в соответствии с EN 55011.

Примеры соответствующих мер для достижения уровня радиочастотных помех, Класса В:

- установка S7-1200 в заземленном шкафу управления;
- использование противопомеховых фильтров в линиях питания.

Гарантируйте, что излучение радиопомех соответствует Классу В в соответствии с EN 55011.

Требуется индивидуальная приемка (законченная установка должна удовлетворять всем требованиям по безопасности и ЭМС для установки в жилых помещениях).

Электромагнитная совместимость

Электромагнитная совместимость (ЭМС) -это способность электрического устройства работать по назначению в электромагнитной среде и работать, не излучая электромагнитные помехи (EMI), которые могут негативно повлиять на другие электрические устройства поблизости.

Таблица А- 2 Невосприимчивость согласно EN 61000-6-2

Электромагнитная совместимость - невосприимчивость согласно EN 61000-6-2	
EN 61000-4-2 Электростатический разряд	8 кВ разряд через воздух на всех поверхностях 6 кВ разряд через контакт с открытыми проводящими поверхностями
EN 61000-4-3 Тест на невосприимчивость к излученному, радиочастотному, электромагнитному полю	от 80 до 1000 МГц, 10 В/м, 80%-ая ампл.мод. при 1 кГц от 1-4 до 2,0 ГГц, 3 В/м, 80%-ая ампл.мод. при 1 кГц от 2,0 до 2,7 ГГц, 1 В/м, 80%-ая ампл.мод. при 1 кГц
EN 61000-4-4 Быстрые переходные импульсы	2 кВ, 5 кГц в цепи связи с системой питания пер. и пост. тока 2 кВ, 5 кГц на клемме входа или выхода
EN 61000-4-5 Устойчивость к динамическим изменениям напряжения питания	Системы пер.т. - 2 кВ синфазная / 1 кВ симметричная помеха Системы пост.т. - 2 кВ синфазная / 1 кВ симметричная помеха Для систем пост.т., обратитесь далее к Устойчивость к динамическим изменениям напряжения питания
EN 61000-4-6 Наводки по цепям питания	от 150 кГц до 80 МГц, 10 В _{эфф} , 80%-ая ампл.мод. при 1 кГц
EN 61000-4-11 Падения напряжения	Системы переменного тока 0% в течение 1 цикла, 40% в течение 12 циклов и 70% в течение 30 циклов при 60 Гц

Проводные системы, подвергающиеся скачкам от взаимодействия с разрядом молнии, должны быть оборудованы внешней защитой. Одна из спецификаций для оценки защиты от скачков типа молнии приведена в EN 61000-4-5 с эксплуатационными пределами, установленными EN 61000-6-2. ЦПУ и сигнальные модули S7-1200 пост.т. требуют, чтобы внешняя защита поддерживала бесперебойную работу при воздействии импульсных напряжений, определенных этим стандартом.

Ниже перечислены некоторые из устройств, которые поддерживают необходимую защиту от скачков. Эти устройства обеспечивают защиту только в том случае, если они должным образом установлены согласно рекомендациям производителя. Устройства, произведенные другими поставщиками с теми же или лучшими характеристиками, также могут использоваться:

Таблица А- 3 Устройства, поддерживающие защиту от скачков

Подсистема	Устройство защиты
Питание +24 В пост.т.	BLITZDUCTOR VT, BVT AVD 24, Номер по каталогу 918 422
Industrial Ethernet	DEHNpatch DPA M CLE RJ45B 48, Номер по каталогу 929 121
RS-485	BLITZDUCTOR XT, Базовый блок BXT BAS, Номер по каталогу 920 300
	BLITZDUCTOR XT, Модуль BXT ML2 BD HFS 5, Номер по каталогу 920 271
RS-232	BLITZDUCTOR XT, Базовый блок BXT BAS, Номер по каталогу 920 300
	BLITZDUCTOR XT, Модуль BXT ML2 BE S 12, Номер по каталогу 920 222
Цифровые входы+24 В пост.т.	DEHN, Inc., Тип DCO SD2 E 24, Номер по каталогу 917 988
Цифровые входы+24 В пост.т. и питание датчика	DEHN, Inc., Тип DCO SD2 E 24, Номер по каталогу 917 988

Подсистема	Устройство защиты
Аналоговый ввод-вывод	DEHN, Inc., Тип DCO SD2 E 12, Номер по каталогу 917 987
Релейные выходы	Не требуется

Таблица А- 4 Наведенные и излученные помехи согласно EN 61000-6-4

Электромагнитная совместимость - Наведенные и излученные помехи согласно EN 61000-6-4		
Наведенные помехи EN 55011, класс А, группа 1	от 0.15 МГц до 0.5 МГц	<79 дБ (мкВ) квазипиковое; <66 дБ (мкВ) среднее значение
	от 0.5 МГц до 5 МГц	<73 дБ (мкВ) квазипиковое; <60 дБ (мкВ) среднее значение
	от 5 МГц до 30 МГц	<73 дБ (мкВ) квазипиковое; <60 дБ (мкВ) среднее значение
Излученные помехи EN 55011, класс А, группа 1	от 30 МГц до 230 МГц	<40 дБ (мкВ/м) квазипиковое на 10 м
	от 230 МГц до 1 ГГц	<47 дБ (мкВ/м) квазипиковое на 10 м
	от 1 ГГц до 3 ГГц	<76 дБ (мкВ/м) квазипиковое на 10 м

Условия окружающей среды

Таблица А- 5 Транспортировка и хранение

Условия окружающей среды - Транспортировка и хранение	
EN 60068-2-2, тест Bb, сухое тепло и EN 60068-2-1, тест Ab, холод	от -40 °С до +70 °С
EN 60068-2-30, тест Db, влажное тепло	от 25 °С до 55 °С, влажность 95%
EN 60068-2-14, тест Na, температурный удар	от -40 °С до +70 °С, время выдержки 3 часа, 5 циклов
EN 60068-2-32, свободное падение	0,3 м, 5 раз, в упаковке для отправки
Атмосферное давление	от 1080 до 660 ГПа (соответствует высоте от -1000 до 3500 м)

Таблица А- 6 Условия эксплуатации

Условия окружающей среды - Эксплуатация	
Диапазон температур окружающей среды (забор воздуха 25 мм под устройством)	от -20 °С до 60 °С при горизонтальном монтаже от -20 °С до 50 °С при вертикальном монтаже Неконденсируемая влажность 95%. Если не определено иначе.
Атмосферное давление	от 1080 до 795 ГПа (соответствует высоте от -1000 до 2000 м)
Концентрация загрязнений	S02: < 0,5 %; H2S: < 0,1 %; относит. влажность < 60% без конденсации
	ISA-S71.04 уровень серьезности G1, G2, G3
EN 60068-2-14, тест Nb, изменение температуры	от 5 °С до 55 °С, 3 °С/мину

Условия окружающей среды - Эксплуатация	
EN 60068-2-27 Механический удар	15 G, импульс 11 мс, 6 ударов по каждой из 3 осей
EN 60068-2-6 Гармонические колебания	Монтаж на DIN-рейке: 3.5 мм в диапазоне 5-9 Гц, 1G в диапазоне 9 - 150 Гц. Монтаж на панели: 7.0 мм в диапазоне 5-9 Гц, 2G в диапазоне 9 - 150 Гц; 10 качаний по каждой оси, 1 октава в минуту

Таблица А- 7 Испытание изоляции высоким напряжением

Испытание изоляции высоким напряжением	
Цепи номиналом 24 В / 5 В пост.т.	520 В пост.т. (типовое испытание оптических границ изоляции)
Цепи 115 В / 230 В пер.т. относительно земли	1500 В пер.т.
Цепи 115 В / 230 В пер.т. относительно цепей 115 В / 230 В пер.т.	1500 В пер.т.
Цепи 115 В / 230 В пер.т. относительно цепей 24 В / 5 В пост.т.	1500 пер.т. (3000 пер.т./4242 пост.т. типовой тест)
Порт Ethernet относительно цепей 24 В / 5 В пост.т. и земли ¹	1500 пер.т. (только типовой тест)

¹ Изоляция порта Ethernet разработана таким образом, чтобы уменьшить опасность воздействия опасных напряжений во время краткосрочных сетевых отказов. Это не соответствует требованиям техники безопасности для стандартной изоляции напряжения линии переменного тока.

Класс защиты

- Класс защиты II согласно EN 61131-2 (Защитный проводник, не требуется)

Степень защиты

- IP20 Механическая защита, EN 60529
- Защита от прямого контакта с высоким напряжением, как например, при тестировании стандартным пробником. Необходима внешняя защита от пыли, грязи, воды и посторонних предметов диаметром < 12,5 мм.

Номинальные напряжения

Таблица А- 8 Номинальные напряжения

Номинальное напряжение	Допуск
24 В пост.т.	от 20.4 до 28.8 В пост.т.
120/230 В пер.т.	от 85 до 264 В пер.т., от 47 до 63 Гц

Защита от обратного напряжения

Схемой защиты от обратного напряжения оснащается каждая из пар выходных клемм питания +24 В пост.т. или внешнего питания для ЦПУ, сигнальных модулей (SM) и сигнальных плат (SB). Однако существует вероятность повреждения системы при соединении различных выходных клемм с противоположной полярностью.

Некоторые порты входного питания на 24 В пост.т. в системе S7-1200 внутренне подключены к общей логической схеме, соединяющей несколько клемм М. К примеру, следующие схемы внутренне соединены, когда определяются в спецификациях как "неизолированные": источник питания 24 В пост.т. для ЦПУ, питание датчика ЦПУ, вход питания для обмотки реле в SM, а также источник питания для неизолированного аналогового входа. Все неизолированные клеммы М, должны быть соединены с одним и тем же опорным потенциалом.



ПРЕДУПРЕЖДЕНИЕ

Соединение неизолированных клемм М с различными опорными потенциалами, вызовет нежелательные электрические токи, которые могут нанести ущерб или привести к непредсказуемой работе ПЛК и любого подключенного оборудования.

Отказ следовать настоящим инструкциям мог бы нанести ущерб или вызвать непредсказуемую работу, которая могла бы привести к смерти или тяжелым травмам и/или материальному ущербу.

При любых обстоятельствах удостоверьтесь, что все неизолированные клеммы М в системе S7-1200, соединены с одним и тем же опорным потенциалом.

Выходы постоянного тока

Выходы постоянного тока ЦПУ, сигнальных модулей (SM) и сигнальных плат (SB) не оснащены схемой защиты цепи от короткого замыкания.

Электрический ресурс реле

Типовые характеристики, сделанные на основании выборочных испытаний, приведены ниже. Фактические данные могут варьироваться в

зависимости от того или иного приложения. Внешняя цепь защиты, согласованная с нагрузкой, увеличивает срок службы контактов. У нормально-замкнутых контактов типовой ресурс составляет приблизительно одну треть от ресурса нормально-разомкнутых контактов при индуктивных нагрузках и коммутации ламп.

Внешняя защитная схема увеличивает ресурс контактов.

Таблица А- 9 Типовые характеристики

Данные для выбора исполнительного элемента				
Непрерывный тепловой ток		2 А макс.		
Коммутационная способность и ресурс контактов				
	Для резистивной нагрузки	Напряжение	Ток	Кол-во рабочих циклов (типовое)
		24 В пост.т.	2.0 А	0.1 миллиона
		24 В пост.т.	1.0 А	0.2 миллиона
		24 В пост.т.	0.5 А	1.0 миллион
		48 В пер.т.	1.5 А	1.5 миллиона
		60 В пер.т.	1.5 А	1.5 миллиона
		120 В пер.т.	2.0 А	1.0 миллион
		120 В пер.т.	1.0 А	1.5 миллиона
		120 В пер.т.	0.5 А	2.0 миллиона
		230 В пер.т.	2.0 А	1.0 миллион
		230 В пер.т.	1.0 А	1.5 миллиона
		230 В пер.т.	0.5 А	2.0 миллиона
			Для индуктивной нагрузки (согласно IEC 947-5-1 DC13/AC15)	Напряжение
24 В пост.т.	2.0 А			0.05 миллиона
24 В пост.т.	1.0 А			0.1 миллиона
24 В пост.т.	0.5 А			0.5 миллиона
24 В пер.т.	1.5 А			1.0 миллион
48 В пер.т.	1.5 А			1.0 миллион
60 В пер.т.	1.5 А			1.0 миллион
120 В пер.т.	2.0 А			0.7 миллиона
120 В пер.т.	1.0 А			1.0 миллион
120 В пер.т.	0.5 А			1.5 миллиона
230 В пер.т.	2.0 А			0.7 миллиона
230 В пер.т.	1.0 А			1.0 миллион
230 В пер.т.	0.5 А			1.5 миллиона
Активация цифрового входа		Возможна		
Частота коммутации				
	Механически	Макс. 10 Гц		
	При резистивной нагрузке	Макс. 1 Гц		
	При индуктивной нагрузке (согласно IEC 947-5-1 DC13/AC15)	Макс. 0.5 Гц		
	При нагрузке в виде ламп	Макс. 1 Гц		

Сохраняемость внутренней памяти ЦПУ

- Время удержания сохраняемых данных и данных из журналов: 10 лет
- Данные, сохраняемые при отключении питания; количество циклов записи: 2 миллиона
- Данные из журналов, до 2 Кбайт на запись; количество циклов записи: 500 миллионов записей в журнале данных

Примечание

Эффект журналов данных во внутренней памяти ЦПУ

Каждая запись в журнале данных использует как минимум 2 Кб памяти. Если Ваша программа пишет небольшие объемы данных, она использует по крайней мере 2 Кб памяти при каждой записи. Наилучшим решением должна было бы стать накопление небольших элементов данных в блоке данных (DB) и запись блока данных в журнал с более продолжительным интервалом.

Если Ваша программа вносит много записей в журнал данных с высокой частотой, рассмотрите вариант использования сменной SD карты памяти.

A.2. CPU 1211C

A.2.1. Общие технические данные и характеристики

Таблица А- 10 Общие технические данные

Технические данные			
Модель	CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/Relay	CPU 1211C DC/DC/DC
Номер для заказа	6ES7 211-1BE40-0XB0	6ES7 211-1HE40-0XB0	6ES7 211-1AE40-0XB0
Размеры Ш x В x Г (мм)	90 x 100 x 75		
Вес с упаковкой	420 грамм	380 грамм	370 грамм
Рассеиваемая мощность	10 Вт	8 Вт	
Доступный ток (шина CM)	макс. 750 мА (5 В пост.т.)		
Доступный ток (24 В пост т.)	макс. 300 мА (питание датчиков)		
Потребление тока цифровым входом (24 пост т.)	4 мА/используемый вход		

Таблица А- 11 Характеристики ЦПУ

Технические данные		Описание
Пользовательская память (Обратитесь к "Общим техническим данным" (стр. 1099), "Сохраняемость внутренней памяти ЦПУ".)	Рабочая	50 Кбайт
	Загрузочная	1 Мбайт, внутренняя, расширяемая до объема SD-карты
	Сохраняемая	10 Кбайт
Встроенные цифровые входы/выходы		6 входов/4 выхода
Встроенные аналоговые входы/выходы		2 входа
Величина образа процесса		1024 байта входов (I) /1024 байта выходов (Q)
Битовая память (M)		4096 байт
Временная (локальная) память		16 кБайт для запуска и программного цикла (включая вызываемые FB и FC) 6 кБайт для каждого из других уровней приоритета прерываний (включая вызываемые FB и FC)
Дополнительные сигнальные модули		Нет
Расширение с помощью SB, CB, BB		макс. 1 SB
Дополнительные коммуникационные модули		макс. 3 CM
Скоростные счетчики		До 6 сконфигурированных для использования входов (любых встроенных или на SB). Обратитесь к таблице, ЦПУ 1211C: Назначение адреса высоко-скоростного счетчика по умолчанию (стр. 463) 100/180 кГц (от Ia.0 до Ia.5)

A.2 CPU 1211C

Технические данные	Описание
Импульсные выходы ²	До 4 сконфигурированных для использования выходов (любых встроенных или на SB) 100 кГц (от Qa.0 до Qa.3)
Входы для улавливания импульсов	6
Прерывания с задержкой по времени	Всего 4 с разрешением 1 мс
Циклические прерывания	Всего 4 с разрешением 1 мс
Прерывания по фронту	6 по нарастающему и 6 по спадающему (10 и 10 с дополнительной сигнальной платой)
Карта памяти	Карта памяти SIMATIC (дополнительно)
Точность часов реального времени	+/- 60 секунд/месяц
Длительность сохранения времени для часов реального времени	Тип.знач. 20 дней /мин.знач. 12 дней при 40°C (не требующий обслуживания суперконденсатор)

¹ Более медленная скорость применима, когда высокоскоростной счетчик сконфигурирован для квадратурного режима работы.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

Таблица А- 12 Производительность

Тип инструкции	Скорость выполнения
Булевая	0,08 мкс на команду
Передача слова	1,7 мкс на команду
Арифметика над действительными числами	2,3 мкс на команду

A.2.2. Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1211C

Таблица А- 13 Блоки, таймеры и счетчики, поддерживаемые CPU 1211C

Элемент		Описание
Блоки	Тип	OB, FB, FC, DB
	Размер	30 кБайт
	Количество	До 1024 блоков всего (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложения	16 от OB программного цикла или запуска 6 от OB любого другого прерывания по событию
	Наблюдение	Состояние 2 блоков может наблюдаться одновременно
Организационные блоки OB	Программный цикл	Произвольное количество
	Стартовые	Произвольное количество
	Прерывания с задержкой	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке	1

	времени исполнения	
Элемент		Описание
Организационные блоки OB	Диагностические прерывания	1
	Снятие / установка модулей	1
	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета • SInt, USInt: 3 Байта • Int, UInt: 6 Байт • DInt, UDInt: 12 Байт

Таблица A- 14 Коммуникации

Технические характеристики	Описание
Количество портов	1
Тип	Ethernet
Устройство человеко-машинного интерфейса	4
Программатор (PG)	1
Соединения	• 8 для Открытых пользовательских коммуникаций (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 3 для серверных GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций • 8 для клиентских GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций
Скорость передачи данных	10/100 Мб/с
Электрическая развязка (внешний сигнал относительно логики ПЛК)	Изоляция с помощью трансформатора, 1500 В пер.т., только для краткосрочной безопасности
Тип кабеля	CAT5e экранированный

A.2 CPU 1211C

Таблица А- 15 Источник питания

Технические характеристики		CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/Relay	CPU 1211C DC/DC/DC
Диапазон напряжений		от 85 до 264 В пер.т.	от 20.4 до 28.8 В пост.т.	
Частота сети		от 47 до 63 Гц	--	
Входной ток	только CPU при макс. нагрузке	60 мА при 120 В пер.т. 30 мА при 240 В пер.т.	30 мА при 24 В пост.т.	300 мА при 24 В пост.т.
	CPU со всеми модулями расширения при макс. нагрузке	180 мА при 120 В пер.т. 90 мА при 240 В пер.т.	900 мА при 24 В пост.т.	
Бросок тока при включении (макс.)		20 А при 264 В пер.т.	12 А при 28.8 В пост.т.	
Электрическая развязка (входного питания относительно логики)		1500 В пер.т.	Не изолировано	
Ток утечки на землю, от линии переменного тока на функциональную землю		0.5 мА макс.	--	
Время удержания (при сбое питания)		20 мс при 120 В пер.т. 80 мс при 240 В пер.т.	10 мс при 24 В пост.т.	
Внутренний предохранитель, не заменяемый пользователем		3 А, 250 В, плавкий предохранитель		

Таблица А- 16 Питание датчиков

Технические характеристики	CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/Relay	CPU 1211C DC/DC/DC
Диапазон напряжений	от 20.4 до 28.8 В пост.т.	L+ минус 4 В пост.т. мин.	
Номинальный выходной ток (макс.)	300 mA (с защитой от короткого замыкания)		
Максимальная пульсирующая помеха (<10 МГц)	< 1 В между пиками	Как на входном проводе	
Электрическая развязка (логики CPU относительно питания датчиков)	Не изолировано		

А.2.3. Цифровые входы и выходы

Таблица А- 17 Цифровые входы

Технические характеристики	CPU 1211C AC/DC/Relay, CPU 1211C DC/DC/Relay и CPU 1211C DC/DC/DC
Количество входов	6
Тип	Потребитель/источник (МЭК тип 1 потребитель)
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение
Длительно допустимое напряжение	30 В пост.т., макс.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин
Потенциально развязанные группы	1
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.) (Уровень лог.1 = от 15 до 26 В пост.т.)	100/80 кГц (от Ia0 до Ia.5)
Количество одноврем.активных входов	6 при 60 °С для горизонтальной, 50 °С для вертикальной установки
Длина кабеля (в метрах)	500 экранированный, 300 неэкранированный, 50 экранированный для входов высокоскоростных счетчиков

Таблица А- 18 Цифровые выходы

Технические характеристики	CPU 1211C AC/DC/Relay и CPU 1211C DC/DC/Relay	CPU 1211C DC/DC/DC
Количество выходов	4	
Тип	релейный, механический	Твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	--	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	--	0.1 В пост.т. макс.
Ток (макс.)	2.0 А	0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.	5 Вт
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля	0.6 Ом макс.
Ток утечки на выход	--	10 мкА макс.
Ток включения	7 А с закрытыми контактами	8 А в течение 100 мс макс.
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в течение 1 мин (катушка относительно контакта) Не изолировано (катушка отн.логики)	500 В пер.т. в течение 1 мин
Сопротивление изоляции	100 МОм мин. для нового модуля	--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин	--
Потенциально развязанные группы	1	
Индуктивное фиксирующее напряжение	--	L+ минус 48 В пост.т., мощность рассеяния 1 Вт

A.2 CPU 1211C

Технические характеристики	CPU 1211C AC/DC/Relay и CPU 1211C DC/DC/Relay	CPU 1211C DC/DC/DC
Максимальная частота переключения	1 Гц	--
Задержка переключения (от Qa.0 до Qa.3)	10 мс макс.	1.0 мкс макс., из выкл. во вкл. 3.0 мкс макс., из вкл. в выкл.
Частота пачки импульсов	Не рекомендуется ¹	100 кГц (от Qa.0 до Qa.3) ² , 2 Гц мин.
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.	--
Срок службы контактов при номинальной нагрузке	100,000 циклов замык./размык.	--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	
Количество одноврем.активных выходов	4 при 60 °С для горизонтальной, 50 °С для вертикальной установки	
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный	

¹ Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

² В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

A.2.4. Аналоговые входы

Таблица A- 19 Аналоговые входы

Технические характеристики	Описание
Количество входов	2
Тип	напряжение (асимметричный)
Диапазон измерений	от 0 до 10 В
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 10.001 до 11.759 В
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	от 11.760 до 11.852 В
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное выдерживаемое напряжение	35 В пост.т.
Сглаживание	Отсутствует, слабое, среднее или сильное Смотри таблицу для реакции на скачок (мс) для аналоговых входов ЦПУ (стр. 1115).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °С / от 0 до 55 °С)	3.0% / 3.5% полного диапазона
Длина кабеля (метры)	100 м, экранированная витая пара

A.2.4.1. Реакция на скачок встроенных аналоговых входов ЦПУ

Таблица А- 20 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов): 16 значений	200 мс	240 мс	1150 мс
Сильное (32 циклов): 32 значения	400 мс	480 мс	2300 мс
Время выборки	4.17 мс	5 мс	25 мс

A.2.4.2. Время выборки для встроенных аналоговых входов ЦПУ

Таблица А- 21 Время выборки для встроенных аналоговых входов ЦПУ

Частота подавления (выбор времени интегрирования)	Время выборки
60 Гц (16.6 мс)	4.17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.2.4.3. Диапазоны измерений для аналоговых входов напряжения (ЦПУ)

Таблица А- 22 Представление аналогового входа для напряжения (ЦПУ)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11.851 В	Переполнение
32512	7F00		
32511	7EFF	11.759 В	Диапазон превышения
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.2.5. Схемы электрических соединений ЦПУ 1211

Таблица А- 23 CPU 1211C AC/DC/Relay (6ES7 211-1BE40-0XB0)

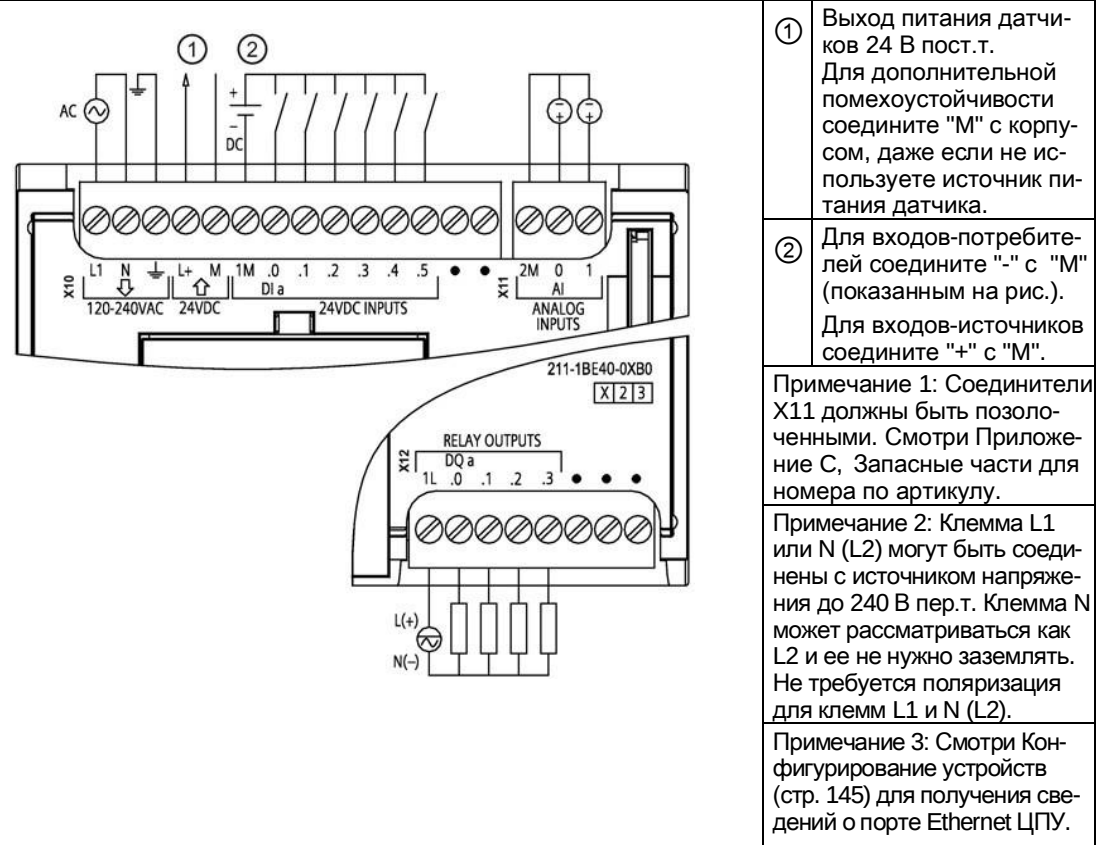


Таблица А- 24 Привязка контактов разъема CPU 1211C AC/DC/Relay (6ES7 211-1BE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L1 / 120-240 В пер.т.	2 М	1L
2	N / 120-240 В пер.т.	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	Не подключено
7	DI a.0	--	Не подключено
8	DI a.1	--	Не подключено
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--

Контакт	X10	X11 (с позолотой)	X12
13	Не подключено	--	--
14	Не подключено	--	--

Таблица A- 25 CPU 1211C DC/DC/Relay (6ES7 211-1HE40-0XB0)

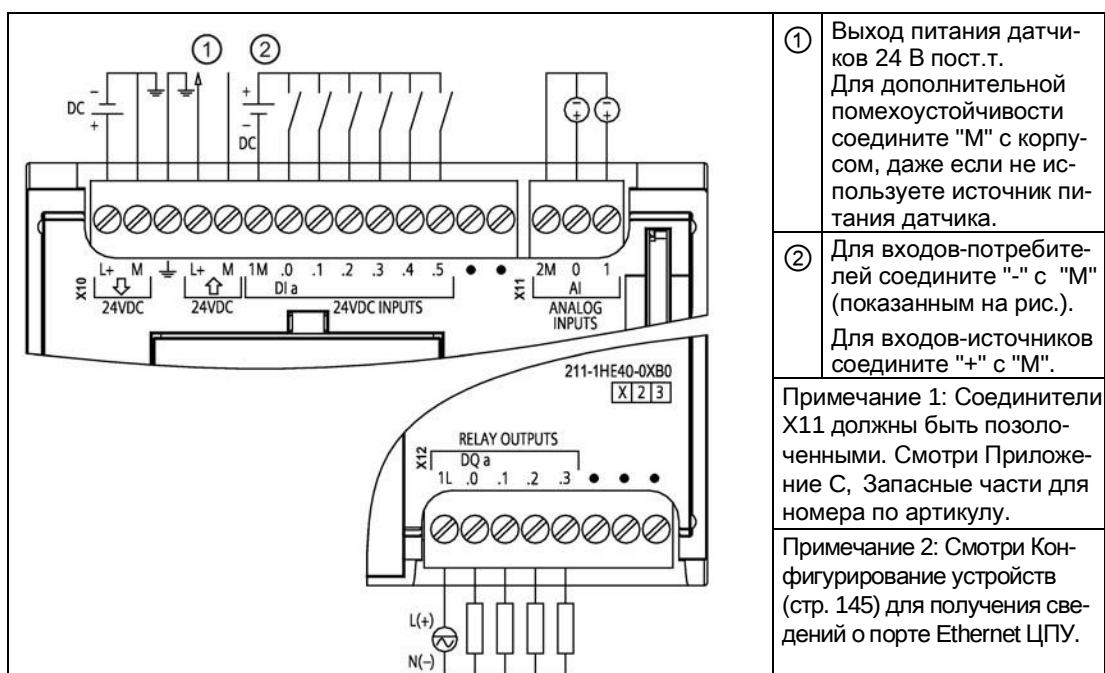


Таблица A- 26 Привязка контактов разъема CPU 1211C DC/DC/Relay (6ES7 211-1HE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	1L
2	M / 24 VDC	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	Не подключено
7	DI a.0	--	Не подключено
8	DI a.1	--	Не подключено
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Не подключено	--	--
14	Не подключено	--	--

Таблица А- 27 CPU 1211C DC/DC/DC (6ES7 211-1AE40-0XB0)

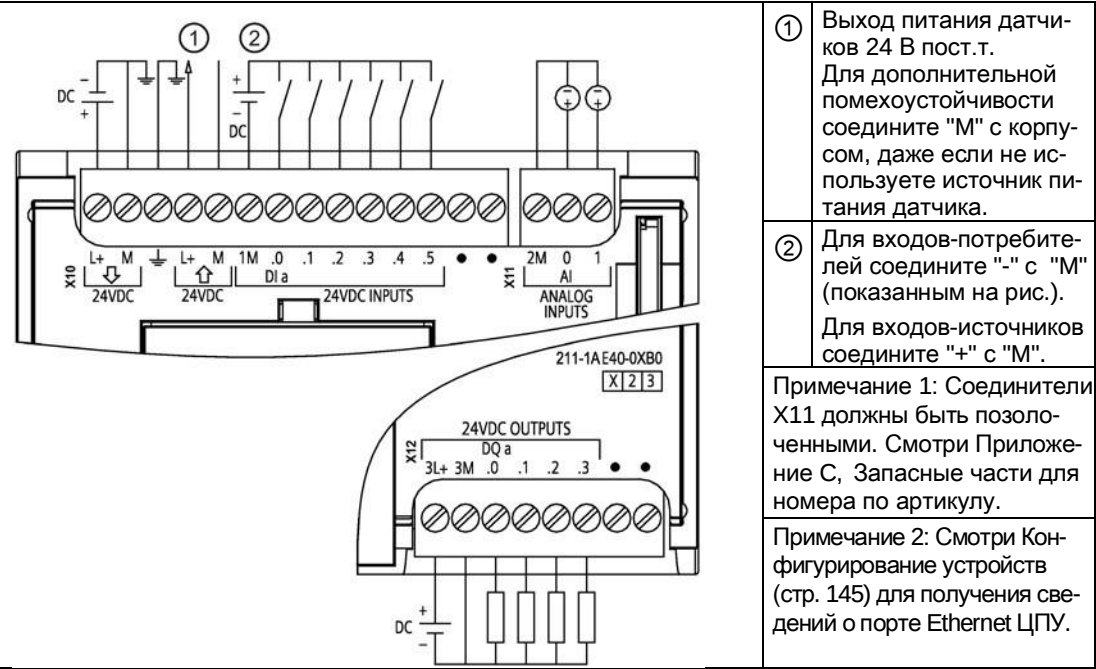


Таблица А- 28 Привязка контактов разъема CPU 1211C DC/DC/DC (6ES7 211-1AE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	3L+
2	M / 24 VDC	AI 0	3M
3	Функциональная земля	AI 1	DQ a.0
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.1
5	M / 24 В пост.т. питание датчиков	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	Нет соединения
8	DI a.1	--	Нет соединения
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Нет соединения	--	--
14	Нет соединения	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены

A.3. CPU 1212C

A.3.1. Общие технические данные и характеристики

Таблица А- 29 Общие технические данные

Технические данные			
Модель	CPU 1212C AC/DC/Relay	CPU 1212C DC/DC/Relay	CPU 1212C DC/DC/DC
Номер для заказа	6ES7 212-1BE40-0XB0	6ES7 212-1HE40-0XB0	6ES7 212-1AE40-0XB0
Размеры Ш x В x Г (мм)	90 x 100 x 75		
Вес с упаковкой	425 грамм	385 грамм	370 грамм
Рассеиваемая мощность	11 Вт	9 Вт	
Доступный ток (шина CM)	макс. 1000 мА (5 В пост.т.)		
Доступный ток (24 В пост т.)	макс. 300 мА (питание датчиков)		
Потребление тока цифровым входом (24 пост т.)	4 мА/используемый вход		

Таблица А- 30 Характеристики ЦПУ

Технические данные		Описание
Пользовательская память (Обратитесь к "Общим техническим данным" (стр. 1099), "Сохраняемость внутренней памяти ЦПУ".)	Рабочая	75 Кбайт
	Загрузочная	1 Мбайт, внутренняя, расширяемая до объема SD-карты
	Сохраняемая	10 Кбайт
Встроенные цифровые входы/выходы		8 входов/6 выходов
Встроенные аналоговые входы/выходы		2 входа
Величина образа процесса		1024 байта входов (I) /1024 байта выходов (Q)
Битовая память (M)		4096 байт
Временная (локальная) память		16 кБайт для запуска и программного цикла (включая вызываемые FB и FC) 6 кБайт для каждого из других уровней приоритета прерываний (включая вызываемые FB и FC)
Дополнительные сигнальные модули		макс.2 SM
Расширение с помощью SB, CB, BB		макс. 1
Дополнительные коммуникационные модули		макс. 3 CM
Скоростные счетчики		До 6 сконфигурированных для использования входов (любых встроенных или на SB). Обратитесь к таблице, ЦПУ 1212C: Назначение адреса высоко-скоростного счетчика по умолчанию (стр. 463) 100/180 кГц (от Ia.0 до Ia.5) 30/120 кГц (от Ia.6 до Ia.7)

A.3 CPU 1212C

Технические данные	Описание
Импульсные выходы ²	До 4 сконфигурированных для использования выходов (любых встроенных или на SB) 100 кГц (от Qa.0 до Qa.3) 30 кГц (от Qa.4 до Qa.5)
Входы для улавливания импульсов	8
Прерывания с задержкой по времени	Всего 4 с разрешением 1 мс
Циклические прерывания	Всего 4 с разрешением 1 мс
Прерывания по фронту	8 по нарастающему и 8 по спадающему (12 и 12 с дополнительной сигнальной платой)
Карта памяти	Карта памяти SIMATIC (дополнительно)
Точность часов реального времени	+/- 60 секунд/месяц
Длительность сохранения времени для часов реального времени	Тип.знач. 20 дней /мин.знач. 12 дней при 40°C (не требующий обслуживания суперконденсатор)

¹ Более медленная скорость применима, когда высокоскоростной счетчик сконфигурирован для квадратурного режима работы.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

Таблица А- 31 Производительность

Тип инструкции	Скорость выполнения
Булевая	0,08 мкс на команду
Передача слова	1,7 мкс на команду
Арифметика над действительными числами	2,3 мкс на команду

A.3.2. Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1212C

Таблица А- 32 Блоки, таймеры и счетчики, поддерживаемые CPU 1212C

Элемент	Описание
Блоки	Тип
	Размер
	Количество
	Диапазон адресов для FB, FC и DB
	Глубина вложения
	Наблюдение
Организационные блоки OB	Программный цикл
	Стартовые
	Прерывания с задержкой
	Циклические прерывания
	Аппаратные прерывания
	Прерывания по ошибке времени исполнения
	Диагностические прерывания
	Снятие / установка модулей

Элемент		Описание
Организационные блоки OB	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета - SInt, USInt: 3 Байта - Int, UInt: 6 Байт - DInt, UDInt: 12 Байт

Таблица А- 33 Коммуникации

Технические характеристики	Описание
Количество портов	1
Тип	Ethernet
Устройство человеко-машинного интерфейса	4
Программатор (PG)	1
Соединения	8 для Открытых пользовательских коммуникаций (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV 3 для серверных GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций 8 для клиентских GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций
Скорость передачи данных	10/100 Мб/с
Электрическая развязка (внешний сигнал относительно логики ПЛК)	Изоляция с помощью трансформатора, 1500 В пер.т., только для краткосрочной безопасности
Тип кабеля	CAT5e экранированный

Таблица А- 34 Источник питания

Технические характеристики		CPU 1212C AC/DC/Relay	CPU 1212C DC/DC/Relay	CPU 1212C DC/DC/DC
Диапазон напряжений		от 85 до 264 В пер.т.	от 20.4 до 28.8 В пост.т.	
Частота сети		от 47 до 63 Гц	--	
Входной ток (макс. нагрузка)	только CPU	80 мА при 120 В пер.т. 40 мА при 240 В пер.т.	400 мА при 24 В пост.т.	
	CPU со всеми модулями расшир. при макс. нагрузке	240 мА при 120 В пер.т. 120 мА при 240 В пер.т.	1200 мА при 24 В пост.т.	
Бросок тока при включении (макс.)		20 А при 264 В пер.т.	12 А при 28.8 В пост.т.	
Электрическая развязка (входного питания относительно логики)		1500 В пер.т.	Не изолировано	
Ток утечки на землю, от линии переменного тока на функциональную землю		0.5 мА макс.	--	

A.3 CPU 1212C

Время удержания (при сбое питания)	20 мс при 120 В пер.т. 80 мс при 240 В пер.т.	10 мс при 24 В пост.т.
Внутренний предохранитель, не заменяемый пользователем	3 А, 250 В, плавкий предохранитель	

Таблица А- 35 Питание датчиков

Технические характеристики	CPU 1212C AC/DC/Relay	CPU 1212C DC/DC/Relay	CPU 1212C DC/DC/DC
Диапазон напряжений	от 20.4 до 28.8 В пост.т.	L+ минус 4 В пост.т. мин.	
Номинальный выходной ток (макс.)	300 mA (с защитой от короткого замыкания)		
Максимальная пульсирующая помеха (<10 МГц)	< 1 В между пиками	Как на входном проводе	
Электрическая развязка (логики CPU относительно питания датчиков)	Не изолировано		

A.3.3. Цифровые входы и выходы

Таблица А- 36 Цифровые входы

Технические характеристики	CPU 1212C AC/DC/Relay, DC/DC/Relay и DC/DC/DC
Количество входов	8
Тип	Потребитель/источник (МЭК тип 1 потребитель)
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение
Длительно допустимое напряжение	30 В пост.т., макс.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин
Потенциально развязанные группы	1
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.) (Уровень лог.1 = от 15 до 26 В пост.т.)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ia.7)
Количество одноврем.активных входов	4 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 8 при 55 °С для горизонтальной или 45 °С для вертикальной установки
Длина кабеля (в метрах)	500 экранированный, 300 неэкранированный, 50 экранированный для входов высокоскоростных счетчиков

Таблица A- 37 Цифровые выходы

Технические характеристики	CPU 1212C AC/DC/Relay и DC/DC/Relay	CPU 1212C DC/DC/DC
Количество выходов	6	
Тип	релейный, механический	Твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	--	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	--	0.1 В пост.т. макс.
Ток (макс.)	2.0 А	0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.	5 Вт
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля	0.6 Ом макс.
Ток утечки на выход	--	10 мкА макс.
Ток включения	7 А с закрытыми контактами	8 А в течение 100 мс макс.
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в течение 1 мин (катушка относительно контакта) Не изолировано (катушка отн. логики)	500 В пер.т. в течение 1 мин
Сопротивление изоляции	100 МОм мин. для нового модуля	--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин	--
Потенциально развязанные группы	2	1
Индуктивное фиксирующее напряжение	--	L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения (от Qa.0 до Qa.3)	10 мс макс.	1.0 мкс макс., из выкл. во вкл. 3.0 мкс макс., из вкл. в выкл.
Задержка переключения (от Qa.4 до Qa.5)	10 мс макс.	50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключения реле	1 Гц	--
Частота пачки импульсов	Не рекомендуется ¹	100 кГц (от Qa.0 до Qa.3) ² , 2 Гц мин. 20 кГц (от Qa.4 до Qa.5) ²
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.	--
Срок службы контактов при номинальной нагрузке	100,000 циклов замык./размык.	--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	3 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 6 при 55 °С для горизонтальной или 45 °С для вертикальной установки	
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный	

¹ Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

² В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

A.3.4. Аналоговые входы

Таблица А- 38 Аналоговые входы

Технические характеристики	Описание
Количество входов	2
Тип	напряжение (асимметричный)
Диапазон измерений	от 0 до 10 В
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 10.001 до 11.759 В
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	от 11.760 до 11.852 В
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное выдерживаемое напряжение	35 В пост.т.
Сглаживание	Отсутствует, слабое, среднее или сильное Смотри таблицу для реакции на скачок (мс) для аналоговых входов ЦПУ (стр. 1124).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от 0 до 55 °C)	3.0% / 3.5% полного диапазона
Длина кабеля (метры)	100 м, экранированная витая пара

A.3.4.1. Реакция на скачок встроенных аналоговых входов ЦПУ

Таблица А- 39 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов): 16 значения	200 мс	240 мс	1150 мс
Сильное (32 циклов): 32 значения	400 мс	480 мс	2300 мс
Время выборки	4.17 мс	5 мс	25 мс

A.3.4.2. Время выборки для встроенных аналоговых входов ЦПУ

Таблица А- 40 Время выборки для встроенных аналоговых входов ЦПУ

Частота подавления (выбор времени интегрирования)	Время выборки
60 Гц (16.6 мс)	4.17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.3.4.3. Диапазоны измерений для аналоговых входов напряжения (ЦПУ)

Таблица А- 41 Представление аналогового входа для напряжения (ЦПУ)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11.851 В	Переполнение
32512	7F00		
32511	7EFF	11.759 В	Диапазон превышения
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

А.3.5. Схемы электрических соединений ЦПУ 1212

Таблица А- 42 CPU 1212C AC/DC/Relay (6ES7 212-1BE40-0XB0)

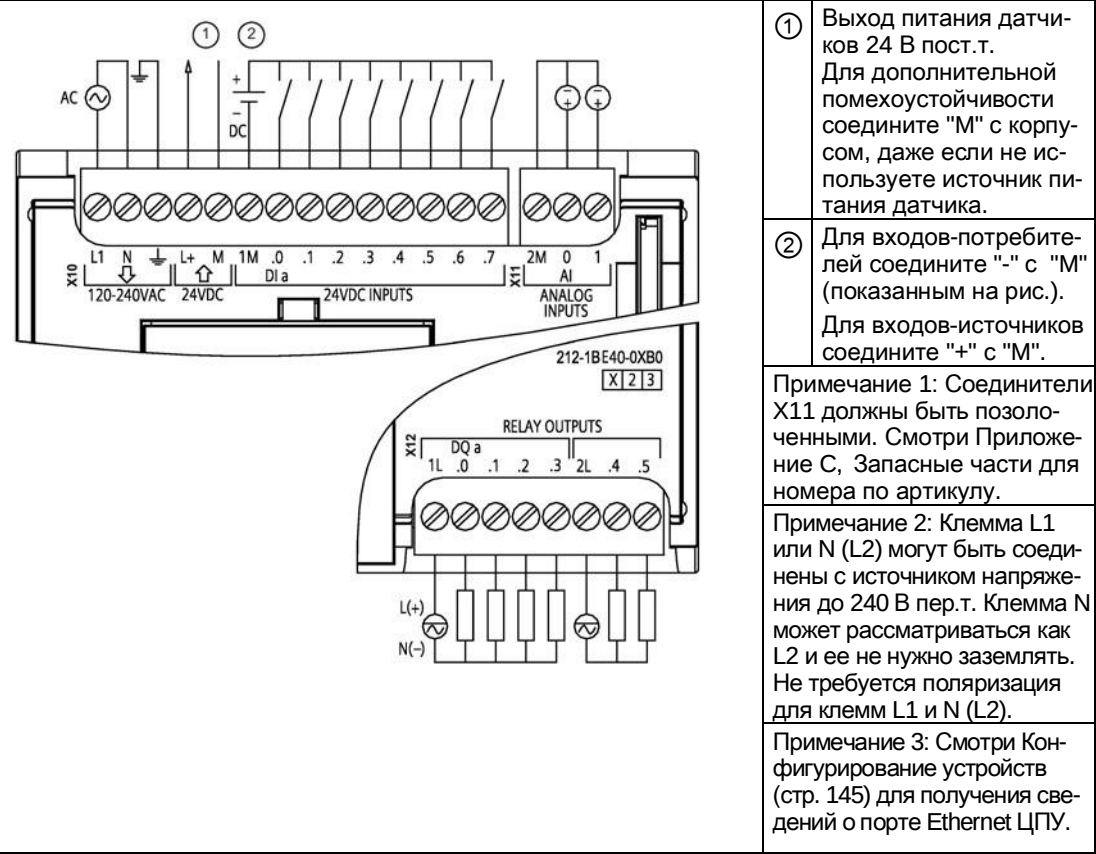


Таблица А- 43 Привязка контактов разъема CPU 1212C AC/DC/Relay (6ES7 212-1BE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L1 / 120-240 В пер.т.	2 М	1L
2	N / 120-240 В пер.т.	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--

Контакт	X10	X11 (с позолотой)	X12
13	DI a.6	--	--
14	DI a.7	--	--

Таблица А- 44 CPU 1212C DC/DC/Relay (6ES7 212-1HE40-0XB0)

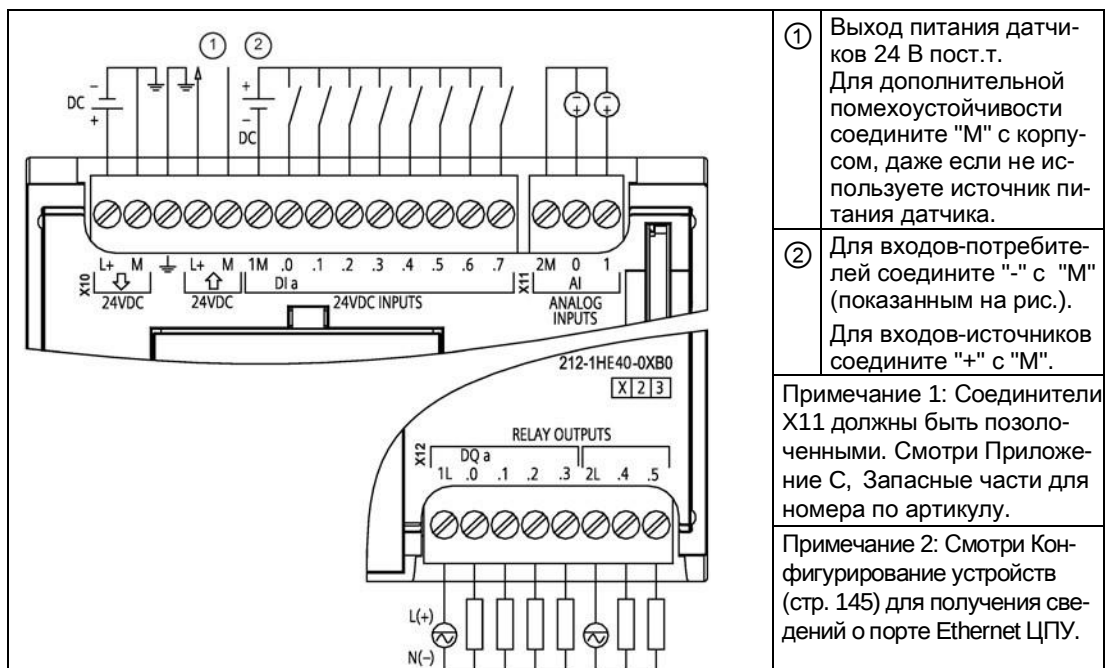


Таблица А- 45 Привязка контактов разъема CPU 1212C DC/DC/Relay (6ES7 212-1HE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	1L
2	M / 24 VDC	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Таблица А- 46 CPU 1212C DC/DC/DC (6ES7-212-1AE40-0XB0)

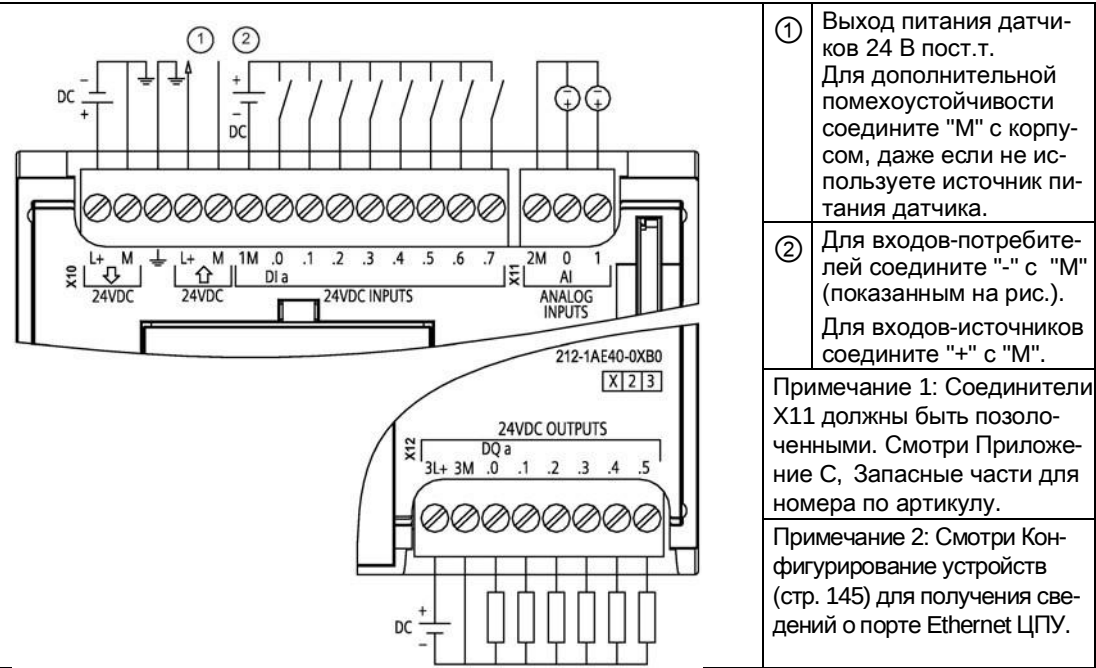


Таблица А- 47 Привязка контактов разъема CPU 1212C DC/DC/DC (6ES7 212-1AE40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	3L+
2	M / 24 VDC	AI 0	3M
3	Функциональная земля	AI 1	DQ a.0
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.1
5	M / 24 В пост.т. питание датчиков	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены

A.4. CPU 1214C

A.4.1. Общие технические данные и характеристики

Таблица А- 48 Общие технические данные

Технические данные			
Модель	CPU 1214C AC/DC/Relay	CPU 1214C DC/DC/Relay	CPU 1214C DC/DC/DC
Номер для заказа	6ES7 214-1BG40-0XB0	6ES7 214-1HG40-0XB0	6ES7 214-1AG40-0XB0
Размеры Ш x В x Г (мм)	110 x 100 x 75		
Вес с упаковкой	475 грамм	435 грамм	415 грамм
Рассеиваемая мощность	14 Вт	12 Вт	
Доступный ток (шина CM)	макс. 1600 мА (5 В пост.т.)		
Доступный ток (24 В пост т.)	макс. 400 мА (питание датчиков)		
Потребление тока цифровым входом (24 пост т.)	4 мА/используемый вход		

Таблица А- 49 Характеристики ЦПУ

Технические данные		Описание
Пользовательская память (Обратитесь к "Общим техническим данным" (стр. 1099), "Сохраняемость внутренней памяти ЦПУ".)	Рабочая	100 Кбайт
	Загрузочная	4 Мбайт, внутренняя, расширяемая до объема SD-карты
	Сохраняемая	10 Кбайт
Встроенные цифровые входы/выходы		14 входов/10 выходов
Встроенные аналоговые входы/выходы		2 входа
Величина образа процесса		1024 байта входов (I) /1024 байта выходов (Q)
Битовая память (M)		8192 байт
Временная (локальная) память		16 кБайт для запуска и программного цикла (включая вызываемые FB и FC) 6 кБайт для каждого из других уровней приоритета прерываний (включая вызываемые FB и FC)
Дополнительные сигнальные модули		макс.8 SM
Расширение с помощью SB, CB, BB		макс. 1
Дополнительные коммуникационные модули		макс. 3 CM
Скоростные счетчики		До 6 сконфигурированных для использования входов (любых встроенных или на SB). Обратитесь к таблице, ЦПУ 1212C: Назначение адреса высоко-скоростного счетчика по умолчанию (стр. 463) 100/180 кГц (от Ia.0 до Ia.5) 30/120 кГц (от Ib.6 до Ib.5)

A.4 CPU 1214C

Технические данные	Описание
Импульсные выходы ²	До 4 сконфигурированных для использования выходов (любых встроенных или на SB) • 100 кГц (от Qa.0 до Qa.3) • 30 кГц (от Qa.4 до Qa.5)
Входы для улавливания импульсов	14
Прерывания с задержкой по времени	Всего 4 с разрешением 1 мс
Циклические прерывания	Всего 4 с разрешением 1 мс
Прерывания по фронту	12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти	Карта памяти SIMATIC (дополнительно)
Точность часов реального времени	+/- 60 секунд/месяц
Длительность сохранения времени для часов реального времени	Тип.знач. 20 дней /мин.знач. 12 дней при 40°C (не требующий обслуживания суперконденсатор)

¹ Более медленная скорость применима, когда высокоскоростной счетчик сконфигурирован для квадратурного режима работы.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

Таблица A- 50 Производительность

Тип инструкции	Скорость выполнения
Булевая	0,08 мкс на команду
Передача слова	1,7 мкс на команду
Арифметика над действительными числами	2,3 мкс на команду

A.4.2. Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1214C

Таблица A- 32 Блоки, таймеры и счетчики, поддерживаемые CPU 1214C

Элемент	Описание
Блоки	Тип
	Размер
	Количество
	Диапазон адресов для FB, FC и DB
	Глубина вложения
	Наблюдение
Организационные блоки OB	Программный цикл
	Стартовые
	Прерывания с задержкой
	Циклические прерывания
	Аппаратные прерывания
	Прерывания по ошибке времени исполнения
	Диагностические прерывания
	Снятие / установка модулей

Элемент	Описание	
Организационные блоки OB	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета • SInt, USInt: 3 Байта • Int, UInt: 6 Байт • DInt, UDInt: 12 Байт

Таблица A- 52 Коммуникации

Технические характеристики	Описание
Количество портов	1
Тип	Ethernet
Устройство человеко-машинного интерфейса	4
Программатор (PG)	1
Соединения	• 8 для Открытых пользовательских коммуникаций (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 3 для серверных GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций • 8 для клиентских GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций
Скорость передачи данных	10/100 Мб/с
Электрическая развязка (внешний сигнал относительно логики ПЛК)	Изоляция с помощью трансформатора, 1500 В пер.т., только для краткосрочной безопасности
Тип кабеля	CAT5e экранированный

A.4 CPU 1214C

Таблица А- 53 Источник питания

Технические характеристики		CPU 1214C AC/DC/Relay	CPU 1214C DC/DC/Relay	CPU 1214C DC/DC/DC
Диапазон напряжений		от 85 до 264 В пер.т.	от 20.4 до 28.8 В пост.т.	
Частота сети		от 47 до 63 Гц	--	
Входной ток (макс. нагрузка)	только CPU	100 мА при 120 В пер.т. 50 мА при 240 В пер.т.	500 мА при 24 В пост.т.	
	CPU со всеми модулями расшир. при макс. нагрузке	300 мА при 120 В пер.т. 150 мА при 240 В пер.т.	1500 мА при 24 В пост.т.	
Бросок тока при включении (макс.)		20 А при 264 В пер.т.	12 А при 28.8 В пост.т.	
Электрическая развязка (входного питания относительно логики)		1500 В пер.т.	Не изолировано	
Ток утечки на землю, от линии перемен- ного тока на функциональную землю		0.5 мА макс.	--	
Время удержания (при сбое питания)		20 мс при 120 В пер.т. 80 мс при 240 В пер.т.	10 мс при 24 В пост.т.	
Внутренний предохранитель, не заме- няемый пользователем		3 А, 250 В, плавкий предохранитель		

Таблица А- 54 Питание датчиков

Технические характеристики	CPU 1214C AC/DC/Relay	CPU 1214C DC/DC/Relay	CPU 1214C DC/DC/DC
Диапазон напряжений	от 20.4 до 28.8 В пост.т.	L+ минус 4 В пост.т. мин.	
Номинальный выходной ток (макс.)	400 mA (с защитой от короткого замыкания)		
Максимальная пульсирующая помеха (<10 МГц)	< 1 В между пиками	Как на входном проводе	
Электрическая развязка (логики CPU относительно питания датчиков)	Не изолировано		

А.4.3. Цифровые входы и выходы

Таблица А- 55 Цифровые входы

Технические характеристики	CPU 1214C AC/DC/Relay	CPU 1214C DC/DC/Relay	CPU 1214C DC/DC/DC
Количество входов	14		
Тип	Потребитель/источник (МЭК тип 1 потребитель)		
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение		
Длительно допустимое напряжение	30 В пост.т., макс.		
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с		
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА		
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА		
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин		
Потенциально развязанные группы	1		
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0		
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.) (Уровень лог.1 = от 15 до 26 В пост.т.)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.5)		
Количество одноврем.активных входов	7 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 14 при 55 °С для горизонтальной или 45 °С для вертикальной установки		
Длина кабеля (в метрах)	500 экранированный, 300 неэкранированный, 50 экранированный для входов высокоскоростных счетчиков		

Таблица А- 56 Цифровые выходы

Технические характеристики	CPU 1214C AC/DC/Relay и DC/DC/Relay	CPU 1214C DC/DC/DC
Количество выходов	10	
Тип	релейный, механический	Твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	--	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	--	0.1 В пост.т. макс.
Ток (макс.)	2.0 А	0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.	5 Вт
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля	0.6 Ом макс.
Ток утечки на выход	--	10 мкА макс.
Ток включения	7 А с закрытыми контактами	8 А в течение 100 мс макс.
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в течение 1 мин (катушка относительно контакта) Не изолировано (катушка отн.логики)	500 В пер.т. в течение 1 мин
Сопротивление изоляции	100 МОм мин. для нового модуля	--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин	--
Потенциально развязанные группы	2	1

A.4 CPU 1214C

Технические характеристики	CPU 1214C AC/DC/Relay и DC/DC/Relay	CPU 1214C DC/DC/DC
Электрическая развязка (между группами)	1500 В пер.т. ¹	--
Индуктивное фиксирующее напряжение	--	L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения (от Qa.0 до Qa.3)	10 мс макс.	1.0 мкс макс., из выкл. во вкл. 3.0 мкс макс., из вкл. в выкл.
Задержка переключения (от Qa.4 до Qa.5)	10 мс макс.	50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключения реле	1 Гц	--
Частота пачки импульсов	Не рекомендуется ²	100 кГц (от Qa.0 до Qa.3) ³ , 2 Гц мин. 20 кГц (от Qa.4 до Qb.1) ³
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.	--
Срок службы контактов при номинальной нагрузке	100,000 циклов замык./размык.	--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	5 (без смежных точек) при 60 °C для горизонтальной или 50 °C для вертикальной установки 10 при 55 °C для горизонтальной или 45 °C для вертикальной установки	
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный	

¹ Электрическая развязка релейной группы по отношению к другим группам разделяет линейное напряжение от SELVPELV и разделяет различные фазы до 250 В пер.т. относительно земли.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

³ В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

A.4.4. Аналоговые входы

Таблица A- 57 Аналоговые входы

Технические характеристики	Описание
Количество входов	2
Тип	напряжение (асимметричный)
Диапазон измерений	от 0 до 10 В
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 10.001 до 11.759 В
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	от 11.760 до 11.852 В
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное выдерживаемое напряжение	35 В пост.т.
Сглаживание	Отсутствует, слабое, среднее или сильное Смотри таблицу для реакции на скачок (мс) для аналоговых входов ЦПУ (стр. 1135).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от 0 до 55 °C)	3.0% / 3.5% полного диапазона
Длина кабеля (метры)	100 м, экранированная витая пара

A.4.4.1. Реакция на скачок встроенных аналоговых входов ЦПУ

Таблица A- 58 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов): 16 значений	200 мс	240 мс	1150 мс
Сильное (32 циклов): 32 значения	400 мс	480 мс	2300 мс
Время выборки	4.17 мс	5 мс	25 мс

A.4.4.2. Время выборки для встроенных аналоговых входов ЦПУ

Таблица А- 59 Время выборки для встроенных аналоговых входов ЦПУ

Частота подавления (выбор времени интегрирования)	Время выборки
60 Гц (16.6 мс)	4.17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.4.4.3. Диапазоны измерений для аналоговых входов напряжения (ЦПУ)

Таблица А- 60 Представление аналогового входа для напряжения (ЦПУ)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11.851 В	Переполнение
32512	7F00		
32511	7EFF	11.759 В	Диапазон превышения
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

А.4.5. Схемы электрических соединений ЦПУ 1214

Таблица А- 61 CPU 1214C AC/DC/Relay (6ES7 214-1BG40-0XB0)

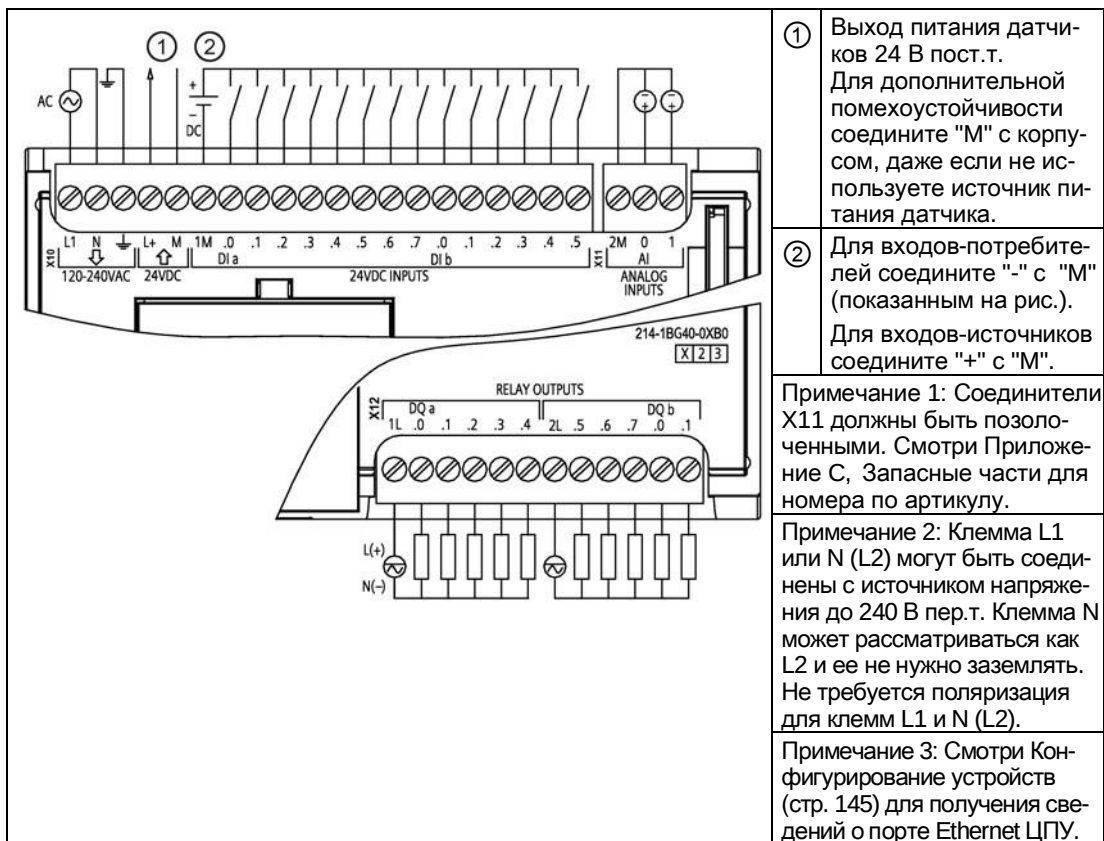


Таблица А- 62 Привязка контактов разъема CPU 1214C AC/DC/Relay (6ES7 214-1BG40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L1 / 120-240 В пер.т.	2 M	1L
2	N / 120-240 В пер.т.	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.5	--	--

Контакт	X10	X11 (с позолотой)	X12
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица А- 63 CPU 1214C DC/DC/Relay (6ES7 214-1HG40-0XB0)

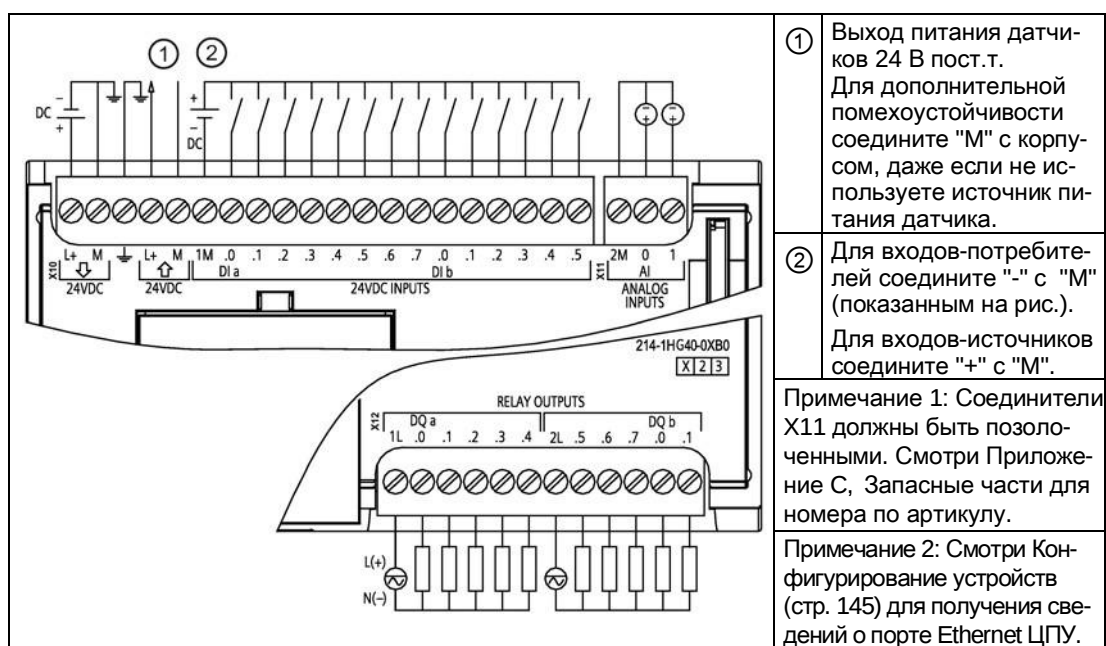


Таблица А- 64 Привязка контактов разъема CPU 1214C DC/DC/Relay (6ES7 214-1HG40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	1L
2	M / 24 VDC	AI 0	DQ a.0
3	Функциональная земля	AI 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.2
5	M / 24 В пост.т. питание датчиков	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
Контакт	X10	X11 (с позолотой)	X12

12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица А- 65 CPU 1214C DC/DC/DC (6ES7-214-1AG40-0XB0)

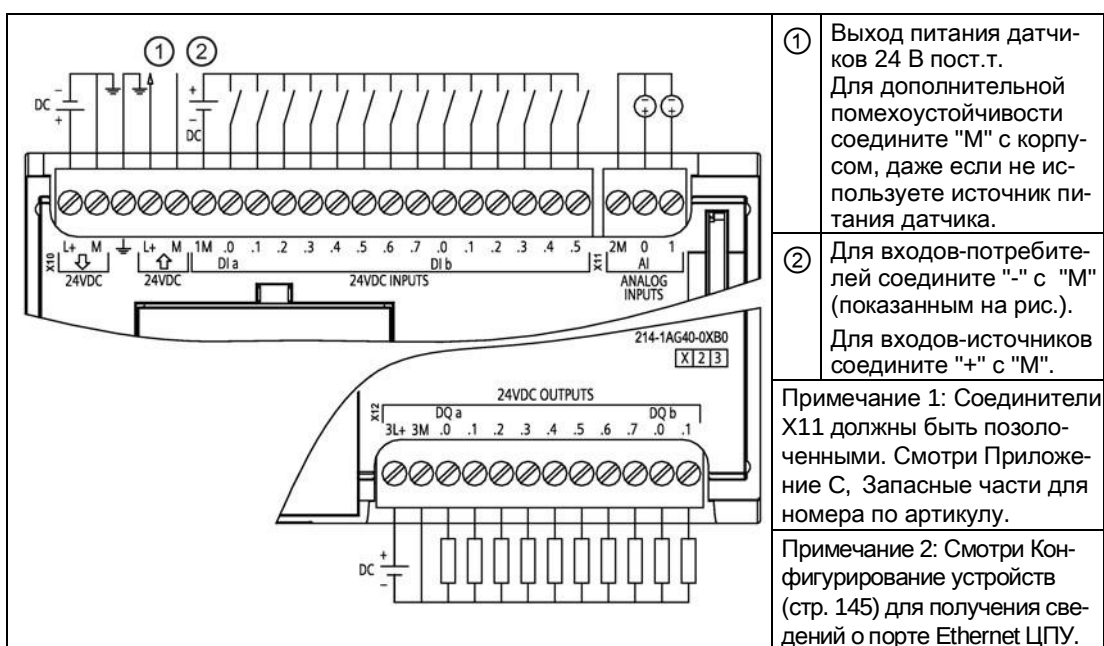


Таблица А- 66 Привязка контактов разъема CPU 1214C DC/DC/DC (6ES7-214-1AG40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	3L+
2	M / 24 VDC	AI 0	3M
3	Функциональная земля	AI 1	DQ a.0
4	L+ / 24 В пост.т. питание датчиков	--	DQ a.1
5	M / 24 В пост.т. питание датчиков	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
Контакт	X10	X11 (с позолотой)	X12

10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены

A.5. CPU 1215C

A.5.1. Общие технические данные и характеристики

Таблица А- 67 Общие технические данные

Технические данные			
Модель	CPU 1215C AC/DC/Relay	CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Номер для заказа	6ES7 215-1BG40-0XB0	6ES7 215-1HG40-0XB0	6ES7 215-1AG40-0XB0
Размеры Ш x В x Г (мм)	130 x 100 x 75		
Вес с упаковкой	585 грамм	550 грамм	520 грамм
Рассеиваемая мощность	14 Вт	12 Вт	
Доступный ток (шина CM)	макс. 1600 мА (5 В пост.т.)		
Доступный ток (24 В пост т.)	макс. 400 мА (питание датчиков)		
Потребление тока цифровым входом (24 пост т.)	4 мА/используемый вход		

Таблица А- 68 Характеристики ЦПУ

Технические данные		Описание
Пользовательская память (Обратитесь к "Общим техническим данным" (стр. 1099), "Сохраняемость внутренней памяти ЦПУ".)	Рабочая	125 Кбайт
	Загрузочная	4 Мбайт, внутренняя, расширяемая до объема SD-карты
	Сохраняемая	10 Кбайт
Встроенные цифровые входы/выходы		14 входов/10 выходов
Встроенные аналоговые входы/выходы		2 входа / 2 выхода
Величина образа процесса		1024 байта входов (I) /1024 байта выходов (Q)
Битовая память (M)		8192 байт
Временная (локальная) память		16 кБайт для запуска и программного цикла (включая вызываемые FB и FC) 6 кБайт для каждого из других уровней приоритета прерываний (включая вызываемые FB и FC)
Дополнительные сигнальные модули		макс.8 SM
Расширение с помощью SB, CB, BB		макс. 1
Дополнительные коммуникационные модули		макс. 3 CM
Скоростные счетчики		До 6 сконфигурированных для использования входов (любых встроенных или на SB). Обратитесь к таблице, ЦПУ 1215C: Назначение адреса высоко-скоростного счетчика по умолчанию (стр. 463) 100/180 кГц (от Ia.0 до Ia.5) 30/120 кГц (от Ib.6 до Ib.5)

A.5 CPU 1215C

Технические данные	Описание
Импульсные выходы ²	До 4 сконфигурированных для использования выходов (любых встроенных или на SB) • 100 кГц (от Qa.0 до Qa.3) • 30 кГц (от Qa.4 до Qa.5)
Входы для улавливания импульсов	14
Прерывания с задержкой по времени	Всего 4 с разрешением 1 мс
Циклические прерывания	Всего 4 с разрешением 1 мс
Прерывания по фронту	12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти	Карта памяти SIMATIC (дополнительно)
Точность часов реального времени	+/- 60 секунд/месяц
Длительность сохранения времени для часов реального времени	Тип.знач. 20 дней /мин.знач. 12 дней при 40°C (не требующий обслуживания суперконденсатор)

¹ Более медленная скорость применима, когда высокоскоростной счетчик сконфигурирован для квадратурного режима работы.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

Таблица А- 69 Производительность

Тип инструкции	Скорость выполнения
Булевая	0,08 мкс на команду
Передача слова	1,7 мкс на команду
Арифметика над действительными числами	2,3 мкс на команду

A.5.2. Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1215C

Таблица А- 70 Блоки, таймеры и счетчики, поддерживаемые CPU 1215C

Элемент	Описание
Блоки	Тип
	Размер
	Количество
	Диапазон адресов для FB, FC и DB
	Глубина вложения
	Наблюдение
Организационные блоки OB	Программный цикл
	Стартовые
	Прерывания с задержкой
	Циклические прерывания
	Аппаратные прерывания
	Прерывания по ошибке времени исполнения
	Диагностические прерывания
	Снятие / установка модулей

Элемент		Описание
Организационные блоки OB	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета • SInt, USInt: 3 Байта • Int, UInt: 6 Байт • DInt, UDInt: 12 Байт

Таблица А- 71 Коммуникации

Технические характеристики	Описание
Количество портов	2
Тип	Ethernet
Устройство человеко-машинного интерфейса	4
Программатор (PG)	1
Соединения	• 8 для Открытых пользовательских коммуникаций (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 3 для серверных GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций • 8 для клиентских GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций
Скорость передачи данных	10/100 Мб/с
Электрическая развязка (внешний сигнал относительно логики ПЛК)	Изоляция с помощью трансформатора, 1500 В пер.т., только для краткосрочной безопасности
Тип кабеля	CAT5e экранированный

A.5 CPU 1215C

Таблица А- 72 Источник питания

Технические характеристики		CPU 1215C AC/DC/Relay	CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Диапазон напряжений		от 85 до 264 В пер.т.	от 20.4 до 28.8 В пост.т.	
Частота сети		от 47 до 63 Гц	--	
Входной ток (макс. нагрузка)	только CPU	100 мА при 120 В пер.т. 50 мА при 240 В пер.т.	500 мА при 24 В пост.т.	
	CPU со всеми модулями расшир. при макс. нагрузке	300 мА при 120 В пер.т. 150 мА при 240 В пер.т.	1500 мА при 24 В пост.т.	
Бросок тока при включении (макс.)		20 А при 264 В пер.т.	12 А при 28.8 В пост.т.	
Электрическая развязка (входного питания относительно логики)		1500 В пер.т.	Не изолировано	
Ток утечки на землю, от линии перемен- ного тока на функциональную землю		0.5 мА макс.	--	
Время удержания (при сбое питания)		20 мс при 120 В пер.т. 80 мс при 240 В пер.т.	10 мс при 24 В пост.т.	
Внутренний предохранитель, не заме- няемый пользователем		3 А, 250 В, плавкий предохранитель		

Таблица А- 73 Питание датчиков

Технические характеристики	CPU 1215C AC/DC/Relay	CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Диапазон напряжений	от 20.4 до 28.8 В пост.т.	L+ минус 4 В пост.т. мин.	
Номинальный выходной ток (макс.)	400 mA (с защитой от короткого замыкания)		
Максимальная пульсирующая помеха (<10 МГц)	< 1 В между пиками	Как на входном проводе	
Электрическая развязка (логики CPU относительно питания датчиков)	Не изолировано		

А.5.3. Цифровые входы и выходы

Таблица А- 74 Цифровые входы

Технические характеристики	CPU 1215C AC/DC/Relay	CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Количество входов	14		
Тип	Потребитель/источник (МЭК тип 1 потребитель)		
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение		
Длительно допустимое напряжение	30 В пост.т., макс.		
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с		
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА		
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА		
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин		
Потенциально развязанные группы	1		
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0		
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.) (Уровень лог.1 = от 15 до 26 В пост.т.)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.5)		
Количество одноврем.активных входов	7 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 14 при 55 °С для горизонтальной или 45 °С для вертикальной установки		
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков		

Таблица А- 75 Цифровые выходы

Технические характеристики	CPU 1215C AC/DC/Relay и CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Количество выходов	10	
Тип	релейный, механический	Твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	--	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	--	0.1 В пост.т. макс.
Ток (макс.)	2.0 А	0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.	5 Вт
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля	0.6 Ом макс.
Ток утечки на выход	--	10 мкА макс.
Ток включения	7 А с закрытыми контактами	8 А в течение 100 мс макс.
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в течение 1 мин (катушка относительно контакта) Не изолировано (катушка отн.логики)	500 В пер.т. в течение 1 мин
Сопротивление изоляции	100 МОм мин. для нового модуля	--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин	--
Потенциально развязанные группы	2	1

A.5 CPU 1215C

Технические характеристики	CPU 1215C AC/DC/Relay и CPU 1215C DC/DC/Relay	CPU 1215C DC/DC/DC
Электрическая развязка (между группами)	1500 В пер.т. ¹	--
Индуктивное фиксирующее напряжение	--	L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения (от Qa.0 до Qa.3)	10 мс макс.	1.0 мкс макс., из выкл. во вкл. 3.0 мкс макс., из вкл. в выкл.
Задержка переключения (от Qa.4 до Qa.5)	10 мс макс.	50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключения реле	1 Гц	--
Частота пачки импульсов	Не рекомендуется ²	100 кГц (от Qa.0 до Qa.3) ³ , 2 Гц мин. 20 кГц (от Qa.4 до Qb.1) ³
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.	--
Срок службы контактов при номинальной нагрузке	100,000 циклов замык./размык.	--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	5 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 10 при 55 °С для горизонтальной или 45 °С для вертикальной установки	
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный	

¹ Электрическая развязка релейной группы по отношению к другим группам разделяет линейное напряжение от SELVPELV и разделяет различные фазы до 250 В пер.т. относительно земли.

² Для моделей ЦПУ с релейными выходами Вы должны установить цифровую сигнальную плату (SB), чтобы использовать импульсные выходы.

³ В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

A.5.4. Аналоговые входы и выходы

Таблица A- 76 Аналоговые входы

Технические характеристики	Описание
Количество входов	2
Тип	напряжение (асимметричный)
Диапазон измерений	от 0 до 10 В
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 10.001 до 11.759 В
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	от 11.760 до 11.852 В
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное выдерживаемое напряжение	35 В пост.т.
Сглаживание	Отсутствует, слабое, среднее или сильное Смотри таблицу для реакции на скачок (мс) для аналоговых входов ЦПУ (стр. 1147).
Подавление помех	10, 50 или 60 Гц

Технические характеристики	Описание
Полное сопротивление	≥100 кОм
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от 0 до 55 °C)	3.0% / 3.5% полного диапазона
Длина кабеля (метры)	100 м, экранированная витая пара

A.5.4.1. Реакция на скачок встроенных аналоговых входов ЦПУ

Таблица А- 77 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов): 16 значения	200 мс	240 мс	1150 мс
Сильное (32 циклов): 32 значения	400 мс	480 мс	2300 мс
Время выборки	4.17 мс	5 мс	25 мс

A.5.4.2. Время выборки для встроенных аналоговых входов ЦПУ

Таблица А- 78 Время выборки для встроенных аналоговых входов ЦПУ

Частота подавления (выбор времени интегрирования)	Время выборки
60 Гц (16.6 мс)	4.17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.5.4.3. Диапазоны измерений для аналоговых входов напряжения (ЦПУ)

Таблица А- 79 Представление аналогового входа для напряжения (ЦПУ)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11.851 В	Переполнение
32512	7F00		
32511	7EFF	11.759 В	Диапазон превышения
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.5.4.4. Характеристики аналоговых выходов

Таблица А- 80 Аналоговые выходы

Технические характеристики	Описание
Количество выходов	2
Тип	токовый
Диапазон измерений	от 0 до 20 мА
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 20.01 до 23.52 мА
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	Смотри сноску ¹
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Полное сопротивление выходного силового элемента	≤500 Ом макс.
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от -20 до 60 °C)	3.0% / 3.5% полного диапазона
Время установки	2 мс
Длина кабеля (метры)	100 м, экранированная витая пара

¹ В состоянии переполнения аналоговые выходы будут вести себя согласно настройкам свойств конфигурации устройства. В параметре "Reaction to CPU STOP" (*Реакция на переход в состояние ОСТАНОВ; Прим.перев.*), выберите либо: "Use substitute value" (*Использовать значение подстановки; Прим.перев.*) или "Keep last value" (*Сохранить последнее значение ; Прим.перев.*).

Таблица A- 81 Представление аналогового выхода для тока (CPU 1215C и CPU 1217C)

Система		Диапазон выходного тока	
Десятичная	Шестнадцатеричная	от 0 до 20 мА	
32767	7FFF	Смотри сноску ¹	Переполнение
32512	7F00	Смотри сноску ¹	
32511	7EFF	23.52 мА	Диапазон превышения
27649	6C01		
27648	6C00	20 мА	Номинальный диапазон
20736	5100	15 мА	
34	22	0.0247 мА	
0	0	0 мА	
Отрицательные значения		Отрицательные значения не поддерживаются	

¹ В состоянии переполнения аналоговые выходы будут вести себя согласно настройкам свойств конфигурации устройства. В параметре "Reaction to CPU STOP" (*Реакция на переход в состояние ОСТАНОВ; Прим.перев.*), выберите либо: "Use substitute value" (*Использовать значение подстановки; Прим.перев.*) или "Keep last value" (*Сохранить последнее значение ; Прим.перев.*).

А.5.5. Схемы электрических соединений ЦПУ 1215

Таблица А- 82 CPU 1215C AC/DC/Relay (6ES7 215-2BG40-0XB0)

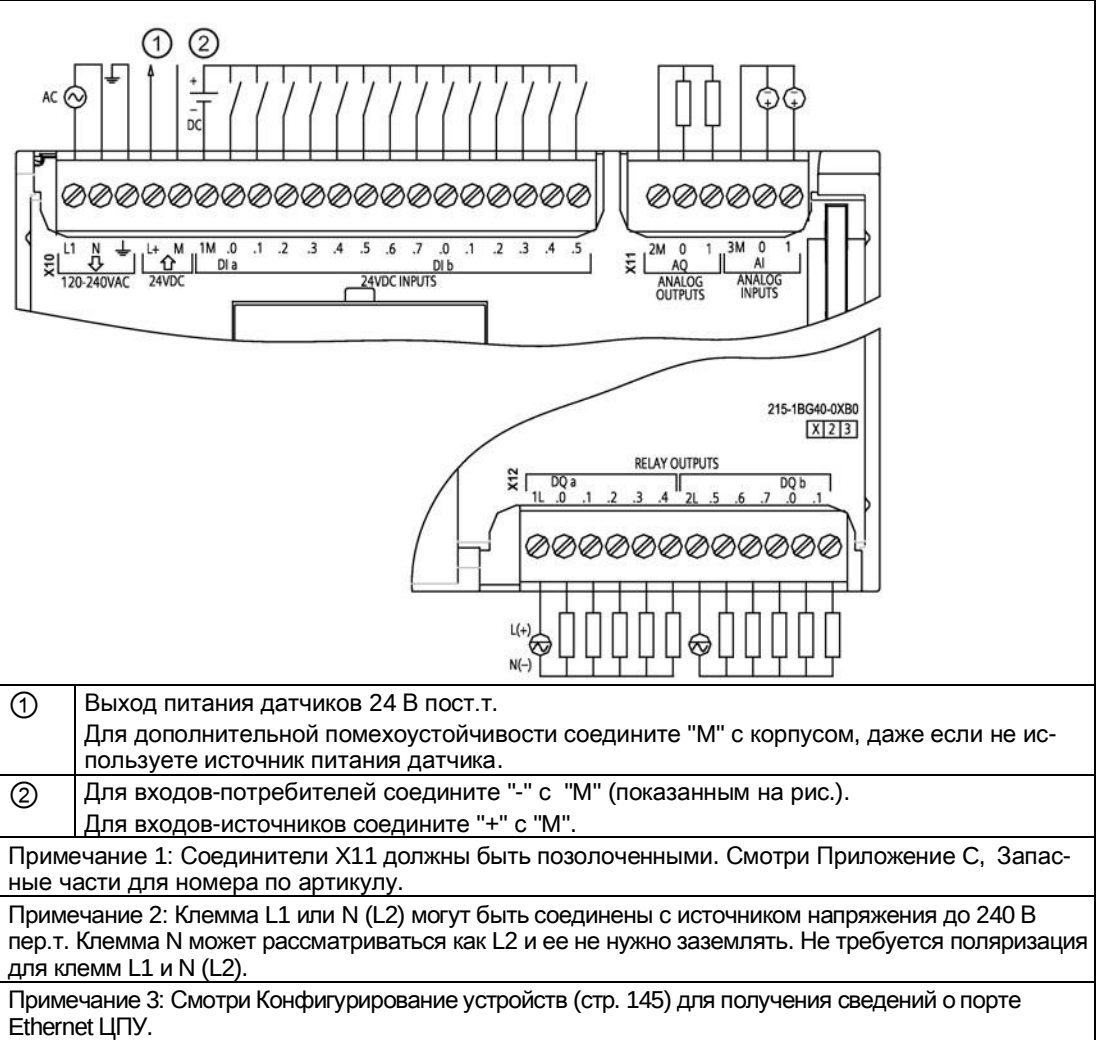


Таблица А- 83 Привязка контактов разъема CPU 1215C AC/DC/Relay (6ES7 215-2BG40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L1 / 120-240 В пер.т.	2 М	1L
2	N / 120-240 В пер.т.	AQ 0	DQ a.0
3	Функциональная земля	AQ 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	3М	DQ a.2
5	М / 24 В пост.т. питание датчиков	AI 0	DQ a.3
6	1М	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6

Контакт	X10	X11 (с позолотой)	X12
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.5	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица А- 84 CPU 1215C DC/DC/Relay (6ES7 215-1HG40-0XB0)

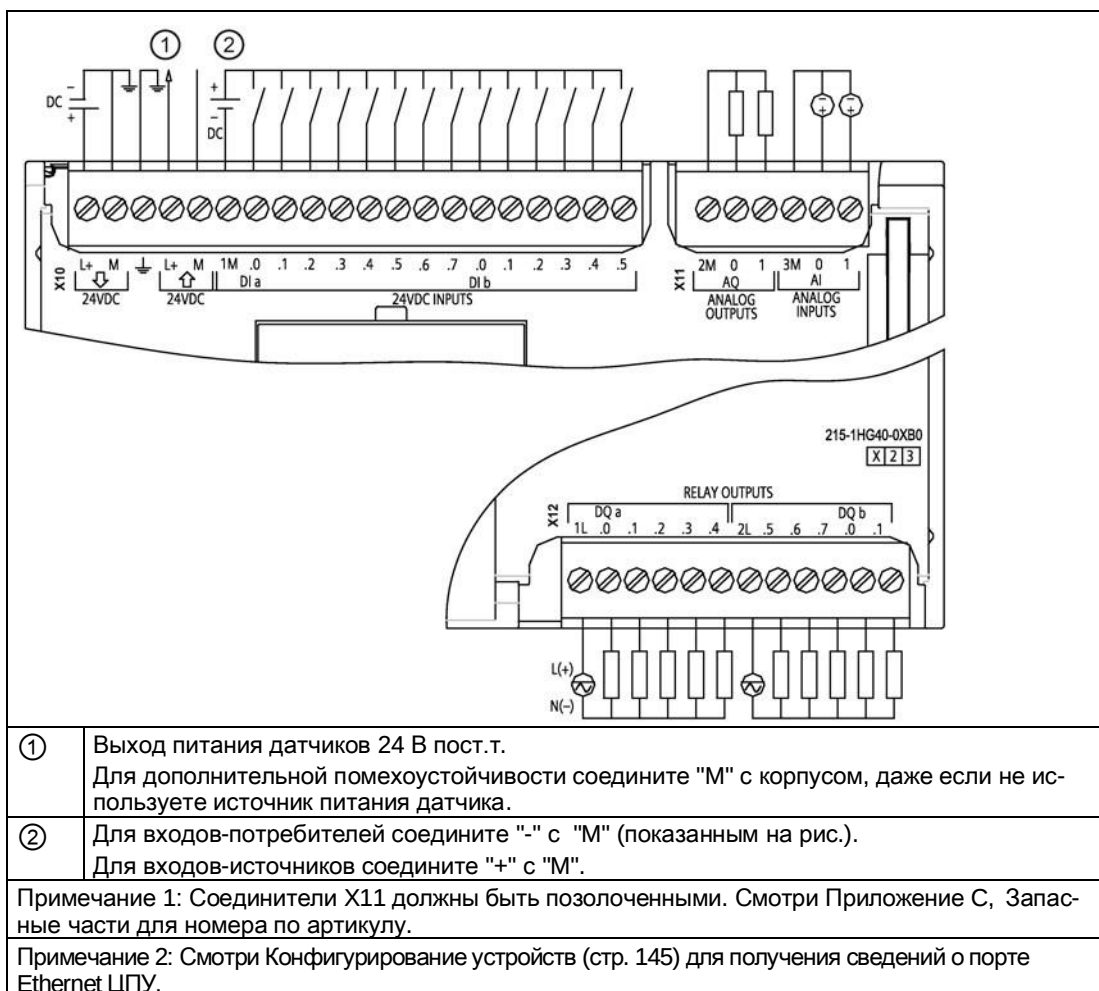


Таблица А- 85 Привязка контактов разъема CPU 1215C DC/DC/Relay (6ES7 215-1HG40-0XB0)

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	1L
2	M / 24 VDC	AQ 0	DQ a.0
3	Функциональная земля	AQ 1	DQ a.1
4	L+ / 24 В пост.т. питание датчиков	3 M	DQ a.2
5	M / 24 В пост.т. питание датчиков	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

The diagram illustrates the terminal block connections for the 215-1AG40-0XB0 module. It shows two main sections: the top section for 24VDC inputs and analog signals, and the bottom section for 24VDC outputs.

Top Section (24VDC INPUTS and ANALOG):

- 24VDC INPUTS:** Labeled "24VDC INPUTS", it includes terminals for "DI a" (0 to 5) and "DI b" (0 to 5). It also shows "L+" and "M" terminals for "24VDC" and "1M" terminals for "1M".
- ANALOG:** Labeled "ANALOG OUTPUTS" and "ANALOG INPUTS", it includes terminals for "2M", "0", "1", "3M", "0", and "1".
- DC Connections:** A "DC" connection is shown at the top left, with a "1" label. A "2" label is also present.

Bottom Section (24VDC OUTPUTS):

- 24VDC OUTPUTS:** Labeled "24VDC OUTPUTS", it includes terminals for "DQ a" (4L+, 4M, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1) and "DQ b" (4L+, 4M, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1).
- DC Connections:** A "DC" connection is shown at the bottom left, with a "+" and "-" terminal.

The module is identified as "215-1AG40-0XB0" with a label "X23" below it.

- | | |
|---|---|
| ① | Выход питания датчиков 24 В пост.т.
Для дополнительной помехоустойчивости соедините "М" с корпусом, даже если не используете источник питания датчика. |
| ② | Для входов-потребителей соедините "-" с "М" (показанным на рис.).
Для входов-источников соедините "+" с "М". |

Примечание 2: См. Конфигурирование устройств (стр. 145) для получения сведений о порте Ethernet ЦПУ.

Контакт	X10	X11 (с позолотой)	X12
1	L+ / 24 VDC	2 M	3L+
2	M / 24 VDC	AQ 0	3M
3	Функциональная земля	AQ 1	DQ a.0
4	L+ / 24 В пост.т. питание датчиков	3 M	DQ a.1
5	M / 24 В пост.т. питание датчиков	AI 0	DQ a.2
6	1M	AI 1	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--

Контакт	X10	X11 (с позолотой)	X12
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены

A.6. CPU 1217C

A.6.1. Общие технические данные и характеристики

Таблица А- 88 Общие технические данные

Технические данные	CPU 1217C DC/DC/DC
Номер для заказа	6ES7 217-1AG40-0XB0
Размеры Ш x В x Г (мм)	150 x 100 x 75
Вес с упаковкой	530 г
Рассеиваемая мощность	12 Вт
Доступный ток (шина CM)	макс. 1600 мА (5 В пост.т.)
Доступный ток (24 В пост т.)	макс. 400 мА (питание датчиков)
Потребление тока цифровым входом (24 пост т.)	4 мА/используемый вход

Таблица А- 89 Характеристики ЦПУ

Технические данные		Описание
Пользовательская память (Обратитесь к "Общим техническим данным" (стр. 1099), "Сохраняемость внутренней памяти ЦПУ".)	Рабочая	150 Кбайт
	Загрузочная	4 Мбайт, внутренняя, расширяемая до объема SD-карты
	Сохраняемая	10 Кбайт
Встроенные цифровые входы/выходы		14 входов/10 выходов
Встроенные аналоговые входы/выходы		2 входа / 2 выхода
Величина образа процесса		1024 байта входов (I) /1024 байта выходов (Q)
Битовая память (M)		8192 байт
Временная (локальная) память		16 кБайт для запуска и программного цикла (включая вызываемые FB и FC) 6 кБайт для каждого из других уровней приоритета прерываний (включая вызываемые FB и FC)
Дополнительные сигнальные модули		макс.8 SM
Расширение с помощью SB, CB, BB		макс. 1
Дополнительные коммуникационные модули		макс. 3 CM
Скоростные счетчики		До 6 сконфигурированных для использования входов (любых встроенных или на SB). Обратитесь к таблице, ЦПУ 1217C: Таблица аппаратного конфигурирования цифровых входов (DI) (стр. 1158). 1 МГц (от Ib.2 до Ib.5) 100/180 кГц (от Ia.0 до Ia.5) 30/120 кГц (от Ia.6 до Ib.1)

A.6 CPU 1217C

Технические данные	Описание
Импульсные выходы ²	До 4 сконфигурированных для использования любых выходов встроенных или на SB. Обратитесь к таблице, ЦПУ 1217C: Таблица аппаратного конфигурирования цифровых выходов (DQ) (стр. 1158). • 1 МГц (от Qa.0 до Qa.3) • 100 кГц (от Qa.4 до Qb.1)
Входы для улавливания импульсов	14
Прерывания с задержкой по времени	Всего 4 с разрешением 1 мс
Циклические прерывания	Всего 4 с разрешением 1 мс
Прерывания по фронту	12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти	Карта памяти SIMATIC (дополнительно)
Точность часов реального времени	+/- 60 секунд/месяц
Длительность сохранения времени для часов реального времени	Тип.знач. 20 дней /мин.знач. 12 дней при 40°C (не требующий обслуживания суперконденсатор)

¹ Более медленная скорость применима, когда высокоскоростной счетчик сконфигурирован для квадратурного режима работы.

Таблица А- 90 Производительность

Тип инструкции	Скорость выполнения
Булевая	0,08 мкс на команду
Передача слова	1,7 мкс на команду
Арифметика над действительными числами	2,3 мкс на команду

A.6.2. Таймеры, счетчики и кодовые блоки, поддерживаемые CPU 1217C

Таблица А- 70 Блоки, таймеры и счетчики, поддерживаемые CPU 1217C

Элемент		Описание
Блоки	Тип	OB, FB, FC, DB
	Размер	64 кБайт
	Количество	До 1024 блоков всего (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложения	16 от OB программного цикла или запуска 6 от OB любого другого прерывания по событию
	Наблюдение	Состояние 2 блоков может наблюдаться одновременно
Организационные блоки OB	Программный цикл	Произвольное количество
	Стартовые	Произвольное количество
	Прерывания с задержкой	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени исполнения	1
	Диагностические прерывания	1
	Снятие / установка модулей	1

Элемент		Описание
Организационные блоки OB	Ошибка стойки / станции	1
	По времени дня	Произвольное количество
	Состояние	1
	Обновление	1
	Параметрирование	1
Таймеры	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	МЭК
	Количество	Ограничено только объемом памяти
	Хранение	Структура в DB, размер зависит от типа счета • SInt, USInt: 3 Байта • Int, UInt: 6 Байт • DInt, UDInt: 12 Байт

Таблица A- 92 Коммуникации

Технические характеристики	Описание
Количество портов	2
Тип	Ethernet
Устройство человеко-машинного интерфейса	4
Программатор (PG)	1
Соединения	• 8 для Открытых пользовательских коммуникаций (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 3 для серверных GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций • 8 для клиентских GET/PUT (ЦПУ-с-ЦПУ) S7 коммуникаций
Скорость передачи данных	10/100 Мб/с
Электрическая развязка (внешний сигнал относительно логики ПЛК)	Изоляция с помощью трансформатора, 1500 В пер.т., только для краткосрочной безопасности
Тип кабеля	CAT5e экранированный

Таблица A- 93 Источник питания

Технические характеристики		CPU 1217C DC/DC/DC
Диапазон напряжений		от 20.4 до 28.8 В пост.т.
Частота сети		--
Входной ток (макс. нагрузка)	только CPU	600 мА при 24 В пост.т.
	CPU со всеми модулями расширения	1600 мА при 24 В пост.т.
Бросок тока при включении (макс.)		12 А при 28.8 В пост.т.
Электрическая развязка (входного питания относительно логики)		Не изолировано
Время удержания (при сбое питания)		10 мс при 24 В пост.т.
Внутренний предохранитель, не заменяемый пользователем		3 А, 250 В, плавкий предохранитель

A.6 CPU 1217C

Таблица А- 94 Питание датчиков

Технические характеристики	CPU 1217C DC/DC/DC
Диапазон напряжений	L+ минус 4 В пост.т. мин.
Номинальный выходной ток (макс.)	400 мА (с защитой от короткого замыкания)
Максимальная пульсирующая помеха (<10 МГц)	Как на входном проводе
Электрическая развязка (логики CPU относительно питания датчиков)	Не изолировано

A.6.3. Цифровые входы и выходы

Таблица А- 95 Цифровые входы

Технические характеристики	CPU 1217C DC/DC/DC
Количество входов	14: общее количество 10: потребитель/источник (МЭК тип 1 потребитель) 4: дифференциальный (RS422/RS485)
Тип: потребитель/источник (МЭК тип 1 источник)	от Ia.0 до Ia.7, от Ib.0 до Ib.1
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение
Длительно допустимое напряжение	30 В пост.т., макс.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин
Потенциально развязанные группы	1
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.) (Уровень лог.1 = от 15 до 26 В пост.т.)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.1)
Тип: дифференциальный вход (RS422/RS485)	от Ib.2 до Ib.5 (от .2+ .2- до .5+ .5-)
Динамический диапазон в режиме синфазного сигнала	от -7 до +12 В, 1 секунда, 3 среднеквадр. знач. непрерывно (технические характеристики RS422/RS485)
Встроенная концевая нагрузка и смещение	390 Ом относительно 2М на Ib'-, 390 Ом относительно +5 V на Ib'-, (OFF со смещением при T/B разомкнутой цепи) 220 Ом между Ib'+ и Ib'-
Входной импеданс приемника	100 Ом включая смещение и концевую нагрузку
Порог/чувствительность дифференциального приемника	+/- 0.2 В мин., 60 мВ типовой гистерезис (технические характеристики RS422/RS485)
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин
Потенциально развязанные группы	1
Постоянные времени фильтра	us настройки: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0 ms настройки: 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0

Технические характеристики	CPU 1217C DC/DC/DC
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Одиночная фаза: 1 МГц (от Ib.2 до Ib.5) Фаза со сдвигом на 90°: 1 МГц (от Ib.2 до Ib.5)
Межканальная расфазировка дифференциального входа	40 нс макс.
Общие характеристики (все цифровые входы)	
Количество одноврем.активных входов	5 потребитель/источник (без смежных точек) и 4 дифференциальных при 60 °С для горизонтальной или 50 °С для вертикальной установки 14 при 55 °С для горизонтальной или 45 °С для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 300 неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков (потребитель/источник) 50 м экранированная витая пара для всех дифференциальных входов

Таблица А- 96 CPU 1217C Таблица аппаратного конфигурирования цифровых входов (DI)

Вход	Тип и скорость
DIa.0	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.1	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.2	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.3	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.4	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.5	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.6	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIa.7	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIb.0	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIb.1	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: 100 кГц макс.
DIb.2+ .2-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: 1 МГц макс.
DIb.3+ .3-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: 1 МГц макс.

A.6 CPU 1217C

Вход	Тип и скорость
Dlb.4+ .4-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: 1 МГц макс.
Dlb.5+ .5-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: 1 МГц макс.

Таблица А- 97 Цифровые выходы

Технические характеристики	CPU 1217C DC/DC/DC
Количество выходов	10: общее количество 6: твердотельный - MOSFET (источник) 4: дифференциальный (RS422/RS485)
Тип: твердотельный – MOSFET (источник)	Qa.4 to Qb.1
Диапазон напряжения	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	0.1 В пост.т. макс.
Ток (макс.)	0.5 А
Ламповая нагрузка	5 Вт
Сопротивление в активном состоянии	0.6 Ом макс.
Ток утечки на выход	10 мкА макс.
Ток включения	8 А в течение 100 мс макс.
Защита от перегрузки	Нет
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин (функциональная развязка)
Потенциально развязанные группы	1
Индуктивное фиксирующее напряжение	L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения (от Qa.0 до Qa.3)	1.0 мкс макс., из выкл. во вкл. 3.0 мкс макс., из вкл. в выкл.
Задержка переключения (от Qa.4 до Qb.1)	50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключения реле	--
Частота пачки импульсов	100 кГц (от Qa.4 до Qb.1) ¹ , 2 Гц мин.
Тип: дифференциальный выход (RS422/RS485)	от Qa.0 до Qa.3 (от .0+ .0- до .3+ .3-)
Динамический диапазон в режиме син-фазного сигнала	от -7 до +12 В, 1 секунда, 3 среднеквадр. знач. непрерывно (технические характеристики RS422/RS485)
Встроенная концевая нагрузка	100 Ом между Qa'+ и Qa'-'
Выходной импеданс силового элемента	100 Ом включая концевую нагрузку
Электрическая развязка	500 В пер.т. в течение 1 мин (функциональная развязка)
Потенциально развязанные группы	1
Задержка переключения (от DQa.0 до DQa.3)	100 нс макс.

Технические характеристики	CPU 1217C DC/DC/DC
Межканальная расфазировка дифференциального входа	40 нс макс.
Частота пачки импульсов	1 МГц (от Qa.0 до Qa.3), 2 Гц мин.
Общие характеристики (все цифровые выходы)	
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	3 твердотельных MOSFET (источник) выхода (без смежных точек) и 4 дифференциальных выхода при 60 °С для горизонтальной или 50 °С для вертикальной установки 10 при 55 °С для горизонтальной или 45 °С для вертикальной установки
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный

¹ В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

Таблица А- 98 CPU 1217C Таблица аппаратного конфигурирования цифровых выходов (DQ)

Вход	Тип и скорость
DQa.0+ .0-	Тип: RS422/RS485 дифференциальных выход Частота выходной пачки импульсов: 1 МГц макс., 2 Гц мин.
DQa.1+ .1-	Тип: RS422/RS485 дифференциальных выход Частота выходной пачки импульсов: 1 МГц макс., 2 Гц мин.
DQa.2+ .2-	Тип: RS422/RS485 дифференциальных выход Частота выходной пачки импульсов: 1 МГц макс., 2 Гц мин.
DQa.3+ .3-	Тип: RS422/RS485 дифференциальных выход Частота выходной пачки импульсов: 1 МГц макс., 2 Гц мин.
DQa.4	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.
DQa.5	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.
DQa.6	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.
DQa.7	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.
DQb.0	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.
DQb.1	Тип: 24 В, выход источник Частота выходной пачки импульсов: 100 кГц макс., 2 Гц мин.

A.6.4. Аналоговые входы и выходы

A.6.4.1. Характеристики аналоговых входов

Таблица A- 99 Аналоговые входы

Технические характеристики	Описание
Количество входов	2
Тип	напряжение (асимметричный)
Диапазон измерений	от 0 до 10 В
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 10.001 до 11.759 В
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	от 11.760 до 11.852 В
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное выдерживаемое напряжение	35 В пост.т.
Сглаживание	Отсутствует, слабое, среднее или сильное Смотри таблицу для реакции на скачок (мс) для аналоговых входов ЦПУ (стр. 1162).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от 0 до 55 °C)	3.0% / 3.5% полного диапазона
Длина кабеля (метры)	100 м, экранированная витая пара

A.6.4.2. Реакция на скачок встроенных аналоговых входов ЦПУ

Таблица A- 100 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов): 16 значения	200 мс	240 мс	1150 мс
Сильное (32 циклов): 32 значения	400 мс	480 мс	2300 мс
Время выборки	4.17 мс	5 мс	25 мс

А.6.4.3. Время выборки для встроенных аналоговых входов ЦПУ

Таблица А- 101 Время выборки для встроенных аналоговых входов ЦПУ

Частота подавления (выбор времени интегрирования)	Время выборки
60 Гц (16.6 мс)	4.17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

А.6.4.4. Диапазоны измерений для аналоговых входов напряжения (ЦПУ)

Таблица А- 102 Представление аналогового входа для напряжения (ЦПУ)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11.851 В	Переполнение
32512	7F00		
32511	7EFF	11.759 В	Диапазон превышения
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

А.6.4.5. Характеристики аналоговых выходов

Таблица А- 103 Аналоговые выходы

Технические характеристики	Описание
Количество выходов	2
Тип	токовый
Диапазон измерений	от 0 до 20 мА
Диапазон измерений (слово данных)	от 0 до 27648
Диапазон превышения	от 20.01 до 23.52 мА
Диапазон превышения (слово данных)	от 27649 до 32511
Диапазон переполнения	Смотри сноску ¹
Диапазон переполнения (слово данных)	от 32512 до 32767
Разрешение	10 бит
Полное сопротивление выходного силового элемента	≤500 Ом макс.
Электрическая развязка (полевая сторона относительно логики)	Отсутствует
Точность (25 °C / от -20 до 60 °C)	3.0% / 3.5% полного диапазона

A.6 CPU 1217C

Технические характеристики	Описание
Время установки	2 мс
Длина кабеля (метры)	100 м, экранированная витая пара

- ¹ В состоянии переполнения аналоговые выходы будут вести себя согласно настройкам свойств конфигурации устройства. В параметре "Reaction to CPU STOP" (*Реакция на переход в состояние ОСТАНОВ; Прим.перев.*), выберите либо: "Use substitute value" (*Использовать значение подстановки; Прим.перев.*) или "Keep last value" (*Сохранить последнее значение ; Прим.перев.*).

Таблица А- 104 Представление аналогового выхода для тока (CPU 1215C и CPU 1217C)

Система		Диапазон выходного тока	
Десятичная	Шестнадцатеричная	от 0 до 20 мА	
32767	7FFF	Смотри сноску ¹	Переполнение
32512	7F00	Смотри сноску ¹	
32511	7EFF	23.52 мА	Диапазон превышения
27649	6C01		
27648	6C00	20 мА	Номинальный диапазон
20736	5100	15 мА	
34	22	0.0247 мА	
0	0	0 мА	
Отрицательные значения		Отрицательные значения не поддерживаются	

- ¹ В состоянии переполнения аналоговые выходы будут вести себя согласно настройкам свойств конфигурации устройства. В параметре "Reaction to CPU STOP" (*Реакция на переход в состояние ОСТАНОВ; Прим.перев.*), выберите либо: "Use substitute value" (*Использовать значение подстановки; Прим.перев.*) или "Keep last value" (*Сохранить последнее значение ; Прим.перев.*).

А.6.5. Схемы электрических соединений ЦПУ 1217

Таблица А- 105 CPU 1217C DC/DC/DC (6ES7 217-1AG40-0XB0)

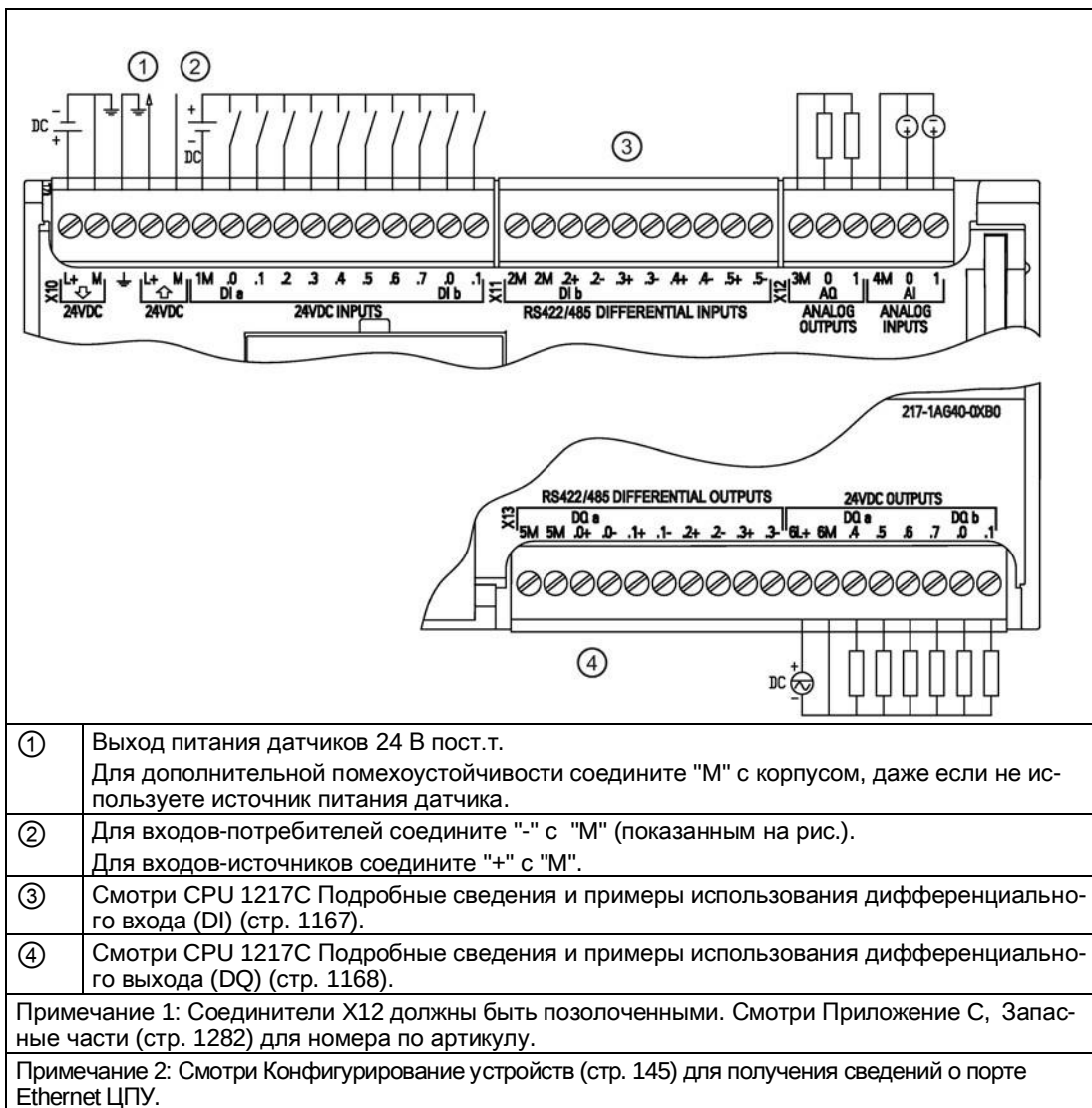


Таблица А- 106 Привязка контактов разъема 1217C DC/DC/DC (6ES7 217-1AG40-0XB0)

Контакт	X10	X11	X12 (с позолотой)	X13
1	L+ / 24 В пост.т.	2M	3M	5M
2	M / 24 В пост.т.	2M	AQ 0	5M
3	Функциональная земля	DI b.2+	AQ 1	DQ a.0+
4	L+ / 24 В пост.т. пит.датчиков	DI b.2-	4M	DQ a.0-
5	M / 24 В пост.т. пит.датчиков	DI b.3+	AI 0	DQ a.1+
6	1M	DI b.3-	AI 1	DQ a.1-
7	DI a.0	DI b.4+	--	DQ a.2+
8	DI a.1	DI b.4-	--	DQ a.2-
9	DI a.2	DI b.5+	--	DQ a.3+

Контакт	X10	X11	X12 (с позолотой)	X13
10	DI a.3	DI b.5-	--	DQ a.3-
11	DI a.4	--	--	6L+
12	DI a.5	--	--	6M
13	DI a.6	--	--	DQ a.4
14	DI a.7	--	--	DQ a.5
15	DI b.0	--	--	DQ a.6
16	DI b.1	--	--	DQ a.7
17	--	--	--	DQ b.0
18	--	--	--	DQ b.1

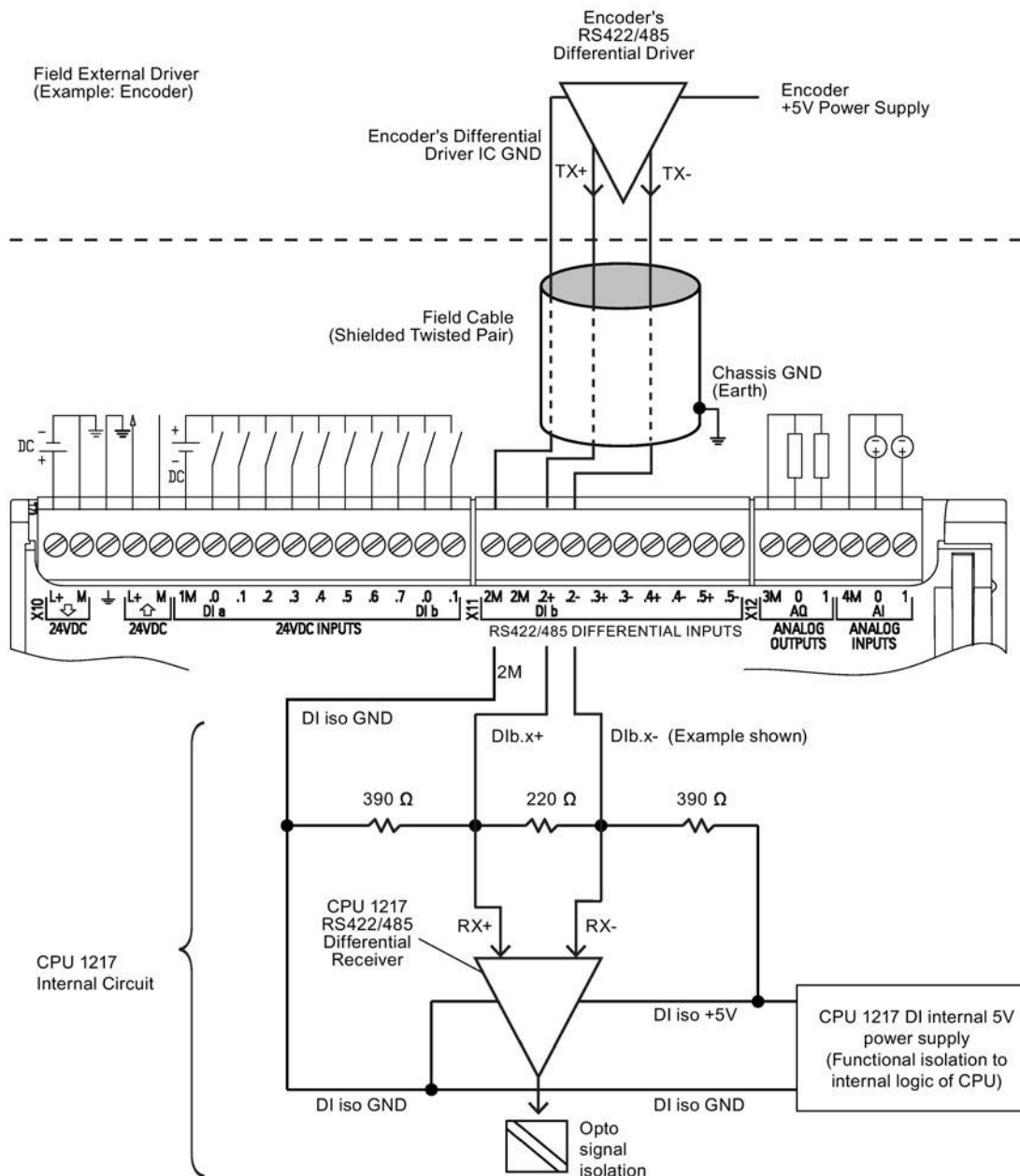
Примечание

Неиспользуемые аналоговые входы должны быть закорочены

Смотри также

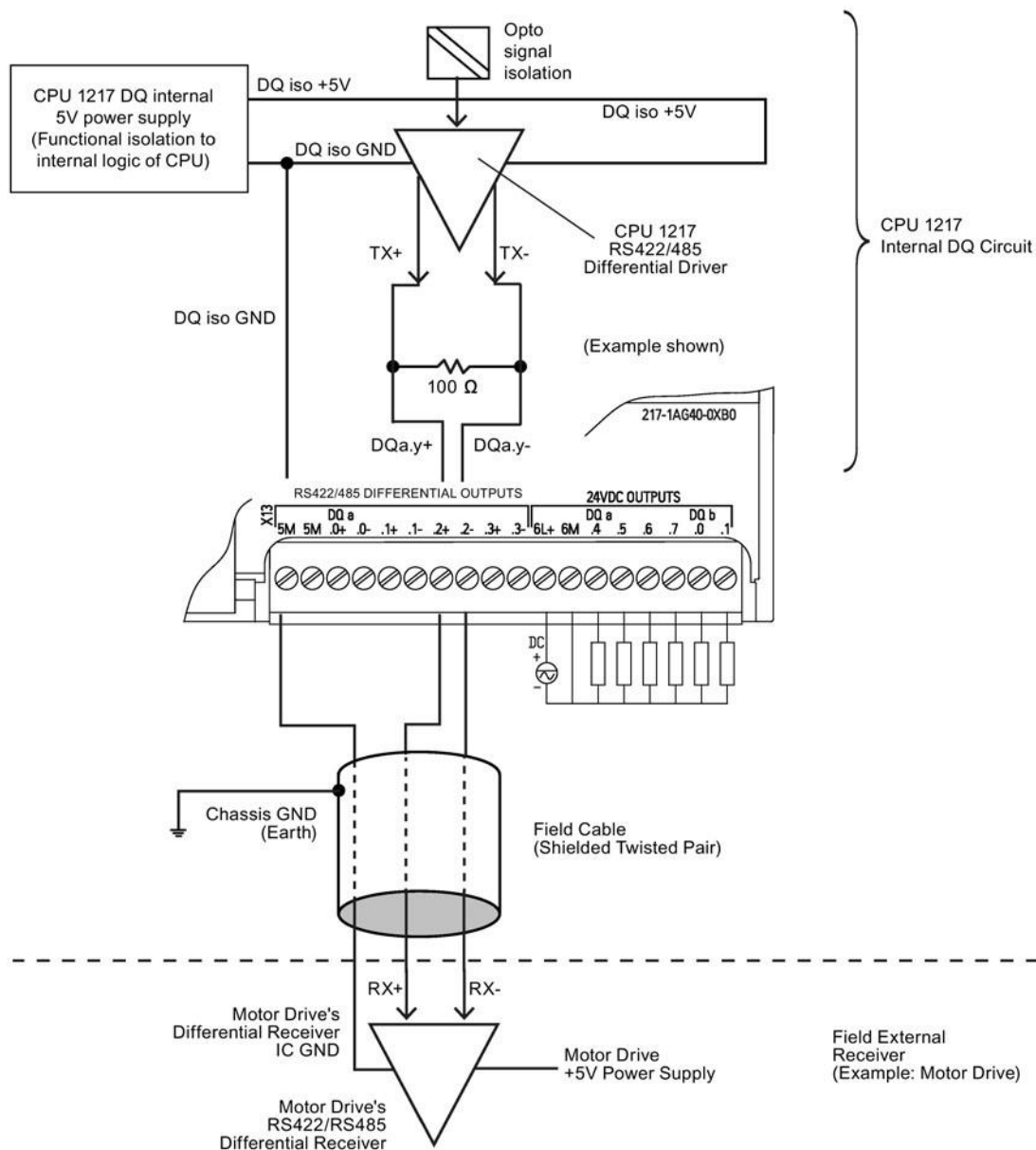
Аналоговые входы и выходы (стр. 1146).

A.6.6. CPU 1217C. Подробные сведения и примеры использования дифференциального входа (DI)



- Каждый дифференциальный цифровой вход смещен в состояние "ВЫКЛ.", когда винтовые зажимы клеммной колодки разомкнуты.
- Встроенные оконечная нагрузка и смещение DI имеют эквивалентный импеданс 100 Ом.
- Встроенные резисторы оконечной нагрузки и смещения DI ограничивают непрерывный динамический диапазон в режиме синфазного сигнала. Сммотри электрические характеристики для получения подробных сведений.

A.6.7. CPU 1217C. Подробные сведения и примеры использования дифференциального выхода (DQ)



- Встроенные резисторы оконечной нагрузки и смещения DQ ограничивают непрерывный динамический диапазон в режиме синфазного сигнала. См. электрические характеристики для получения подробных сведений.

А.7. Цифровые сигнальные модули (SM)

А.7.1. Технические данные цифровых входов SM 1221

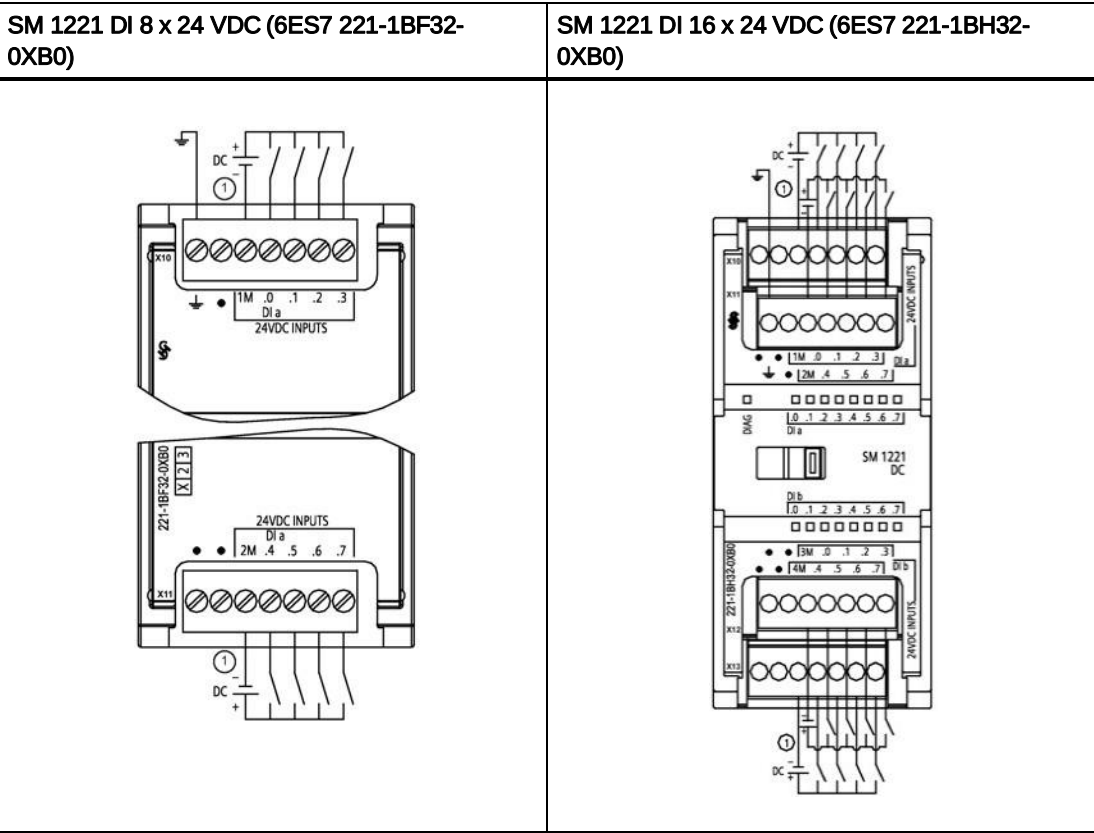
Таблица А- 107 Общие технические данные

Модель	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
Номер для заказа	6ES7 221-1BF32-0XB0	6ES7 221-1BH32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	
Вес	170 грамм	210 грамм
Рассеиваемая мощность	1.5 Вт	2.5 Вт
Потребление тока (SM шина)	105 мА	130 мА
Потребление тока (24 В пост.т.)	4 мА/используемый вход	

Таблица А- 108 Цифровые входы

Модель	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
Количество входов	8	16
Тип	Потребитель/источник (МЭК тип 1 потребитель)	
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение	
Длительно допустимое напряжение	30 В пост.т., макс.	
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с	
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА	
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА	
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин	
Потенциально развязанные группы	2	4
Постоянные времени фильтра	0.2, 0.4, 0.8, 1.6, 3.2, 6.4 и 12.8 мс (выбирается группами по 4)	0.2, 0.4, 0.8, 1.6, 3.2, 6.4 и 12.8 мс (выбирается группами по 4)
Количество одноврем.активных входов	8	16
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный	

Таблица А- 109 Схемы электрических соединений для цифрового входа SM



① Для входов потребителей соедините "-" с "М" (показано на рис.). Для входов источников, соедините "+" с "М".

Таблица А- 110 Привязка контактов разъема для SM 1221 DI 8 x 24 VDC (6ES7 221-1BF32-0XB0)

Контакт	X10	X11
1	Функциональная земля	Нет соединения
2	Нет соединения	Нет соединения
3	1M	2M
4	DI a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

Таблица А- 111 Привязка контактов разъема для SM 1221 DI 16 x 24 VDC (6ES7 221-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	Нет соединения	Функциональная земля	Нет соединения	Нет соединения
2	Нет соединения	No connection	Нет соединения	Нет соединения
3	1M	2M	3 M	4 M

Контакт	X10	X11	X12	X13
4	DI a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

A.7.2. Технические данные 8-канальных цифровых выходов SM 1222

Таблица А- 112 Общие технические данные

Модель	SM 1222 DQ 8 x Relay	SM 1222 DQ 8 RLY Changeover	SM 1222 DQ 8 x 24 VDC
Номер для заказа	6ES7 222-1HF32-0XB0	6ES7 222-1XF32-0XB0	6ES7 222-1BF32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Вес	190 грамм	310 грамм	180 грамм
Рассеиваемая мощность	4.5 Вт	5 Вт	1.5 Вт
Потребление тока (SM шина)	120 мА	140 мА	120 мА
Потребление тока (24 В пост.т.)	11 мА / на катушку реле	16.7 мА / на катушку реле	50 мА

Таблица А- 113 Цифровые выходы

Модель	SM 1222 DQ 8 x Relay	SM 1222 DQ8 RLY Changeover	SM 1222 DQ 8 x 24 VDC
Количество выходов	8		
Тип	релейный, механический	релейный переключающий контакт	твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.		от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	--		20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	--		0.1 В пост.т. макс
Ток (макс.)	2.0 А		0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.		5 Вт
Сопrotивление в активном состоянии	0.2 Ом макс. для нового модуля		0.6 Ом макс.
Ток утечки на выход	--		10 мкА макс.
Ток включения	7 А с закрытыми контактами		8 А в течение 100 мс макс.
Защита от перегрузки	нет		
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в теч. 1 мин (катушка отн. контакта) Не изолировано (катушка отн. логики)	1500 В пер.т. в теч. 1 мин (катушка отн. контакта)	500 В пер.т. в течение 1 мин.
Сопrotивление изоляции	100 МОм мин. для нового модуля		--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин.		--
Потенциально развязанные группы	2	8	1
Ток на общий провод (макс.)	10 А	2 А	4 А

A.7 Цифровые сигнальные модули (SM)

Модель	SM 1222 DQ 8 x Relay	SM 1222 DQ8 RLY Changeover	SM 1222 DQ 8 x 24 VDC
Индуктивное фиксирующее напряжение	--		L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения	10 мс макс.		50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключения реле	1 Гц		--
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.		--
Срок службы контактов при номинальной нагрузке (норм.откр.контакт)	100,000 циклов замык./размык.		--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)		
Количество одноврем.активных выходов	8	4 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 8 при 55 °С для горизонтальной или 45 °С для вертикальной установки	8
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный		

A.7.3. Технические данные 16-канальных цифровых выходов SM 1222

Таблица А- 114 Общие технические данные

Модель	SM 1222 DQ 16 x Relay	SM 1222 DQ 16 x 24 VDC
Номер для заказа	6ES7 222-1BH32-0XB0	6ES7 222-1BH32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	
Вес	260 грамм	220 грамм
Рассеиваемая мощность	8.5 Вт	2.5 Вт
Потребление тока (SM шина)	135 мА	140 мА
Потребление тока (24 В пост.т.)	11 мА / на катушку реле	100 мА

Таблица А- 115 Цифровые выходы

Модель	SM1222 DQ 16 x Relay	SM1222 DQ 16 x 24 VDC
Количество выходов	16	
Тип	релейный, механический	твердотельный - MOSFET (источник)
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	-	20 В пост.т. мин.
Сигнал логического 0 с нагрузкой 10 кОм	-	0.1 В пост.т. макс.

Модель	SM1222 DQ 16 x Relay	SM1222 DQ 16 x 24 VDC
Ток (макс.)	2.0 А	0.5 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.	5 Вт
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля	0.6 Ом макс.
Ток утечки на выход	--	10 мкА макс.
Ток включения	7 А с закрытыми контактами	8 А в течение 100 мс макс.
Защита от перегрузки	нет	
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в теч. 1 мин (катушка отн. контакта) Не изолировано (катушка отн. логики)	500 В пер.т. в течение 1 мин.
Сопротивление изоляции	100 МОм мин. для нового модуля	-
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин.	-
Потенциально развязанные группы	4	1
Ток на общий провод (макс.)	10 А	8 А
Индуктивное фиксирующее напряжение	--	L+ минус 48 В пост.т., мощность рассеяния 1 Вт
Задержка переключения	10 мс макс.	50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.
Максимальная частота переключе-	1 Гц	--
Механический срок службы	10,000,000 циклов замык./размык.	--
Срок службы контактов при номинальной нагрузке (норм.откр.контакт)	100,000 циклов замык./размык.	--
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	8 (без смежных точек) при 60 °C для горизонтальной или 50 °C для вертикальной установки 16 при 55 °C для горизонтальной или 45 °C для вертикальной установки	16
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный	

Таблица А- 116 Схемы электрических соединений для SM 8-канальных цифровых выходов

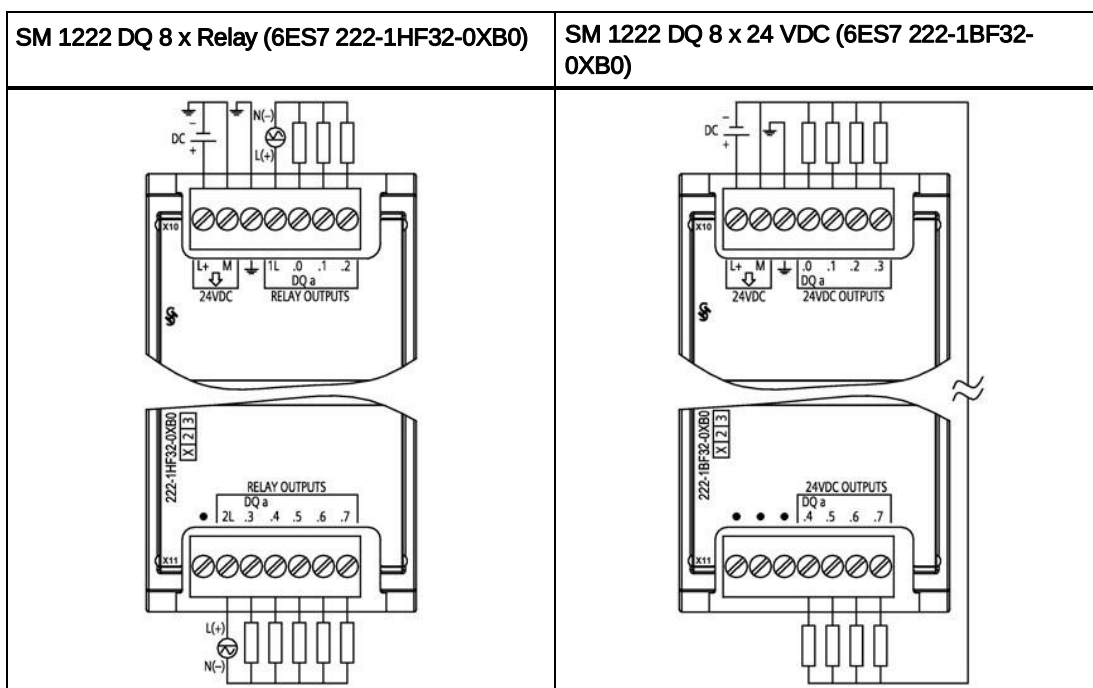


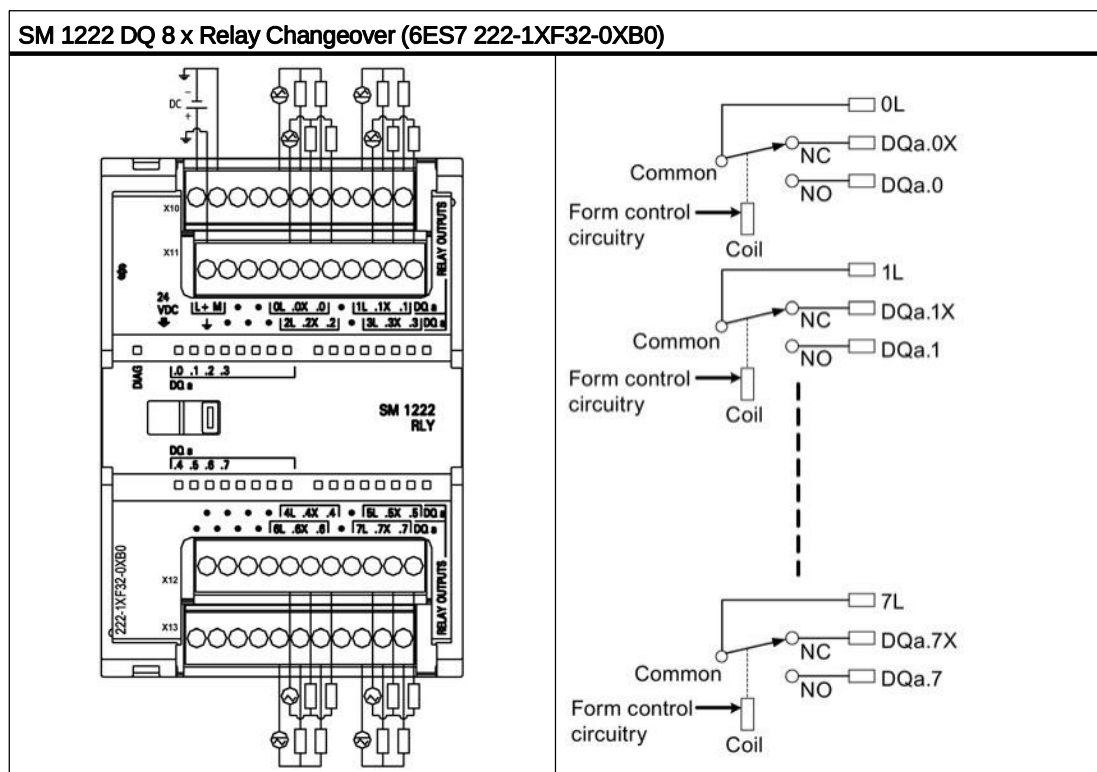
Таблица А- 117 Привязка контактов разъема для SM 1222 DQ 8 x Relay (6ES7 222-1HF32-0XB0)

Контакт	X10	X11
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	2L
3	Функциональная земля	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

Таблица А- 118 Привязка контактов разъема для SM 1222 DQ 8 x 24 VDC (6ES7 222-1BF32-0XB0)

Контакт	X10	X11
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	Нет соединения
3	Функциональная земля	Нет соединения
4	DQ a.0	DQ a.4
5	DQ a.1	DQ a.5
6	DQ a.2	DQ a.6
7	DQ a.2	DQ a.7

Таблица А- 119 Схемы электрических соединений для SM 8-канальных цифровых релейных выходов с переключающим контактом



Переключающий релейный выход управляет двумя цепями с помощью соединения с общим проводом: одного нормально замкнутого и одного нормально разомкнутого контакта. Например, с помощью выхода "0", в состоянии OFF, общий провод (0L) соединяется с нормально замкнутым контактом (.0X) и отключается от нормально разомкнутого контакта (.0). В состоянии выхода ON общий провод (0L) отключается нормально замкнутого контакта (.0X) и соединяется с нормально разомкнутым контактом (.0).

Таблица А- 120 Привязка контактов разъема для SM 1222 DQ 8 x Relay Changeover (6ES7 222-1XF32- 0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Нет соединения	Нет соединения	Нет соединения	Нет соединения
4	Нет соединения	Нет соединения	Нет соединения	Нет соединения
5	0L	2L	4L	6L
6	DQ a.0X	DQ a.2X	DQ a.4X	DQ a.6X
7	DQ a.0	DQ a.2	DQ a.4	DQ a.6
8	Нет соединения	Нет соединения	Нет соединения	Нет соединения
9	1L	3L	5L	7L
10	DQ a.1X	DQ a.3X	DQ a.5X	DQ a.7X
11	DQ a.1	DQ a.3	DQ a.5	DQ a.7

Таблица А- 121 Схемы электрических соединений для SM 16-канальных цифровых выходов

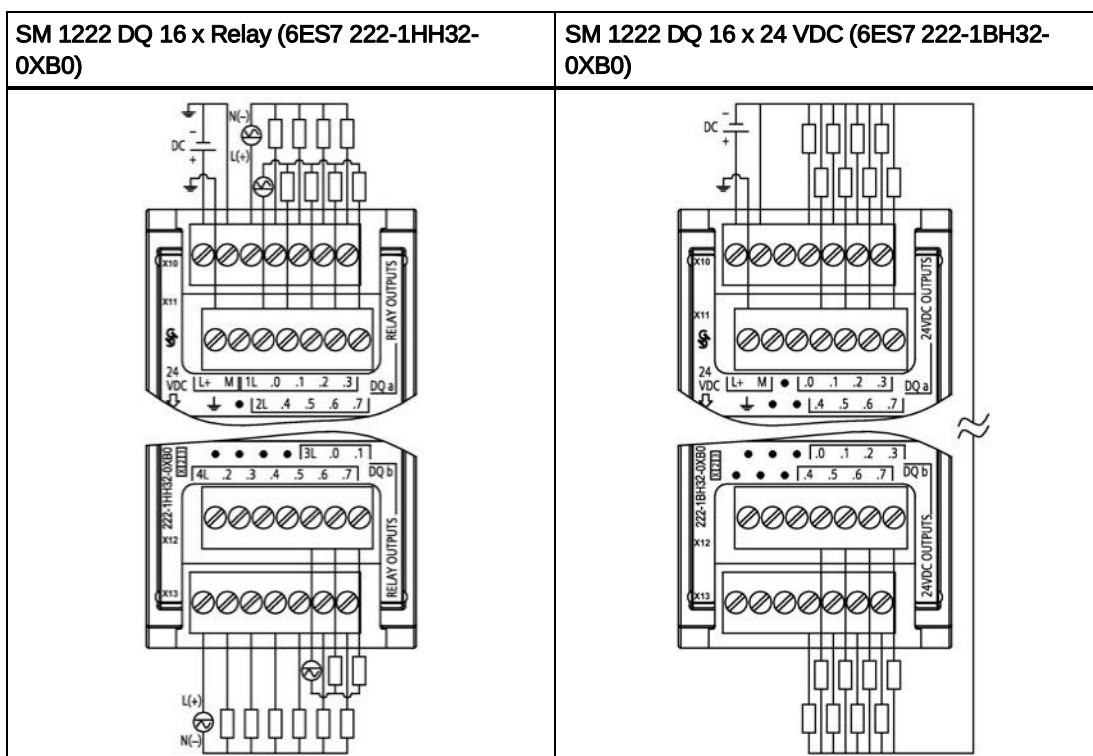


Таблица А- 122 Привязка контактов разъема для SM 1222 DQ 16 x Relay (6ES7 222-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	4L
2	M / 24 В пост.т.	Нет соединения	Нет соединения	DQ b.2
3	1L	2L	Нет соединения	DQ b.3
4	DQ a.0	DQ a.4	Нет соединения	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5
6	DQ a.2	DQ a.6	DQ b.0	DQ b.6
7	DQ a.3	DQ a.7	DQ b.1	DQ b.7

Таблица А- 123 Привязка контактов разъема для SM 1222 DQ 16 x VDC (6ES7 222-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Нет соединения	Нет соединения	Нет соединения	Нет соединения
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5

Контакт	X10	X11	X12	X13
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

A.7.4. Технические данные цифровых входов/выходов постоянного тока SM 1223

Таблица А- 124 Общие технические данные

Модель	SM 1223 DI 8 x 24 VDC, DQ 8 x Relay	SM 1223 DI 16 x 24 VDC, DQ 16 x Relay	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Номер для заказа	6ES7 223-1PH32-0XB0	6ES7 223-1PL32-0XB0	6ES7 223-1BH32-0XB0	6ES7 223-1BL32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
Вес	230 грамм	350 грамм	210 грамм	310 грамм
Рассеиваемая мощность	5.5 Вт	10 Вт	2.5 Вт	4.5 Вт
Потребление тока (SM шина)	145 мА	180 мА	145 мА	185 мА
Потребление тока (24 В пост.т.)	4 мА/используемый вход 11 мА / на катушку реле		150 мА	200 мА

Таблица А- 125 Цифровые входы

Модель	SM 1223 DI 8 x 24 VDC, DQ 8 x Relay	SM 1223 DI 16 x 24 VDC, DQ 16 x Relay	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Количество входов	8	16	8	16
Тип	Потребитель/источник (МЭК тип 1 потребитель)			
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение			
Длительно допустимое напряжение	30 В пост.т., макс.			
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с			
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА			
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА			
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин			
Потенциально развязанные группы	2			
Постоянные времени фильтра	0.2, 0.4, 0.8, 1.6, 3.2, 6.4 и 12.8 мс, выбирается группами по 4			
Количество одноврем.активных входов	8	8 (без смежных точек) при 60 °С для гориз. или 50 °С для вертик. установки 16 при 55 °С для гориз. или 45 °С для вертик. установки	8	16
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный			

A.7 Цифровые сигнальные модули (SM)

Таблица А- 126 Цифровые выходы

Модель	SM 1223 DI 8 x 24 VDC, DQ 8 x Relay	SM 1223 DI 16 x 24 VDC, DQ 16 x Relay	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
Количество выходов	8	16	8	16
Тип	релейный, механический		твердотельный - MOSFET (источник)	
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.		от 20.4 до 28.8 В пост.т.	
Сигнал логической 1 при макс. токе	--		20 В пост.т. мин.	
Сигнал логического 0 с нагрузкой 10 кОм	--		0.1 В пост.т. макс.	
Ток (макс.)	2.0 А		0.5 А	
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.		5 Вт	
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля		0.6 Ом макс.	
Ток утечки на выход	--		10 мкА макс.	
Ток включения	7 А с закрытыми контактами		8 А в течение 100 мс макс.	
Защита от перегрузки	Нет			
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в теч. 1 мин (катушка отн. контакта) Не изолировано (катушка отн. логики)		500 В пер.т. в течение 1 мин.	
Сопротивление изоляции	100 МОм мин. для нового модуля			--
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин.			--
Потенциально развязанные группы	2	4	1	
Ток на общий провод (макс.)	10А	8 А	4 А	8 А
Индуктивное фиксирующее напряжение	--		L+ минус 48 В пост.т., мощность рассеяния 1 Вт	
Задержка переключения	10 мс макс.		50 мкс макс., из выкл. во вкл. 200 мкс макс., из вкл. в выкл.	
Максимальная частота переключения	1 Гц		--	
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.		--	
Срок службы контактов при номинальной нагрузке (норм.откр.контакт)	100,000 циклов замык./размык.		--	
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)			
Количество одноврем.активных выходов	8	8 (без смежных точек) при 60 °С для горизонт. или 50 °С для вертик. установки 16 при 55 °С для горизонт. или 45 °С для вертикальной установки	8	16
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный			

Таблица А- 127 Схемы электрических соединений для SM цифровых входов постоянного тока /релейных выходов

SM 1223 DI 8 x 24 VDC, DQ 8 x Relay (6ES7 223-1PH32-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x Relay (6ES7 223-1PL32-0XB0)	Примечание
		① Для входов-потребителей, соедините "-" с "М" (показано на рис.). Для входов-источников, соедините "+" с "М".

Таблица А- 128 Привязка контактов разъема для SM 1223 DI 8 x 24 VDC, DQ 8 x Relay (6ES7 223-1PH32- 0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

А.7 Цифровые сигнальные модули (SM)

Таблица А- 129 Привязка контактов разъема для SM 1223 DI 16 x 24 VDC, DQ 16 x Relay (6ES7 223- 1PL32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	1L	3L
2	M / 24 В пост.т.	Нет соединения	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1
4	DI a.0	DI b.0	DQ a.2	DQ b.2
5	DI a.1	DI b.1	DQ a.3	DQ b.3
6	DI a.2	DI b.2	Нет соединения	Нет соединения
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

Таблица А- 130 Схемы электрических соединений для SM цифровых входов / выходов постоянного тока

SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7 223-1BH32-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7 223-1BL32-0XB0)	Примечание
		① Для входо-потребителей, соедините "-" с "М" (показано на рис.). Для входо-источников, соедините "+" с "М".

Таблица A- 131 Привязка контактов разъема для SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC
(6ES7 223- 1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	1M	2M	Нет соединения	Нет соединения
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Таблица A- 132 Привязка контактов разъема для SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
(6ES7 223- 1BL32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	1M	2M	Нет соединения	Нет соединения
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

A.7.5. Технические данные цифровых входов/выходов переменного тока SM 1223

Таблица А- 133 Общие технические данные

Модель	SM 1223 DI 8 x120/230 VAC / DQ 8 x Relay
Номер для заказа	6ES7 223-1QH32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75 мм
Вес	190 грамм
Рассеиваемая мощность	7.5 Вт
Потребление тока (SM шина)	120 мА
Потребление тока (24 В пост.т.)	11 мА на активный вход

Таблица А- 134 Цифровые входы

Модель	SM 1223 DI 8 x 120/230 VAC / DQ 8 x Relay
Количество входов	8
Тип	МЭК тип 1
Номинальное напряжение	120 В пер.т. при 6 мА, 230 В пер.т. при 9 мА
Длительно допустимое напряжение	264 В пер.т.
Импульсное перенапряжение	--
Сигнал логической 1 (мин.)	79 В пер.т. при 2.5 мА
Сигнал логического 0 (макс.)	20 В пер.т. при 1 мА
Ток утечки (макс.)	1 мА
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в течение 1 мин
Потенциально развязанные группы ¹	4
Время задержки входа	типовое: от 0.2 до 12.8 мс, выбирается пользователем максимальное: -
Подключение 2-проводных датчиков приближения (Всего) (макс.)	1 мА
Длина кабеля (в метрах)	300 м неэкранированный 500 м экранированный,
Количество одновременно активных входов	8

¹ Каналы из одной группы должны подключаться к одной фазе.

Таблица A- 135 Цифровые выходы

Модель	SM 1223 DI 8 x 120/230 VAC / DQ 8 x Relay
Количество выходов	8
Тип	релейный, механический
Диапазон напряжения	от 5 до 30 В пост.т. или от 5 до 250 В пер.т.
Сигнал логической 1 при макс. токе	--
Сигнал логического 0 с нагрузкой 10 кОм	--
Ток (макс.)	2.0 А
Ламповая нагрузка	30 Вт пост.т. / 200 Вт пер.т.
Сопротивление в активном состоянии	0.2 Ом макс. для нового модуля
Ток утечки на выход	--
Ток включения	7 А с закрытыми контактами
Защита от перегрузки	Нет
Электрическая развязка (полевая сторона относительно логики)	1500 В пер.т. в теч. 1 мин (катушка отн. контакта) Не изолировано (катушка отн. логики)
Сопротивление изоляции	100 МОм мин. для нового модуля
Электрическая развязка между открытыми контактами	750 В пер.т. в течение 1 мин.
Потенциально развязанные группы	2
Ток на общий провод (макс.)	10 А
Индуктивное фиксирующее напряжение	--
Задержка переключения (макс.)	10 мс
Максимальная частота переключения реле	1 Гц
Механический срок службы (без нагрузки)	10,000,000 циклов замык./размык.
Срок службы контактов при номинальной нагрузке (норм.откр.контакт)	100,000 циклов замык./размык.
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	4 (без смежных точек) при 60 °С для горизонтальной или 50 °С для вертикальной установки 8 при 55 °С для горизонтальной или 45 °С для вертикальной установки
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный

А.7 Цифровые сигнальные модули (SM)

Таблица А-136 SM 1223 DI 8 x 120/230 VAC, DQ 8 x Relay (6ES7 223-1QH32-0XB0)

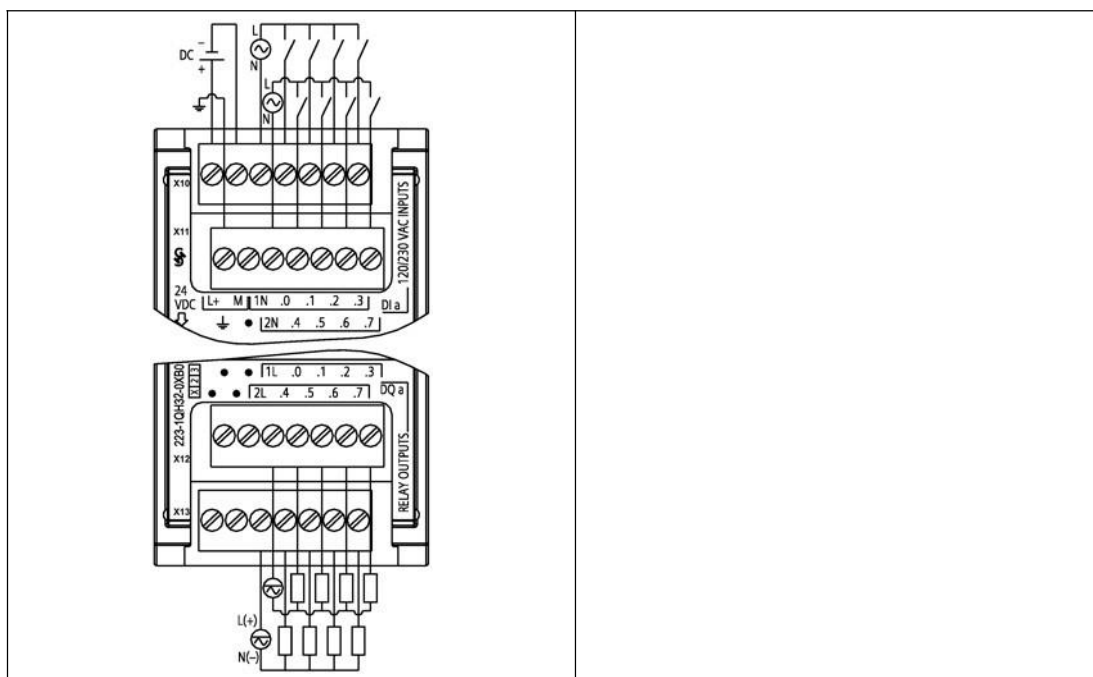


Таблица А- 137 Привязка контактов разъема для SM 1223 DI 8 x 120/240 VAC, DQ 8 x Relay (6ES7 223- 1QH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В пост.т.	Функциональная земля	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

А.8. Аналоговые сигнальные модули (SM)

А.8.1. Технические данные модулей аналоговых входов SM 1231

Таблица А- 138 Общие технические данные

Модель	SM 1231 AI 4 x 13 bit	SM 1231 AI 8 x 13 bit	SM 1231 AI 4 x 16 bit
Номер для заказа	6ES7 231-4HD32-0XB0	6ES7 231-4HF32-0XB0	6ES7 231-5ND30-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75		
Вес	180 грамм		
Рассеиваемая мощность	2.2 Вт	2.3 Вт	2.0 Вт
Потребление тока (SM шина)	80 мА	90 мА	80 мА
Потребление тока (24 В пост.т.)	45 мА		65 мА

Таблица А- 139 Аналоговые входы

Модель	SM 1231 AI 4 x 13 bit	SM 1231 AI 8 x 13 bit	SM 1231 AI 4 x 16 bit
Количество входов	4	8	4
Тип	Напряжение или ток (дифференциальный): конфигурируется группами по 2		Напряжение или ток (дифференциальный)
Диапазон	± 10 В, ± 5 В, ± 2.5 В, 0...20 мА или 4... 20 мА		± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 0...20 мА или 4... 20 мА
Диапазон измерений (слово данных)	от -27648 до 27648 для напряжения / от 0 до 27648 для тока		
Диапазон превышения полож./ отриц. значения (слово данных) Обратитесь к разделу диапазонов аналогового входа для напряжения и тока (стр. 1194).	Напряжение: от 32511 до 27649 / от -27649 до -32512 Ток: от 32511 до 27649 / от 0 до -4864		
Диапазон переполнения полож./ отриц. значения (слово данных) Обратитесь к разделу диапазонов аналогового входа для напряжения и тока (стр. 1194).	Напряжение: от 32767 до 32512 / от -32513 до -32768 Ток 0 ... 20 мА: от 32767 до 32512 / от -4865 до -32768 Ток 4 ... 20 мА: от 32767 до 32512 (значения меньше -4864 соответствуют обрыву провода)		
Разрешение ¹	12 бит + знаковый бит		15 бит + знаковый бит
Максимальная прочность по напряжению/току	± 35 В / ± 40 мА		
Сглаживание	Отсутствует, слабое, среднее или сильное Обратитесь к разделу для реакции на скачок (стр. 1194).		
Подавление помех	400, 60, 50 или 10 Гц Обратитесь к разделу для интервала выборки (стр. 1194).		
Входной импеданс	≥ 9 МОм (напряжение) / 280 Ом (ток)		≥ 1 МОм (напряжение) / < 315 Ом, > 280 Ом (ток)

A.8 Аналоговые сигнальные модули (SM)

Модель	SM 1231 AI 4 x 13 bit	SM 1231 AI 8 x 13 bit	SM 1231 AI 4 x 16 bit
Электрическая развязка Полевая сторона отн. логики Логика отн. 24 В пост.т. Полевая сторона отн. 24 В пост.т. Канал относительно канала	Нет		500 В пер.т. 500 В пер.т. 500 В пер.т. Нет
Точность (25 °C / от -20 до 60 °C)	$\pm 0.1\%$ / $\pm 0.2\%$ полного диапазона		$\pm 0.1\%$ / $\pm 0.3\%$ полного диапазона
Принцип измерения	Преобразование фактического значения		
Подавление синфазной помехи	40 дБ, от пост.т. до 60 Гц		
Рабочий диапазон сигналов ¹	Сигнал плюс напряжение синфазной помехи должно быть меньше +12 В и больше -12 В		
Длина кабеля (метры)	100 м, экранированная витая пара		

¹ Напряжение вне рабочего диапазона приложенное к одному из каналов, может повлиять на другие каналы.

Таблица А- 140 Диагностика

Модель	SM 1231 AI 4 x 13 bit	SM 1231 AI 8 x 13 bit	SM 1231 AI 4 x 16 bit
Переполнение для полож./ отриц. значений	Да		
Пониженное напряжение 24 В пост.т.	Да		
Обрыв провода	Только в диапазоне от 4 до 20 мА (если на входе меньше -4164; 1.185 мА)		

Таблица А- 141 Схемы электрических соединений для SM аналоговых входов

SM 1231 AI 4 x 13 bit (6ES7 231-4HD32-0XB0)	SM 1231 AI 8 x 13 bit (6ES7 231-4HF32-0XB0)
Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.	

Таблица A- 142 Привязка контактов разъема для SM 1231 AI 4 x 13 bit (6ES7 231-4HD32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	Нет соединения
3	Функциональная земля	Нет соединения
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Таблица A- 143 Привязка контактов разъема для SM 1231 AI 8 x 13 bit (6ES7 231-4HF32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-

Таблица A- 144 Схема электрических соединений для SM аналоговых входов

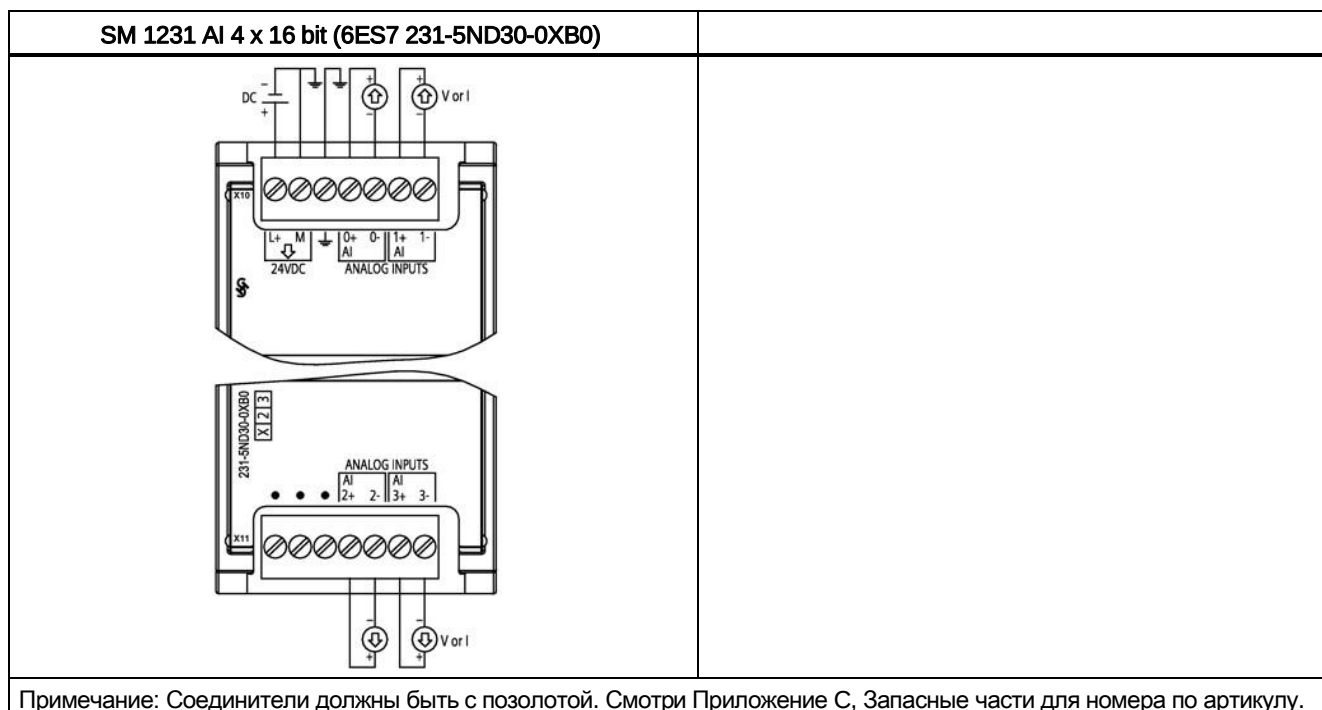


Таблица A- 145 Привязка контактов разъема для SM 1231 AI 4 x 16 bit (6ES7 231-5ND30-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	Нет соединения
3	Функциональная земля	Нет соединения
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Примечание

Неиспользуемые входы напряжения должны быть закорочены.

Для неиспользуемых входных токовых каналов необходимо сконфигурировать диапазон 0 ... 20 мА и/или отменить диагностику обрыва провода.

Входы, сконфигурированные для измерения тока, не генерируют ток в контуре, пока модуль не будет запитан и сконфигурирован.

Токовые входные каналы не будут работать, пока внешнее питание не будет подано на преобразователь.

A.8.2. Технические данные модулей аналоговых выходов SM 1232

Таблица A- 146 Общие технические данные

Технические характеристики	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Номер для заказа	6ES7 232-4NB32-0XB0	6ES7 232-4HD32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	
Вес	180 грамм	
Рассеиваемая мощность	1.8 Вт	2.0 Вт
Потребление тока (SM шина)	80 мА	
Потребление тока (24 В пост.т.)	45 мА (без нагрузки)	

Таблица А- 147 Аналоговые выходы

Технические характеристики	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Количество выходов	2	4
Тип	Напряжение или ток	
Диапазон	± 10 В, 0 ... 20 мА или 4 ... 20 мА	
Разрешение	Напряжение: 14 бит Ток: 13 бит	
Диапазон (слово данных)	Напряжение: от -27 648 до 27 648 ; Ток: от 0 до 27 648 Обратитесь к выходным диапазонам для напряжения и тока (стр. 1195).	
Точность (25 °С / от -20 до 60 °С)	$\pm 0.3\%$ / $\pm 0.6\%$ диапазона	
Время установки (95% от нового значения)	Напряжение: 300 мкс (резистор), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мкс (10 мГн)	
Импеданс нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом	
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	
Электрическая развязка Полевая сторона отн. логики	Нет	
Длина кабеля (метры)	100 м, экранированная витая пара	

Таблица А- 148 Диагностика

Технические характеристики	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Переполнение для полож./ отриц. значений	Да	
Замыкание на землю (только для напряжения)	Да	
Обрыв провода (только для тока)	Да	
Пониженное напряжение 24 В пост.т.	Да	

Таблица А- 149 Схема электрических соединений для SM аналоговых выходов

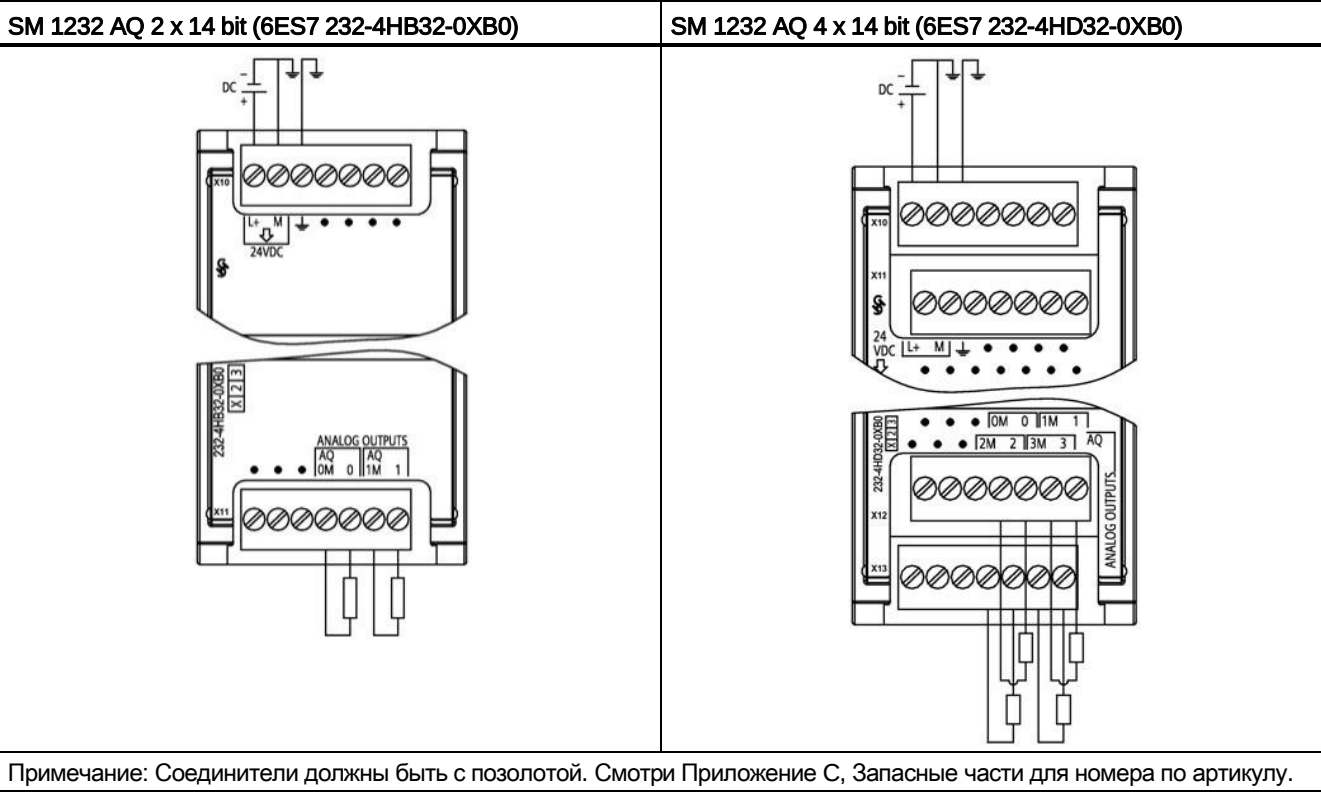


Таблица А- 150 Привязка контактов разъема для SM 1232 AQ 2 x 14 bit (6ES7 232-4HB32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	Нет соединения
3	Функциональная земля	Нет соединения
4	Нет соединения	AQ 0M
5	Нет соединения	AQ 0
6	Нет соединения	AQ 1M
7	Нет соединения	AQ 1

Таблица А- 151 Привязка контактов разъема для SM 1232 AQ 4 x 14 bit (6ES7 232-4HD32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	Нет соединения	Нет соединения	AQ 0M	AQ 2M
5	Нет соединения	Нет соединения	AQ 0	AQ 2

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
6	Нет соединения	Нет соединения	AQ 1M	AQ 3M
7	Нет соединения	Нет соединения	AQ 1	AIQ 3

A.8.3. Технические данные модулей аналоговых входов/выходов SM 1234

Таблица A- 152 Общие технические данные

Технические характеристики	SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit
Номер для заказа	6ES7 234-4HE32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75
Вес	220 грамм
Рассеиваемая мощность	2.4 Вт
Потребление тока (SM шина)	80 мА
Потребление тока (24 В пост.т.)	60 мА (без нагрузки)

Таблица A- 153 Аналоговые входы

Модель	SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit
Количество входов	4
Тип	Напряжение или ток (дифференциальный): конфигурируется группами по 2
Диапазон	± 10 В, ± 5 В, ± 2.5 В, 0...20 мА или 4... 20 мА
Диапазон измерений (слово данных)	от -27648 до 27648
Диапазон превышения положительные / отрицательные значения (слово данных)	Напряжение: от 32511 до 27649 / от -27649 до -32512 Ток: от 32511 до 27649 / от 0 до -4864 Обратитесь к разделу диапазонов аналогового входа для напряжения и тока (стр. 1194).
Диапазон переполнения положительные / отрицательные значения (слово данных)	Напряжение: от 32767 до 32512 / от -32513 до -32768 Ток: от 32767 до 32512 / от -4865 до -32768 Обратитесь к разделу диапазонов аналогового входа для напряжения и тока (стр. 1194).
Разрешение	12 бит + знаковый бит
Максимальная прочность по напряжению/току	± 35 В / ± 40 мА
Сглаживание	Отсутствует, слабое, среднее или сильное Обратитесь к разделу для реакции на скачок (стр. 1194).
Подавление помех	400, 60, 50 или 10 Гц Обратитесь к разделу для интервала выборки (стр. 1194).
Входной импеданс	≥ 9 МОм (напряжение) / 280 Ом (ток)
Электрическая развязка Полевая сторона отн. логики	Нет
Точность (25 °C / от -20 до 60 °C)	$\pm 0.1\%$ / $\pm 0.2\%$ полного диапазона
Время преобразования аналогового значения в цифровое	625 мкс (подавление 400 Гц)
Подавление синфазной помехи	40 дБ, от пост.т. до 60 Гц

A.8 Аналоговые сигнальные модули (SM)

Модель	SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit
Рабочий диапазон сигналов ¹	Сигнал плюс напряжение синфазной помехи должно быть меньше +12 В и больше -12 В
Длина кабеля (метры)	100 м, экранированная витая пара

¹ Напряжение вне рабочего диапазона приложенное к одному из каналов, может повлиять на другие каналы.

Таблица А- 154 Аналоговые выходы

Технические характеристики	SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit
Количество выходов	2
Тип	Напряжение или ток
Диапазон	±10 В или 0 ... 20 мА
Разрешение	Напряжение: 14 бит; Ток: 13 бит
Диапазон (слово данных)	Напряжение: от -27 648 до 27 648 ; Ток: от 0 до 27 648 Обратитесь к выходным диапазонам для напряжения и тока (стр. 1195).
Точность (25 °C / от -20 до 60 °C)	±0.3% / ±0.6% диапазона
Время установки (95% от нового значения)	Напряжение: 300 мкс (резистор), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мкс (10 мГн)
Импеданс нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)
Электрическая развязка Полевая сторона отн. логики	Нет
Длина кабеля (метры)	100 м, экранированная витая пара

Таблица А- 155 Диагностика

Модель	SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit
Переополнение для полож./ отриц. значений	Да
Замыкание на землю (только для напряжения)	Да для выходов
Обрыв провода (только для тока)	Да для выходов
Пониженное напряжение 24 В пост.т.	Да

Таблица A- 156 Схема электрических соединений для SM аналоговых входов/выходов

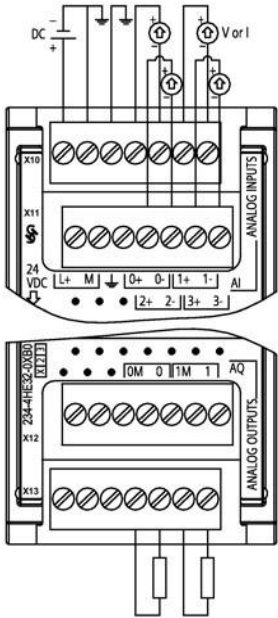
SM 1234 AI 4 x 13 Bit / AQ 2 x 14 bit (6ES7 234-4HE32-0XB0)	
	
Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.	

Таблица A- 157 Привязка контактов разъема для SM 1234 AI 4 x 13 Bit / AQ 2 x 14 bit (6ES7 234-4HE32- 0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	AI 0+	AI 2+	Нет соединения	AQ 0M
5	AI 0-	AI 2-	Нет соединения	AQ 0
6	AI 1+	AI 3+	Нет соединения	AQ 1M
7	AI 1-	AI 3-	Нет соединения	AQ 1

Примечание

Неиспользуемые входы напряжения должны быть закорочены.

Для неиспользуемых входных токовых каналов необходимо сконфигурировать диапазон 0 ... 20 мА и/или отменить диагностику обрыва провода.

Входы, сконфигурированные для измерения тока, не генерируют ток в контуре, пока модуль не будет запитан и сконфигурирован.

Токовые входные каналы не будут работать, пока внешнее питание не будет подано на преобразователь.

A.8.4. Реакция на скачок аналоговых входов

Таблица А- 158 Реакция на скачок (мс) от 0 до границы диапазона, измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Частота подавления (время интегрирования)			
	400 Гц (2.5 мс)	60 Гц (16.6 мс)	50 Гц (20 мс)	10 Гц (100 мс)
Отсутствует (1 цикл): без усреднения	4 мс	18 мс	22 мс	100 мс
Слабое (4 цикла): 4 значения	9 мс	52 мс	63 мс	320 мс
Среднее (16 циклов): 16 значения	32 мс	203 мс	241 мс	1200 мс
Сильное (32 циклов): 32 значения	61 мс	400 мс	483 мс	2410 мс
Время выборки				
• 4 AI x 13 bits	• 0.625 мс	• 4.17 мс	• 5 мс	• 25 мс
• 8 AI x 13 bits	• 1.25 мс	• 4.17 мс	• 5 мс	• 25 мс
• 4 AI4 x 16 bits	• 0.417 мс	• 0.397 мс	• 0.400 мс	• 0.400 мс

A.8.5. Время выборки и время обновления для аналоговых входов

Таблица А- 159 Время выборки и время обновления

Частота подавления (время интегрирования)	Время выборки	Время обновления модуля для всех каналов	
		4-канальный SM	8-канальный SM
400 Гц (2.5 мс)	4-канальный SM: 0.625 мс 8-канальный SM: 1.250 мс	0.625 мс	1.250 мс
60 Гц (16.6 мс)	4.170 мс	4.17 мс	4.17 мс
50 Гц (20 мс)	5.000 мс	5 мс	5 мс
10 Гц (100 мс)	25.000 мс	25 мс	25 мс

A.8.6. Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM)

Таблица А- 160 Представление аналогового входа для напряжения (SB и SM)

Система		Диапазон измерения напряжения				
Десятичная	Шестнадцатеричная	±10 В	±5 В	±2.5 В	±1.25 В	
32767	7FFF ¹	11.851 В	5.926 В	2.963 В	1.481 В	Переполнение для полож.значений
32512	7F00					
32511	7EFF	11.759 В	5.879 В	2.940 В	1.470 В	Диапазон превыше- ния для полож.знач.
27649	6C01					
27648	6C00	10 В	5 В	2.5 В	1.250 В	Номинальный диапазон
20736	5100	7.5 В	3.75 В	1.875 В	0.938 В	
1	1	361.7 мВ	180.8 мВ	90.4 мВ	45.2 мВ	

Система		Диапазон измерения напряжения			
Десятичная	Шестнадцатеричная	±10 В	±5 В	±2.5 В	±1.25 В
0	0	0 В	0 В	0 В	0 В
-1	FFFF				
-20736	AF00	-7.5 В	-3.75 В	-1.875 В	-0.938 В
-27648	9400	-10 В	-5 В	-2.5 В	-1.250 В
-27649	93FF				
-32512	8100	-11.759 В	-5.879 В	-2.940 В	-1.470 В
-32513	80FF				
-32768	8000	-11.851 В	-5.926 В	-2.963 В	-1.481 В

¹ 7FFF может быть возвращено по одной из следующих причин: переполнение (как отмечено в этом столе), прежде чем станут доступны действительные значения (например, непосредственно после подачи питания) или если обнаружен обрыв провода.

Таблица А- 161 Представление аналогового входа для тока (SB и SM)

Система		Диапазон измерения тока		
Десятичная	Шестнадцатер.	от 0 до 20 мА	от 4 до 20 мА	
32767	7FFF	23.70 мА	22.96 мА	Переполнение для положительных значений
32512	7F00			
32511	7EFF	23.52 мА	22.81 мА	Диапазон превышения для положительных значений
27649	6C01			
27648	6C00	20 мА	20 мА	Номинальный диапазон
20736	5100	15 мА	16 мА	
1	1	723.4 нА	4 мА + 578.7 нА	
0	0	0 мА	4 мА	
-1	FFFF			Диапазон превышения для отрицательных значений
-4864	ED00	-3.52 мА	1.185 мА	
-4865	ECFF			Переполнение для отрицательных значений
-32768	8000			

А.8.7. Диапазоны аналоговых выходов для напряжения и тока (SB и SM)

Таблица А- 162 Представление аналогового выхода для напряжения (SB и SM)

Система		Диапазон выходного напряжения	
Десятичная	Шестнадцатер.	± 10 В	
32767	7FFF	Смотри примечание 1	Переполнение для положительных значений
32512	7F00	Смотри примечание 1	
32511	7EFF	11.76 В	Диапазон превышения для положительных значений
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	

A.8 Аналоговые сигнальные модули (SM)

Система		Диапазон выходного напряжения	
Десятичная	Шестнадцатер.	± 10 В	
1	1	361.7 мкВ	
0	0	0 В	
-1	FFFF	-361.7 мкВ	
-20736	AF00	-7.5 В	
-27648	9400	-10 В	
-27649	93FF		Диапазон превышения для отрицательных значений
-32512	8100	-11.76 В	
-32513	80FF	Смотри примечание 1	Переполнение для отрицательных значений
-32768	8000	Смотри примечание 1	

¹ В состоянии переполнения для положительных или отрицательных значений аналоговые выходы принимают значение подстановки для режима STOP.

Таблица А- 163 Представление аналогового выхода для тока (SB и SM)

Система		Диапазон выходного тока		
Десятичная	Шестнадцатер.	от 0 до 20 мА	от 4 до 20 мА	
32767	7FFF	Смотри примечание 1	Смотри примечание 1	Переполнение для положительных значений
32512	7F00	Смотри примечание 1	Смотри примечание 1	
32511	7EFF	23.52 мА	22.81 мА	Диапазон превышения для положительных значений
27649	6C01			
27648	6C00	20 мА	20 мА	Номинальный диапазон
20736	5100	15 мА	16 мА	
1	1	723.4 нА	4 мА + 578.7 нА	
0	0	0 мА	4 мА	
-1	FFFF		4 мА ... 578.7 нА	Диапазон превышения для отрицательных значений
-6912	E500		0 мА	
-6913	E4FF			Невозможно. Выходное значение ограничено 0 мА.
-32512	8100			
-32513	80FF	Смотри примечание 1	Смотри примечание 1	Переполнение для отрицательных значений
-32768	8000	Смотри примечание 1	Смотри примечание 1	

¹ В состоянии переполнения для положительных или отрицательных значений аналоговые выходы принимают значение подстановки для режима STOP.

A.9. Сигнальные модули для термопар и термосопротивлений (SM)

A.9.1. SM 1231 Thermocouple

Таблица A- 164 Общие технические данные

Модель	SM 1231 AI 4 x 16 bit TC	SM 1231 AI 8 x 16 bit TC
Номер для заказа	6ES7 231-5QD32-0XB0	6ES7 231-5QF32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	
Вес	180 грамм	190 грамм
Рассеиваемая мощность	1.5 Вт	
Потребление тока (SM шина)	80 мА	
Потребление тока (24 В пост.т.) ¹	40 мА	

¹ от 20.4 до 28.8 В пост.т. (Класс 2, ограниченное питание или питание датчиков от ПЛК)

Таблица A- 165 Аналоговые входы

Модель	SM 1231 AI 4 x 16 bit TC	SM 1231 AI 8 x 16 bit TC
Количество входов	4	8
Диапазон Номинальный диапазон (слово данных) Диапазон превышения полож./отриц. значений (слово данных) Диапазон переполнения полож./отриц. значений (слово данных)	Смотри таблицу выбора термопары (стр. 1200).	
Разрешение	Температура	0.1 °C/0.1 °F
	Напряжение	15 бит плюс знак
Максимальная прочность по напряжению	± 35 В	
Подавление помех	85 дБ для выбранной настройки фильтра (10 Гц, 50 Гц, 60 Гц или 400 Гц)	
Подавление синфазной помехи	> 120 дБ при 120 В пер.т.	
Полное сопротивление	≥ 10 МОм	
Электрическая развязка	Полевая сторона относительно логики	500 В пер.т.
	Полевая сторона относительно 24 В пост.т.	500 В пер.т.
	24 В пост.т. относительно логики	500 В пер.т.
Канал относительно канала	120 В пер.т.	
Точность	Смотри таблицу выбора термопары (стр. 1200).	
Повторяемость	±0.05% диапазон	
Принцип измерения	Интегрирование	
Время обновления модуля	Смотри таблицу выбора способа подавления помех (стр. 1200).	
Ошибка холодного спая	±1.5 °C	
Длина кабеля (метры)	100 метров до датчика макс.	
Сопротивление проводника	100 Ом макс.	

А.9 Сигнальные модули для термопар и термосопротивлений (SM)

Таблица А- 166 Диагностика

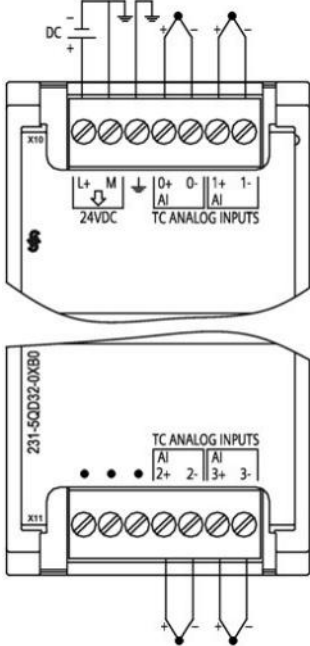
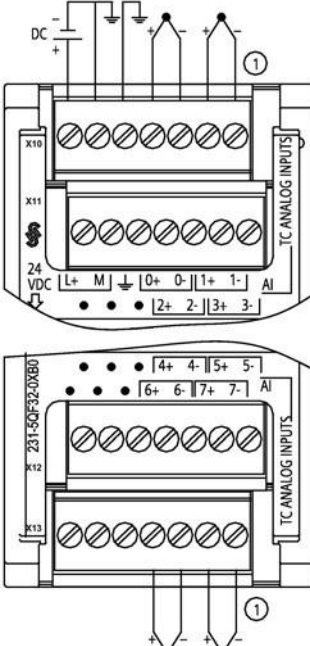
Модель	SM 1231 AI 4 x 16 bit TC	SM 1231 AI 8 x 16 bit TC
Переполнение для положительных / отрицательных значений ¹	Да	
Обрыв провода (только для тока) ²	Да	
Пониженное напряжение 24 В пост.т. ¹	Да	

- ¹ Диагностическая сигнальная информация о переполнении положительных и отрицательных значений и пониженном напряжении, будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.
- ² Когда тревога обрыва провода отключена, и присутствует обрыв в проводке датчика, модуль может выдавать случайные значения.

Аналоговый сигнальный модуль SM 1231 Thermocouple (TC) измеряет значение на-пряжения, поданного на входы модуля. Тип измерения температуры может быть или "Термопара" или "Напряжение".

- "Термопара": Значение будет передано в градусах, умноженных на десять (напри-мер, 25.3 градуса будет представлено как десятичное число 253).
- "Напряжение": Значением предела номинального диапазона будет десятичное 27648.

Таблица А- 167 Схема электрических соединений для SM с подключением термопар

SM 1231 AI 4 x TC 16 bit (6ES7 231-5QD32-0XB0)	SM 1231 AI 8 x TC bit (6ES7 231-5QF32-0XB0)
	
Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.	

① TC 2, 3, 4 и 5 не показаны подключенными для ясности.

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Таблица А- 168 Привязка контактов разъема для SM 1231 AI 4 x TC 16 bit (6ES7 231-5QD32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения
2	M / 24 В пост.т.	Нет соединения
3	Функциональная земля	Нет соединения
4	AI 0+ /TC	AI 2+ /TC
5	AI 0- /TC	AI 2- /TC
6	AI 1+ /TC	AI 3+ /TC
7	AI 1- /TC	AI 3- /TC

Таблица А- 169 Привязка контактов разъема для SM 1231 AI 8 x TC bit (6ES7 231-5QF32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	AI 0+ /TC	AI 2+ /TC	AI 4 I- /TC	AI 6 I- /TC
5	AI 0- /TC	AI 2- /TC	AI 4 I+ /TC	AI 6 I+ /TC
6	AI 1+ /TC	AI 3+ /TC	AI 5 M- /TC	AI 7 M- /TC
7	AI 1- /TC	AI 3- /TC	AI 5 M+ /TC	AI 7 M+ /TC

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

Неиспользуемые каналы для термопар могут быть деактивированы. Деактивация неиспользуемых каналов не приводит к ошибкам

A.9.1.1. Базовый режим работы для термопар

Термопары образуются, когда два разнородных металла электрически соединены друг с другом. Вырабатывается напряжение, которое пропорционально температуре спая. Это напряжение мало; один микровольт может представлять много градусов. Измерение напряжения от термопары, компенсация дополнительных спаев, и последующая линеаризация результата являются основанием для измерения температуры с помощью термопар.

Когда Вы подключаете термопару к модулю SM 1231 Thermocouple, два разнородных металлических провода подключены к модулю посредством сигнального соединителя модуля. Место, где два разнородных провода присоединены друг к другу, образует термопару датчика.

Еще две термопары образуются, когда два разнородных провода подключены к сигнальному соединителю. Температура соединителя вызывает напряжение, которое добавляется к напряжению от термопары датчика. Если это напряжение не скорректировано, то передаваемая температура, будет отличаться от температуры датчика.

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Компенсация холодного спая используется для корректировки влияния термопары соединителя. Таблицы термопар основаны на справочной температуре соединения, обычно для нуля градусов по Цельсию. Компенсация холодного спая дает корректировку соединителю до нуля градусов по Цельсию. Компенсация холодного спая восстанавливает напряжение, измененное термопарами соединителя. Температура модуля измеряется внутренне, и затем преобразуется в значение, которое будет добавлено к значению от датчика. Затем исправленное значение датчика линеаризуется, используя таблицы термопар.

Для оптимальной работы компенсации холодного спая модуль термопары должен быть расположен в термически стабильной окружающей среде. Медленное изменение (менее чем 0.1 °C/мин) в окружении модуля правильно компенсируется согласно техническим требованиям модуля. Движение воздуха через модуль также приведет к ошибкам компенсации холодного спая.

Если необходима наилучшая компенсация ошибки холодного спая, может использоваться внешняя изотермическая клеммная колодка. Модуль термопары предусматривает использование клеммной колодки с эталонной температурой 0 °C или 50 °C.

A.9.1.2. Таблицы выбора для SM 1231 с подключением термопар

Диапазоны и точность для различных типов термопар, поддерживаемых сигнальным модулем SM 1231 Thermocouple, представлены в таблице ниже.

Таблица A- 170 Таблица выбора термопары для SM 1231

Тип	Минимум превышения отриц.знач. ¹	Нижний предел номинального диапазона	Верхний предел номинального диапазона	Максимум превышения полож.знач. ²	Точность в нормальном диапазоне ^{3,4} @ 25 °C	Точность в нормальном диапазоне ^{1,2} от -20 °C до 60 °C
J	-210.0 °C	-150.0 °C	1200.0 °C	1450.0 °C	±0.3 °C	±0.6 °C
K	-270.0 °C	-200.0 °C	1372.0 °C	1622.0 °C	±0.4 °C	±1.0 °C
T	-270.0 °C	-200.0 °C	400.0 °C	540.0 °C	±0.5 °C	±1.0 °C
E	-270.0 °C	-200.0 °C	1000.0 °C	1200.0 °C	±0.3 °C	±0.6 °C
R & S	-50.0 °C	100.0 °C	1768.0 °C	2019.0 °C	±1.0 °C	±2.5 °C
B	0.0 °C	200.0 °C	800.0 °C	--	±2.0 °C	±2.5 °C
	--	800.0 °C	1820.0 °C	1820.0 °C	±1.0 °C	±2.3 °C
N	-270.0 °C	-200.0 °C	1300.0 °C	1550.0 °C	±1.0 °C	±1.6 °C
C	0.0 °C	100.0 °C	2315.0 °C	2500.0 °C	±0.7 °C	±2.7 °C
TXK/XK(L)	-200.0 °C	-150.0 °C	800.0 °C	1050.0 °C	±0.6 °C	±1.2 °C
Напряжение	-32512	-27648 -80 мВ	27648 80 мВ	32511	±0.05%	±0.1%

¹ Значения от термопары ниже минимального превышения отрицательного значения передаются как -32768.

² Значения от термопары выше максимального превышения положительного значения передаются как 32767.

³ Внутренняя ошибка холодного спая составляет ±1.5 °C для всех диапазонов. Она добавляется к ошибке согласно этой таблице. Модуль требуется по крайней мере 30 минут разогрева, чтобы удовлетворять настоящей спецификации.

⁴ При наличии излучения в диапазоне от 970 МГц до 990 МГц точность SM 1231 AI 4 x 16 bit TC может ухудшиться.

Примечание**Канал для подключения термопары**

Каждый из каналов на сигнальном модуле термопар может конфигурироваться для различных типов термопар (можно выбрать в программном обеспечении во время конфигурации модуля).

Таблица A- 171 Подавление помех и время обновления для SM 1231 Thermocouple

Выбор частоты подавления	Время интегрирования	Время обновления 4-канального модуля (секунды)	Время обновления 8-канального модуля (секунды)
400 Гц (2.5 мс)	10 мс ¹	0.143	0.285
60 Гц (16.6 мс)	16.67 мс	0.223	0.445
50 Гц (20 мс)	20 мс	0.263	0.525
10 Гц (100 мс)	100 мс	1.225	2.450

- ¹ Для того, чтобы достичь заявленной точности и разрешения при фильтре 400 Гц, время интегрирования устанавливается равным 10 мс. Этот фильтр также отсекает помехи частот 100 Гц и 200 Гц.

Для измерительных термопар рекомендуется использовать время интегрирования 100 мс. Использование меньших значений времени интегрирования увеличит ошибку воспроизводимости регистрируемых значений температуры.

Примечание

После подачи питания модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль выдает значение 32767 на каждом канале, пока действительные данные не станут доступны на этом канале. Ваша пользовательская программа, возможно, должна допускать это время инициализации. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, Вы должны проверить поведение модуля в Вашей конфигурации. При необходимости Вы можете включать логику в свою пользовательскую программу, чтобы подстроиться под время инициализации модуля.

Представление аналоговых значений для термопары Тип J

Представление аналоговых значений для термопары типа J приведено в таблице ниже.

Таблица А- 172 Представление аналоговых значений для термопары типа J

Тип J в °C	Единицы		Тип J в °F	Единицы		Диапазон
	Десятичн.	Шестнадцат.		Десятичн.	Шестнадцат.	
> 1450.0	32767	7FFF	> 2642.0	32767	7FFF	Переполюсовка значений
1450.0	14500	38A4	2642.0	26420	6734	Превышение
:	:	:	:	:	:	
1200.1	12001	2EE1	2192.2	21922	55A2	Номинальный диапазон
1200.0	12000	2EE0	2192.0	21920	55A0	
:	:	:	:	:	:	Переполюсовка отриц. значений ¹
-150.0	-1500	FA24	-238.0	-2380	F6B4	
< -150.0	-32768	8000	< -238.0	-32768	8000	

¹ Неправильное подключение (к примеру, обратная полярность или разомкнутый вход) или ошибка датчика в диапазоне отрицательных температур (к примеру, неверный тип термопары) может стать причиной переполюсовки отрицательных значений в сигнальном модуле для подключения термопар.

A.9.2. SM 1231 RTD

Технические данные для SM 1231 RTD

Таблица А- 173 Общие технические данные

Модель	SM 1231 AI 4 x RTD x 16 bit	SM 1231 AI 8 x RTD x 16 bit
Номер для заказа	6ES7 231-5PD32-0XB0	6ES7 231-5PF32-0XB0
Размеры Ш x В x Г (мм)	45 x 100 x 75	70 x 100 x 75
Вес	220 грамм	270 грамм
Рассеиваемая мощность	1.5 Вт	
Потребление тока (SM шина)	80 мА	90 мА
Потребление тока (24 В пост.т.) ¹	40 мА	

¹ от 20.4 до 28.8 В пост.т. (Класс 2, ограниченное питание или питание датчиков от ПЛК)

Таблица А- 174 Аналоговые входы

Технические данные	SM 1231 AI 4 x RTD x 16 bit	SM 1231 AI 8 x RTD x16 bit
Количество входов	4	8
Тип	Рекомендованные для модуля термосопротивления и сопротивления	

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Технические данные		SM 1231 AI 4 x RTD x 16 bit	SM 1231 AI 8 x RTD x16 bit
Диапазон Номинальный диапазон (слово данных) Диапазон превышения полож./отриц. значений (слово данных) Диапазон переполнения полож./отриц. значений (слово данных)		Смотри таблицу выбора терморезистора (стр. 1205).	
Разрешение	Температура	0.1 °C/0.1 °F	
	Напряжение	15 бит плюс знак	
Максимальная прочность по напряжению		± 35 В	
Подавление помех		85 дБ для выбранной настройки фильтра (10 Гц, 50 Гц, 60 Гц или 400 Гц)	
Подавление синфазной помехи		> 120 дБ	
Полное сопротивление		≥ 10 МОм	
Электрическая развязка	Полевая сторона относительно логики	500 В пер.т.	
	Полевая сторона относительно 24 В пост.т.	500 В пер.т.	
	24 В пост.т. относительно логики	500 В пер.т.	
Канал относительно канала		нет	
Точность		Смотри таблицу выбора терморезистора (стр. 1205).	
Повторяемость		±0.05% диапазон	
Максимальная мощность рассеяния на датчике		0.5 мВт	
Принцип измерения		Интегрирование	
Время обновления модуля		Смотри таблицу выбора способа подавления помех (стр. 1205).	
Длина кабеля (метры)		100 метров до датчика макс.	
Сопротивление проводника		20 Ом, 2.7 Ом для терморезистора 10 Ом макс.	

Таблица A- 175 Диагностика

Технические данные	SM 1231 AI 4 x RTD x 16 bit	SM 1231 AI 8 x RTD x16 bit
Переполнение для положительных / отрицательных значений ^{1,2}	Да	
Обрыв провода ³	Да	
Пониженное напряжение 24 В пост.т. ¹	Да	

¹ Диагностическая сигнальная информация о переполнении положительных и отрицательных значений и пониженном напряжении, будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

² Для диапазонов сопротивлений никогда не активируется обнаружение переполнения отрицательных значений.

³ Когда тревога обрыва провода отключена, и есть обрыв провода датчика, модуль может выдавать случайные значения.

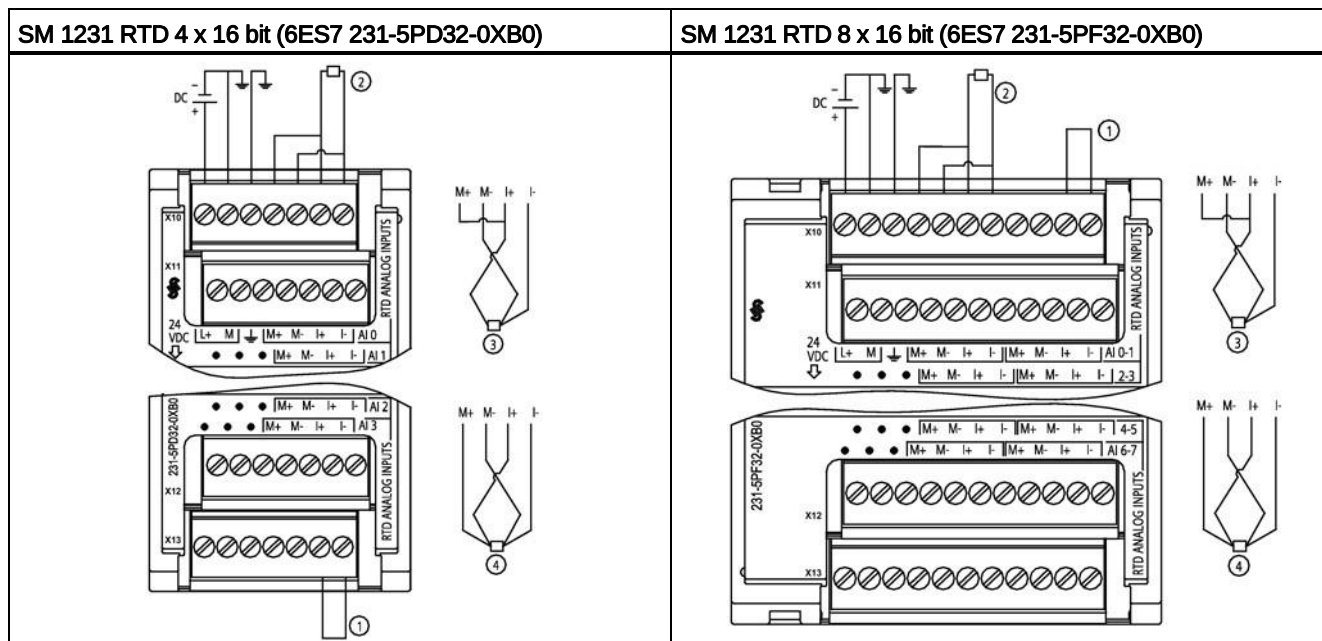
Аналоговый сигнальный модуль SM 1231 RTD измеряет значение сопротивления, подключенного ко входам модуля. Тип измерения температуры может быть или "Резистор" или "Терморезистор".

- "Резистор": Значением предела номинального диапазона будет десятичное 27648.
- "Терморезистор": Значения будут переданы в градусах, умноженных на десять (например, 25.3 градуса будут представлены как десятичное число 253). Значения климатического диапазона будут переданы в градусах, умноженных на сто (например, 25.34 градуса будут представлены как десятичное число 2534).

А.9 Сигнальные модули для термопар и термосопротивлений (SM)

Модуль SM 1231 RTD поддерживает измерения с помощью 2-проводного, 3-проводного и 4-проводного подключения датчика сопротивления.

Таблица А- 176 Схема электрических соединений для SM с подключением терморезисторов



① Замкните неиспользуемые входы терморезисторов

② 2-проводный терморезистор ③ 3- проводный терморезистор ④ 4- проводный терморезистор

ПРИМЕЧАНИЕ: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.

Таблица А- 177 Привязка контактов разъема для SM 1231 RTD 4 x 16 bit (6ES7 231-5PD32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	AI 0 M+ /RTD	AI 1 M+ /RTD	AI 2 M+ /RTD	AI 3 M+ /RTD
5	AI 0 M- /RTD	AI 1 M- /RTD	AI 2 M- /RTD	AI 3 M- /RTD
6	AI 0 I+ /RTD	AI 1 I+ /RTD	AI 2 I+ /RTD	AI 3 I+ /RTD
7	AI 0 I- /RTD	AI 1 I- /RTD	AI 2 I- /RTD	AI 3 I- /RTD

Таблица А- 178 Привязка контактов разъема для SM 1231 RTD 8 x 16 bit (6ES7 231-5PF32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
4	AI 0 M+ /RTD	AI 2 M+ /RTD	AI 4 M+ /RTD	AI 6 M+ /RTD
5	AI 0 M- /RTD	AI 2 M- /RTD	AI 4 M- /RTD	AI 6 M- /RTD
6	AI 0 I+ /RTD	AI 2 I+ /RTD	AI 4 I+ /RTD	AI 6 I+ /RTD
7	AI 0 I- /RTD	AI 2 I- /RTD	AI 4 I- /RTD	AI 6 I- /RTD
8	AI 1 M+ /RTD	AI 3 M+ /RTD	AI 5 M+ /RTD	AI 7 M+ /RTD
9	AI 1 M- /RTD	AI 3 M- /RTD	AI 5 M- /RTD	AI 7 M- /RTD
10	AI 1 I+ /RTD	AI 3 I+ /RTD	AI 5 I+ /RTD	AI 7 I+ /RTD
11	AI 1 I- /RTD	AI 3 I- /RTD	AI 5 I- /RTD	AI 7 I- /RTD

Примечание

Неиспользуемые каналы для термопар могут быть деактивированы. Деактивация неиспользуемых каналов не приводит к ошибкам.

Модуль для подключения терморезисторов должен иметь замкнутую токовую петлю, чтобы устранить дополнительное время стабилизации, которое автоматически добавляется к неиспользованному каналу, который не деактивирован. Для стабильности у модуля RTD должен иметь подключенный резистор (как в 2-проводном подключении терморезистора).

A.9.2.1. Таблицы выбора для SM 1231 RTD

Таблица A- 179 Диапазоны и точность для различных датчиков, поддерживаемых модулем RTD

Температурный коэффициент	Тип терморезистора	Минимум превышения отрицательного значения ¹	Нижний предел номинального диапазона	Верхний предел номинального диапазона	Максимум превышения положительного значения ²	Точность в нормальном диапазоне @ 25 °C	Точность в нормальном диапазоне от -20 °C до 60 °C
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 climatic	-145.00 °C	-120.00 °C	145.00 °C	155.00 °C	±0.20 °C	±0.40 °C
	Pt 10	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±1.0 °C	±2.0 °C
	Pt 50	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	± 0.5 °C	±1.0 °C
	Pt 200	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	± 0.5 °C	±1.0 °C
	Pt 500						
	Pt 1000						
Pt 0.003910	Pt 10	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±1.0 °C	±2.0 °C
	Pt 50	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±0.8 °C	±1.6 °C

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Температурный коэффициент	Тип терморезистора	Минимум превышения отрицательного знач. ¹	Нижний предел номинального диапазона	Верхний предел номинального диапазона	Максимум превышения положительного знач. ²	Точность в нормальном диапазоне @ 25 °C	Точность в нормальном диапазоне от -20 °C до 60 °C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
Ni 0.006170	Ni 100	-105.0 °C	-60.0 °C	180.0 °C	212.4 °C	±0.5 °C	±1.0 °C
Cu 0.004270	Cu 10	-240.0 °C	-200.0 °C	260.0 °C	312.0 °C	±1.0 °C	±2.0 °C
Cu 0.004260	Cu 10	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±0.6 °C	±1.2 °C
	Cu 100						
Cu 0.004280	Cu 10	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±0.7 °C	±1.4 °C
	Cu 100						

¹ Значения от терморезистора ниже минимального превышения отрицательного значения передаются как -32768.

² Значения от терморезистора выше максимального превышения положительного значения передаются как 32767.

Таблица A- 180 Сопротивление

Диапазон	Минимум превышения отрицательн. значения	Нижний предел номин. диапазона	Верхний предел номинального диапазона	Максимум превышения положительн. значения ¹	Точность в нормальном диапазоне @ 25 °C	Точность в нормальном диапазоне от -20 °C до 60 °C
150 Ом	неприменимо	0 (0 Ом)	27648 (150 Ом)	176.383 Ом	±0.05%	±0.1%
300 Ом	неприменимо	0 (0 Ом)	27648 (300 Ом)	352.767 Ом	±0.05%	±0.1%
600 Ом	неприменимо	0 (0 Ом)	27648 (600 Ом)	705.534 Ом	±0.05%	±0.1%

¹ Значения сопротивления выше максимального превышения положительного значения передаются как +32767.

Примечание

Модуль выдает 32767 для любого активированного канала без подключенного датчика. Если также разрешено обнаружение обрыва провода, модуль активирует соответствующие красные светодиоды.

Когда используются диапазоны терморезисторов 500 и 1000 Ом с другими резисторами меньшей величины, ошибка может увеличиться до двух раз от указанного значения ошибки.

Для терморезисторов диапазона 10 Ом наилучшая точность будет достигнута при использовании 4-проводного подключения.

Сопротивление соединительных проводов при 2-проводном способе вызовет ошибку интерпретации показаний датчика, и поэтому точность не гарантирована.

A.9 Сигнальные модули для термопар и термосопротивлений (SM)

Таблица A- 181 Шумоподавление и время отклика для модулей подключения терморезисторов

Выбор частоты подавления	Время интегрирования	Время обновления (секунды)	
		4-канальный модуль	8-канальный модуль
400 Гц (2.5 мс)	10 мс ¹	4-/2-пров.: 0.142 3-пров.: 0.285	4-/2-пров.: 0.285 3-пров.: 0.525
60 Гц (16.6 мс)	16.67 мс	4-/2-пров.: 0.222 3-пров.: 0.445	4-/2-пров.: 0.445 3-пров.: 0.845
50 Гц (20 мс)	20 мс	4-/2-пров.: 0.262 3-пров.: .505	4-/2-пров.: 0.524 3-пров.: 1.015
10 Гц (100 мс)	100 мс	4-/2-пров.: 1.222 3-пров.: 2.445	4-/2-пров.: 2.425 3-пров.: 4.845

- ¹ Для того, чтобы достичь заявленной точности и разрешения при фильтре 400 Гц, время интегрирования устанавливается равным 10 мс. Этот фильтр также отсекает помехи частот 100 Гц и 200 Гц.

Примечание

После подачи питания модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль выдает значение 32767 на каждом канале, пока действительные данные не станут доступны на этом канале. Ваша пользовательская программа, возможно, должна допускать это время инициализации. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, Вы должны проверить поведение модуля в Вашей конфигурации. При необходимости Вы можете включать логику в свою пользовательскую программу, чтобы подстроиться под время инициализации модуля.

Представление аналоговых значений для терморезисторов

Представление оцифрованного измеренного значения терморезисторов стандартных температурных диапазонов приведено в таблице ниже.

Таблица A- 182 Представление аналоговых значений для термометров сопротивлений
PT 100 200, 500, 1000 и PT 10, 50, 100, 500 ГОСТ (0.003850) стандарт

Pt x00 стандарт в °C (1 цифра = 0.1 °C)	Единицы		Pt x00 стандарт в ° F (1 цифра = 0.1 ° F)	Единицы		Диапазон
	Десятичн.	Шестнадцат.		Десятичн.	Шестнадцат.	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	Переполнение полож. значений
1000.0	10000	2710	1832.0	18320	4790	Превышение положит. значений
: 850.1	: 8501	: 2135	: 1562.1	: 15621	: 3D05	
850.0	8500	2134	1562.0	15620	3D04	Номинальный диапазон
: -200.0	: -2000	: F830	: -328.0	: -3280	: F330	

Pt x00 стандарт в °C (1 цифра = 0.1 °C)	Единицы		Pt x00 стандарт в ° F (1 цифра = 0.1 ° F)	Единицы		Диапазон
	Десятичн.	Шестнадцат.		Десятичн.	Шестнадцат.	
-200.1 :	-2001 :	F82F :	-328.1 :	-3281 :	F32F :	Превышение отрицат. значений
-243.0	-2430	F682	-405.4	-4054	F02A	
< -243.0	-32768	8000	< -405.4	-32768	8000	Переполнение отриц. значений

А.10. Технологические модули

А.10.1. SM 1278 4xIO-Link Master

Таблица А- 183 Общие технические данные

Технические характеристики		Сигнальный модуль SM 1278 4xIO-Link Master
Номер для заказа		6ES7 278-4BD32-0XB0
Размеры Ш x В x Г (мм)		45 x 100 x 75
Вес		150 грамм
Общая информация		
	I&M данные	Да; от IM0 до IM3
Напряжение питания		
	Номинальное напряжение (пост.т.)	24 В пост.т.
	Нижний предел допустимого диапазона (пост.т.)	19.2 В; 20.5 В если используется IO-Link (напряжение питания для устройств IO-Link на ведущем устройстве должно составлять по крайней мере 20 В)
	Верхний предел допустимого диапазона (пост.т.)	28.8 пост.т.
	Защита от обратной полярности	Да
Входной ток		
	Потребление тока	65 мА; без нагрузки
Питание энкодера		
	Количество выходов	4
	Выходной ток, номинальное значение	200 мА
Потери мощности		
	Потери мощности, типовое значение.	1 Вт, исключая нагрузку порта
Цифровые входы/выходы		
	Длина кабеля (метры)	20 м, неэкранированный, макс.
SDLC		
	Длина кабеля (метры)	20 м, неэкранированный, макс.
IO-Link		
	Количество портов	4

Технические характеристики		Сигнальный модуль SM 1278 4xIO-Link Master
	Кол-во портов, которые могут управляться одновр.	4
	IO-Link протокол 1.0	Да
	IO-Link протокол 1.1	Да
Режим работы		
	IO-Link	Да
	DI	Да
	DQ	Да; макс. 100 мА
Подключение IO-Link устройств		
	Порт типа А	Да
	Скорость передачи	4.8 кБод (COM1)
		38.4 кБод (COM2)
		230.4 кБод (COM3)
	Время цикла, мин.	2 мс, динамически, в зависимости от длины пользовательских данных
	Объем технологических данных, входов на порт	32 байта; макс.
	Объем технологических данных, входов на модуль	32 байта
	Объем технологических данных, выходов на порт	32 байта; макс.
	Объем технологических данных, выходов на модуль	32 байта
	Объем памяти для параметров устройства	2 кбайт
	Длина незэкранированного кабеля, макс. (метры)	20 м
Информация о прерываниях/диагностике/состоянии		
	Индикация состояния	Да
Прерывания		
	Диагностическое прерывание	Да; диагностика порта доступна только в режиме IO-Link
Диагностические тревоги		
	Диагностика	
	Контроль напряжения питания	Да
	Короткое замыкание	Да
Светодиод диагностики		
	Контроль напряжения питания	Да; мигающий красный светодиод DIAG
	Индикация состояния канала	Да; один зеленый светодиод Qn на канал для состояния канала (SIO режим) и Sp для состояния порта (IO-Link режим)
	Для диагностики канала	Да; красный светодиод Fn
	Для диагностики модуля	Да; зеленый/красный светодиод DIAG
Электрическая развязка		
	Электрическая развязка каналов	
	Между каналами	Нет
	Между каналами и внутренней шиной	Да
Допустимая разность потенциалов		
	Между различными цепями	75 В пост.т./ 60 В пер.т. (основная изоляция)
Изоляция		
	Изоляция протестирована с помощью	707 В пост.т. (типовой тест)

Технические характеристики		Сигнальный модуль SM 1278 4xIO-Link Master
Окружающие условия		
	Рабочая температура	
	Мин.	-20 °C
	Макс.	60 °C
	Горизонтальная установка, мин.	-20 °C
	Горизонтальная установка, макс.	60 °C
	Вертикальная установка, мин.	-20 °C
	Вертикальная установка, макс.	50 °C

Обзор времени отклика

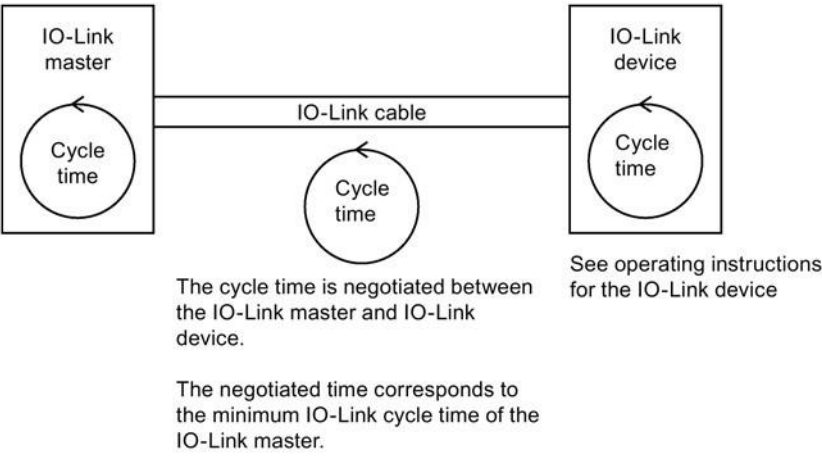


Таблица А- 184 Схема электрических соединений для сигнального модуля SM 1278 IO-Link Master

SM 1278 IO-Link Master (6ES7 278-4BD32-0XB0)	

Таблица А- 185 Привязка контактов разъема для SM 1278 IO-Link Master (6ES7 278-4BD32-0XB0)

Контакт	X10 (с позолотой)	X11 (с позолотой)	X12 (с позолотой)	X13 (с позолотой)
1	L+ / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
2	M / 24 В пост.т.	Нет соединения	Нет соединения	Нет соединения
3	Функциональная земля	Нет соединения	Нет соединения	Нет соединения
4	Нет соединения	Нет соединения	Нет соединения	Нет соединения
5	L ₁	L ₂	L ₃	L ₄
6	C/Q ₁	C/QL ₂	C/Q ₃	C/QL ₄
7	ML ₁	ML ₂	M ₃	ML ₄

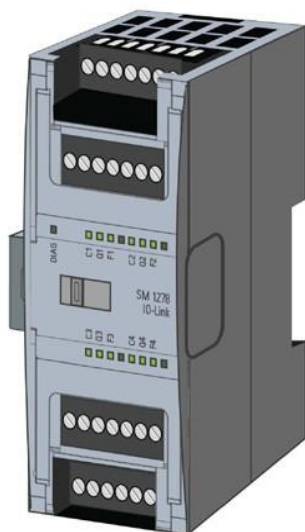
А.10.1.1. Обзор модуля SM 1278 4xIO-Link Master

SM 1278 4xIO-Link Master - это модуль с 4 портами, который работает и как сигнальный модуль, и как коммуникационный модуль. Каждый порт может работать в IO-Link режиме, как одиночный цифровой вход 24 В пост.т. или цифровой выход на 24 В пост.т.

Ведущее устройство IO-Link программирует ациклический обмен с IO-Link устройством, используя функциональный блок (FB) IOL_CALL в Вашей STEP 7 программе контроллера S7-1200. IOL_CALL FB указывает какое ведущее устройство IO-Link использует Ваша программа, и какие порты использует ведущее устройство для обмена данными.

Посетите веб-сайт промышленной он-лайн поддержки Сименс (<http://support.automation.siemens.com>) для получения более подробных сведений относительно работы с IOL_CALL FB. Введите "IO-Link" в поисковой строке веб-сайта, чтобы получить доступ к информации об IO-Link продуктах и их использовании.

Вид модуля



Свойства

Технические свойства

- Ведущее устройство IO-Link согласно IO-Link спецификациям V1.1 (смотри веб-сайт объединения IO-Link (<http://io-link.com/en/index.php>) для получения более подробных сведений)
- Последовательный коммуникационный модуль с четырьмя портами (каналами)
- Скорость обмена данными COM1 (4.8 кБод), COM2 (38.4 кБод), COM3 (230.4 кБод)
- SIO режим (стандартный режим ввода-вывода)
- Подключение до четырех устройств IO-Link (3-проводное соединение) или четырех стандартных исполнительных устройств или стандартных энкодеров
- Программируемая функция диагностики портом

Поддерживаемые функции

- I&M (инсталляция и обслуживание) идентификационные данные
- Обновление встроенного ПО
- Назначение IO-Link параметров посредством инструмента конфигурации S7-PCT порта, STEP 7 Professional и ЦПУ S7-1200 V4.0 или более поздней версии

IO-Link это соединение точка-к-точке между ведущим и ведомым устройством. Как обычные, так и интеллектуальные датчики/исполнительные механизмы могут использоваться в качестве устройств на IO-Link через неэкранированные стандартные кабели, используя проверенную 3-проводную технологию. IO-Link обратно совместима с обычными цифровыми датчиками и исполнительными механизмами. Структура схемы и канал данных разработаны согласно испытанной технологии 24 В пост.т.

Для получения дополнительной информации относительно технологии SIMATIC IO-Link обратитесь к "Руководству по функционированию IO-Link системы" на веб-сайте промышленной он-лайн поддержки Сименс (<http://support.automation.siemens.com>).

Примечание

Данные параметров IO-Link

Когда Вы заменяете SM 4xIO-Link Master, данные параметров не назначаются ему автоматически.



ВНИМАНИЕ

Удаление и вставка

Если Вы вставляете SM 4xIO-Link Master с включенной нагрузкой, это может привести к созданию опасных условий на Вашем производстве.

В результате может произойти физическое повреждение в системе автоматизации S7-1200.

Удаляйте или вставляйте SM 4xIO-Link Master только при выключенной нагрузке.

Эффект сброса на заводские установки

Используйте функцию "Сброс на заводские установки", чтобы восстановить назначение параметров, выполненное Вами с помощью S7-PCT к состоянию при поставке.

После "Сброса на заводские установки" параметры SM 1278 4xIO-Link модуля будут назначены следующим образом:

- Порты находятся в режиме DI
- Порты привязаны к относительным адресам от 0.0 до 0.3
- PortQualifier отменен
- Данные обслуживания с 1 до 3 удалены

Примечание

Когда Вы выполняете сброс на заводские установки, параметры устройства удаляются, и состояние при поставке восстанавливается.

Если Вы удаляете сигнальный модуль SM 1278 4xIO-Link, выполните сброс на заводские установки прежде, чем поместить его на хранение.

Процедура

Для "сброса на заводские установки" поступайте как описано в интерактивной справке для S7-PCT на вкладке "Master Configuration > 'Commands'".

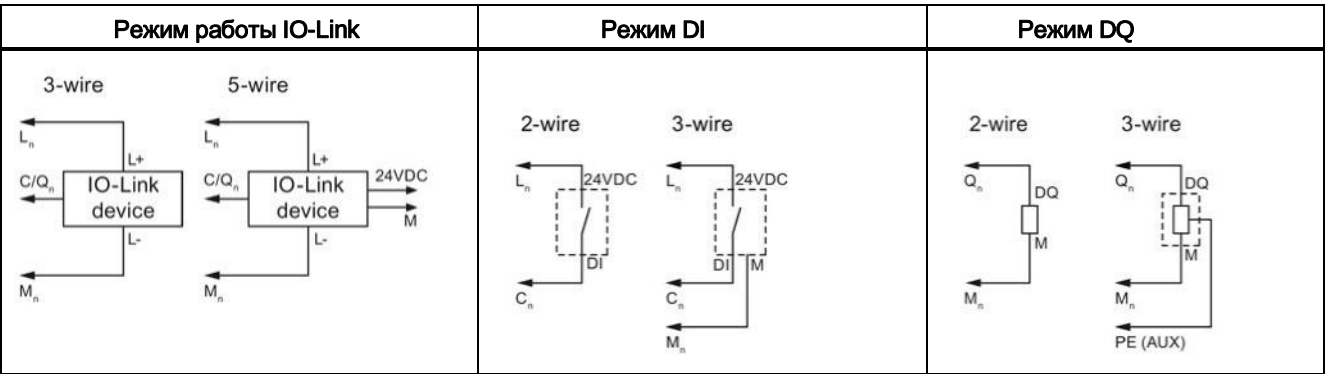
А.10.1.2. Подключение

Для получения подробных сведений относительно привязки контактов смотри таблицу Привязка контактов разъема для SM 1278 IO-Link Master (6ES7 278-4BD32-0XB0) (стр. 1208).

Следующая таблица показывает привязку контактов для SM 1278 4xIO-Link Master:

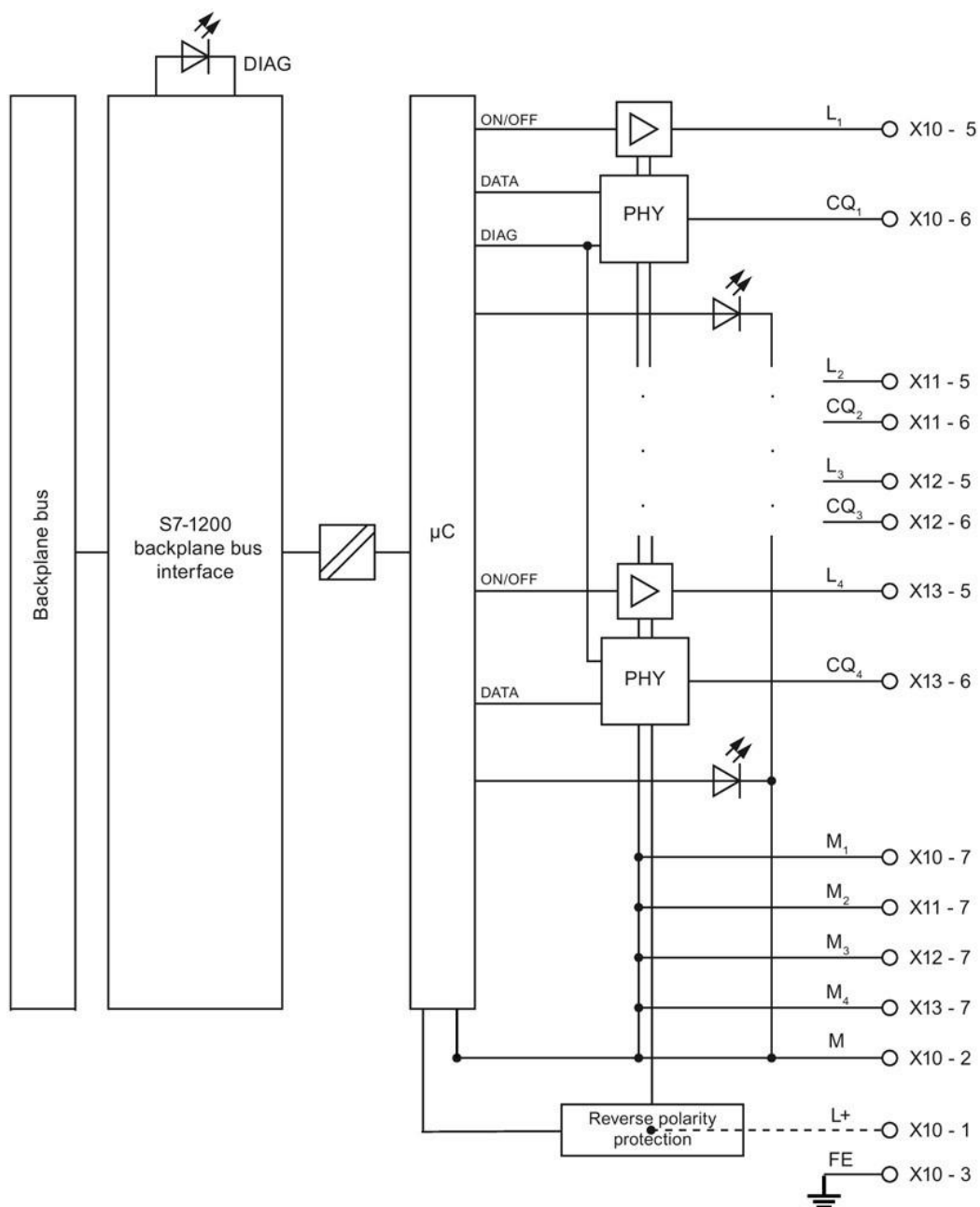
Кон- такт	X10	X11	X12	X13	Примечания	Базовые блоки
7	M ₁	M ₂	M ₃	M ₄	- M_n: земля к ведомому - C/Q_n: SDLC, DI или DQ - L_n: 24 В пост.т. к ведомому - M: земля - L+: 24 В пост.т. к ведущему - RES: зарезервирован; может не подключаться	A1
6	C/Q ₁	C/Q ₂	C/Q ₃	C/Q ₄		
5	L ₁	L ₂	L ₃	L ₄		
4	RES	RES	RES	RES		
3	 (функц.земля)	RES	RES	RES		
2	M	RES	RES	RES		
1	L+	RES	RES	RES		

Следующая таблица содержит иллюстрации примеров подключения, где n=номер порта:



Примечание

Подключенные датчики должны использовать источник питания посредством соединения L_n с ведущим модулем.



А.10.1.3. Параметры / адресное пространство

Конфигурирование SM 1278 4xIO-Link Master

Для интеграции модуля Вам нужен инструмент разработки TIA Portal V13 или выше от Сименс. Вам также нужен S7-PCT V3.2 или выше для интеграции IO-Link.

При вводе в эксплуатацию Вам требуется как инженерный пакет, так и S7-PCT V3.2 или выше для назначения параметров.

В следующей таблице представлены параметры для SM 1278 4xIO-Link Master:

Параметры	Диапазон значений	По умолчанию	Конфигурирование в RUN	Диапазон эффективности
Diagnostics port 1	- Disable - Enable	Disable	Да	Порт (канал)
Diagnostics port 2	- Disable - Enable	Disable	Да	Порт (канал)
Diagnostics port 3	- Disable - Enable	Disable	Да	Порт (канал)
Diagnostics port 4	- Disable - Enable	Disable	Да	Порт (канал)

Параметр для разрешения диагностики портов с 1 по 4

Этот параметр позволяет активировать диагностику для конкретных портов из имеющихся четырех портов IO-Link. Назначения порта следующие:

Port 1 → channel 1

Port 2 → channel 2

Port 3 → channel 3

Port 4 → channel 4

Максимальный размер адресов входа и выхода модуля SM 4xIO-Link Master составляет 32 байта в каждом случае. Вы назначаете адресные пространства, используя инструмент конфигурирования порта S7-PCT.

Запись данных параметра

Назначение параметров в пользовательской программе

Вы можете конфигурировать устройство во времени работы.

Изменение параметров во время работы

Параметры модуля включены в запись данных 128. Вы можете передать модифицируемые параметры в модуль с помощью инструкции WRREC.

Когда Вы перезагружаете (цикл включения-выключения) ЦПУ, он переписывает параметры, которые были переданы в модуль инструкцией WRREC во время параметризации.

Инструкция для назначения параметров

Следующая инструкция предлагается для назначения параметров модулю ввода / вывода в пользовательской программе:

Инструкция	Применение
SFB 53 WRREC	Передача изменяемых параметров в модуль.

Сообщение об ошибке

Следующее возвращаемое значение передается в сообщении об ошибке::

Код ошибки	Значение
80B1 _n	Ошибка длины данных
80E0 _n	Ошибка в заголовке информации
80E1 _n	Ошибка параметра

Структура записи данных

Следующая таблица показывает параметры IO-Link:

Смещение	Метка	Тип	По умолчанию	Описание
0	Version	1 байт	0x02	Показывает структуру записи 0x02 для IO-Link Master в соответствии с IO-Link V1.1
1	Parameter length	1 байт	0x02	Длина параметра (2 байта + 2 заголовка)
Стартовые параметры IO-Link				
2	Port diagnostics (Port1 от 1 до n)	1 байт	0x00	Активация диагностики для портов с 1 до n
3	IOL properties	1 байт	0x00	Свойства модуля

В следующей таблице представлена версия из записи данных:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Зарезервировано		Старшая версия (00)		Дополнительная версия (0010)			

В следующей таблице представлена диагностика порта из записи данных:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Зарезервировано				EN_Port4	EN_Port3	EN_Port2	EN_Port1

EN_Portx:

0 = Диагностика деактивирована

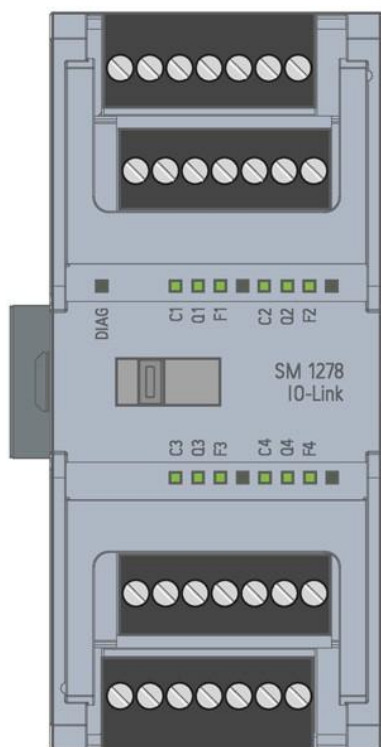
1 = Диагностика активирована

В следующей таблице представлены IOL свойства:

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Зарезервировано							

A.10.1.4. Прерывание, ошибка и системные тревоги

Светодиодный дисплей



Значение светодиодных индикаторов

Следующая таблица поясняет значение индикаторов ошибок и состояния. Вы можете найти меры по устранению для диагностических тревог в разделе "Диагностические тревоги".

Светодиод DIAG

DIAG	Значение
 Выкл.	Питание внутренней шины S7-1200 не в порядке.
 Мигает	Модуль не сконфигурирован
 Вкл.	Параметры модуля заданы и отсутствуют условия для срабатывания диагностики модуля
 Мигает	Параметры модуля заданы и сработала диагностика модуля или питание L+ не подключено

Светодиод состояния порта

Действительно для IO-Link порта, находящегося в режиме IO-Link port.

COM/1 ... COM/4	Значение
 Выкл.	Порт деактивирован
 Мигает	Порт активирован, устройство не подключено или Порт не подключен к сконфигурированному устройству
 Вкл.	Порт активирован, устройство подключено

Светодиод состояния канала

Действительно для IO-Link порта, находящегося в режиме DI/Q.

DI/Q1 ... DI/Q4	Значение
 Выкл.	Технологический сигнал = 0
 Вкл.	Технологический сигнал = 1

Светодиод ошибки порта

F1 ... F4	Значение
 Выкл.	Нет ошибки
 Вкл.	Ошибка

Ошибки модуля индицируются как диагностика (статус модуля) только в IO-Link режиме.

Диагностическая тревога	Код ошибки (десятичн.)	STATUS (W#16#...)	Значение (код ошибки IO-Link)	IO-Link ведущ. устр-во	IO-Link ведом. устр-во
Короткое замыкание	1	1804	Короткое замыкание в технологических кабелях устройства IO-Link	X	
		7710	Короткое замыкание в IO устр-ве		X
Недонапряжение	2	5111 5112	Напряжение питания слишком мало		X
Перенапряжение	3	5110	Напряжение питания слишком велико		X
Перегрев	5	1805	Превышение температуры на ведущем устройстве	X	
		4000 4210	Превышение температуры на устройстве		X
Обрыв провода	6	1800	- IO-Link устр-во не подключено - Обрыв сигнального провода IO-Link устройства - IO-Link устройство не способно к обмену из-за другой ошибки	X	
Переполнение полож. значений	7	8C10 8C20	Выход за пределы максимального значения технологического тэга		X
		8C20	Превышен диапазон измерения		
Переполнение отриц. значений	8	8C30	Выход за пределы минимального значения технологического тэга		X
Ошибка	9	---	Все коды ошибок IO-Link, не перечисленные здесь отображаются в эту PROFIBUS DP ошибку		X
Ошибка назначения параметра	16	1882 1883	Ведущее устройство IO-Link нельзя сконфигурировать	X	
		1802	Некорректное устройство		
		1886	Ошибка сохранения		
		6320 6321 6350	Устройство было сконфигурировано неправильно		X
Отсутствие напряжения питания	17	1806	Отсутствует напряжение питания устройства L+	X	
		1807	Напряжение питания устройства L+ слишком мало (<20 В)		

Диагностическая тревога	Код ошибки (десятичн.)	STATUS (W#16#...)	Значение (код ошибкиIO-Link)	IO-Link ведущ. устр-во	IO-Link ведом. устр-во
Неисправен предохран.	18	5101	Предохранитель устройства неисправен		X
Защитное выключение	25	1880	Серьезная ошибка (требуется замена ведущего устройства)	X	
Внешняя ошибка	26	1809 180A 180B 180C 180D	Ошибка в хранении данных	X	
		1808	Больше 6 ошибок возникают одновременно на устройстве IO-Link		

A.11. Цифровые сигнальные платы (SB)

A.11.1. Технические характеристики платы цифровых входов SB 1221 200 kHz

Таблица А- 186 Общие технические данные

Технические характеристики	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
Номер для заказа	6ES7 221-3BD30-0XB0	6ES7 221-3AD30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21	
Вес	35 грамм	
Рассеиваемая мощность	1.5 Вт	1.0 Вт
Потребление тока (SM шина)	40 мА	
Потребление тока (24 В пост.т.)	7 мА / вход + 20 мА	15 мА / вход + 15 мА

Таблица А- 187 Цифровые входы

Технические характеристики	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
Количество входов	4	
Тип	Источник	
Номинальное напряжение	24 В пост.т. при 7 мА, номинальное значение	5 В пост.т. при 15 мА, номинальное значение
Длительно допустимое напряжение	28.8 В пост.т.	6 В пост.т.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с	6 В пост.т.
Сигнал логической 1 (мин.)	L+ минус 10 В пост.т. при 2.9 мА	L+ минус 2.0 В пост.т. при 5.1 мА
Сигнал логического 0 (макс.)	L+ минус 5 В пост.т. при 1.4 мА	L+ минус 1.0 В пост.т. при 2.2 мА
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Одиочная фаза: 200 кГц Фаза со сдвигом на 90°: 160 кГц	
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин	
Потенциально развязанные группы	1	
Постоянные времени фильтра	us установки 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0	

A.11 Цифровые сигнальные платы (SB)

Технические характеристики		SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
	ms установки	0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0	
Количество одноврем.активных входов		• 2 (без смежных точек) при 60 °C для горизонтальной или 50 °C для вертикальной установки • 4 при 55 °C для горизонтальной или 45 °C для вертикальной установки	4
Длина кабеля (в метрах)		50 м экранированная витая пара	

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрите следующие варианты улучшения качества сигнала на входах:

- Минимизируйте длину кабеля
- Смените тип генератора с потребляющего ток на потребляющий/отдающий ток
- Выберите кабель лучшего качества
- Перейдите от схемы/компонентов 24 В на 5 В
- Подключите внешнюю нагрузку на входе

Таблица А- 188 Схема электрических соединений для платы цифровых входов 200 кГц

SB 1221 DI 4 x 24 VDC, 200 kHz (6ES7 221-3BD30-0XB0)	SB 1221 DI 4 x 5 VDC, 200 kHz (6ES7 221-3AD30-0XB0)
SB 1221 DC 200 KHz DI 4x24VDC 6ES7 221-3BD30-0XB0 24VDC	SB 1221 DC 200 KHz DI 4x5VDC 6ES7 221-3AD30-0XB0 5VDC

① Поддерживает только входы источники

Таблица А- 189 Привязка контактов разъема для SB 1221 DI 4 x 24 VDC, 200 kHz (6ES7 221-3BD30-0XB0)

Контакт	X19
1	L+ / 24 В пост.т.
2	M / 24 В пост.т.
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Таблица А- 190 Привязка контактов разъема для SB 1221 DI 4 x 5 VDC, 200 kHz (6ES7 221-3AD30-0XB0)

Контакт	X19
1	L+ / 5 В пост.т.
2	M / 5 В пост.т.
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

А.11.2. Технические характеристики платы цифровых выходов SB 1222 200 kHz

Таблица А- 191 Общие технические данные

Технические характеристики	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Номер для заказа	6ES7 222-1BD30-0XB0	6ES7 222-1AD30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21	
Вес	35 грамм	
Рассеиваемая мощность	0.5 Вт	
Потребление тока (SM шина)	35 мА	
Потребление тока (24 В пост.т.)	15 мА	

Таблица А- 192 Цифровые выходы

Технические характеристики	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Количество выходов	4	
Тип выхода	твердотельный – MOSFET потребитель и источник ¹	
Диапазон напряжения	от 20.4 до 28.8 В пост.т.	от 4.25 до 6.0 В пост.т.
Сигнал логической 1 при макс. токе	L+ минус 1.5 В	L+ минус 0.7 В
Сигнал логического 0 при макс. токе	1.0 В пост.т., макс.	0.2 В пост.т., макс.
Ток (макс.)	0.1 А	
Ламповая нагрузка	--	

A.11 Цифровые сигнальные платы (SB)

Технические характеристики	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
Сопротивление контакта в активном состоянии	11 Ом макс.	7 Ом макс.
Сопротивление в неактивном состоянии	6 Ом макс.	0.2 Ом макс.
Ток утечки на выход	--	
Частота пачки импульсов	200 кГц макс., 2 Гц мин.	
Ток включения	0.11 А	
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин.	
Потенциально развязанные группы	1	
Ток на общий провод (макс.)	0.4 А	
Индуктивное фиксирующее напряжение	Нет	
Задержка переключения	1.5 мкс + 300 нс нарастание 1.5 мкс + 300 нс спад	200 нс + 300 нс нарастание 200 нс + 300 нс спад
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	
Количество одноврем.активных выходов	2 (без смежных точек) при 60 °C для горизонтальной или 50 °C для вертикальной установки 4 при 55 °C для горизонтальной или 45 °C для вертикальной установки	4
Длина кабеля (метры)	50 м экранированная витая пара	

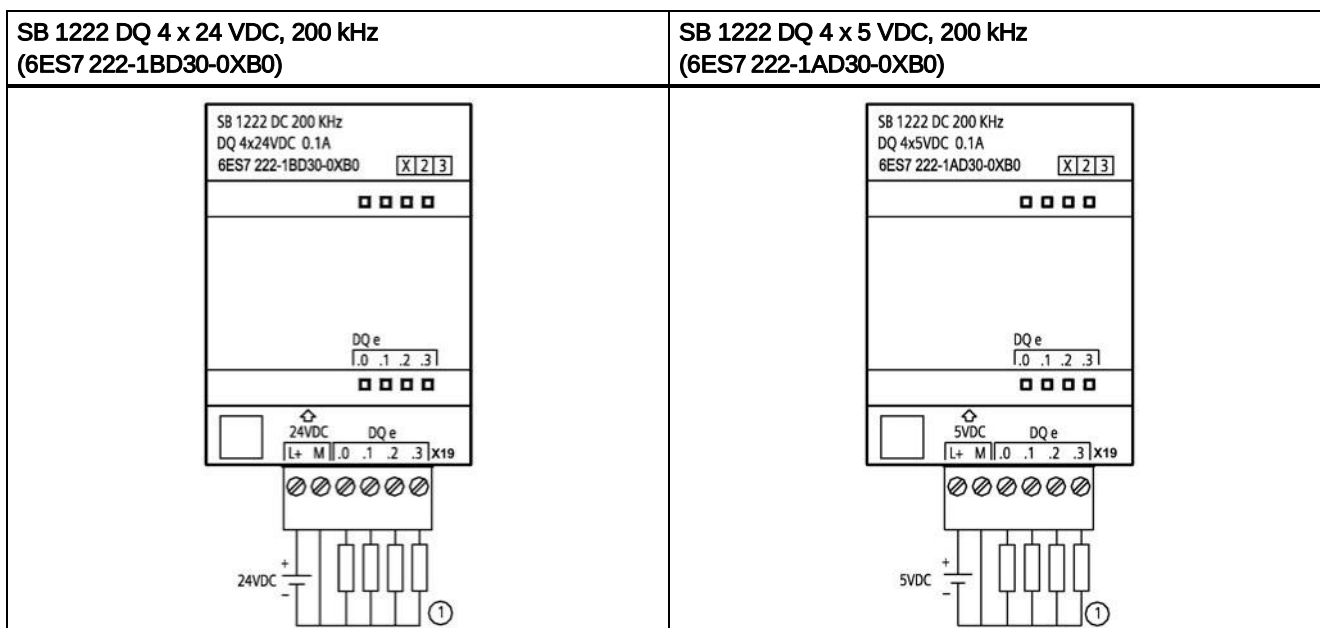
- ¹ Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузку), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузку). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрите следующие варианты улучшения качества сигнала на входах:

- Минимизируйте длину кабеля
- Смените тип генератора с потребляющего ток на потребляющий/отдающий ток
- Выберите кабель лучшего качества
- Перейдите от схемы/компонентов 24 В на 5 В
- Подключите внешнюю нагрузку на входе

Таблица А- 193 Схема электрических соединений для платы цифровых выходов 200 кГц



- ① Для отдающих ток выходов подключите "Нагрузку" к "-" (показано на рис.). Для потребляющих ток выходов подключите "Нагрузку" к "+". Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузку), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Таблица А- 194 Привязка контактов разъема для SB 1222 DQ 4 x 24 VDC, 200 kHz (6ES7 222-1BD30-0XB0)

Контакт	X19
1	L+ / 24 В пост.т.
2	M / 24 В пост.т.
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

Таблица А- 195 Привязка контактов разъема для SB 1222 DQ 4 x 5 VDC, 200 kHz (6ES7 222-1AD30-0XB0)

Контакт	X19
1	L+ / 5 В пост.т.
2	M / 5 В пост.т.
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

A.11.3. Технические характеристики платы цифровых входов / выходов SB 1223 200 kHz

Таблица А- 196 Общие технические данные

Технические характеристики	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Номер для заказа	6ES7 223-3BD30-0XB0	6ES7 223-3AD30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21	
Вес	35 грамм	
Рассеиваемая мощность	1.0 Вт	0.5 Вт
Потребление тока (SM шина)	35 мА	
Потребление тока (24 В пост.т.)	7 мА / вход + 30 мА	15 мА / вход + 15 мА

Таблица А- 197 Цифровые входы

Технические характеристики	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Количество входов	2	
Тип	Источник	
Номинальное напряжение	24 В пост.т. при 7 мА, номинальное	24 В пост.т. при 7 мА, номинальное
Длительно допустимое напряжение	28.8 В пост.т.	28.8 В пост.т.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с	35 В пост.т. в течение 0.5 с
Сигнал логической 1 (мин.)	L+ минус 10 В пост.т. при 2.9 мА	L+ минус 10 В пост.т. при 2.9 мА
Сигнал логического 0 (макс.)	L+ минус 5 В пост.т. при 1.4 мА	L+ минус 5 В пост.т. при 1.4 мА
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Одиночная фаза: 200 кГц Фаза со сдвигом на 90°: 160 кГц	
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин	
Потенциально развязанные группы	1 (нет развязки с выходами)	
Постоянные времени фильтра	us установки	0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0
	ms установки	0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 10.0, 12.8, 20.0
Количество одноврем.активных входов	2	
Длина кабеля (метры)	50, экранированная витая пара	

Таблица А- 198 Цифровые выходы

Технические характеристики	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Количество выходов	2	
Тип выхода	твердотельный – MOSFET потребитель и источник ¹	
Диапазон напряжения	от 20.4 до 28.8 В пост.т.	от 4.25 до 6.0 В пост.т.
Номинальное значение	24 В пост.т.	5 В пост.т.
Сигнал логической 1 при макс. токе	L+ минус 1.5 В	L+ минус 0.7 В
Сигнал логического 0 при макс. токе	1.0 В пост.т., макс.	0.2 В пост.т., макс.

Технические характеристики	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz
Ток (макс.)	0.1 A	
Ламповая нагрузка	--	
Сопротивление контакта в активном состоянии	11 Ом макс.	7 Ом макс.
Сопротивление в неактивном состоянии	6 Ом макс.	0.2 Ом макс.
Ток утечки на выход	--	
Частота пачки импульсов	200 кГц макс., 2 Гц мин.	
Ток включения	0.11 A	
Защита от перегрузки	Нет	
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин.	
Потенциально развязанные группы	1 (нет развязки со входами)	
Ток на общий провод (макс.)	0.2 A	
Индуктивное фиксирующее напряжение	Нет	
Задержка переключения	1.5 мкс + 300 нс нарастание 1.5 мкс + 300 нс спад	200 нс + 300 нс нарастание 200 нс + 300 нс спад
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)	
Количество одноврем.активных выходов	2	
Длина кабеля (метры)	50 м экранированная витая пара	

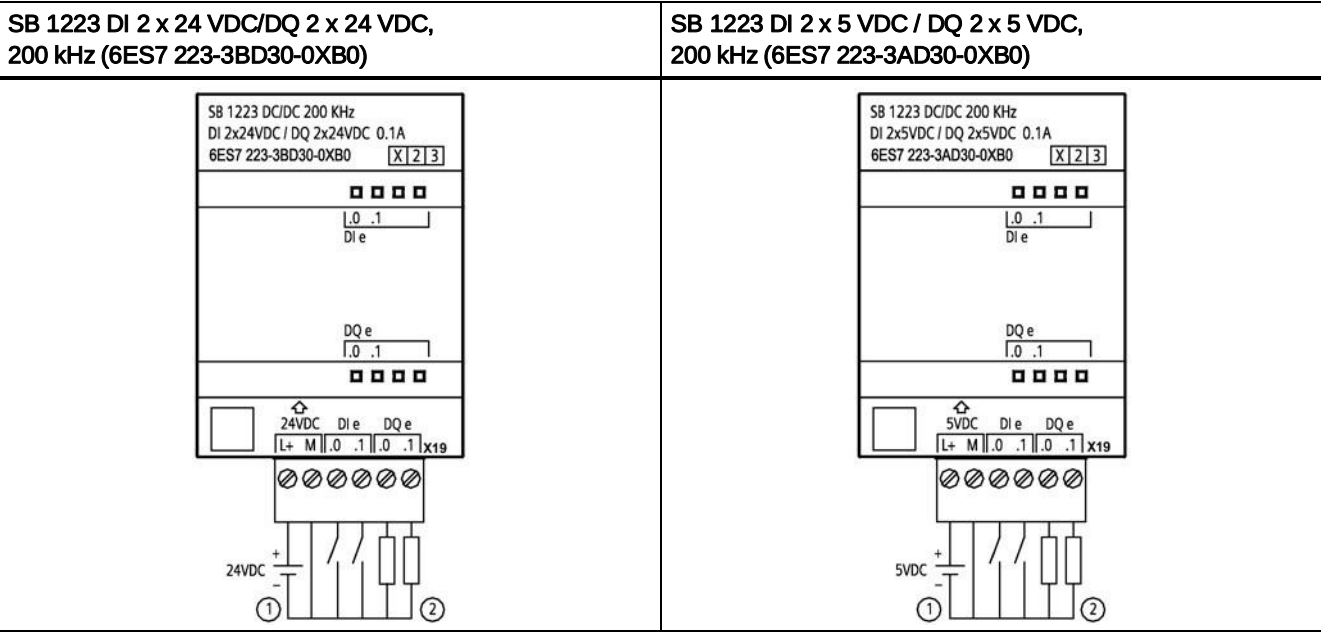
- ¹ Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузку), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрите следующие варианты улучшения качества сигнала на входах:

- Минимизируйте длину кабеля
- Смените тип генератора с потребляющего ток на потребляющий/отдающий ток
- Выберите кабель лучшего качества
- Перейдите от схемы/компонентов 24 В на 5 В
- Подключите внешнюю нагрузку на входе

Таблица А- 199 Схема электрических соединений для платы цифровых входов/выходов SB 200 кГц



- ① Поддерживает только входы-источники
- ② Для отдающих ток выходов подключите "Нагрузку" к "-" (показано на рис.). Для потребляющих ток выходов подключите "Нагрузку" к "+". Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузку), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Таблица А- 200 Привязка контактов разъема для SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC,
200 kHz (6ES7 223-3BD30-0XB0)

Контакт	X19
1	L+ / 24 В пост.т.
2	M / 24 В пост.т.
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

Таблица А- 201 Привязка контактов разъема для SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 kHz (6ES7 223- 3AD30-0XB0)

Контакт	X19
1	L+ / 5 В пост.т.
2	M / 5 В пост.т.
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

А.11.4. Технические характеристики платы 1223 с двумя входами 24 В пост.т. / двумя выходами 24 В пост.т.

Таблица А- 202 Общие технические данные

Технические характеристики	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Номер для заказа	6ES7 223-0BD30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	40 грамм
Рассеиваемая мощность	1.0 Вт
Потребление тока (SM шина)	50 мА
Потребление тока (24 В пост.т.)	4 мА / на используемый вход

Таблица А- 203 Цифровые входы

Технические характеристики	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Количество входов	2
Тип	МЭК Тип 1 потребитель
Номинальное напряжение	24 В пост.т. при 4 мА, номинальное значение
Длительно допустимое напряжение	30 В пост.т., макс.
Импульсное перенапряжение	35 В пост.т. в течение 0.5 с
Сигнал логической 1 (мин.)	15 В пост.т. при 2.5 мА
Сигнал логического 0 (макс.)	5 В пост.т. при 1 мА
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Одиночная фаза: 30 кГц (от 15 до 26 В пост.т.) Фаза со сдвигом на 90°: 20 кГц (от 15 до 26 В пост.т.)
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин
Потенциально развязанные группы	1
Постоянные	us установки
времени фильтра	ms установки
Количество одноврем.активных входов	2
Длина кабеля (метры)	500 экранированный, 300 неэкранированный

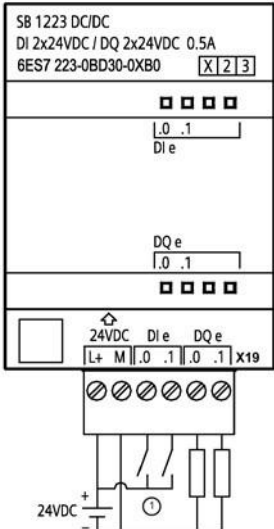
A.11 Цифровые сигнальные платы (SB)

Таблица А- 204 Цифровые выходы

Технические характеристики	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Количество выходов	2
Тип выхода	твердотельный – MOSFET (источник)
Диапазон напряжения	от 20.4 до 28.8 В пост.т.
Сигнал логической 1 при макс. токе	20 В пост.т. мин.
Сигнал логического 0 при нагрузке 10 кОм	0.1 В пост.т. макс.
Ток (макс.)	0.5 А
Ламповая нагрузка	5 Вт
Сопротивление контакта в активном состоянии	0.6 Ом макс.
Ток утечки на выход	10 мкА макс.
Частота пачки импульсов (РТО)	20 кГц макс., 2 Гц мин. ¹
Ток включения	5 А в течение 100 мс макс.
Защита от перегрузки	Нет
Электрическая развязка (полевая сторона относительно логики)	500 В пер.т. в течение 1 мин.
Потенциально развязанные группы	1
Ток на общий провод	1 А
Индуктивное фиксирующее напряжение	L+ минус 48 В, мощность рассеяния 1 Вт
Задержка переключения	2 мкс макс. из выкл во вкл. 10 мкс макс. из вкл. в выкл.
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)
Количество одноврем.активных выходов	2
Длина кабеля (метры)	500 м экранированный, 150 м неэкранированный

¹ В зависимости от Вашего приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивость.

Таблица А- 205 Схема электрических соединений для платы цифровых входов/выходов

SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC (6ES7 223-0BD30-0XB0)	
	

① Поддерживает только входы-потребители

Таблица А- 206 Привязка контактов разъема для SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC (6ES7 223-0BD30- 0XB0)

Контакт	X19
1	L+ / 24 В пост.т.
2	M / 24 В пост.т.
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.12. Аналоговые сигнальные платы (SB)

A.12.1. Технические характеристики SB 1231 с 1 аналоговым входом

Примечание

Чтобы использовать эту сигнальную плату Ваш ЦПУ должен иметь встроенное ПО версии V2.0 или выше.

Таблица A- 207 Общие технические данные

Технические характеристики	SB 1231 AI 1 x 12 bit
Номер для заказа	6ES7 231-4HA30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	35 грамм
Рассеиваемая мощность	0.4 Вт
Потребление тока (SM шина)	55 мА
Потребление тока (24 В пост.т.)	отсутствует

Таблица A- 208 Аналоговые входы

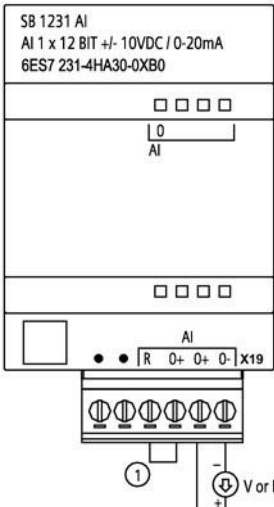
Технические характеристики	SB 1231 AI 1x12 bit
Количество входов	1
Тип	Напряжение или ток (дифференциальный)
Диапазон	± 10 В, ± 5 В, ± 2.5 В или 0... 20 мА
Разрешение	11 бит + знаковый бит
Диапазон измерений (слово данных)	от -27648 до 27648
Диапазон превышения положительных / отрицательных значений (слово данных)	Напряжение: от 32511 до 27649 / от -27649 до -32512 Ток: от 32511 до 27649 / от 0 до -4864 (Обратитесь к разделам Представление аналогового входа для напряжения / тока (стр. 1236))
Диапазон переполнения положительных / отрицательных значений (слово данных)	Напряжение: от 32767 до 32512 / от -32513 до -32768 Ток: от 32767 до 32512 / от -4865 до -32768 (Обратитесь к разделам Представление аналогового входа для напряжения / тока (стр. 1236))
Максимальная прочность по напряжению/току	± 35 В / ± 40 мА
Сглаживание	Отсутствует, слабое, среднее или сильное. (Обратитесь к разделу Реакция аналогового входа на скачок (стр.1236))
Подавление помех	400, 60, 50 или 10 Гц. (Обратитесь к разделу Время реакции аналогового входа для времени выборки (стр. 1236))
Точность (25 °C / от -20 до 60 °C)	$\pm 0.3\%$ / $\pm 0.6\%$ полного диапазона

Технические характеристики	SB 1231 AI 1x12 bit
Входной импеданс	Напряжение: 220 кОм; Ток: 250 Ом
Дифференциальный режим	Напряжение: 55 кОм; Ток: 55 кОм
Синфазный режим	
Поведение при переходе из RUN в STOP	Последнее или подстановочное значение (по умолчанию 0)
Принцип измерения	Преобразование фактического значения
Подавление синфазной помехи	40 дБ, от пост.т. до 60 Гц
Рабочий диапазон сигналов	Сигнал плюс напряжение синфазной помехи должно быть меньше +35 В и больше -35 В
Электрическая развязка (полевая сторона относительно логики)	Нет
Длина кабеля (метры)	100 м, экранированная витая пара

Таблица А- 209 Диагностика

Технические характеристики	SB 1231 AI 1 x 12 bit
Переполнение для положительных / отрицательных значений	Да
Пониженное напряжение 24 В пост.т.	Нет

Таблица А- 210 Схема электрических соединений для платы аналогового входа

SB 1231 AI x 12 bit (6ES7 231-4HA30-0XB0)	
	① Соедините "R" и "0+" в режиме тока. Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.

A.12 Аналоговые сигнальные платы (SB)

Таблица А- 211 Привязка контактов разъема для SB 1231 AI x 12 bit (6ES7 231-4HA30-0XB0)

Контакт	X19 (с позолотой)
1	Нет соединения
2	Нет соединения
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

A.12.2. Технические характеристики SB 1231 с 1 аналоговым выходом

Таблица А- 212 Общие технические данные

Технические характеристики	SB 1232 AQ 1 x 12 bit
Номер для заказа	6ES7 232-4HA30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	40 грамм
Рассеиваемая мощность	1.5 Вт
Потребление тока (SM шина)	15 мА
Потребление тока (24 В пост.т.)	40 мА (без нагрузки)

Таблица А- 213 Аналоговые выходы

Технические характеристики	SB 1232 AQ 1 x 12 bit
Количество выходов	1
Тип	Напряжение или ток
Диапазон	±10 В или 0 ... 20 мА
Разрешение	Напряжение: 12 бит Ток: 11 бит
Диапазон (слово данных) Обратитесь к выходным диапазонам для напряжения и тока (стр. 1237).	Напряжение: от -27 648 до 27 648 ; Ток: от 0 до 27 648
Точность (25 °C / от -20 до 60 °C)	±0.5% / ±1% диапазона
Время установки (95% от нового значения)	Напряжение: 300 мкс (резистор), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мкс (10 мГн)
Импеданс нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом
Поведение при переходе из RUN в STOP	Последнее значение или подстановочное значение (по умолчанию 0)
Электрическая развязка Полевая сторона относительно логики	Нет
Длина кабеля (метры)	100 м, экранированная витая пара

Таблица А- 214 Диагностика

Технические характеристики	SB 1232 AQ 1 x 12 bit
Переполнение для положит./ отрицат. значений	Да
Замыкание на землю (только для напряжения)	Да
Обрыв провода (только для тока)	Да

Таблица А- 215 Схема электрических соединений для платы SB 1232 AQ 1 x 12 bit

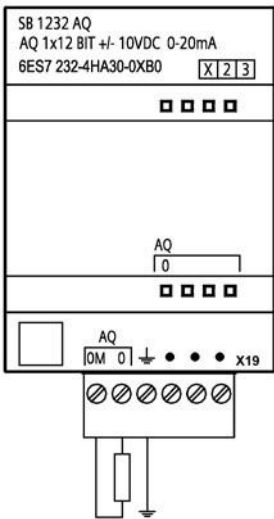
SB 1232 AQ 1 x 12 bit (6ES7 232-4HA30-0XB0)	
	
Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.	

Таблица А- 216 Привязка контактов разъема для SB 1232 AQ 1 x 12 bit (6ES7 232-4HA30-0XB0)

Контакт	X19 (с позолотой)
1	AQ 0M
2	AQ 0
3	Функциональная земля
4	Нет соединения
5	Нет соединения
6	Нет соединения

A.12.3. Диапазоны аналоговых входов и выходов

A.12.3.1. Реакция на скачок аналоговых входов

Таблица А- 217 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Настройка сглаживания (усреднение по выборке)	Настройка времени интегрирования			
	400 Гц (2.5 мс)	60 Гц (16.6 мс)	50 Гц (20 мс)	10 Гц (100 мс)
Отсутствует (1 цикл): без усреднения	4.5 мс	18.7 мс	22.0 мс	102 мс
Слабое (4 цикла): 4 значения	10.6 мс	59.3 мс	70.8 мс	346 мс
Среднее (16 циклов): 16 значений	33.0 мс	208 мс	250 мс	1240 мс
Сильное (32 циклов): 32 значения	63.0 мс	408 мс	490 мс	2440 мс
Время выборки	0.156 мс	1.042 мс	1.250 мс	6.250 мс

A.12.3.2. Время выборки и обновления для аналоговых входов

Таблица А- 218 Время выборки и обновления

Настройка	Время выборки	Время обновления сигнальной платы
400 Гц (2.5 мс)	0.156 мс	0.156 мс
60 Гц (16.6 мс)	1.042 мс	1.042 мс
50 Гц (20 мс)	1.250 мс	1.25 мс
10 Гц (100 мс)	6.250 мс	6.25 мс

A.12.3.3. Диапазоны измерения аналоговых входов для напряжения и тока (SB и SM)

Таблица А- 219 Представление аналогового входа для напряжения (SB и SM)

Система		Диапазон измерения напряжения				
Десятичная	Шестнадцатеричная	±10 В	±5 В	±2.5 В	±1.25 В	
32767	7FFF ¹	11.851 В	5.926 В	2.963 В	1.481 В	Переполнение для полож.значений
32512	7F00					
32511	7EFF	11.759 В	5.879 В	2.940 В	1.470 В	Диапазон превыше- ния для полож.знач.
27649	6C01					
27648	6C00	10 В	5 В	2.5 В	1.250 В	Номинальный диапазон
20736	5100	7.5 В	3.75 В	1.875 В	0.938 В	
1	1	361.7 мВ	180.8 мВ	90.4 мВ	45.2 мВ	
0	0	0 В	0 В	0 В	0 В	
-1	FFFF					
-20736	AF00	-7.5 В	-3.75 В	-1.875 В	-0.938 В	
-27648	9400	-10 В	-5 В	-2.5 В	-1.250 В	

-27649	93FF					Диапазон превышения для отриц.знач.
-32512	8100	-11.759 В	-5.879 В	-2.940 В	-1.470 В	
-32513	80FF					Переполнение для отриц.значений
-32768	8000	-11.851 В	-5.926 В	-2.963 В	-1.481 В	

¹ 7FFF может быть возвращено по одной из следующих причин: переполнение (как отмечено в этом столе), прежде чем станут доступны действительные значения (например, непосредственно после подачи питания) или если обнаружен обрыв провода.

Таблица A- 220 Представление аналогового входа для тока (SB и SM)

Система		Диапазон измерения тока		
Десятичная	Шестнадцатер.	от 0 до 20 мА	от 4 до 20 мА	
32767	7FFF	23.70 мА	22.96 мА	Переполнение для положительных значений
32512	7F00			
32511	7EFF	23.52 мА	22.81 мА	Диапазон превышения для положительных значений
27649	6C01			
27648	6C00	20 мА	20 мА	Номинальный диапазон
20736	5100	15 мА	16 мА	
1	1	723.4 нА	4 мА + 578.7 нА	
0	0	0 мА	4 мА	
-1	FFFF			Диапазон превышения для отрицательных значений
-4864	ED00	-3.52 мА	1.185 мА	
-4865	ECFF			Переполнение для отрицательных значений
-32768	8000			

A.12.3.4. Диапазоны аналоговых выходов для напряжения и тока (SB и SM)

Таблица A- 221 Представление аналогового выхода для напряжения (SB и SM)

Система		Диапазон выходного напряжения	
Десятичная	Шестнадцатер.	± 10 В	
32767	7FFF	Смотри примечание 1	Переполнение для положительных значений
32512	7F00	Смотри примечание 1	
32511	7EFF	11.76 В	Диапазон превышения для положительных значений
27649	6C01		
27648	6C00	10 В	Номинальный диапазон
20736	5100	7.5 В	
1	1	361.7 мкВ	
0	0	0 В	
-1	FFFF	-361.7 мкВ	
-20736	AF00	-7.5 В	
-27648	9400	-10 В	

A.12 Аналоговые сигнальные платы (SB)

Система		Диапазон выходного напряжения	
Десятичная	Шестнадцатер.	± 10 В	
-27649	93FF		Диапазон превышения для отрицательных значений
-32512	8100	-11.76 В	
-32513	80FF	Смотри примечание 1	Переполнение для отрицательных значений
-32768	8000	Смотри примечание 1	

¹ В состоянии переполнения для положительных или отрицательных значений аналоговые выходы принимают значение подстановки для режима STOP.

Таблица A- 222 Представление аналогового выхода для тока (SB и SM)

Система		Диапазон выходного тока		
Десятичная	Шестнадцатер.	от 0 до 20 мА	от 4 до 20 мА	
32767	7FFF	Смотри примечание 1	Смотри примечание 1	Переполнение для положительных значений
32512	7F00	Смотри примечание 1	Смотри примечание 1	
32511	7EFF	23.52 мА	22.81 мА	Диапазон превышения для положительных значений
27649	6C01			
27648	6C00	20 мА	20 мА	Номинальный диапазон
20736	5100	15 мА	16 мА	
1	1	723.4 нА	4 мА + 578.7 нА	
0	0	0 мА	4 мА	
-1	FFFF		4 мА ... 578.7 нА	Диапазон превышения для отрицательных значений
-6912	E500		0 мА	
-6913	E4FF			Невозможно. Выходное значение ограничено 0 мА.
-32512	8100			
-32513	80FF	Смотри примечание 1	Смотри примечание 1	Переполнение для отрицательных значений
-32768	8000	Смотри примечание 1	Смотри примечание 1	

¹ В состоянии переполнения для положительных или отрицательных значений аналоговые выходы принимают значение подстановки для режима STOP.

А.12.4. Сигнальные платы для термопар (SB)

А.12.4.1. Технические характеристики SB 1231 с 1 аналоговым входом для термопар

Примечание

Чтобы использовать эту сигнальную плату Ваш ЦПУ должен иметь встроенное ПО версии V2.0 или выше.

Таблица А- 223 Общие технические данные

Технические характеристики	SB 1231 AI 1 x 16 bit Thermocouple
Номер для заказа	6ES7 231-5QA30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	35 грамм
Рассеиваемая мощность	0.5 Вт
Потребление тока (SM шина)	5 мА
Потребление тока (24 В пост.т.)	20 мА

Таблица А- 224 Аналоговые входы

Технические характеристики		SB 1231 AI 1x16 bit Thermocouple
Количество входов		1
Тип		Плавающая термопара и мВ
Диапазон - Номинальный диапазон (слово данных) - Диапазон превышения положительных / отрицательных значений (слово данных) - Диапазон переполнения положительных / отрицательных значений (слово данных)		Смотри таблицу выбора фильтра термопары (стр. 1240).
Разрешение	Температура	0.1° C / 0.1° F
	Напряжение	15 бит плюс знак
Максимальная прочность по напряжению		± 35 В
Подавление помех		85 дБ для выбранной настройки фильтра (10 Гц, 50 Гц, 60 Гц или 400 Гц)
Подавление синфазной помехи		> 120 дБ при 120 В пер.т.
Полное сопротивление		≥ 10 МОм
Точность		Смотри таблицу выбора термопары (стр. 1240).
Повторяемость		±0.05% диапазона
Принцип измерения		Интегрирование
Время обновления модуля		Смотри таблицу выбора фильтра термопары (стр. 1240).
Ошибка холодного спая		±1.5 °C
Электрическая развязка Полевая сторона относительно логики		500 В пер.т.

A.12 Аналоговые сигнальные платы (SB)

Технические характеристики	SB 1231 AI 1x16 bit Thermocouple
Длина кабеля (метры)	100 метров до датчика макс.
Сопротивление проводника	100 Ом макс.

Таблица А- 225 Диагностика

Технические характеристики	SB 1231 AI 1 x 16 bit Thermocouple
Переполнение для положительных / отрицательных значений ¹	Да
Обрыв провода (только для тока) ²	Да

¹ Диагностическая сигнальная информация о переполнении положительных и отрицательных значений и пониженном напряжении, будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

² Когда тревога обрыва провода отключена, и присутствует обрыв в проводке датчика, модуль может выдавать случайные значения.

Аналоговая сигнальная плата SB 1231 Thermocouple (TC) измеряет значение напряжения, поданного на входы модуля. Тип измерения температуры может быть или "Термопара" или "Напряжение".

- "Термопара": Значение будет передано в градусах, умноженных на десять (например, 25.3 градуса будет представлено как десятичное число 253).
- "Напряжение": Значением предела номинального диапазона будет десятичное 27648.

A.12.4.2 Базовый режим работы для термопар

Термопары образуются, когда два разнородных металла электрически соединены друг с другом. Вырабатывается напряжение, которое пропорционально температуре спая. Это напряжение мало; один микровольт может представлять много градусов. Измерение напряжения от термопары, компенсация дополнительных спаев, и последующая линеаризация результата являются основанием для измерения температуры с помощью термопар.

Когда Вы подключаете термопару к плате SB 1231 Thermocouple, два разнородных металлических провода подключены к модулю посредством сигнального соединителя модуля. Место, где два разнородных провода присоединены друг к другу, образует термопару датчика.

Еще две термопары образуются, когда два разнородных провода подключены к сигнальному соединителю. Температура соединителя вызывает напряжение, которое добавляется к напряжению от термопары датчика. Если это напряжение не скорректировано, то передаваемая температура, будет отличаться от температуры датчика.

Компенсация холодного спая используется для корректировки влияния термопары соединителя. Таблицы термопар основаны на справочной температуре соединения, обычно для нуля градусов по Цельсию. Компенсация холодного спая дает корректировку соединителю до нуля градусов по Цельсию. Компенсация холодного спая восстанавливает напряжение, измененное термопарами соединителя. Температура модуля измеряется внутренне, и затем преобразуется в значение, которое будет добавлено к значению от датчика. Затем исправленное значение датчика линеаризуется, используя таблицы термопар.

Для оптимальной работы компенсации холодного спая модуль термопары должен быть расположен в термически стабильной окружающей среде. Медленное изменение (менее чем 0.1 °C/мин) в окружении модуля правильно компенсируется согласно техническим требованиям модуля. Движение воздуха через модуль также приведет к ошибкам компенсации холодного спая.

Если необходима наилучшая компенсация ошибки холодного спая, может использоваться внешняя изотермическая клеммная колодка. Модуль термопары предусматривает использование клеммной колодки с эталонной температурой 0 °C или 50 °C.

Таблицы выбора для SB 1231 с подключением термопар

Диапазоны и точность для различных типов термопар, поддерживаемых сигнальной платой SB 1231 Thermocouple, представлены в таблице ниже.

Таблица A- 226 Таблица выбора термопары для SB 1231

Тип	Минимум превышения отриц.знач. ¹	Нижний предел номинального диапазона	Верхний предел номинального диапазона	Максимум превышения полож.знач. ²	Точность в нормальном диапазоне ³ , @ 25 °C	Точность в нормальном диапазоне ³ от -20 °C до 60 °C
J	-210.0 °C	-150.0 °C	1200.0 °C	1450.0 °C	±0.3 °C	±0.6 °C
K	-270.0 °C	-200.0 °C	1372.0 °C	1622.0 °C	±0.4 °C	±1.0 °C
T	-270.0 °C	-200.0 °C	400.0 °C	540.0 °C	±0.5 °C	±1.0 °C
E	-270.0 °C	-200.0 °C	1000.0 °C	1200.0 °C	±0.3 °C	±0.6 °C
R & S	-50.0 °C	100.0 °C	1768.0 °C	2019.0 °C	±1.0 °C	±2.5 °C
B	0.0 °C	200.0 °C	800.0 °C	--	±2.0 °C	±2.5 °C
	--	800.0 °C	1820.0 °C	1820.0 °C	±1.0 °C	±2.3 °C
N	-270.0 °C	0.0 °C	1300.0 °C	1550.0 °C	±1.0 °C	±1.6 °C
C	0.0 °C	100.0 °C	2315.0 °C	2500.0 °C	±0.7 °C	±2.7 °C
ТХК/ХК(L)	-200.0 °C	-150.0 °C	800.0 °C	1050.0 °C	±0.6 °C	±1.2 °C
Напряжение	-32512	-27648 -80 mV	27648 80 mV	32511	±0.05%	±0.1%

¹ Значения от термопары ниже минимального превышения отрицательного значения передаются как -32768.

² Значения от термопары выше максимального превышения положительного значения передаются как 32767.

³ Внутренняя ошибка холодного спая составляет ±1.5 °C для всех диапазонов. Она добавляется к ошибке согласно этой таблице. Плате требуется по крайней мере 30 минут разогрева, чтобы удовлетворять настоящей спецификации.

Таблица A- 227 Подавление помех и время обновления для SB 1231 с подключением термопар

Частота подавления (Гц)	Время интегрирования (мс)	Время обновления сигнальной платы (с)
10	100	0.306
50	20	0.066
60	16.67	0.056
400 ¹	10	0.036

¹ Для того, чтобы достичь заявленной точности и разрешения при фильтре 400 Гц, время интегрирования устанавливается равным 10 мс. Этот фильтр также отсекает помехи частот 100 Гц и 200 Гц.

Для измерительных термопар рекомендуется использовать время интегрирования 100 мс. Использование меньших значений времени интегрирования увеличит ошибку воспроизводимости регистрируемых значений температуры.

Примечание

После подачи питания плата выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени плата выдает значение 32767 на каждом канале, пока действительные данные не станут доступны на этом канале. Ваша пользовательская программа, возможно, должна допускать это время инициализации.

Таблица А- 228 Схема электрических соединений для SB 1231 AI 1 x 16 thermocouple

SB 1231 TC
AI 1 x TC x 16 BIT
6ES7 231-5QA30-0XB0

□ □ □ □

0
AI

□ □ □ □

AIO - TC
0+ 0- x19

⦿ ⦿ ⦿ ⦿ ⦿ ⦿

+

+

-

Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.

Таблица А- 229 Привязка контактов разъема для SB 1231 AI 1 x 16 bit thermocouple (6ES7 231-5QA30- 0XB0)

Контакт	X19 (с позолотой)
1	Нет соединения
2	Нет соединения
3	Нет соединения
4	Нет соединения
5	AI 0- /TC
6	AI 0+ /TC

А.12.5. Сигнальные платы для подключения терморезисторов (SB)

А.12.5.1. Технические характеристики SB 1231 с 1 аналоговым входом для терморезисторов

Примечание

Чтобы использовать эту сигнальную плату Ваш ЦПУ должен иметь встроенное ПО версии V2.0 или выше.

Таблица А- 230 Общие технические данные

Технические характеристики	SB 1231 AI 1 x 16 bit RTD
Номер для заказа	6ES7 231-5QA30-0XB0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	35 грамм
Рассеиваемая мощность	0.7 Вт
Потребление тока (SM шина)	5 мА
Потребление тока (24 В пост.т.)	25 мА

Таблица А- 231 Аналоговые входы

Технические характеристики		SB 1231 AI 1 x 16 bit RTD
Количество входов		1
Тип		Рекомендованные для модуля термосопротивления и сопротивления
Диапазон - Номинальный диапазон (слово данных) - Диапазон превышения положительных / отрицательных значений (слово данных) - Диапазон переполнения положительных /		Смотри таблицу выбора (стр. 1246).
Разрешение	Температура	0.1° C / 0.1° F
	Напряжение	15 бит плюс знак
Максимальная прочность по напряжению		± 35 В
Подавление помех		85 дБ (10 Гц, 50 Гц, 60 Гц или 400 Гц)
Подавление синфазной помехи		> 120 дБ
Полное сопротивление		≥ 10 МОм
Точность		Смотри таблицу выбора (стр. 1246).
Повторяемость		±0.05% диапазона
Максимальная мощность рассеяния на датчике		0.5 мВт
Принцип измерения		Интегрирование
Время обновления платы		Смотри таблицу выбора (стр. 1246).
Электрическая развязка Полевая сторона относительно логики		500 В пер.т.
Длина кабеля (метры)		100 метров до датчика макс.
Сопротивление проводника		20 Ом, 2.7 Ом для терморезистора 10 Ом макс.

Таблица А- 232 Диагностика

Технические данные	SB 1231 AI 1 x 16 bit RTD
Переполнение для положительных / отрицательных значений ^{1,2}	Да
Обрыв провода ³	Да

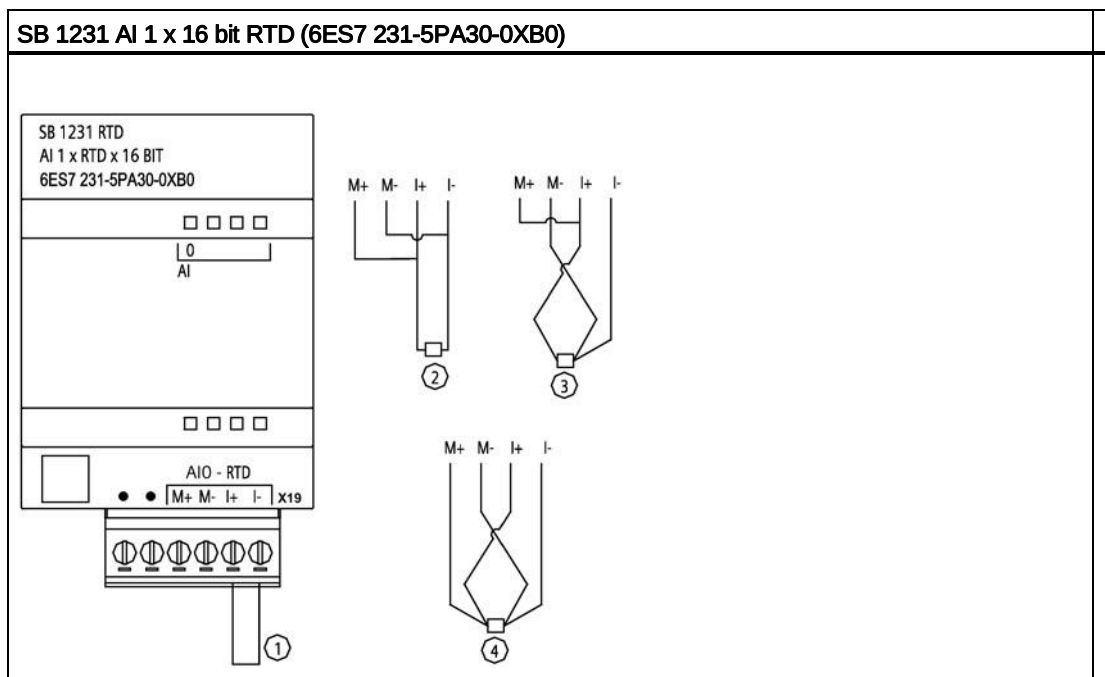
- ¹ Диагностическая сигнальная информация о переполнении положительных и отрицательных значений и пониженном напряжении, будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.
- ² Для диапазонов сопротивлений никогда не активируется обнаружение переполнения отрицательных значений.
- ³ Когда тревога обрыва провода отключена, и есть обрыв провода датчика, модуль может выдавать случайные значения.

Аналоговая сигнальная плата SM 1231 RTD измеряет значение сопротивления, подключенного ко входу платы. Тип измерения температуры может быть или "Резистор" или "Терморезистор".

- "Резистор": Значением предела номинального диапазона будет десятичное 27648.
- "Терморезистор": Значения будут переданы в градусах, умноженных на десять (например, 25.3 градуса будут представлены как десятичное число 253). Значения климатического диапазона будут переданы в градусах, умноженных на сто (например, 25.34 градуса будут представлены как десятичное число 2534).

Плата SM 1231 RTD поддерживает измерения с помощью 2-проводного, 3-проводного и 4-проводного подключения датчика сопротивления.

Таблица А- 233 Схема электрических соединений для SB 1231 AI 1 x 16 bit RTD



① Замкните неиспользуемые входы терморезисторов

② 2-проводный терморезистор

③ 3- проводный терморезистор

④ 4- проводный терморезистор

Примечание: Соединители должны быть с позолотой. Смотри Приложение С, Запасные части для номера по артикулу.

Таблица А- 234 Привязка контактов разъема для SB 1231 AI 1 x 16 bit RTD (6ES7 231-5PA30-0XB0)

Контакт	X19 (с позолотой)
1	Нет соединения
2	Нет соединения
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD

A.12.5.2. Таблицы выбора для SB 1231 с подключением терморезисторов

Таблица А- 235 Диапазоны и точность для различных датчиков, поддерживаемых модулем RTD

Температурный коэффициент	Тип RTD	Минимум превыш. отрицат. знач. ²	Нижний предел номин. диапазона	Верхний предел номин. диапазона	Максимум превыш. положит. знач. ²	Точность в норм.диапазоне @ 25 °C	Точность в норм. диапазоне от -20 до 60 °C
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 climatic	-145.00 °C	-120.00 °C	-145.00 °C	-155.00 °C	±0.20 °C	±0.40 °C
	Pt 10	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±1.0 °C	±2.0 °C
	Pt 50	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	± 0.5 °C	±1.0 °C
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003910	Pt 10	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±1.0 °C	±2.0 °C
	Pt 50	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±0.8 °C	±1.6 °C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0.005000	LG-Ni 1000	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
Ni 0.006170	Ni 100	-105.0 °C	-60.0 °C	180.0 °C	212.4 °C	±0.5 °C	±1.0 °C
Cu 0.004270	Cu 10	-240.0 °C	-200.0 °C	260.0 °C	312.0 °C	±1.0 °	±2.0 °C
Cu 0.004260	Cu 10	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±0.6 °C	±1.2 °C
	Cu 100						
Cu 0.004280	Cu 10	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±0.7 °C	±1.4 °C
	Cu 100						

¹ Значения от терморезистора ниже минимального превышения отрицательного значения передаются как -32768.² Значения от терморезистора выше максимального превышения положительного значения передаются как 32767.

Таблица A- 236 Сопротивление

Диапазон	Минимум превышения отрицател. значения	Нижний предел номин. диапазона	Верхний предел номинального диапазона	Максимум превышения положит. значения ¹	Точность в нормальном диапазоне @ 25 °C	Точность в нормальном диапазоне от -20 °C до 60 °C
150 Ом	неприменимо	0 (0 Ом)	27648 (150 Ом)	176.383 Ом	±0.05%	±0.1%
300 Ом	неприменимо	0 (0 Ом)	27648 (300 Ом)	352.767 Ом	±0.05%	±0.1%
600 Ом	неприменимо	0 (0 Ом)	27648 (600 Ом)	705.534 Ом	±0.05%	±0.1%

¹ Значения сопротивления выше максимального превышения положительного значения передаются как +32767.

Примечание

Модуль выдает 32767 для любого активированного канала без подключенного датчика. Если также разрешено обнаружение обрыва провода, модуль активирует соответствующие красные светодиоды.

Для терморезисторов диапазона 10 Ом наилучшая точность будет достигнута при использовании 4-проводного подключения.

Сопротивление соединительных проводов при 2-проводном способе вызовет ошибку интерпретации показаний датчика, и поэтому точность не гарантирована.

Таблица A- 237 Шумоподавление и время отклика для модулей подключения терморезисторов

Выбор частоты подавления	Время интегрирования	4-/2-проводное, 1-канальная плата Время обновления (секунды)	3-проводное, 1-канальная плата Время обновления (секунды)
400 Гц (2.5 мс)	10 мс ¹	0.036	0.071
60 Гц (16.6 мс)	16.67 мс	0.056	0.111
50 Гц (20 мс)	20 мс	0.066	1.086
10 Гц (100 мс)	100 мс	0.306	0.611

¹ Для того, чтобы достичь заявленной точности и разрешения при фильтре 400 Гц, время интегрирования устанавливается равным 10 мс. Этот фильтр также отсекает помехи частот 100 Гц и 200 Гц.

Примечание

После подачи питания плата выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль выдает значение 32767 на каждом канале, пока действительные данные не станут доступны на этом канале. Ваша пользовательская программа, возможно, должна допускать это время инициализации. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, Вы должны проверить поведение модуля в Вашей конфигурации. При необходимости Вы можете включать логику в свою пользовательскую программу, чтобы подстроиться под время инициализации модуля.

A.13. Батарейная плата BB 1297

Батарейная плата BB 1297

Батарейная плата S7-1200 BB 1297 разработана для долговременного резервирования часов реального времени. Она устанавливается в слот для сигнальных плат ЦПУ S7-1200 (встроенное ПО версии 3.0 и выше). Вы должны добавить BB 1297 в конфигурацию устройства и загрузить аппаратную конфигурацию в ЦПУ для того чтобы батарейная плата выполняла свои функции.

Батарея (типа CR1025) не включена в объем поставки BB 1297 и должна приобретаться пользователем.

Примечание

BB 1297 конструктивно разработана для оснащения ЦПУ с версией встроенного ПО 3.0 и выше.

Не используйте BB 1297 с более ранними версиями ЦПУ, поскольку соединитель BB 1297 не подходит к ЦПУ.



ПРЕДУПРЕЖДЕНИЕ

Установка непригодной батареи в плату BB 1297, или иначе подключение непригодной батареи к схеме могут привести к возгоранию или повреждению компонентов и непредсказуемой работе оборудования.

Возгорание или непредсказуемая работа оборудования могут привести к смертельному случаю, тяжелым телесным повреждениям или материальному ущербу.

Используйте только специальную батарею CR1025 для резервирования часов реального времени.

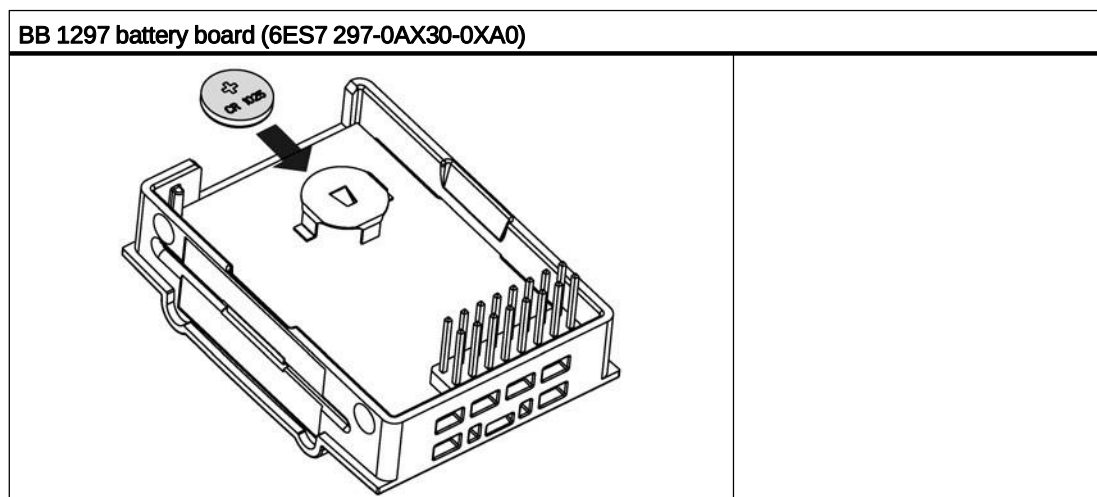
Таблица A- 238 Общие технические данные

Технические характеристики	BB 1297 Battery Board
Номер для заказа	6ES7 297-0AX30-0XA0
Размеры Ш x В x Г (мм)	38 x 62 x 21
Вес	28 грамм
Рассеиваемая мощность	0.5 Вт
Потребление тока (SM шина)	11 мА
Потребление тока (24 В пост.т.)	отсутствует

Батарея (не включена в поставку)	BB 1297 Battery Board
Время удержания	Приблизительно 1 год
Тип батареи	CR1025. Обратитесь к разделу Установка или замена батареи в батарейной плате BB 1297 (стр. 60)
Номинальное напряжение	3 В
Номинальная емкость	По крайней мере 30 мА/ч

Диагностика	BB 1297 Battery Board
Критическое напряжение батареи	< 2.5 В
Диагностика батареи	Индикатор пониженного напряжения: - Пониженное напряжение батареи вызывает постоянное свечение желтого светодиода MAINT на ЦПУ. - Событие в диагностическом буфере: 16#06:2700 "Требуется обслуживание субмодуля: По меньшей мере одна батарея разряжена (BATTF)"
Состояние батареи	Предусмотрен бит состояния батареи 0 = Батарея в порядке 1 = Батарея разряжена
Обновление статуса батареи	Статус батареи обновляется при подаче питания и затем один раз в сутки пока ЦПУ находится в RUN..

Таблица А- 239 Схема сборки для батарейной платы BB 1297



A.14. Коммуникационные интерфейсы

A.14.1. PROFIBUS

A.14.1.1. CM 1242-5 ведомое устройство PROFIBUS DP

Таблица А- 240 Технические характеристики CM 1242-5

Технические характеристики	
Номер для заказа	6GK7 242-5DX30-0XE0
Интерфейсы	
Подключение к шине PROFIBUS	9-контактное D-sub гнездо
Максимальное потребление тока на интерфейсе PROFIBUS при подключении сетевых компонентов (например, оптических сетевых компонентов)	15 мА при 5 В (только для шинного окончания) *)
Допустимые условия окружающей среды	
Окружающая температура - Во время хранения - Во время транспортировки - Во время работы при вертикальной установке (DIN-рейка горизонтально) - Во время работы при горизонтальной установке (DIN-рейка вертикально)	-40 °C ... 70 °C -40 °C ... 70 °C 0 °C ... 55 °C 0 °C ... 45 °C
Относительная влажность при 25 °C во время эксплуатации, без конденсации, максимум	95 %
Степень защиты	IP20
Источник питания, потребление тока и потери мощности	
Тип источника питания	Постоянный ток
Питание от внутренней шины	5 В
Потребление тока (типичное значение)	150 мА
Эффективная рассеиваемая мощность (типичное значение)	0.75 Вт
Электрическая развязка - PROFIBUS интерфейс относительно земли - PROFIBUS интерфейс относительно внутренних цепей	710 В пер.т. в течение 1 минуты
Габариты и масса	
- Ширина - Высота - Глубина	30 мм 100 мм 75 мм

Технические характеристики	
Масса	
• Масса без упаковки	• 115 г
• Масса с упаковкой	• 152 г

*) Токовая нагрузка от внешнего потребителя, подключенного между VP (контакт 6) и DGND (контакт 5), не должна превышать максимум 15 мА (защита от короткого замыкания) для завершения шины.

А.14.1.2. Привязка контактов D-sub гнезда модуля CM 1242-5

PROFIBUS интерфейс

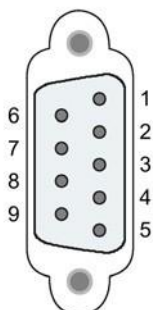


Таблица А- 241 Привязка контактов D-sub гнезда

Контакт	Описание	Контакт	Описание
1	- не используется -	6	P5V2: +5В источника питания
2	- не используется -	7	- не используется -
3	RxD/TxD-P: Линия данных В	8	RxD/TxD-N: Линия данных А
4	RTS	9	- не используется -
5	M5V2: Опорный потенциал канала данных (земля DGND)	Корпус	Клемма заземления

А.14.1.3. CM 1243-5 ведущее устройство PROFIBUS DP

Технические характеристики	
Номер для заказа	6GK7 243-5DX30-0XE0
Интерфейсы	
Подключение к шине PROFIBUS	9-контактное D-sub гнездо
Максимальное потребление тока на интерфейсе PROFIBUS при подключении сетевых компонентов (например, оптических сетевых компонентов)	15 мА при 5 В (только для шинного окончания) *)
Допустимые условия окружающей среды	
Окружающая температура - Во время хранения - Во время транспортировки - Во время работы при вертикальной установке (DIN- рейка горизонтально) - Во время работы при горизонтальной установке (DIN- рейка вертикально)	-40 °C ... 70 °C -40 °C ... 70 °C 0 °C ... 55 °C 0 °C ... 45 °C
Относительная влажность при 25 °C во время эксплуатации, без конденсации, максимум	95 %
Степень защиты	IP20
Источник питания, потребление тока и потери мощности	
Тип источника питания	Постоянный ток
Напряжение питания / внешнее - минимум - максимум	24 В 19.2 В 28.8 В
Потребление тока (типовое значение) - от 24 В пост.т. - от внутренней шины S7-1200	100 мА 0 мА
Эффективная рассеиваемая мощность (типовое значение) - от 24 В пост.т. - от внутренней шины S7-1200	2.4 Вт 0 Вт
Источник питания 24 В пост.т. / внешний - Мин. поперечное сечение кабеля - Макс. поперечное сечение кабеля - Момент затяжки винтовых соединителей	- мин.: 0.14 мм² (AWG 25) - макс.: 1.5 мм² (AWG 15) 0.45 Нм (4 фунта на дюйм)
Электрическая развязка - PROFIBUS интерфейс относительно земли - PROFIBUS интерфейс относительно внут-	710 В пер.т. в течение 1 минуты
Габариты и масса	

Технические характеристики	
• Ширина • Высота • Глубина	• 30 мм • 100 мм • 75 мм
Масса • Масса без упаковки • Масса с упаковкой	• 134 г • 171 г

*) Токовая нагрузка от внешнего потребителя, подключенного между VP (контакт 6) и DGND (контакт 5), не должна превышать максимум 15 мА (защита от короткого замыкания) для завершения шины.

Примечание

СМ 1243-5 (модуль ведущего устройства PROFIBUS) должен получать питание от источника 24 В пост. т. для питания датчиков ЦПУ..

А.14.1.4. Привязка контактов D-sub гнезда модуля СМ 1243-5

PROFIBUS интерфейс

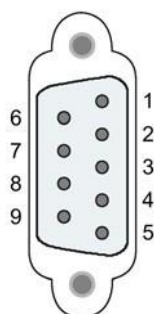


Таблица А- 243 Привязка контактов D-sub гнезда

Контакт	Описание	Контакт	Описание
1	- не используется -	6	VP: +5В источника питания, только для концевых резисторов шины; не для питания внешних устройств
2	- не используется -	7	- не используется -

Контакт	Описание	Контакт	Описание
3	RxD/TxD-P: Линия данных В	8	RxD/TxD-N: Линия данных А
4	CNTR-P: RTS	9	- не используется -
5	DGND: Земля сигналов данных и VP	Корпус	Клемма заземления

PROFIBUS кабель**Примечание****Контакт с экраном PROFIBUS-кабеля**

Экран PROFIBUS-кабеля необходимо подключить.

Чтобы сделать это, снимите изоляцию на конце PROFIBUS-кабеля и соедините экран с функциональной землей.

A.14.2. CP 1242-7**Примечание**

Модуль CP 1242-7 не одобрен к применению в приложениях, связанных с морем

CP 1242-7 не имеет морского сертификата.

Примечание

Чтобы использовать эти модули Ваш ЦПУ должен иметь встроенное ПО версии V2.0 или выше.

A.14.2.1. CP 1242-7 GPRS

Таблица А- 244 Технические характеристики модуля CP 1242-7 GPRS V2

Технические характеристики	
Номер для заказа	6GK7 242-7KX3-0XE0
Беспроводный интерфейс	
Антенный соединитель	SMA гнездо
Номинальный импеданс	50 Ом
Беспроводное подключение	
Максимальная излучаемая мощность	- GSM 850, класс 4: +33 дБм ± 2 дБм - GSM 900, класс 4: +33 дБм ± 2 дБм - GSM 1800, класс 1: +30 дБм ± 2 дБм - GSM 1900, класс 1: +30 дБм ± 2 дБм

Технические характеристики	
GPRS	Многоканальное класса 10 устройство класса В схема кодирования 1...4 (GMSK)
SMS	Способ отправки: MO сервис: точка-к точке
Допустимые условия окружающей среды	
Окружающая температура - Во время хранения - Во время транспортировки - Во время работы при вертикальной установке (DIN-рейка горизонтально) - Во время работы при горизонтальной установке (DIN-рейка вертикально)	-40 °C ... 70 °C -40 °C ... 70 °C 0 °C ... 55 °C 0 °C ... 45 °C
Относительная влажность при 25 °C во время эксплуатации, без конденсации, максимум	95 %
Степень защиты	IP20
Источник питания, потребление тока и потери мощности	
Тип источника питания	Постоянный ток
Напряжение питания / внешнее - минимум - максимум	24 В 19.2 В 28.8 В
Потребление тока (типовое значение) - от 24 В пост.т. - от внутренней шины S7-1200	100 мА 0 мА
Эффективная рассеиваемая мощность (типовое значение) - от 24 В пост.т. - от внутренней шины S7-1200	2.4 Вт 0 Вт
Источник питания 24 В пост.т. / внешний - Мин. поперечное сечение кабеля - Макс. поперечное сечение кабеля - Момент затяжки винтовых соединителей	- мин.: 0.14 мм² (AWG 25) - макс.: 1.5 мм² (AWG 15) 0.45 Нм (4 фунта на дюйм)
Электрическая развязка Модуль источника питания относительно внутренних цепей	710 В пер.т. в течение 1 минуты
Габариты и масса	
- Ширина - Высота - Глубина	30 мм 100 мм 75 мм
Масса - Масса без упаковки - Масса с упаковкой	133 г 170 г

A.14.2.2. GSM/GPRS антенна ANT794-4MR

Технические характеристики антенны ANT794-4MR GSM/GPRS

ANT794-4MR	
Номер для заказа	6NH9860-1AA00
Мобильные беспроводные сети	GSM/GPRS
Частотные диапазоны	- от 824 до 960 МГц (GSM 850, 900) - от 1 710 до 1 880 МГц (GSM 1 800) - от 1 900 до 2 200 МГц (GSM / UMTS)
Характеристика	ненаправленная
Коэффициент усиления	0 дБ
Импеданс	50 Ом
Коэффициент стоячей волны (КСВ)	< 2,0
Макс. мощность	20 Вт
Поляризация	линейная вертикальная
Соединитель	SMA
Длина антенного кабеля	5 м
Наружная оболочка	Жесткий ПВХ, с защитой от УФ-излучения
Степень защиты	IP20
Допустимые условия окружающей среды	- Рабочая температура - Температура транспортировки / хранения - Относительная влажность
Конструкция	Антенна с 5 м кабелем фиксированной длины и SMA штекерный соединитель
Габариты (Г х В) в мм	25 x 193
Вес	- Антенна вместе с кабелем - Крепеж
Установка	С помощью прилагаемой скобы

А.14.2.3. Плоская антенна ANT794-3M

Технические характеристики плоской антенны ANT794-3M

ANT794-3M		
Номер для заказа	6NH9870-1AA00	
Мобильные беспроводные сети	GSM 900	GSM 1800/1900
Частотные диапазоны	890 - 960 МГц	1710 - 1990 МГц
Коэффициент стоячей волны (КСВ)	≤ 2:1	≤ 1,5:1
Обратные потери (Тх)	≈ 10 дБ	≈ 14 дБ
Коэффициент усиления	0 дБ	
Импеданс	50 Ом	
Макс. мощность	10 Вт	
Антенный кабель	В/ч кабель RG 174 (фикс.длины) с SMA штекерным соединителем	
Длина кабеля	1.2 м	
Степень защиты	IP64	
Допустимый температурный диапазон	от -40°C до +75°C	
Огнестойкость	UL 94 V2	
Наружная оболочка	ABS Polylac PA-765, светло-серый (RAL 7035)	
Габариты (Ш x Д x В) в мм	70.5 x 146.5 x 20.5	
Вес	130 г	

А.14.3. Ведущее устройство CM 1243-2 AS-i шины

А.14.3.1. Технические характеристики ведущего устройства AS-i шины CM 1243-2

Таблица А- 245 Технические характеристики для ведущего устройства AS-i шины CM 1243-2

Технические характеристики	
Номер для заказа	3RK7243-2AA30-0XB0
Версия встроенного ПО	V1.0
Дата	01.12.2011
Интерфейсы	
Максимальный потребляемый ток От внутренней шины S7-1200	Макс. 250 мА, Напряжение питания коммуникационной шины S7-1200 5 В пост.т.
От AS-i-кабеля	Макс. 100 мА
Максимально допустимая нагрузка по току между контактами ASI+/ASI-	8 А
Привязка контактов	Смотри раздел Электрические соединения ведущего устройства AS-i шины (стр. 1259)
Поперечное сечение проводника	0.2 мм ² (AWG 24) ... 3.3 мм ² (AWG 12)

Технические характеристики	
Момент затяжки ASI соединителя	0.56 Нм
Допустимые условия окружающей среды	
Окружающая температура	
Во время хранения	-40 °C ... 70 °C
Во время транспортировки	-40 °C ... 70 °C
Во время работы, с вертикальной установкой (горизонтальная стандартная монтажная рейка)	0 °C ... 55 °C
Во время работы, с горизонтальной установкой (вертикальная стандартная монтажная рейка)	0 °C ... 45 °C
Относительная влажность при 25 °C во время эксплуатации, без конденсации, максимум	95 %
Степень защиты	IP20
Источник питания, потребление тока и потери мощности	
Тип источника питания	Постоянный ток
Потребление тока (типовое значение)	
От внутренней шины S7-1200	200 мА
Суммарные потери мощности (типовое значение):	
• От внутренней шины S7-1200	1 Вт
• От AS-i кабеля	2.4 Вт
Габариты и масса	
Ширина	30 мм
Высота	100 мм
Глубина	75 мм
Масса	
Масса без упаковки	122 г
Масса с упаковкой	159 г

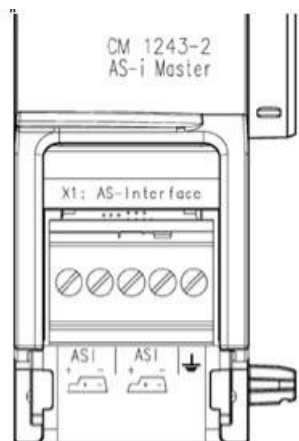
А.14.3.2. Электрические соединения ведущего устройства AS-i шины

Источник питания ведущего устройства AS-i шины CM 1243-2

Ведущее устройство AS-i шины CM 1243-2 получает питание от коммуникационной шины S7-1200. Это означает, что диагностическое сообщение может быть послано в S7-1200 после сбоя напряжения питания AS-i шины. Соединение с коммуникационной шиной находится на ведущем устройстве AS-i шины CM 1243-2 справа.

Терминалы AS-интерфейса

Съемный терминал для подключения AS-i кабеля расположен за нижней крышкой на передней стороне ведущего устройства AS-i шины CM 1243-2.



Если используется AS-i кабель специальной формы, Вы можете узнать правильную полярность с помощью символа



Информацию о том, как удалить и повторно установить клеммную колодку, можно найти в главе по установке (стр. 65).

Примечание

Пропускная способность соединительных контактов составляет максимум 8 А. Если это значение превышено в AS-i кабеле, то ведущее устройство AS-i шины CM 1243-2 не должно быть "закольцовано" через AS-i кабель, а должно подключаться через ответвление (только одна соединительная пара, подведенная к ведущему устройству AS-i шины CM 1243-2).

Кроме того убедитесь, что используемые кабели пригодны для рабочих температур не менее 75°C, если ток проводится через ведущее устройство AS-i шины и присутствуют токи больше 4 ампер.

Вы найдете дополнительную информацию о подключении AS-i кабеля в разделе "Монтаж, подключение и ввод в эксплуатацию модулей" в руководстве "Ведущее устройство AS-i шины CM 1243-2 и Блок развязки AS-i данных DCM 1271 для SIMATIC S7-1200".

Привязка контактов разъема

Метка	Значение
ASI+	Подключение к AS-i- положительная полярность
ASI-	Подключение к AS-i- отрицательная полярность
	Функциональная земля

A.14.4. RS232, RS422 и RS485**A.14.4.1. Технические характеристики коммуникационной платы CB 1241 RS485****Примечание**

Чтобы использовать эти модули Ваш ЦПУ должен иметь встроенное ПО версии V2.0 или выше.

Таблица А- 246 Общие технические данные

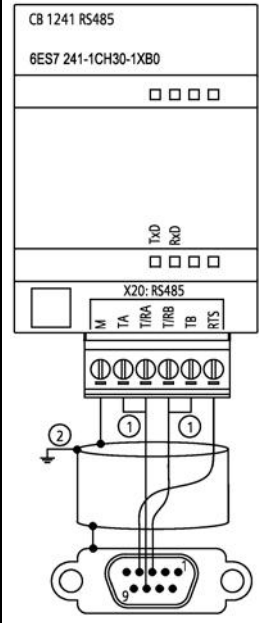
Технические характеристики	CB 1241 RS485
Номер для заказа	6ES7 241-1CH30-1XB0
Габариты Ш x В x Г (мм)	38 x 62 x 21
Масса	40 грамм

Таблица А- 247 Передатчик и приемник

Технические характеристики	СВ 1241 RS485
Тип	RS485 (2-проводный полу-дуплекс)
Динамический диапазон для синфазного сигнала	от -7 до +12 В, 1 секунда, 3 действ. непрерывно
Напряжение дифференциального выхода передатчика	2 В мин. при $R_L = 100 \text{ Ом}$ 1.5 В мин. при $R_L = 54 \text{ Ом}$
Нагрузка и смещение	10 кОм на +5 В для В, RS485 контакт 3 10 кОм на GND для А, RS485 контакт 4
Дополнительная нагрузка	Замкнуть контакт ТВ на Т/ВВ, эффективный импеданс нагрузки составляет 127 Ом, подключается к RS485 контакт 3 Замкнуть контакт ТА на Т/РА, эффективный импеданс нагрузки составляет 127 Ом, подключается к RS485 контакт 4
Входной импеданс приемника	5.4 кОм мин. включая нагрузку
Пороговая чувствительность приемника	+/- 0.2 В мин., 60 мВ типовой гистерезис
Электрическая развязка RS485 сигнал относительно корпуса RS485 сигнал относительно общего провода логической схемы ЦПУ	500 В пер.т., 1 минута
Длина экранированного кабеля	1000 м макс.
Скорость обмена	300 бод, 600 бод, 1.2 кбит/с, 2.4 кбит/с, 4.8 кбит/с, 9.6 кбит/с (по умолчанию), 19.2 кбит/с, 38.4 кбит/с, 57.6 кбит/с, 76.8 кбит/с, 115.2 кбит/с,
Четность	Без контроля четности (по умолчанию), четный, нечетный, метка (бит четности всегда установлен в 1), пробел (бит четности всегда установлен в 0)
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	не поддерживается
Время ожидания	от 0 до 65535 мс

Таблица А- 248 Источник питания

Технические характеристики	СВ 1241 RS485
Потеря мощности (рассеяние)	1.5 Вт
Потребление тока (SM шина), макс.	50 мА
Потребление тока (24 В пост.т.) макс.	80 мА

CB 1241 RS485 (6ES7 241-1CH30-1XB0)	
	
① Подключите "ТА" и "ТВ" как показано на рисунке, чтобы подключить концевую нагрузку. (Включайте концевую нагрузку только на оконечных устройствах RS485 сети.) ② Используйте кабель на основе экранированной витой пары и соедините экран кабеля с корпусом.	

Вы подключаете концевую нагрузку только на двух концах RS485 сети. К устройствам, находящимся между двумя оконечными устройствами, не подключают ни концевую нагрузку ни смещение. Смотри раздел "Смещение и концевая нагрузка соединителя сети RS485" (стр. 862).

Таблица А- 249 Привязка контактов соединителя для CB 1241 RS485 (6ES7 241-1CH30-1XB0)

Контакт	9-контактный соединитель	X20
1	RS485 / логическая земля	--
2	RS485 / не используется	--
3	RS485 / TxD+	3 - T/RB
4	RS485 / RTS	1 - RTS
5	RS485 / логическая земля	--
6	RS485 / питание 5 В	--
7	RS485 / не используется	--
8	RS485 / TxD-	4 - T/RA
9	RS485 / не используется	--
Корпус		7 - M

А.14.4.2. Технические характеристики коммуникационного модуля CM 1241 RS232

Таблица А- 250 Общие технические данные

Технические характеристики	CM 1241 RS232
Номер для заказа	6ES7 241-1AH32-0XB0
Габариты Ш x В x Г (мм)	30 x 100 x 75
Масса	150 г

Таблица А- 251 Передатчик и приемник

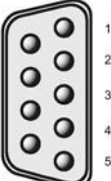
Технические характеристики	CM 1241 RS232
Тип	RS232 (полный дуплекс)
Выходное напряжение передатчика	+/- 5 В мин. при $R_L = 3 \text{ кОм}$
Выходное напряжение передатчика	+/- 15 В макс.
Входной импеданс приемника	3 кОм мин.
Чувствительность приемника	0.8 В мин. низкий уровень, 2.4 В макс. высокий уровень 0.5 В типовой гистерезис
Входное напряжение приемника	+/- 30 В пост.т. макс.
Электрическая развязка RS485 сигнал относительно корпуса RS485 сигнал относительно общего провода ло-	500 В пер.т., 1 минута
Длина экранированного кабеля	10 м макс.
Скорость обмена	300 бод, 600 бод, 1.2 кбит/с, 2.4 кбит/с, 4.8 кбит/с, 9.6 кбит/с (по умолчанию), 19.2 кбит/с, 38.4 кбит/с, 57.6 кбит/с, 76.8 кбит/с, 115.2 кбит/с,
Четность	Без контроля четности (по умолчанию), четный, нечетный, метка (бит четности всегда установлен в 1), пробел (бит четности всегда установлен в 0)
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	Аппаратное, программное
Время ожидания	от 0 до 65535 мс

Таблица А- 252 Источник питания

Технические характеристики	CM 1241 RS232
Потеря мощности (рассеяние)	1 Вт
От +5 В пост.т.	200 мА

A.14 Коммуникационные интерфейсы

Таблица А- 253 RS232 соединитель (штекер)

Контакт	Описание	Соединитель (штекер)	Контакт	Описание
1 DCD	Обнаруж.инф.сигнала: вход		6 DSR	Источник данных готов: вход
2 RxD	Данные, принятые от DCE: вход		7 RTS	Запрос на передачу: выход
3 TxD	Данные, переданные в DCE: выход		8 CTS	Разрешение на передачу: вход
4 DTR	Терминал данных готов: выход		9 RI	Кольцевой индикатор (не используется)
5 GND	Логическая земля		SHELL	Корпусная земля

A.14.4.3. Технические характеристики коммуникационного модуля CM 1241 RS422/485

Технические характеристики CM 1241 RS422/485

Таблица А- 254 Общие технические данные

Технические характеристики	CM 1241 RS232
Номер для заказа	6ES7 241-1CH32-0XB0
Габариты Ш x В x Г (мм)	30 x 100 x 75
Масса	155 г

Таблица А- 255 Передатчик и приемник

Технические характеристики	CM 1241 RS422/485
Тип	RS422 или RS485, 9-контактный D- sub гнездовой соединитель
Динамический диапазон для синфазного сигнала	от -7 до +12 В, 1 секунда, 3 действ. непрерывно
Напряжение дифференциального выхода передатчика	2 В мин. при $R_L = 100 \text{ Ом}$ 1.5 В мин. при $R_L = 54 \text{ Ом}$
Нагрузка и смещение	10 кОм на +5 В для В, PROFIBUS контакт 3 10 кОм на GND для А, PROFIBUS контакт 8 Предлагается опция с внутренним смещением или без смещения. Во всех случаях требуется внешняя концевая нагрузка, смотри Смещение и концевая нагрузка соединителя сети RS485 (стр. 862) и Конфигурирование RS422 и RS485 (стр. 905) в системном руководстве по программируемому контроллеру S7-1200.
Входной импеданс приемника	5.4 кОм мин. включая нагрузку
Пороговая чувствительность приемника	+/- 0.2 В мин., 60 мВ типовой гистерезис
Электрическая развязка RS485 сигнал относительно корпуса RS485 сигнал относительно общего провода	500 В пер.т., 1 минута
Длина экранированного кабеля	1000 м макс. (в зависимости от скорости обмена)
Скорость обмена	300 бод, 600 бод, 1.2 кбит/с, 2.4 кбит/с, 4.8 кбит/с, 9.6 кбит/с (по умолчанию), 19.2 кбит/с, 38.4 кбит/с, 57.6 кбит/с, 76.8 кбит/с, 115.2 кбит/с,
Четность	Без контроля четности (по умолчанию), четный, нечетный, метка (бит четности всегда установлен в 1), пробел (бит четности всегда установлен в 0)

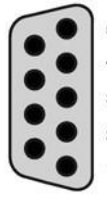
A.15 Удаленное управление (TS адаптер и модульный TS адаптер)

Технические характеристики	CM 1241 RS422/485
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	XON/XOFF поддерживается для режима RS422
Время ожидания	от 0 до 65535 мс

Таблица А- 256 Источник питания

Технические характеристики	CM 1241 RS422/485
Потеря мощности (рассеяние)	1,1 Вт
От +5 В пост.т.	220 мА

Таблица А- 257 RS485 или RS422 соединитель (гнездо)

Контакт	Описание	Соединитель (гнездо)	Контакт	Описание
1	Логическая или коммуникационная земля		6 PWR	+5 В с последовательным подключением резистора 100 Ом: выход
2 TxD+ ¹	Подключается для RS422 Не используется для RS485: выход		7	Не подключается
3 TxD+	Сигнал В (RxD/TxD+): вход/выход		8 TXD-	Сигнал А (RxD/TxD-): вход/выход
4 RTS ²	Запрос на передачу (TTL уровень): выход		9 TXD- ¹	Подключается для RS422 Не используется для RS485: выход
5 GND	Логическая или коммуникационная земля		Корпус	Корпусная земля

¹ Контакты 2 и 9 используются только как сигналы передачи для RS422.

² RTS - это сигнал уровня TTL и может использоваться для управления другим полудуплексным устройством на базе этого сигнала. Он активен, когда Вы осуществляете передачу, и неактивен все остальное время.

A.15. Удаленное управление (TS адаптер и модульный TS адаптер)

Следующие руководства содержат технические характеристики TS адаптера IE Basic и модульного TS адаптера:

- Промышленные инструменты программирования
Модульный TS адаптер
- Промышленные инструменты программирования
TS адаптер IE Basic

Для получения дополнительной информации и документации по продукту, обратитесь к веб-сайту каталога продукции для TS адаптера

(<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapt%20e%20IE%20basic&tab=>).

A.16. Карта памяти SIMATIC

Номер для заказа	Емкость
6ES7 954-8LP01-0AA0	2 ГБ
6ES7 954-8LL02-0AA0	256 МБ
6ES7 954-8LF02-0AA0	24 МБ
6ES7 954-8LE02-0AA0	12 МБ
6ES7 954-8LC02-0AA0	4 МБ

A.17. Имитатор входов

Таблица А- 258 Общие технические данные

Технические характеристики	Имитатор на 8 позиций	Имитатор на 14 позиций	Имитатор для CPU 1217C
Номер для заказа	6ES7 274-1XF30-0XA0	6ES7 274-1XH30-0XA0	6ES7 274-1XK30-0XA0
Габариты Ш x В x Г (мм)	43 x 35 x 23	67 x 35 x 23	93 x 40 x 23
Масса	20 грамм	30 грамм	43 грамм
Точки ввода	8	14	14
Используется с ЦПУ	CPU 1211C, CPU 1212C	CPU 1214C, CPU 1215C	CPU 1217C

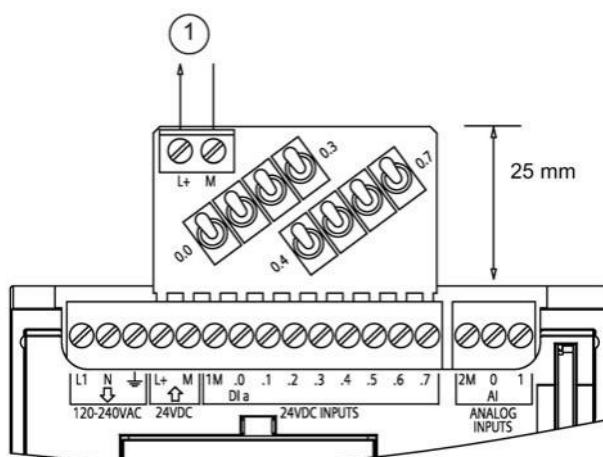
ПРЕДУПРЕЖДЕНИЕ

Безопасное использование имитаторов

Эти имитаторы входов не одобрены для использования во взрывоопасных производственных помещениях класса I, раздела 2, и класса I, зоны 2. Переключатели представляют потенциальную опасность возникновения искры / взрыва, если используются в помещениях класса I, раздела 2 или класса I, зоны 2. Несанкционированное использование могло бы привести к смерти или тяжелым травмам персонала и/или повреждению оборудования.

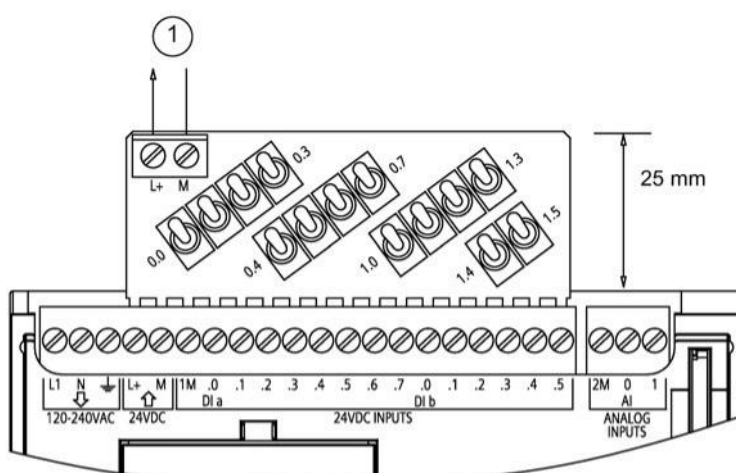
Используйте эти имитаторы входов только в безопасных помещениях. Не используйте их в помещениях класса I, раздела 2, и класса I, зоны 2.

8-позиционный имитатор (6ES7 274-1XF30-0XA0)



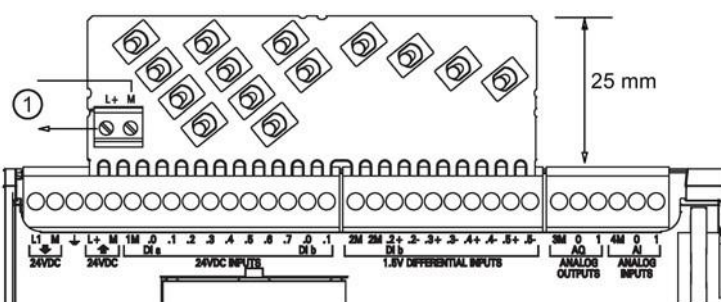
① Питание датчиков
24 В пост.т.

14-позиционный имитатор (6ES7 274-1XF30-0XA0)



① Питание датчиков
24 В пост.т.

Имитатор для ЦПУ 1217C (6ES7 274-1XK30-0XA0)



① Питание датчиков
24 В пост.т.

A.18. Модуль потенциометра для S7-1200

Модуль потенциометра S7-1200 является дополнительной принадлежностью для ЦПУ S7-1200. Каждый потенциометр создает выходное напряжение, пропорциональное положению потенциометра, чтобы управлять каждым из двух аналоговых входов ЦПУ, от 0 до 10 В постоянного тока. Чтобы установить потенциометр:

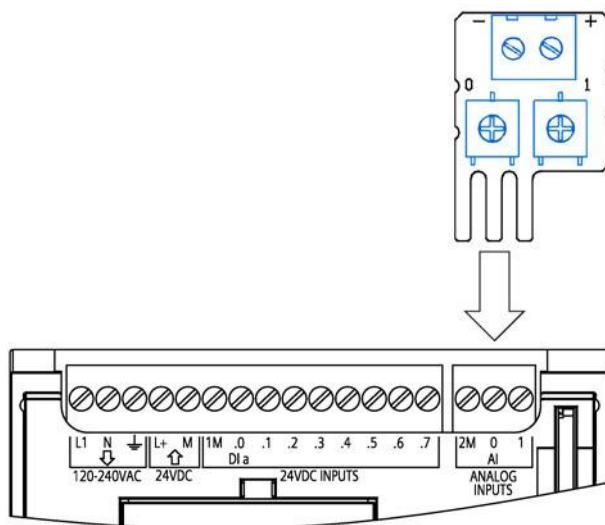
1. Вставьте 'пальцы' печатной платы в любую клеммную колодку аналогового входа ЦПУ S7-1200 и подключите внешний источник питания постоянного тока к 2-позиционному соединителю на модуле потенциометра.
2. Используйте небольшую отвертку, чтобы внести корректировки: поверните потенциометр по часовой стрелке (вправо), чтобы увеличить выходное напряжение, и против часовой стрелки (влево), чтобы уменьшить выходное напряжение.

Примечание

Следуйте инструкциям по работе с электрически чувствительными устройствами при обращении с модулем потенциометра S7-1200.

Технические данные	Модуль потенциометра S7-1200
Номер для заказа	6ES7 274-1XA30-0XA0
Используется с ЦПУ	Любые ЦПУ S7-1200
Количество потенциометров	2
Габариты Ш x В x Г (мм)	20 x 33 x 14
Вес	26 г
Подаваемое пользователем на 2-позиционный соединитель входное напряжение ¹ (Класс 2, ограниченная мощность или питание датчиков от ПЛК)	от 16.4 до 28.8 В пост.т.
Длина кабеля (метры)/тип	<30 м, экранированная витая пара
Входной потребляемый ток	10 мА макс.
Выходное напряжение потенциометра на аналоговые входы ЦПУ S7-1200 ¹	от 0 до 10.5 В пост.т. мин.
Электрическая развязка	Не изолировано
Диапазон температур окружающей среды	от -20 °C до 60 °C

¹ Стабильность выходного напряжения модуля потенциометра зависит от качества подаваемого пользователем на 2-позиционный соединитель входного напряжения - рассматривайте его как напряжение аналогового входа.

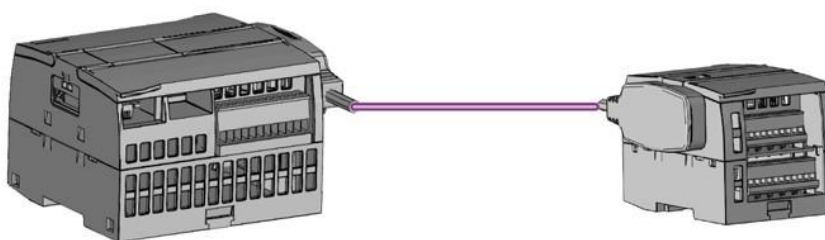


A.19. Кабель для расширения ввода/вывода

Таблица A- 259 Кабели расширения

Технические данные	
Номер для заказа	6ES7 290-6AA30-0XA0
Длина кабеля	2 м
Вес	200 г

Обратитесь к разделу установки (стр. 66) для получения информации об установке и удалении кабеля расширения S7-1200.



А.20. Сопутствующие продукты

А.20.1. Модуль питания РМ 1207

РМ 1207 является модулем источника питания для SIMATIC S7-1200. Это имеет следующие характеристики:

- Вход 120/230 В пер.тока, выход 24 В пост.т./2.5 А
- Номер для заказа 6ESP 332-1SH71-4AA0

Для получения дополнительной информации относительно этого продукта и документации по нему обратитесь на веб-сайт каталога продукции относительно РМ 1207 (<https://eb.automation.siemens.com/mall/en/de/Catalog/Product/6AG1332-1SH71-4AA0>).

А.20.2. Компактный коммутационный модуль CSM 1277

CSM1277 является компактным коммутационным модулем Industrial Ethernet. Он может использоваться, чтобы размножить Ethernet-интерфейс S7-1200, чтобы позволить одновременные коммуникации с операторскими панелями, программаторами или другими контроллерами. Он обладает следующими характеристиками:

- 4 x RJ45 гнезда для подключения к Industrial Ethernet
- 3 контактная съемная клеммная колодка для подключения внешнего питания 24 В пост.тока в верхней части
- Светодиоды для диагностики и отображения состояния портов Industrial Ethernet
- Номер для заказа 6GK7 277-1AA00-0AA0

Для получения дополнительной информации об этом продукте и документации по продукту, обратитесь на веб-сайт каталога продукции относительно CSM1277 ([https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277 & tab=](https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=)).

А.20.3. Коммуникационный модуль CANopen

Коммуникационный модуль CANopen является сменным модулем между ПЛК SIMATIC S7-1200 и любым устройством, использующим CANopen. Коммуникационный модуль CANopen может быть сконфигурирован как ведущее, либо как ведомое устройство. Существует два коммуникационных модуля CANopen: модуль CANopen (номер для заказа 021620-B), и модуль CANopen (усиленный) (номер для заказа 021730-B).

Модуль CANopen обладает следующими характеристиками:

- Возможность подключения до 3 модулей на ЦПУ
- Подключение до 16 ведомых узлов CANopen
- 256-байтов ввода и 256-байтов вывода на модуль
- 3 светодиода предоставляющих диагностическую информацию о модуле, сети и состоянии ввода-вывода
- Поддержка хранения конфигурации сети CANopen в ПЛК
- Модуль интегрируется в аппаратном каталоге конфигурационных средств TIA Portal
- Конфигурирование CANopen посредством CANopen Configuration Studio (включено) или через любой другой внешний инструмент конфигурирования CANopen
- Удовлетворяет коммуникационным профилям CANopen CiA 301 rev. 4.2 и CiA 302 rev. 4.1
- Поддерживает прозрачный CAN 2.0 А для обработки пользовательского протокола
- Доступны предварительно созданные функциональные блоки, для программирования ПЛК в TIA Portal
- Коммуникационные модули CANopen включают в себя; DSUB с винтовыми клеммами для подсети. CD с набором конфигурационных инструментов для CANopen и конфигурационный USB-кабель

Для получения дополнительной информации об этом продукте и документации по продукту, обратитесь на веб-сайт каталога продукции относительно CM CANopen.

А.20.4. Коммуникационный модуль RF120C

RF10C позволяет простое подключение систем RFID от Сименс и систем считывания кода к S7-1200. Считыватель подключается к RF120C посредством соединения "точка-к-точке". До трех коммуникационных модулей могут быть подключены к S7-1200 слева от ЦПУ. Коммуникационный модуль RF120C конфигурируется через TIA Portal. Номер заказа для коммуникационного модуля RF120C - 6GT2002-0LA00.

Для получения дополнительной информации об этом продукте и документации по продукту, обратитесь на веб-сайт каталога продукции относительно RF120C.

Расчет баланса мощностей

ЦПУ имеет внутренний источник для питания самого ЦПУ, любых модулей расширения, а также других пользовательских потребителей напряжения 24 В постоянного тока.

Существует три типа модулей расширения:

- Сигнальные модули (SM) установлены справа от ЦПУ. Каждый ЦПУ позволяет подключение максимального количества возможных сигнальных модулей безотносительно к балансу мощности.
 - CPU 1214C, CPU 1215C и CPU 1217C допускают подключение 8 модулей;
 - CPU 1212C допускает подключение 2 модулей;
 - CPU 1211C не допускает подключение сигнальных модулей.
- Коммуникационные модули (CM) устанавливаются слева от ЦПУ. Максимум 3 модуля допустимы для любого ЦПУ безотносительно к балансу мощности.
- Сигнальные платы (SB), коммуникационные платы (CB) и батарейные платы (BB) устанавливаются сверху ЦПУ. Максимум 1 сигнальная, коммуникационная или батарейная плата допустима для любого ЦПУ.

Используйте следующую информацию в качестве руководства для определения того, какую мощность (или ток) ЦПУ может предоставить для Вашей конфигурации.

Каждый ЦП обеспечивает питание как 24 В пост. тока, так и 5 В пост. тока:

- ЦПУ обеспечивает питание 5 В пост. тока для модулей расширения, когда модуль расширения подключен. Если требования по питанию 5 В пост. тока для модулей расширения превышают баланс мощности ЦПУ, Вы должны удалить модули расширения, пока требования не будут удовлетворять балансу мощности.
- Каждый ЦПУ имеет источник питания датчиков 24 В пост.тока, который может предоставить питание для локальных входных каналов или для обмоток реле на модулях расширения. Если требования по питанию 24 В пост. тока превышает баланс мощности ЦПУ, Вы можете добавить внешний источник питания 24 В пост. тока, чтобы обеспечить питание модулей расширения. Вы должны вручную подключить питание 24 В пост. тока к входным каналам или обмоткам реле.



ПРЕДУПРЕЖДЕНИЕ

Подключение внешнего источника питания 24 В пост. тока параллельно с источником питания датчиков может привести к конфликту между двумя источниками, поскольку каждый стремится установить свой собственный предпочтительный уровень выходного напряжения .

Результатом этого конфликта может явиться сокращение срока службы или немедленный отказ одного или обоих источников питания с последующей непредсказуемой работой ПЛК системы.

Непредсказуемая работа может привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Источник питания датчиков в ЦПУ и любой внешний источник питания должны обеспечивать питание для различных каналов. Разрешено соединение только общих проводников.

Некоторые из входных портов питания 24 В в системе ПЛК соединены между собой, причем общий провод логики соединяет несколько клемм М. Вход источника питания 24 В на CPU, вход питания катушек на SM и не имеющий потенциальной развязки вход питания аналогового модуля являются примерами цепей, которые соединяются друг с другом, если в технических данных они обозначены, как не имеющие потенциальной развязки. Все не имеющие потенциальной развязки клеммы М должны быть присоединены к одному и тому же внешнему опорному потенциалу.

ПРЕДУПРЕЖДЕНИЕ

Подключение не имеющих потенциальной развязки М клемм к разным опорным потенциалам вызовет протекание непредусмотренных токов, которые могут вызвать повреждение или непредсказуемое поведение ПЛК и подключенного оборудования.

Такое повреждение или непредсказуемая работа могут привести к смерти, тяжелым телесным повреждениям и/или материальному ущербу.

Всегда будьте уверены, что все неизолированные М терминалы в ПЛК системе, соединены с одним и тем же опорным потенциалом .

Информация о балансах мощности ЦПУ и потребляемой мощности сигнальных модулей представлена в Технических данных (стр. 1099).

Примечание

Превышение баланса мощности ЦПУ может привести к невозможности подключения, максимального количества модулей, допустимого для Вашего ЦПУ.

Примерный баланс мощности

Следующий пример показывает предварительный расчет потребляемой мощности для конфигурации, которая включает в себя CPU 1214C AC/DC/Relay, одну SB 1223 2 x входа 24 В пост.т. / 2 x выхода 24 В пост.т., один CM 1241, три SM 1223 с 8 входами пост.т. / 8 релейными выходами и один SM 1221 8 входами пост.т.. В этом примере представлено в общей сложности 48 входов и 36 выхода.

Примечание

ЦПУ уже выделена мощность, требуемая для управления внутренними обмотками реле. Вы не должны включать мощность, потребляемую внутренними обмотками реле в расчет баланса мощности.

ЦПУ в этом примере обеспечивает достаточный ток от источника 5 В пост. тока для SM, но не обеспечивает достаточный ток от источника питания датчиков 24 В пост. тока для всех входов и обмоток реле модулей расширения. Вводу-выводу требуется 456 мА, а ЦПУ обеспечивает только 400 мА. Этой конфигурации требуется дополнительный источник с током, по крайней мере, в 56 мА при напряжении 24 В пост. тока для управления всеми включенными входами и выходами 24 В пост. тока.

Таблица В- 1 Примерный баланс мощности

Баланс мощности ЦПУ	5 В пост. тока	24 В пост. тока
CPU 1214C AC/DC/Relay	1600 мА	400 мА
<i>Минус</i>		
Потребности системы	5 В пост. тока	24 В пост. тока
CPU 1214C, 14 входов	-	$14 * 4 \text{ мА} = 56 \text{ мА}$
1 SB 1223 2 х вход 24 В пост.тока/ 2 х выход 24 В пост.тока	50 мА	$2 * 4 \text{ мА} = 8 \text{ мА}$
1 CM 1241 RS422/485, питание 5 В	220 мА	
3 SM 1223, питание 5 В	$3 * 145 \text{ мА} = 435 \text{ мА}$	-
1 SM 1221, питание 5 В	$1 * 105 \text{ мА} = 105 \text{ мА}$	-
3 SM 1223, 8 входов в каждом	-	$3 * 8 * 4 \text{ мА} = 96 \text{ мА}$
3 SM 1223, 8 обмоток реле в каждом	-	$3 * 8 * 11 \text{ мА} = 264 \text{ мА}$
1 SM 1221, 8 входов в каждом	-	$8 * 4 \text{ мА} = 32 \text{ мА}$
Суммарные потребности	810 мА	456 мА
<i>Равно</i>		
Текущий баланс	5 В пост. тока	24 В пост. тока
Суммарный текущий баланс	790 мА	(56 мА)

Форма для вычисления Вашего баланса мощности

Используйте следующую таблицу, чтобы определить, какая мощность (или ток) может обеспечить ЦПУ S7-1200 для Вашей конфигурации. Обратитесь к техническим данным (стр. 1099) для получения балансов мощности Вашей модели ЦПУ и потребляемой мощности Ваших сигнальных модулей.

Таблица В- 2 Вычисление баланса мощности

Баланс мощности ЦПУ	5 В пост. тока	24 В пост. тока
<i>Минус</i>		
Потребности системы	5 В пост. тока	24 В пост. тока
Суммарные потребности		
<i>Равно</i>		
Текущий баланс	5 В пост. тока	24 В пост. тока
Суммарный текущий баланс		

Номера для заказа

С.1 Модули ЦПУ

Таблица С- 1 ЦПУ S7-1200

Модель ЦПУ		Номер для заказа
CPU 1211C	CPU 1211C DC/DC/DC	6ES7 211-1AE40-0XB0
	CPU 1211C AC/DC/Relay	6ES7 211-1BE40-0XB0
	CPU 1211C DC/DC/Relay	6ES7 211-1HE40-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7 212-1AE40-0XB0
	CPU 1212C AC/DC/Relay	6ES7 212-1BE40-0XB0
	CPU 1212C DC/DC/Relay	6ES7 212-1HE40-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7 214-1AG40-0XB0
	CPU 1214C AC/DC/Relay	6ES7 214-1BG40-0XB0
	CPU 1214C DC/DC/Relay	6ES7 214-1HG40-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7 215-1AG40-0XB0
	CPU 1215C AC/DC/Relay	6ES7 215-1BG40-0XB0
	CPU 1215C DC/DC/Relay	6ES7 215-1HG40-0XB0
CPU 1217C	CPU 1217C DC/DC/DC	6ES7 217-1AG40-0XB0

С.2 Сигнальные модули (SM), сигнальные платы (SB) и батарейные платы (BB)

Таблица С- 2 Сигнальные модули (SM)

Сигнальные модули		Номер для заказа
Цифровой ввод	SM 1221 8 х вход 24 В пост.т. (потребитель/источник)	6ES7 221-1BF32-0XB0
	SM 1221 16 х вход 24 В пост.т. (потребитель/источник)	6ES7 221-1BH32-0XB0
Цифровой вывод	SM 1222 8 х выход 24 В пост.т. (источник)	6ES7 222-1BF32-0XB0
	SM 1222 16 х выход 24 В пост.т. (источник)	6ES7 222-1BH32-0XB0
	SM 1222 8 х релейный выход	6ES7 222-1HF32-0XB0
	SM 1222 8 х релейный выход (с переключением)	6ES7 222-1XF32-0XB0
	SM 1222 16 х релейный выход	6ES7 222-1HH32-0XB0
	SM 1222 8 х вход 24 В пост.т. (потребитель/источник) / 8 х выход 24 В пост.т. (источник)	6ES7 223-1BH32-0XB0
Цифровой ввод / вывод	SM 1223 16 х вход 24 В пост.т. (потребитель/источник) / 16 х выход 24 В пост.т. (источник)	6ES7 223-1BL32-0XB0
	SM 1223 8 х вход 24 В пост.т. (потребитель/источник) / 8 х релейный выход	6ES7 223-1PH32-0XB0
	SM 1223 16 х вход 24 В пост.т. (потребитель/источник) / 16 х релейный выход	6ES7 223-1PL32-0XB0
	SM 1223 8 х вход 120/230 В пер.т. (потребитель/источник) / 8 х релейный выход	6ES7 223-1QH32-0XB0
	SM 1231 4 х аналоговый вход	6ES7 231-4HD32-0XB0
Аналоговый ввод	SM 1231 8 х аналоговый вход	6ES7 231-4HF32-0XB0
	SM 1231 4 х аналоговый вход х 16 бит (high feature)	6ES7 231-5ND32-0XB0
	SM 1232 2 х аналоговый выход	6ES7 232-4HB32-0XB0
Аналоговый вывод	SM 1232 4 х аналоговый выход	6ES7 232-4HD32-0XB0
	SM 1234 4 х аналоговый вход / 2 х аналоговый выход	6ES7 234-4HE32-0XB0
Терморезисторы (RTD) и термопары	SM 1231 TC 4 х 16 бит	6ES7 231-5QD32-0XB0
	SM 1231 TC 8 х 16 бит	6ES7 231-5QF32-0XB0
	SM 1231 RTD 4 х 16 бит	6ES7 231-5PD32-0XB0
	SM 1231 RTD 8 х 16 бит	6ES7 231-5PF32-0XB0
Технологические модули	SM 1278 4xIO-Link Master	6ES7 278-4BD32-0XB0

Таблица С- 3 Сигнальные платы (SB) и батарейные платы (BB)

Сигнальные и батарейные платы		Номер для заказа
Цифровой ввод	SB 1221 200 kHz 4 x вход 24 В пост.т. (источник)	6ES7 221-3BD30-0XB0
	SB 1221 200 kHz 4 x вход 5 В пост.т. (источник)	6ES7 221-3AD30-0XB0
Цифровой вывод	SB 1222 200 kHz 4 x выход 24 В пост.т. (потребитель / источник)	6ES7 222-1BD30-0XB0
	SB 1222 200 kHz 4 x выход 5 В пост.т. (потребитель / источник)	6ES7 222-1AD30-0XB0
Цифровой ввод / вывод	SB 1223 2 x вход 24 В пост.т. (потребитель) / 2 x выход 24 В пост.т. (источник)	6ES7 223-0BD30-0XB0
	SB 1223 200 kHz 2 x вход 24 В пост.т. (источник) / 2 x выход 24 В пост.т. (потребитель/источник)	6ES7 223-3BD30-0XB0
	SB 1223 200 kHz 2 x вход 5 В пост.т. (источник) / 2 x выход 5 В пост.т. (потребитель/источник)	6ES7 223-3AD30-0XB0
Аналоговая	SB 1232 1 аналоговый выход	6ES7 232-4HA30-0XB0
	SB 1231 1 аналоговый вход	6ES7 231-4HA30-0XB0
	SB 1231 1 аналоговый вход для термопар	6ES7 231-5QA30-0XB0
	SB 1231 1 аналоговый вход для терморезисторов	6ES7 231-5PA30-0XB0
Батарейная	BB 1297 батарейная плата (батарея, Тип CR1025 не включена)	6ES7 297-0AX30-0XA0

С.3 Коммуникации

Таблица С- 4 Коммуникационные модули (СМ)

Коммуникационный модуль (СМ)			Номер для заказа
RS232, RS422 и RS485	CM 1241 RS232	RS232	6ES7 241-1AH32-0XB0
	CM 1241 RS422/485	RS422/485	6ES7 241-1CH32-0XB0
PROFIBUS	CM 1243-5	Ведущее устройство PROFIBUS шины	6GK7 243-5DX30-0XE0
	CM 1242-5	Ведомое устройство PROFIBUS шины	6GK7 242-5DX30-0XE0
AS-i Master	CM 1243-2	Ведущее устройство AS-i шины	3RK7 243-2AA30-0XB0
RF120C CM	RF120C	Считыватель RFID	6GT2 002-0LA00

Таблица С- 5 Коммуникационная плата (СВ)

Коммуникационная плата (СВ)			Номер для заказа
RS485	CB 1241 RS485	RS485	6ES7 241-1CH30-1XB0

Таблица С- 6 Коммуникационный процессор (СР)

СР	Интерфейс	Номер для заказа
CP 1242-7 GPRS V2	GPRS	6GK7 242-7KX31-0XE0
CP 1243-7 LTE-EU	LTE	6GK7 243-7KX30-0XE0
CP 1243-1 DNP3	IE-интерфейс	6GK7 243-1JX30-0XE0
CP 1243-1 IEC	IE- интерфейс	6GK7 243-1PX30-0XE0
CP 1243-1	IE- интерфейс	6GK7 243-1BX30-0XE0
CP 1243-1 PCC	IE- интерфейс	6GK7 243-1HX30-0XE0
CP 1243-8 ST7	IE- и последовательный интерфейс	6GK7 243-8RX30-0XE0

Таблица С- 7 Телесервис

TS адаптер	Номер для заказа
TS адаптер IE Basic	6ES7 972-0EB00-0XA0
TS адаптер IE Advanced	6ES7 972-0EA00-0XA0
TS модуль GSM	6GK7 972-0MG00-0XA0
TS модуль RS232	6ES7 972-0MS00-0XA0
TS модуль Modem	6ES7 972-0MM00-0XA0
TS модуль ISDN	6ES7 972-0MD00-0XA0

Таблица С- 8 Дополнительные принадлежности

Дополнительная принадлежность			Номер для заказа
Антенна	ANT794-4MR	GSM/GPRS антенна	6NH9 860-1AA00
	ANT794-3M	Плоская антенна	6NH9 870-1AA00

Таблица С- 9 Соединители

Тип соединителя		Номер для заказа
RS485	отвод кабеля под углом 35 градусов, винтовое клеммное соединение	6ES7 972-0BA42-0XA0
	отвод кабеля под углом 35 градусов, соединение FastConnect	6ES7 972-0BA60-0XA0

С.4 Отказоустойчивые ЦПУ и сигнальные модули

Таблица С- 10 Отказоустойчивые ЦПУ

Модели отказоустойчивых ЦПУ		Номер для заказа
CPU 1214FC	CPU 1214FC DC/DC/DC	6ES7 214-1AF40-0XB0
	CPU 1214FC DC/DC/Relay	6ES7 214-1HF40-0XB0
CPU 1215FC	CPU 1215FC DC/DC/DC	6ES7 215-1AF40-0XB0
	CPU 1215FC DC/DC/Relay	6ES7 215-1HF40-0XB0

Таблица С- 11 Отказоустойчивые сигнальные модули

Сигнальные модули Functional Safety		Номер для заказа
Цифровой ввод	SM 1226 F-DI 16 x 24 VDC	6ES7 226-6BA32-0XB0
Цифровой вывод	SM 1226 F-DQ 4 x 24 VDC	6ES7 226-6DA32-0XB0
	SM 1226 F-DQ 2 x Relay	6ES7 226-6RA32-0XB0

С.5 Прочие модули

Таблица С- 12 Сопутствующие модули

Изделие		Номер для заказа
Источник питания	Источник питания PM 1207	6EP1 332-1SH71-4AA0
Ethernet-коммутатор	CSM 1277 Ethernet-коммутатор с 4 портами	6GK7 277-1AA10-0AA0
CM CANopen	CANopen для SIMATIC S7-1200	021620-B
	CANopen (усиленный) для SIMATIC S7-1200	021730-B
RF120C	Коммуникационный модуль RF120C	6GT2002-0LA00

С.6 Карты памяти

Таблица С- 13 Карты памяти

Карты памяти SIMATIC	Номер для заказа
SIMATIC MC 2 GB	6ES7 954-8LP01-0AA0
SIMATIC MC 256 MB	6ES7 954-8LL02-0AA0
SIMATIC MC 24 MB	6ES7 954-8LF02-0AA0
SIMATIC MC 12 MB	6ES7 954-8LE02-0AA0
SIMATIC MC 4 MB	6ES7 954-8LC02-0AA0

С.7 Устройства человеко-машинного интерфейса серии Basic

Таблица С- 14 Устройства человеко-машинного интерфейса

Панели для человеко-машинного интерфейса серии Basic	Номер для заказа
KTP400 Basic (монохромная, PN)	6AV2 123-2DB03-0AX0
KTP700 Basic	6AV2 123-2GB03-0AX0
KTP700 Basic DP	6AV2 123-2GA03-0AX0
KTP900 Basic	6AV2 123-2JB03-0AX0
KTP1200 Basic	6AV2 123-2MB03-0AX0
KTP1200 Basic DP	6AV2 123-2MA03-0AX0

С.8 Запасные части и другое аппаратное обеспечение

Таблица С- 15 Кабели расширения, имитаторы и концевые фиксаторы

Элемент	Номер для заказа
Кабель расширения ввода-вывода	Кабель расширения ввода-вывода, 2 м 6ES7 290-6AA30-0XA0
Имитатор ввода-вывода	Имитатор (1211C/1212C - 8 позиций) 6ES7 274-1XF30-0XA0
	Имитатор (1214C/1215C - 14 позиций) 6ES7 274-1XH30-0XA0
	Имитатор, CPU 1217C 6ES7 274-1XK30-0XA0
Модуль потенциометра	Модуль потенциометра для S7-1200 6ES7 274-1XA30-0XA0
Компенсатор натяжения Ethernet	Компенсатор натяжения для одиночного порта RJ45 6ES7 290-3AA30-0XA0
	Компенсатор натяжения для двойного порта RJ45 6ES7 290-3AB30-0XA0
Комплект запасной дверцы	CPU 1211C/1212C 6ES7 291-1AA30-0XA0
	CPU 1214C 6ES7 291-1AB30-0XA0
	CPU 1215C 6ES7 291-1AC30-0XA0
	CPU 1217C 6ES7 291-1AD30-0XA0

Элемент		Номер для заказа
	Сигнальный модуль, 45 мм	6ES7 291-1BA30-0XA0
	Сигнальный модуль, 70 мм	6ES7 291-1BB30-0XA0
	Коммуникационный модуль (для использования с 6ES7 2хх-xxx32-0XB0 и 6ES7 2хх-xxx30-0XB0 модулями)	6ES7 291-1CC30-0XA0
Концевой фиксатор	Концевой фиксатор из термопластика, 10 мм	8WA1808
	Концевой фиксатор из стали, 10.3 мм	8WA1805

Таблица С- 16 Комплекты запасных клеммных колодок для ЦПУ S7-1200 V4.0 и выше

Если у Вас ЦПУ S7-1200 V4.0 и выше (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
CPU 1211C DC/DC/DC (6ES7 211-1AE40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH30-0XA0	8 конт., луженые
	6ES7 292-1AP30-0XA0	14 конт., луженые
CPU 1211C DC/DC/Relay (6ES7 211-1HE40-0XB0)	6ES7 292-1AH30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH40-0XA0	8 конт., луженые, с мех.код.
	6ES7 292-1AP30-0XA0	14 конт., луженые
CPU 1211C AC/DC/Relay (6ES7 211-1BE40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH40-0XA0	8 конт., луженые, с мех.код.
	6ES7 292-1AP40-0XA0	14 конт., луженые, с мех.код.
CPU 1212C DC/DC/DC (6ES7 212-1AE40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH30-0XA0	8 конт., луженые
	6ES7 292-1AP30-0XA0	14 конт., луженые
CPU 1212C DC/DC/Relay (6ES7 212-1HE40-0XB0)	6ES7 292-1AH30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH40-0XA0	8 конт., луженые, с мех.код.
	6ES7 292-1AP30-0XA0	14 конт., луженые
CPU 1212C AC/DC/Relay (6ES7 212-1BE40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AH40-0XA0	8 конт., луженые, с мех.код.
	6ES7 292-1AP40-0XA0	14 конт., луженые, с мех.код.
CPU 1214C DC/DC/DC (6ES7 214-1AG40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AM30-0XA0	12 конт., луженые
	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1214C DC/DC/Relay (6ES7 214-1HG40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AM40-0XA0	12 конт., луженые, с мех.код.
	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1214C AC/DC/Relay (6ES7 214-1BG40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AM40-0XA0	12 конт., луженые, с мех.код.
	6ES7 292-1AV40-0XA0	20 конт., луженые, с мех.код.
CPU 1215C DC/DC/DC (6ES7 215-1AG40-0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
CPU 1215C DC/DC/Relay (6ES7 215-1HG40-0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
	6ES7 292-1AM40-0XA0	12 конт., луженые, с мех.код.
	6ES7 292-1AV30-0XA0	20 конт., луженые

Если у Вас ЦПУ S7-1200 V4.0 и выше (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
CPU 1215C AC/DC/Relay (6ES7 215-1BG40-0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
	6ES7 292-1AM40-0XA0	12 конт., луженые, с мех.код.
	6ES7 292-1AV40-0XA0	20 конт., луженые, с мех.код.
CPU 1217C DC/DC/DC (6ES7 217-1AG40-0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
	6ES7 292-1AK30-0XA0	10 конт., луженые
	6ES7 292-1AR30-0XA0	16 конт., луженые
	6ES7 292-1AT30-0XA0	18 конт., луженые

Таблица С- 17 Комплекты запасных клеммных колодок для сигнальных модулей (SM) S7-1200 V4.0 и выше

Если у Вас сигнальный модуль S7-1200 V4.0 и выше (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
SM1221 DI 8xDC (6ES7 221-1BF32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 8xDC (6ES7 222-1BF32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 8xRelay (6ES7 222-1HF32-0XB0)	6ES7 292-1AG40-0XA1	7 конт., луженые, с мех.кодом слева
SM1231 AI 4x13 bit (6ES7 231-4HD32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1232 AQ 2x14 bit (6ES7 232-4HB32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI4 x TC (6ES7 231-5QD32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI4 x 16 bit (6ES7 231-5ND32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1221 DI 16xDC (6ES7 221-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 16xDC (6ES7 222-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 16xRelay (6ES7 222-1HH32-0XB0)	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа
SM1223 DI 8xDC/DQ 8xDC (6ES7 223-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1223 8xDC/8xRelay (6ES7 223-1PH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа
SM1223 8xAC/8xRelay (6ES7 223-1QH32-0XB0)	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа
SM1234 AI 4 / AQ 2 (6ES7 234-4HE32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI 8x13 BIT (6ES7 231-4HF32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1232 AQ4x14 bit (6ES7 232-4HD32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI4 x RTD (6ES7 231-5PD32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI8 x TC (6ES7 231-5QF32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM 1278 IO LINK (6ES7 278-4BD32 0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 8xRelay (Changeover) (6ES7 222-1XF32-0XB0)	6ES7 292-1AL30-0XA0	11 конт., луженые
SM1223 DI 16xDC/DQ 16xDC (6ES7 223-1BL32-0XB0)	6ES7 292-1AL30-0XA0	11 конт., луженые
	6ES7 292-1AL40-0XA0	11 конт., луженые, с мех.код.
SM1231 AI8 x RTD (6ES7 231-5PF32-0XB0)	6ES7 231-5PF32-0XB0	11 конт., анод.золотом

Таблица С- 18 Комплекты запасных клеммных колодок для отказоустойчивых ЦПУ

Если у Вас отказоустойчивый ЦПУ (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
CPU 1214FC DC/DC/DC (6ES7 214-1AF40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AM30-0XA0	12 конт., луженые
	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1214FC DC/DC/Relay (6ES7 214-1HF40-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
	6ES7 292-1AM40-0XA0	12 конт., луженые, с мех.код.
	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1215FC DC/DC/DC (6ES7 215 1AF40 0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
	6ES7 292-1AM30-0XA0	12 конт., луженые
	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1215FC DC/DC/Relay (6ES7 215 1HF40 0XB0)	6ES7 292-1BF30-0XB01	6 конт., анод.золотом
	6ES7 292-1AM40-0XA0	2 конт., луженые, с мех.код.
	6ES7 292-1AV30-0XA0	20 конт., луженые

Таблица С- 19 Комплекты запасных клеммных колодок для отказоустойчивых сигнальных модулей

Если у Вас отказоустойчивый сигнальный модуль (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
SM 1226 F-DI (6ES7 226-6BA32-0XB0)	6ES7 292-1AL30-0XA0	11 конт., луженые
SM 1226 F-DQ (6ES7 226-6DA32-0XB0)	6ES7 292-1AL30-0XA0	11 конт., луженые
SM 1226 F-Relay (6ES7 226-6RA32-0XB0)	6ES7 292-1AL40-0XA0	11 конт., луженые, с мех.код.

С.9 Средства программирования

Таблица С- 20 Средства программирования

Программное обеспечение SIMATIC		Номер для заказа
Средства про- граммирования	STEP 7 Basic V13	6ES7 822-0AA01-0YA0
	STEP 7 Professional V13	6ES7 822-1AA01-0YA5
Программные средства для визуализации	WinCC Basic V12 SP1	6AV2100-0AA01-0AA0
	WinCC Comfort V12 SP1	6AV2101-0AA01-0AA5
	WinCC Advanced V12 SP1	6AV2102-0AA01-0AA5
	WinCC Professional 512 PowerTags V12 SP1	6AV2103-0DA01-0AA5
	WinCC Professional 4096 PowerTags V12 SP1	6AV2103-0HA01-0AA5
	WinCC Professional max. PowerTags V12 SP1	6AV2103-0XA01-0AA5

Замена устройств и совместимость запасных частей



Вы можете заменить свой ЦПУ V3.0 на ЦПУ V4.1 (стр. 161) и использовать Ваш существующий проект STEP 7, который Вы разработали для ЦПУ V3.0. Вы не можете обновить ЦПУ V3.0 до ЦПУ V4.1 посредством обновления встроенного ПО; Вы должны заменить аппаратные средства. Когда Вы заменяете ЦПУ V3.0 на ЦПУ V4.1, Вы, возможно, могли бы также захотеть проверить наличие и выполнить обновление встроенного ПО (стр. 141) Ваших подключенных сигнальных и коммуникационных модулей.

Примечание

Никакой обмен устройств невозможен с V4.1 на V3.0

Вы можете заменить ЦПУ V3.0 на ЦПУ V4.1, но Вы не можете заменить ЦПУ V4.1 на ЦПУ V3.0 после того, как Вы загружаете конфигурацию. Если Вы хотите просмотреть или иначе использовать Ваш существующий проект STEP 7 V3.0, сделайте архив своего проекта STEP 7 V3.0 до замены устройства.

Обратите внимание на то, что, если Вы не загрузили конфигурацию с заменой устройства, Вы можете отменить ее. Однако, после загрузки, Вы не можете отменить замену V3.0 на V4.1.

Вы должны принять к сведению некоторые изменения в конфигурации и функционировании между двумя версиями ЦПУ:

Организационные блоки

С помощью V4.1 Вы можете сконфигурировать выполнение ОБ с возможностью или без возможности прерывания (стр. 98). Для проектов от прежних ЦПУ V3.0 STEP 7, по умолчанию, устанавливает для всех ОБ отсутствие возможности прерывания.

STEP 7 устанавливает для всех приоритетов ОБ (стр. 98) значения, которые у них были в проекте STEP 7 для ЦПУ V3.0.

Вы можете впоследствии, если хотите, изменить возможность прерывания или настройки приоритета.

Стартовая информация ОБ прерывания по ошибке диагностики (стр. 92) содержит ссылку на субмодуль в целом, если отсутствует активное диагностическое событие.

Защита ЦПУ паролем

STEP 7 устанавливает уровень защиты паролем (стр. 197) для ЦПУ V4.1 эквивалентным уровнем защиты паролем, который был установлен для ЦПУ V3.0 и назначает пароль V3.0 для "Full access (no protection)" пароля для ЦПУ V4.1:

Уровень защиты V3.0	Уровень защиты V4.1
No protection	Полный доступ (отсутствие защиты)
Защита от записи	Доступ на чтение
Защита от записи / чтения	Доступ от HMI

Обратите внимание на то, что уровень доступа V4.1 "Отсутствие доступа (полная защита)" не существовал для V3.0.

Веб-сервер

Если Вы используете пользовательские Веб-страницы в своем проекте V3.0, сохраните их в Вашей папке установки проекта во вложенной папке "UserFiles\Webserver" до обновления Вашего проекта. Если Вы сохраните свои пользовательские страницы в этом месте, то сохранение проекта STEP 7 также приведет к сохранению пользовательских Веб-страниц.

Если Вы заменяете ЦПУ V3.0 на ЦПУ V4.1, Ваш проектные настройки для активации веб-сервера и HTTPS настройки (стр. 787) будут теми же, что и для V3.0. Вы можете затем сконфигурировать пользователей, полномочия, пароли (стр. 789) и языки (стр. 787) по мере необходимости, чтобы использовать веб-сервер. Если Вы не конфигурируете пользователей с дополнительными полномочиями, то Ваши возможности просмотра стандартных Веб-страниц ограничены (стр. 794).

ЦПУ S7-1200 V4.1 не поддерживает прежнего предварительно сконфигурированного пользователя "admin" и пароль.

Страница веб-сервера S7-1200 V3.0 "Data log" обеспечивает операцию "Download and Clear". Страница файлового браузера веб-сервера V4.1 (стр. 810), из которой Вы получаете доступ к журналам данных, больше не обеспечивает эту функцию. Вместо этого веб-сервер обеспечивает возможность загрузить, переименовать и удалить файлы журнала данных.

Несовместимость карт передачи

Вы не можете использовать карту передачи V3.0 (стр.133), чтобы перенести программу V3.0 в ЦПУ V4.1. Вы должны открыть проект V3.0 в STEP 7, изменить устройство на ЦПУ V4.1 (стр. 161) и загрузить проект STEP 7 в Ваш ЦПУ V4.1. После того, как Вы изменили свой проект на проект V4.1, Вы можете затем сделать карту передачи V4.1 для последующих переносов программы.

Коммуникации GET/PUT

По умолчанию коммуникации GET/PUT были активированы в V3.0. Когда Вы заменяете свой ЦПУ V3.0 на ЦПУ V4.1 (стр.161), Вы видите сообщение в разделе информации о совместимости, указывающее на то, что GET/PUT активированы..

Поддержка управления перемещением

ЦПУ S7-1200 V4.1 не поддерживают библиотеки управления перемещением V1.0 и V2.0. Если Вы выполняете замену устройства для проекта STEP 7 с библиотеками V1.0 или V2.0, то в ходе этого процесса они заменяются совместимыми командами управления перемещением V3.0 (стр. 571) при компиляции.

Если Вы выполняете замену устройства от ЦПУ V3.0 до ЦПУ V4.1 для проекта STEP 7, который содержит две различных версии команды управления перемещением (V3.0 и V5.0), обмен устройства заменяет совместимыми командами управления движения V5.0 (Страница 571) в компиляции.

Во время обмена устройства от ЦП V3.0 до ЦП V4.1 управление движением версия технологического объекта (ТО) автоматически не изменяется с V3.0 на V5.0. Если Вы хотите обновить до более поздних версий, Вы должны перейти к дереву инструкций и выбрать требуемую S7-1200 версию управления перемещением для своего проекта как показано в таблице ниже:

Версия ЦПУ	Допустимые версии управления перемещением
V4.1 (motion control V5.0)	V5.0 или V4.0 или V3.0
V4.0 (motion control V4.0)	V4.0 или V3.0
V3.0 (motion control V3.0)	V3.0

Существует различия в структуре для ТО управления перемещением версий V3.0 и V5.0. Все связанные блоки также изменяются. Интерфейсы блоков, таблицы наблюдения, и трассировки обновляются согласно новой структуре V5.0 для управления перемещением. Вы можете найти отличия между параметрами управления перемещением оси для ЦПУ V3.0 и ЦПУ V4.1 в следующих двух таблицах:

ЦПУ V3.0 (Управление перемещением V3.0)	ЦПУ V4.1 (Управление перемещением V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

Единственный параметр "commandtable", который переименован, является массивом инструкций:

V3.0	V4.1
Config.Command[]	Command[]

Примечание: Массив "Command[]" представляет собой UDT с типом "TO_CmdTab_Config_Command" в V3.0 и "TO_Struct_Command" в V4.1.

Изменения инструкций

Следующие инструкции имеют изменения в параметрах или поведении:

- RDREC и WRREC (стр. 351)
- CONV (стр. 269)

Коммуникации с HMI-панелью

Если у Вас была одна или более HMI-панелей (стр. 30), подключенная к Вашему ЦПУ S7-1200 V3.0, то коммуникации с ЦПУ S7-1200 V4.1 зависят от типа коммуникаций, которые Вы используете и версии встроенного ПО HMI-панели. Перекомпилируйте и загрузите свой проект в ЦПУ и HMI и/или обновите свое встроенное ПО Ваших HMI-панелей.

Требование по перекомпиляции программных блоков

После замены ЦПУ V3.0 на ЦПУ V4.1 Вы должны перекомпилировать все программные блоки, прежде чем Вы сможете загрузить их в ЦПУ V4.1. Кроме того, если у какой-либо из блоков имеет защиту ноу-хау (стр. 200) или защиту от копирования, связанную с серийным номером ПЛК (стр. 201), Вы должны снять защиту, прежде чем Вы скомпилируете и загрузите блоки. (Вы не обязаны, однако, деактивировать защиту от копирования, связанную с картой памяти.) После успешной компиляции Вы можете заново сконфигурировать защиту ноу-хау и/или защиту от копирования с привязкой к серийному номеру ПЛК. Обратите внимание на то, что, если Ваш проект содержит какие-либо блоки с защитой ноу-хау, предоставленные OEM (производитель оборудования), Вы должны связаться с OEM относительно возможности получения версии V4.1 для этих блоков.

Обычно Сименс рекомендует, чтобы Вы перекомпилировали аппаратную конфигурацию и программное обеспечение в STEP 7 и выполнили загрузку во все устройства в Вашем проекте после замены устройства. Исправьте любые ошибки, которые обнаруживаются при компиляции проекта, и заново скомпилируйте, пока у Вас не будет ошибок. Затем Вы можете загрузить проект в ЦПУ V4.1.

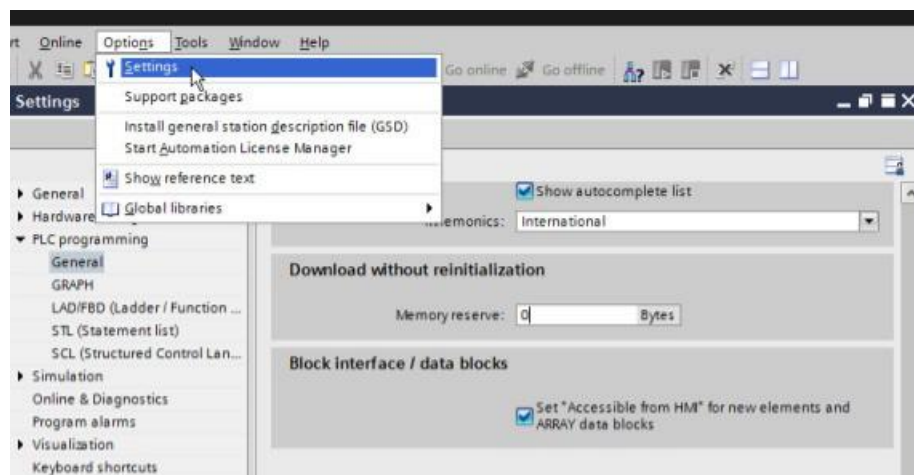
Проекты S7-1200 V3.0 могли бы не подойти для ЦПУ S7-1200 V4.1

S7-1200 V4.0 добавил резервную область в 100 байтов в каждый DB, чтобы поддерживать загрузку без реинициализации.

Вы можете удалить 100-байтовую резервную область из DB до попытки загрузки проекта V3.0 в ЦПУ V4.1.

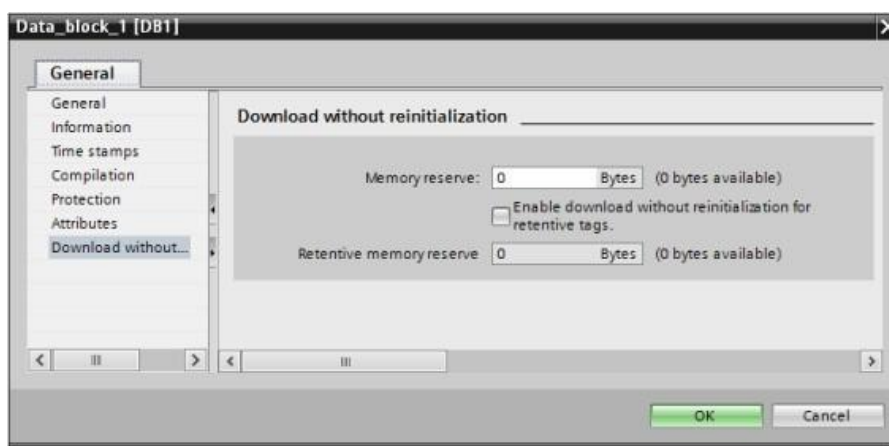
Чтобы удалить 100-байтовую резервную область, выполните следующие шаги, прежде чем Вы выполните замену устройства:

1. Из главного меню TIA Portal выберите команду меню Options > Settings.
2. Из навигационного дерева откройте узел PLC programming > General.
3. В области "Download without reinitialization", установите резерв памяти на 0 байтов.



Если Вы уже выполнили замену устройства, Вы должны удалить 100-байтовый резерв из каждого блока индивидуально:

1. Из дерева проекта щелкните правой кнопкой по блоку данных из папки "Program blocks" и выберите Properties из контекстного меню.
2. В диалоговом окне свойств блока данных выберите раздел "Download without reinitialization".
3. Установите резерв памяти на 0 байтов.
4. Повторите это для каждого блока данных в Вашем проекте.



D.1 Комплекты запасных клеммных колодок S7-1200 V3.0 и V4.0

Таблица D-1 Комплекты запасных клеммных колодок для ЦПУ S7-1200 V3.0 и ниже

Если у Вас ЦПУ S7-1200 V3.0 и ниже (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)	6ES7 292-1BC30-0XA0	3 конт., анод.золотом
CPU 1211C DC/DC/Relay (6ES7 211-1HE31-0XB0)	6ES7 292-1AH30-0XA0	8 конт., анод.золотом
CPU 1211C AC/DC/Relay (6ES7 211-1BE31-0XB0)	6ES7 292-1AP30-0XA0	14 конт., луженые, с мех.код.
CPU 1212C DC/DC/DC (6ES7 212-1AE31-0XB0)		
CPU 1212C DC/DC/Relay (6ES7 212-1HE31-0XB0)		
CPU 1212C AC/DC/Relay (6ES7 212-1BE31-0XB0)		
CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)	6ES7 292-1BC3-0XA0	3 конт., анод.золотом
CPU 1214C DC/DC/Relay (6ES7 214-1HG31-0XB0)	6ES7 292-1AM30-0XA0	12 конт., луженые
CPU 1214C AC/DC/Relay (6ES7 214-1BG31-0XB0)	6ES7 292-1AV30-0XA0	20 конт., луженые
CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)	6ES7 292-1BF30-0XB0	6 конт., анод.золотом
CPU 1215C DC/DC/Relay (6ES7 215-1HG31-0XB0)	6ES7 292-1AM30-0XA0	12 конт., луженые
CPU 1215C AC/DC/Relay (6ES7 215-1BG31-0XB0)	6ES7 292-1AV30-0XA0	20 конт., луженые, с мех.код.

Таблица D-2 Комплекты запасных клеммных колодок для сигнальных модулей S7-1200 V3.0 и ниже

Если у Вас сигнальный модуль S7-1200 V3.0 и ниже (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
SM1221 DI 8xDC (6ES7 221-1BF32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 8xDC (6ES7 222-1BF32-0XB0)		
SM1222 DQ 8xRelay (6ES7 222-1HF32-0XB0)	6ES7 292-1AG40-0XA1	7 конт., луженые, с мех.кодом слева
SM1231 AI 4x13 bit (6ES7 231-4HD32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1232 AQ 2x14 bit (6ES7 232-4HB32-0XB0)		
SM1231 AI4 x TC (6ES7 231-5QD32-0XB0)		
SM1231 AI4 x 16 bit (6ES7 231-5ND32-0XB0)		
SM1221 DI 16xDC (6ES7 221-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 16xDC (6ES7 222-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 16xRelay (6ES7 222-1HH32-0XB0)	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа
SM1223 DI 8xDC/DQ 8xDC (6ES7 223-1BH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1223 8xDC/8xRelay (6ES7 223-1PH32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа

Если у Вас сигнальный модуль S7-1200 V3.0 и ниже (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
SM1223 8xAC/8xRelay (6ES7 223-1QH32-0XB0)	6ES7 292-1AG40-0XA0	7 конт., луженые, с мех.кодом справа
SM1234 AI 4 / AQ 2 (6ES7 234-4HE32-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI 8x13 BIT (6ES7 231-4HF32-0XB0)		
SM1232 AQ4x14 bit (6ES7 232-4HD32-0XB0)		
SM1231 AI4 x RTD (6ES7 231-5PD32-0XB0)		
SM1231 AI8 x TC (6ES7 231-5QF32-0XB0)		
SM 1278 IO Link(6ES7 278-4BD32-0XB0)	6ES7 292-1AG30-0XA0	7 конт., анод.золотом

Таблица D-3 Комплекты запасных клеммных колодок для сигнальных модулей S7-1200 V3.2 и выше

Если у Вас сигнальный модуль S7-1200 V3.2 и выше (номер для заказа)	Используйте комплект запасной клеммной колодки (4/упак.)	
	Номер для заказа клеммной колодки	Описание клеммной колодки
SM1221 DI 8xDC (6ES7 221-1BF30-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 8xDC (6ES7 222-1BF30-0XB0)		
SM1222 DQ 8xRelay (6ES7 222-1HF30-0XB0)		
SM1231 AI 4x13 bit (6ES7 231-4HD30-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1232 AQ 2x14 bit (6ES7 232-4HB30-0XB0)		
SM1231 AI4 x TC (6ES7 231-5QD30-0XB0)		
SM1231 AI4 x 16 bit (6ES7 231-5ND30-0XB0)		
SM1221 DI 16xDC (6ES7 221-1BH30-0XB0)	6ES7 292-1AG30-0XA0	7 конт., луженые
SM1222 DQ 16xDC (6ES7 222-1BH30-0XB0)		
SM1222 DQ 16xRelay (6ES7 222-1HH30-0XB0)		
SM1223 DI 8xDC/DQ 8xDC (6ES7 223-1BH30-0XB0)		
SM1223 8xDC/8xRelay (6ES7 223-1PH30-0XB0)		
SM1223 8xAC/8xRelay (6ES7 223-1QH30-0XB0)		
SM1234 AI 4 / AQ 2 (6ES7 234-4HE30-0XB0)	6ES7 292-1BG30-0XA0	7 конт., анод.золотом
SM1231 AI 8x13 BIT (6ES7 231-4HF30-0XB0)		
SM1232 AQ4x14 bit (6ES7 232-4HD30-0XB0)		
SM1231 AI4 x RTD (6ES7 231-5PD30-0XB0)		
SM1231 AI8 x TC (6ES7 231-5QF30-0XB0)		
SM1222 DQ 8xRelay (Changeover) (6ES7 222-1XF30-0XB0)	6ES7 292-1AL30-0XA0	11 конт., луженые
SM1223 DI 16xDC/DQ 16xDC (6ES7 223-1BL30-0XB0)		
SM 1223 16 xDC/16X Relay (6ES7 223-1PL30-1XB0)		
SM1231 AI8 x RTD (6ES7 231-5PF30-0XB0)	6ES7 292-1BL30-0XA0	11 конт., анод.золотом

Указатель

