



SIEMENS

SIMATIC

Программируемый контроллер S7-200

Руководство по эксплуатации

Предисловие, содержание	
Обзор продукта	1
Первые шаги	2
Монтаж S7-200	3
Основы ПЛК	4
Основы программирования, соглашения и функции	5
Набор команд S7-200	6
Обмен данными через сеть	7
Руководство по устранению неисправностей аппаратуры и инструментальные средства для тестирования программ	8
Управление перемещением без обратной связи с помощью S7-200	9
Создание программы для модема	10
Использование библиотеки протокола USS для управления приводом MicroMaster	11
Использование библиотеки протокола Modbus	12
Использование рецептов	13
Использование протоколов данных	14
Автоматическая настройка PID и панель управления настройкой PID	15
Приложения	
Предметный указатель	

Номер для заказа этого руководства
6ES7298-8FA24-8BH0

Издание 06/2004
A5E00307987-01

Указания по технике безопасности

Данное руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты от повреждений продукта и связанного с ним оборудования. Эти замечания выделены в руководстве предупреждающим треугольником и помечены, как показано ниже, в соответствии с уровнем опасности:



Опасность

Указывает на угрозу возникновения опасной ситуации, которая **приведет** к смерти или тяжким телесным повреждениям, если не будут предприняты соответствующие меры предосторожности.



Предупреждение

Указывает на потенциально опасную ситуацию, которая **может** привести к смерти или тяжким телесным повреждениям, если не будут предприняты соответствующие меры предосторожности.



Предостережение

При использовании с предупреждающим треугольником указывает на потенциально опасную ситуацию, которая может привести к незначительным или умеренным телесным повреждениям, если не будут предприняты соответствующие меры предосторожности.

Предостережение

При использовании без предупреждающего треугольника указывает на потенциально опасную ситуацию, которая может привести к имущественному ущербу, если не будут предприняты соответствующие меры предосторожности.

Внимание

Указывает на потенциальную ситуацию, которая может привести к нежелательному результату или состоянию, если не будут предприняты соответствующие меры.

Квалифицированный персонал

К монтажу и работе на этом оборудовании должен допускаться только **квалифицированный** персонал. Квалифицированный персонал – это люди, которые имеют право вводить в действие, заземлять и маркировать электрические цепи, оборудование и системы в соответствии с установленной практикой и стандартами техники безопасности.

Надлежащее использование

Примите во внимание следующее:



Предупреждение

Это устройство и его компоненты могут использоваться только для приложений, описанных в каталоге или технической документации, и в соединении только с теми устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Этот продукт может правильно и надежно функционировать только в том случае, если он правильно транспортируется, хранится, устанавливается и монтируется, а также эксплуатируется и обслуживается в соответствии с рекомендациями.

Товарные знаки

SIMATIC®, SIMATIC HMI® и SIMATIC NET® это зарегистрированные товарные знаки фирмы SIEMENS AG.

Некоторые другие обозначения, использованные в этой документации, также являются зарегистрированными товарными знаками, использование которых третьими лицами для своих целей может привести к нарушению прав их обладателей.

Copyright Siemens AG 2004 Все права защищены

Воспроизведение, передача или использование этого документа или его содержания не разрешаются без специального письменного разрешения. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права, вытекающие из патента или регистрации практической модели или конструкции, защищены.

Siemens AG
Департамент автоматизации и приводов (A&D)
Промышленные системы автоматизации (AS)
П/я 4848, D- 90327, Нюрнберг

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанным аппаратным и программным обеспечением. Так как отклонения не могут быть полностью исключены, то мы не можем гарантировать полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются, и все необходимые исправления вносятся в последующие издания. Мы будем благодарны за предложения по улучшению содержания.

© Siemens AG 2004
Технические данные могут быть изменены.

Содержание

1	Обзор продукта	1
	Что нового?	2
	CPU S7–200	2
	Модули расширения S7–200	3
	Пакет для программирования STEP 7-Micro/WIN	4
	Возможности обмена данными	4
	Индикаторные панели	5
2	Первые шаги	7
	Подключение CPU S7–200	8
	Создание программы-примера	10
	Загрузка программы-примера	14
	Перевод S7–200 в режим RUN	14
3	Монтаж S7–200	15
	Указания по монтажу устройств S7–200	16
	Монтаж и демонтаж модулей S7–200	17
	Указания по заземлению и подключению	20
4	Основы ПЛК	23
	Выполнение логики управления с помощью S7–200	24
	Доступ к данным S7–200	26
	Сохранение и извлечение данных с помощью S7–200	36
	Установка режима работы CPU S7–200	41
	Использование проводника S7–200	41
	Функции S7–200	42
5	Основы программирования, соглашения и функции	51
	Рекомендации по проектированию системы с микроконтроллером	52
	Основные элементы программы	53
	Использование STEP 7-Micro/WIN для создания вашей программы	55
	Выбор между наборами команд SIMATIC и IEC 1131-3	57
	Соглашения, используемые в редакторах программ	58
	Создание программы управления с помощью мастеров	60
	Устранение ошибок в S7–200	60
	Назначение адресов и начальных значений в редакторе блоков данных	62
	Использование таблицы символов для символической адресации переменных	62
	Использование локальных переменных	63
	Контроль над программой с помощью таблицы состояний	63
	Создание библиотеки команд	64
	Функции тестирования программы	64

6	Набор команд S7–200	65
	Соглашения, используемые для описания команд	67
	Области памяти и функции S7–200	68
	Битовые логические операции	70
	Контакты	70
	Катушки	73
	Стековые логические операции	75
	Триггеры с преимуществом установки и с преимуществом сброса	77
	Команды, связанные с временем	78
	Команды для обмена данными	81
	Команды чтения из сети и записи в сеть	81
	Команды передачи и приема (свободно программируемый обмен данными)	86
	Команды получения и установки адреса порта	95
	Команды сравнения	96
	Сравнение числовых величин	96
	Сравнение строк	98
	Команды преобразования	99
	Стандартные команды преобразования	99
	Команды преобразования ASCII	103
	Команды преобразования строк	107
	Команды кодирования и декодирования	112
	Команды счета	113
	Счетчик SIMATIC	113
	Счетчик IEC	116
	Скоростные счетчики	118
	Команда вывода импульсов	133
	Арифметические операции	140
	Операции сложения, вычитания, умножения и деления	140
	Умножение целых чисел с представлением результата в виде двойного целого числа и деление целых чисел с остатком	142
	Числовые функции	143
	Команды инкрементирования и декрементирования	144
	Пропорционально-интегрально-дифференциальный (PID) регулятор	145
	Команды прерывания	153
	Логические операции	161
	Операции инвертирования	161
	Операции И, ИЛИ и исключающее ИЛИ	162
	Команды пересылки	164
	Пересылка байта, слова, двойного слова или вещественного числа	164
	Непосредственная пересылка байта (чтение и запись)	165
	Команды групповой пересылки	166
	Команды управления программой	167
	Условное завершение	167
	Условное завершение	167
	Сброс времени контроля	167
	Команды формирования программного цикла For-Next	169
	Команды перехода	171
	Команды для реле управления очередностью (SCR)	172
	Команда управления диагностическим светодиодом	178

Команды сдвига и циклического сдвига	179
Команды сдвига вправо и сдвига влево	179
Команды циклического сдвига вправо и циклического сдвига влево	179
Команда перемещения значения бита в регистр сдвига	181
Команда перестановки байтов в слове	183
Строковые команды	184
Табличные команды	189
Ввод значения в таблицу	189
Удаление первой записи из таблицы и удаление последней записи из таблицы	190
Заполнение памяти комбинацией двоичных разрядов	192
Поиск значения в таблице	193
Таймерные команды	196
Таймерные команды SIMATIC	196
Таймерные команды IEC	201
Интервальные таймеры	203
Операции с подпрограммами	204
7 Обмен данными через сеть	209
Основы обмена данными через сеть для S7–200	210
Выбор протокола связи для вашей сети	214
Инсталляция и деинсталляция интерфейсов для обмена данными	220
Построение сети	221
Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными	225
Использование в сети модемов и STEP 7-Micro/WIN	228
Темы для опытных пользователей	233
Конфигурирование кабеля RS–232/CPI Multi–Master для дистанционного режима	239
8 Устранение неисправностей аппаратуры и инструментальные средства для тестирования программ	243
Функции для отладки вашей программы	244
Отображение состояния программы	246
Использование таблицы состояний для контроля и изменения данных в S7–200	247
Принудительное присваивание значений	248
Исполнение программы в течение указанного количества циклов	248
Указания по устранению неисправностей аппаратуры	249
9 Управление перемещением без обратной связи с помощью S7-200	251
Обзор	252
Использование выхода с широтно-импульсной модуляцией (ШИМ)	253
Основы управления перемещением без обратной связи с помощью шаговых двигателей или сервомоторов	255
Команды, создаваемые Мастером управления позиционированием	260
Коды ошибок для команд РТО	264
Функции модуля позиционирования	265
Конфигурирование модуля позиционирования	267
Команды, создаваемые Мастером управления позиционированием для модуля позиционирования	273
Примеры программ для модуля позиционирования	285

	Наблюдение за модулем позиционирования с помощью панели управления EM 253	290
	Коды ошибок для модуля и команд позиционирования	292
	Для опытных пользователей	294
	Режимы поиска опорной точки, поддерживаемые модулем позиционирования	303
10	Создание программы для модемного модуля	307
	Функции модемного модуля	308
	Использование Мастера расширения модема для конфигурирования модемного модуля	314
	Обзор команд модемного модуля и ограничений	318
	Команды для модемного модуля	319
	Пример программы для модемного модуля	323
	CPU S7-200, поддерживающие интеллектуальные модули	323
	Биты специальной памяти для модемного модуля	323
	Темы для опытных пользователей	325
	Формат телефонных номеров для передачи сообщений	327
	Формат текстовых сообщений	328
	Формат сообщений для передачи данных CPU	329
11	Управление приводом MicroMaster с помощью библиотеки протокола USS	331
	Требования для использования протокола USS	332
	Расчет времени, необходимого для обмена данными с приводом	332
	Использование команд USS	333
	Команды для протокола USS	334
	Примеры программ для протокола USS	341
	Коды ошибок выполнения команд USS	342
	Подключение и наладка привода MicroMaster серии 3	342
	Подключение и наладка привода MicroMaster серии 4	345
12	Использование библиотеки протокола Modbus	347
	Требования для использования протокола Modbus	348
	Инициализация и время исполнения протокола Modbus	348
	Адресация Modbus	349
	Использование команд протокола slave-устройств Modbus	350
	Команды протокола slave-устройств Modbus	351
13	Использование рецептов	355
	Обзор	356
	Определение рецепта и терминология	357
	Использование Мастера рецептов	357
	Команды, создаваемые Мастером рецептов	361

14	Использование протоколирования данных	363
	Обзор	364
	Использование Мастера протоколирования данных	365
	Команды, создаваемые Мастером протоколирования данных	369
15	Автоматическая настройка PID и панель управления настройкой PID	371
	Что такое автоматическая настройка PID	372
	Расширенная таблица обратной связи	372
	Предпосылки	375
	Автогистерезис и автодевиация	375
	Последовательность автоматической настройки	376
	Исключительные ситуации	377
	Замечания относительно выхода регулируемой переменной за пределы диапазона (код результата 3)	377
	Панель управления настройкой PID	378
A	Технические данные	381
	Общие технические данные	382
	Технические данные CPU	385
	Технические данные цифровых модулей расширения	394
	Технические данные аналоговых модулей расширения	400
	Технические данные модулей расширения для термопар и RTD	411
	Технические данные модуля EM 277 PROFIBUS-DP	423
	Технические данные модемного модуля EM 241	435
	Технические данные модуля позиционирования EM 253	437
	Технические данные модуля Ethernet (CP 243-1)	443
	Технические данные модуля Internet (CP 243-1 IT)	445
	Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2)	448
	Дополнительные съемные модули	450
	Кабель с разъемами для модулей расширения	451
	Кабель RS-232/PPI Multi-Master и кабель USB/PPI Multi-Master	452
	Имитаторы входов	456
B	Расчет баланса мощностей	457
C	Коды ошибок	461
	Коды фатальных ошибок и сообщения	462
	Ошибки этапа выполнения	463
	Нарушение правил компиляции	464

D	Биты специальной памяти (SM)	465
	SMB0: биты состояния	466
	SMB1: биты состояния	466
	SMB2: принимаемые символы при свободно программируемом обмене данными	467
	SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными	467
	SMB4: переполнение очереди ожидания	467
	SMB5: состояние входов/выходов	468
	SMB6: идентификационный регистр CPU	468
	SMB7: резерв	468
	SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода	469
	SMW22 – SMW26: времена цикла	470
	SMB28 и SMB29: аналоговый потенциометр	470
	SMB30 и SMB130: регистры управления свободно программируемым обменом данными	470
	SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ)	471
	SMB34 и SMB35: регистры интервалов времени для прерываний, управляемых временем	471
	SMB36 – SMB65: регистры HSC0, HSC1 и HSC2	471
	SMB66 – SMB85: регистры PTO/PWM	473
	SMB86 – SMB94 и SMB186 – SMB194: управление приемом сообщений	474
	SMW98: ошибки в шине расширения	475
	SMB130: регистр управления свободно программируемым обменом данными (см. SMB30)	475
	SMB131 – SMB165: регистры HSC3, HSC4 и HSC5	475
	SMB166 – SMB185: таблица определения профилей PTO0, PTO1	476
	SMB166 – SMB185: таблица определения профилей PTO0, PTO1)	476
	SMB200 – SMB549: состояние интеллектуальных модулей	477
E	Номера для заказа S7–200	479
F	Времена выполнения команд STL	483
G	Краткая информация об S7–200	489

Предисловие

Назначение руководства

Серия S7–200 – это ряд микропрограммируемых логических контроллеров (микроконтроллеров), которые могут управлять разнообразными прикладными системами автоматизации. Компактная конструкция, низкая стоимость и мощная система команд делают контроллеры S7–200 идеальным средством решения для управления малыми приложениями. Большое разнообразие моделей S7–200 и инструментальные средства программирования на основе Windows обеспечивают необходимую гибкость при решении ваших задач автоматизации.

Это руководство дает информацию о монтаже и программировании микроконтроллеров S7–200 и предназначено для инженеров, программистов, монтажников и электриков, которые имеют общие знания о программируемых логических контроллерах.

Основные необходимые знания

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

Область применения руководства

Это руководство имеет силу для STEP 7-Micro/WIN версии 4.0 и семейства продуктов CPU S7–200. Полный список продуктов S7–200, описанных в этом руководстве, и номера для заказа вы найдете в приложении А.

Изменения по сравнению с предыдущей версией

Это руководство было пересмотрено для включения следующей информации.

- Модели CPU S7–200: CPU 221, CPU 222, CPU 224, CPU 224XP и CPU 226, включая:
Поддержка новых аппаратных средств CPU: возможность выключения редактирования в режиме RUN для увеличения программной памяти, CPU 224XP поддерживает встроенный аналоговый вход/выход и два коммуникационных порта. CPU 226 включает в себя дополнительные входные фильтры и регистратор импульсов.
- Поддержка нового картриджа памяти: утилита браузер проводника S7–200, переносы, сравнения картриджей памяти и программирование выбранных элементов
- STEP 7-Micro/WIN, версия 4.0, пакет 32-битового программного обеспечения для S7–200, включая:
Новые и усовершенствованные инструментальные средства, поддерживающие самые последние расширения CPU: панель управления автоматической настройкой PID, встроенные в ПЛК Мастер управления позиционированием, Мастер регистрации данных и Мастер рецептов
Новое диагностическое инструментальное средство: диагностический светодиод для конфигурирования
Новые команды: летнее время (READ_RTCX и SET_RTCX), интервальные таймеры (BITIM, CITIM), очистка события, вызвавшего прерывание (CLR_EVNT) и диагностический светодиод (DIAG_LED)
Расширения ПОУ и библиотек: новые строковые константы, добавлена поддержка косвенной адресации на большем количестве типов памяти, улучшена поддержка параметризации чтения и записи библиотеки USS для главных приводов фирмы Siemens
Усовершенствованный блок данных: страницы блока данных, автоматическое инкрементирование блока данных
Увеличенные удобство и простота использования STEP 7-Micro/WIN

Сертификация

Продукты SIMATIC S7-200 имеют следующие сертификаты:

- Underwriters Laboratories [Лаборатории страхователей], Inc.
UL 508, зарегистрировано (Промышленная аппаратура управления)
Регистрационный номер E75310
- Canadian Standards Association [Канадская ассоциация стандартов]: CSA C22.2,
номер 142 (Оборудование для управления процессами)
- Factory Mutual Research [Заводские совместные исследования]: класс номер 3600,
класс номер 3611, FM класс I, раздел 2, группы опасных помещений A, B, C и D, T4A
и класс I, зона 2, IIC, T4



Совет

Серия SIMATIC S7-200 удовлетворяет стандарту CSA.
Логотип cULus указывает, что S7-200 проверен и сертифицирован Лабораториями страхователей (Underwriters Laboratories, UL) на соответствие стандартам UL 508 и CSA 22.2 № 142.

Маркировка ЕС

Продукты SIMATIC S7-200 удовлетворяют требованиям и нормативам защиты следующих директив Европейского сообщества (ЕС):

- Директива Европейского сообщества по устройствам низкого напряжения 73/23/EEC
- Директива Европейского сообщества по электромагнитной совместимости 89/336/EEC

Проконсультируйтесь у своего местного представителя фирмы Siemens относительно соответствия продукта Директиве Европейского сообщества ATEX 94/9/EC.

Метка С

Продукты SIMATIC S7-200 удовлетворяют требованиям стандарта AS/NZS 2064 (Австралия).

Стандарты:

Продукты SIMATIC S7-200 удовлетворяют требованиям и критериям IEC 61131-2, Программируемые контроллеры – Требования к оборудованию.

За полной информацией обратитесь к Приложению А.

Место этой документации в информационной среде

Семейство продуктов	Документация	Номер для заказа
S7-200	Руководство пользователя сенсорной панели TP070 [TP070 Touch Panel User Manual] (на английском языке)	6AV6 591-1DC01-0AB0
	Руководство по эксплуатации TP170 micro [TP170 micro Operating Manual] (на английском языке)	6AV6 691-1DB01-0AB0
	Руководство по обмену данными для двухточечного интерфейса S7-200 [S7-200 Point-to-Point Interface Communication Manual] (на английском и немецком языках)	6ES7 298-8GA00-8XH0
	Руководство CP 243-2 SIMATIC NET AS-Interface Master (на английском языке)	6GK7 243-2AX00-8BA0
	Интернет-модуль CP 243-1 IT (с электронной документацией на CD)	6GK7 243-1GX00-0XE0
	Ethernet-модуль CP 243-1 (с электронной документацией на CD)	6GK7 243-1GX00-0XE0
	Программируемый контроллер S7-200, Системное руководство [S7-200 Programmable Controller, System Manual] (на английском языке)	6ES7 298-8FA24-8BH0

Как пользоваться этим руководством

Если вы новый пользователь микроконтроллеров S7–200, то вам следует прочитать все *Системное руководство по программируемым контроллерам S7–200*. Если вы опытный пользователь, то обратитесь к содержанию или предметному указателю для нахождения конкретной информации.

Системное руководство по программируемым контроллерам S7–200 разбито на следующие темы:

- Глава 1 (Обзор продукта) дает обзор некоторых характеристик семейства микроконтроллеров S7–200.
- Глава 2 (Первые шаги) представляет собой учебное пособие по созданию и загрузке простой программы управления в S7–200.
- Глава 3 (Монтаж S7–200) дает размеры и основные рекомендации по монтажу модулей CPU S7–200 и модулей расширения ввода/вывода.
- Глава 4 (Основы ПЛК) дает информацию о функционировании S7–200.
- Глава 5 (Основы программирования, соглашения и функции) дает информацию о функциях STEP 7-Micro/WIN, о редакторах программ и видах команд (IEC 1131–3 или SIMATIC), о типах данных S7–200, а также содержит руководство по созданию программ.
- Глава 6 (Набор команд S7–200) дает описания и примеры команд программирования, поддерживаемых S7–200.
- Глава 7 (Обмен данными через сеть) дает информацию по созданию различных сетевых конфигураций, поддерживаемых S7–200.
- Глава 8 (Устранение неисправностей аппаратуры и инструментальные средства для тестирования программ) дает информацию по устранению неисправностей аппаратуры S7–200 и о функциях STEP 7-Micro/WIN, с помощью которых вы можете тестировать свою программу.
- Глава 9 (Управление перемещением с помощью S7-200 без обратной связи) дает информацию о трех методах управления перемещением без обратной связи: широтно-импульсная модуляция, вывод последовательности импульсов и модуль управления положением EM 253.
- Глава 10 (Создание программы для модема) дает информацию о командах и мастере, с помощью которых можно создать программу для модема EM 241.
- Глава 11 (Управление приводом MicroMaster с помощью библиотеки протокола USS) дает информацию о командах, используемых для создания программы управления для привода MicroMaster. Она также дает информацию о конфигурировании приводов MicroMaster 3 и MicroMaster 4.
- Глава 12 (Использование библиотеки протокола Modbus) дает информацию о командах, применяемых для создания программы, использующей протокол Modbus для обмена данными.
- Глава 13 (Использование рецептов) дает информацию об организации и загрузке рецептов программ автоматизации в картридж памяти.
- Глава 14 (Использование записей данных) дает информацию о сохранении данных измерений процесса в картридже памяти.
- Глава 15 (Автоматическая настройка PID и панель управления настройкой PID) дает информацию об использовании этих функций для существенного увеличения эффективности и облегчения использования функции PID, предоставляемой S7–200.
- Приложение A (Технические данные) дает техническую информацию и таблицы данных аппаратного обеспечения S7–200.

В других приложениях приведена дополнительная справочная информация, например, описания кодов ошибок, описания области специальных битов памяти (меркеров) (SM), номера для заказа оборудования S7–200 и времена выполнения команд STL.

Кроме этого руководства, STEP 7-Micro/WIN предоставляет обширную оперативную помощь в режиме online для начального знакомства с программированием S7–200. При покупке программного обеспечения STEP 7-Micro/WIN вы получаете бесплатную документацию на компакт-диске. На этом компакт-диске находятся советы по применению, электронная версия данного руководства и другая информация.

Оперативная помощь

Для получения помощи достаточно нажатия на одну клавишу! Нажатием на F1 вы получаете доступ к обширной оперативной помощи в режиме online для STEP 7-Micro/WIN. Онлайн-помощь включает в себя полезную информацию..

Электронное руководство

Электронная версия данного системного руководства по S7-200 имеется на компакт-диске с документацией. Вы можете установить это электронное руководство на своем компьютере, получая тем самым легкий доступ к информации, во время работы с программным обеспечением STEP 7-Micro/WIN.

Советы по программированию

Компакт-диск с документацией содержит советы по программированию, набор примеров применения с образцами программ. Анализ и модификация этих примеров может помочь вам в поисках эффективных и новаторских решений для вашего приложения. Самую последнюю версию советов по программированию также в Интернете на сайте S7-200.

Утилизация и удаление отходов

Для безопасной для окружающей среды утилизации вашего устройства обратитесь к компании, имеющей сертификат на удаление электронного лома.

Дополнительная поддержка

Местный отдел сбыта фирмы Siemens или дистрибьютор

Для получения помощи в ответах на технические вопросы, для обучения работе с продуктами S7-200 или для их заказа обратитесь к своему местному дистрибьютору фирмы Siemens или в отдел сбыта. Так как торговые представители фирмы имеют техническое образование и конкретные знания о вашей деятельности, процессе и отрасли промышленности, а также о конкретных продуктах фирмы Siemens, которые вы используете, они могут наиболее быстро предоставить эффективные ответы на любые проблемы, с которыми вы можете встретиться.

Обслуживание и поддержка в Интернете

Кроме нашей документации, мы предлагаем вам наши знания в режиме online в Интернете по адресу:

<http://www.siemens.com/automation/service&support>,

где вы найдете следующее:

- www.siemens.com/S7-200 для продуктов S7-200
Сайт S7-200 в Интернете содержит ответы на часто задаваемые вопросы (FAQ), советы и приемы (примеры применения и образцы программ), информацию о вновь выпущенных продуктах, а также обновления продуктов или загрузки.
- Информационный бюллетень, который постоянно снабжает вас самой современной информацией о продуктах фирмы.
- Нужную вам документацию через функцию поиска (Search) в разделе обслуживания и поддержки (Service & Support).
- Форум, где пользователи и эксперты со всего мира обмениваются своим опытом.
- Информацию о местных представителях департаменте Автоматизации и приводов.
- Информацию об обслуживании в полевых условиях, ремонте, запасных частях и многом другом в разделе «Services [Услуги]».

Техническое обслуживание

В вашем распоряжении находится также хорошо обученный персонал Центра технического обслуживания S7-200, который может оказать вам помощь в решении любых проблем, с которыми вы можете встретиться. Вы можете обращаться к ним круглосуточно в любой день недели.

Техническая поддержка Департамента A&D

Доступна круглосуточно во всем мире:



<p>По всему миру (Нюрнберг) Техническая поддержка</p> <p>Круглосуточно, 365 дней в году Тел.: +49 (180) 5050-222 Факс: +49 (180) 5050-223 mailto:adsupport@siemens.com Гринвичское время: +1:00</p>	<p>Соединенные Штаты (Джонсон-сити) Техническая поддержка и авторизация</p> <p>Местное время: Пн.– Пт. с 8:00 до 17:00 Тел.: +1 (423) 262 2522 +1 (800) 333-7421 (только США) Факс: +1 (423) 262 2289 mailto:simatic.hotline@sea.siemens.com Гринвичское время: -5:00</p>	<p>Азия / Австралия (Пекин) Техническая поддержка и авторизация</p> <p>Местное время: Пн.– Пт. с 8:00 до 17:00 Тел.: +86 10 64 75 75 75 Факс: +86 10 64 74 74 74 mailto:adsupport.asia@siemens.com Гринвичское время: +8:00</p>
<p>Европа / Африка (Нюрнберг) Авторизация</p> <p>Местное время: Пн.– Пт. с 8:00 до 17:00 Тел.: +49 (180) 5050-222 Факс: +49 (180) 5050-223 mailto:adsupport@siemens.com Гринвичское время: +1:00</p>		
<p>На горячих линиях SIMATIC и горячих линиях авторизации обычно говорят на немецком и английском языке.</p>		

Обзор продукта

1

Семейство программируемых логических микроконтроллеров (микро-ПЛК) S7–200 может управлять широким спектром устройств для решения ваших задач автоматизации.

S7–200 контролирует входы и изменяет выходы под управлением программы пользователя, которая может содержать булевы логические операции, функции счета и времени, сложные математические операции и операции по обмену данными с другими интеллектуальными устройствами. Благодаря компактной конструкции, гибкой конфигурации и мощному набору команд S7–200 в высшей степени пригоден для решения широкого спектра прикладных задач управления.

В этой главе

Что нового?	2
CPU S7–200	2
Модули расширения S7–200	4
Пакет для программирования STEP 7-Micro/WIN	4
Возможности обмена данными	5
Индикаторные панели	5

Что нового?

SIMATIC S7–200 включает в себя следующие новые функции. Таблица 1–1 показывает CPU S7–200, которые поддерживают эти новые функции.

- Модели CPU S7–200 CPU 221, CPU 222, CPU 224, CPU 224XP и CPU 226, включая:
Поддержка нового аппаратного обеспечения CPU: возможность выключить редактирование в режиме RUN, чтобы увеличить размер памяти для программы, CPU 224XP поддерживает встроенный аналоговый ввод/вывод и два коммуникационных порта. CPU 226 содержит дополнительные входные фильтры и регистратор импульсов.
- Поддержка нового картриджа памяти: утилита браузер проводника S7–200, переносы, сравнения картриджей памяти и программирование выбранных элементов
- STEP 7-Micro/WIN, версия 4.0, пакет 32-битового программного обеспечения для S7–200, включая:
Новые и усовершенствованные инструментальные средства, поддерживающие самые последние расширения CPU: панель управления автоматической настройкой PID, встроенные в ПЛК Мастер управления позиционированием, Мастер регистрации данных и Мастер рецептов.
Новое диагностическое инструментальное средство: диагностический светодиод для конфигурирования
Новые команды: летнее время (READ_RTCX и SET_RTCX), интервальные таймеры (BITIM, CITIM), очистка события, вызвавшего прерывание (CLR_EVNT) и диагностический светодиод (DIAG_LED).
Расширения POU и библиотек: новые строковые константы, добавлена поддержка косвенной адресации на большем количестве типов памяти, улучшена поддержка параметризации чтения и записи библиотеки USS для главных приводов фирмы Siemens
Усовершенствованный блок данных: страницы блока данных, автоматическое инкрементирование блока данных
Увеличенные удобство и простота использования STEP 7-Micro/WIN

Таблица 1–1. CPU S7–200

CPU S7–200	Номер для заказа
CPU 221 DC/DC/DC 6 входов/4 выхода	6ES7 211-0AA23-0XB0
CPU 221 AC/DC/Relay 6 входов/4 реле	6ES7 211-0BA23-0XB0
CPU 222 DC/DC/DC 8 входов/6 выходов	6ES7 212-1AB23-0XB0
CPU 222 AC/DC/Relay 8 входов/6 реле	6ES7 212-1BB23-0XB0
CPU 224 DC/DC/DC 14 входов/10 выходов	6ES7 214-1AD23-0XB0
CPU 224 AC/DC/Relay 14 входов/10 реле	6ES7 214-1BD23-0XB0
CPU 224XP DC/DC/DC 14 входов/10 выходов	6ES7 214-2AD23-0XB0
CPU 224XP AC/DC/Relay 14 входов/10 реле	6ES7 214-2BD23-0XB0
CPU 226 DC/DC/DC 24 входа/16 выходов	6ES7 216-2AD23-0XB0
CPU 226 AC/DC/Relay 24 входа/16 реле	6ES7 216-2BD23-0XB0

CPU S7–200

CPU S7–200 состоит из микропроцессора, встроенного источника питания, входных и выходных цепей, находящихся в компактном корпусе и образующих мощный микро-ПЛК (см. рис. 1-1). После загрузки программы S7–200 содержит логику, необходимую для контроля и управления входными и выходными устройствами в вашем приложении.

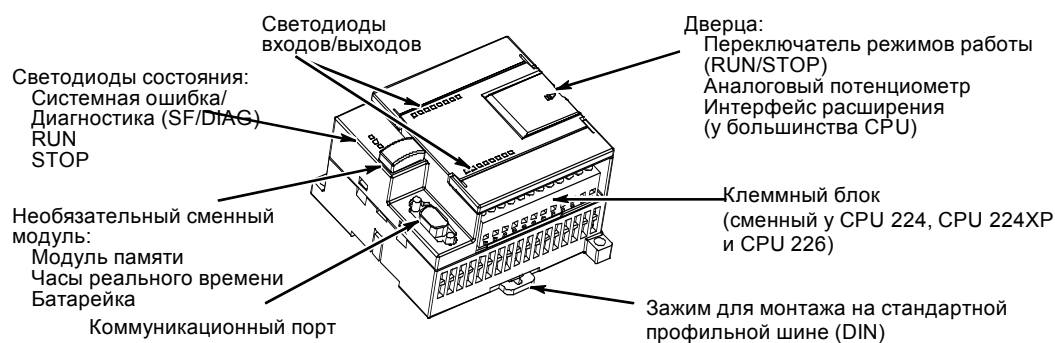


Рис. 1–1. Микро-ПЛК S7-200

Фирма Siemens предлагает различные модели CPU S7–200 с разнообразными характеристиками производительности и функциями, чтобы помочь вам в создании эффективных решений для самых разнообразных приложений. В таблице 1–2 дается краткое сравнение функций различных CPU. Подробную информацию для конкретных CPU вы найдете в Приложении А.

Таблица 1–2. Сравнение моделей CPU S7–200

Характеристика	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Физические размеры (мм)	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
Программная память: с редактированием в режиме RUN без редактирования в режиме RUN	4096 байт	4096 байт	8192 байта	12288 байт	16384 байта
	4096 байт	4096 байт	12288 байт	16384 байта	24576 байт
Память данных	2048 байт	2048 байт	8192 байта	10240 байт	10240 байт
Буферизация памяти	Обычно 50 часов	Обычно 50 часов	Обычно 100 часов	Обычно 100 часов	Обычно 100 часов
Локальные встроенные входы/ выходы цифровые аналоговые	6 вх./4 вых.	8 вх./6 вых.	14 вх./10 вых.	14 вх./10 вых.	24 вх./16 вых.
	-	-	-	2 вх./1 вых.	-
Модули расширения	0 модулей	2 модуля ¹	7 модулей ¹	7 модулей ¹	7 модулей ¹
Скоростные счетчики	4 при 30 кГц	4 при 30 кГц	6 при 30 кГц	4 при 30 кГц	6 при 30 кГц
	2 при 20 кГц	2 при 20 кГц	4 при 20 кГц	2 при 200 кГц 3 при 20 кГц 1 при 100 кГц	4 при 20 кГц
Импульсные выходы (DC)	2 при 20 кГц	2 при 20 кГц	2 при 20 кГц	2 при 100 кГц	2 при 20 кГц
Аналоговые потенциометры	1	1	2	2	2
Часы реального времени	Сменный модуль	Сменный модуль	Встроенные	Встроенные	Встроенные
Коммуникационные порты	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
Арифметика с плавающей точкой	Да				
Цифровые входы/выходы (образ процесса)	256 (128 входов, 128 выходов)				
Времена выполнения булевых операций	0,22 микросекунд/операцию				
<p>¹ Вы должны рассчитать свой энергетический потенциал, чтобы определить, какую мощность (или величину тока) CPU S7–200 может предоставить Вашей конфигурации. Если энергетический потенциал CPU превышен, то вы не сможете подключить максимальное количество модулей. Потребности в мощности CPU и модулей расширения Вы найдете в Приложении А, а данные для расчета своего энергетического потенциала – в Приложении В.</p>					

Модули расширения S7–200

Для лучшего выполнения требований, предъявляемых вашими приложениями, в семействе S7–200 имеется большое количество модулей расширения. С помощью этих модулей расширения вы можете расширить функциональные возможности своего CPU S7–200. В таблице 1–3 приведен список имеющихся в настоящее время модулей расширения. Подробную информацию для конкретных модулей вы найдете в Приложении А.

Таблица 1–3. Модули расширения S7–200

Модули расширения		Виды		
Цифровые модули	Ввод	8 вх. пост. тока	8 вх. перем. тока.	16 вх. пост. тока
		4 вых. пост. тока	4 реле	
	Вывод	8 вых. пост. тока	8 вых. перем. тока	8 реле
		Комбинация	4 вх. пост. тока/ 4 вых. пост. тока	8 вх. пост. тока/ 8 вых. пост. тока
Аналоговые модули	Ввод	4 аналоговых входа	4 входа для термопар	2 входа RTD
		Вывод	2 аналоговых выхода	
	Комбинация	4 аналоговых входа/1 аналоговый выход		
Интеллектуальные модули		Позиционирование Ethernet	Модем Internet	PROFIBUS–DP
Другие модули		AS-интерфейс		

Пакет для программирования STEP 7-Micro/WIN

Пакет для программирования STEP 7-Micro/WIN предоставляет дружелюбную пользователю среду для разработки, редактирования и контроля логики, необходимой для управления вашим приложением. STEP 7-Micro/WIN снабжен тремя редакторами программ, с помощью которых можно удобно и эффективно разрабатывать программы управления для вашего приложения. Для помощи в поиске нужной вам информации STEP 7-Micro/WIN предлагает обширную онлайн-ую систему помощи и компакт-диск с документацией, содержащий электронную версию данного руководства, советы по применению и другую полезную информацию.

Требования к компьютеру

STEP 7-Micro/WIN работает на персональных компьютерах или на устройствах программирования фирмы Siemens, например, PG 760. Ваш компьютер или устройство программирования должны удовлетворять следующим минимальным требованиям:

- Операционная система: Windows 2000, Windows XP (профессиональная или домашняя версия)
- не менее 100 Мбайт свободного пространства на жестком диске
- мышь (рекомендуется)

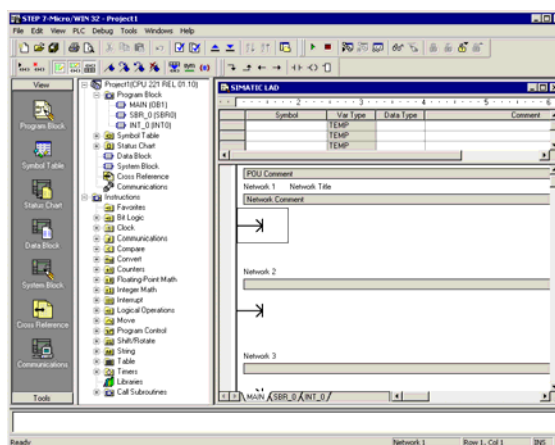


Рис. 1–2. STEP 7-Micro/WIN

Установка STEP 7-Micro/WIN

Вставьте компакт-диск со STEP 7-Micro/WIN в дисковод для компакт-дисков своего компьютера. Мастер установки автоматически запускается и ведет вас через весь процесс установки. За дополнительной информацией об установке STEP 7-Micro/WIN обратитесь к файлу readme.



Совет

Для установки STEP 7-Micro/WIN на ПК с операционной системой Windows 2000 или Windows XP (профессиональная или домашняя версия) вы должны зарегистрироваться с правами администратора.

Возможности обмена данными

Фирма Siemens предоставляет две программные возможности для присоединения вашего компьютера к S7-200: непосредственное соединение с помощью кабеля PPI Multi-Master или с использованием платы коммуникационного процессора (CP) с кабелем MPI.

Использование кабеля PPI Multi-Master является наиболее часто используемым и удобным способом присоединения вашего компьютера к S7-200. Этот кабель соединяет коммуникационный порт S7-200 с последовательным коммуникационным портом вашего компьютера. Кабель PPI Multi-Master может использоваться также для присоединения к S7-200 других устройств для обмена данными.

Индикаторные панели

Текстовые дисплеи (TD 200 и TD 200C)

TD 200 и TD 200C – это 2-строчные текстовые дисплеи с 20 символами в каждой строке, которые могут быть присоединены к S7-200. С помощью мастера для TD 200 вы можете легко запрограммировать свой S7-200 для отображения текстовых сообщений и других данных, относящихся к вашему приложению.

TD 200 и TD 200C представляют собой недорогие интерфейсы с вашим приложением, позволяющие вам просматривать, контролировать и изменять параметры процесса в этом приложении.

За дополнительной информацией о текстовых дисплеях обратитесь к Руководству пользователя *SIMATIC Text Display (TD)* [Текстовый дисплей *SIMATIC*] на компакт-диске с документацией STEP 7-Micro/WIN.

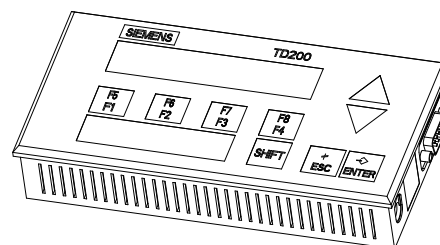


Рис. 1–3. Текстовый дисплей (TD 200 и TD 200C)

Мастер конфигурирования TD 200 в STEP 7-Micro/WIN поможет вам быстро и удобно конфигурировать сообщения TD 200. Для запуска Мастера TD 200 выберите команду меню **Tools > TD 200 Wizard** [Инструментальные средства > Мастер TD 200].

Сенсорные индикаторные панели TP070 и TP170 micro

TP070 и TP170 micro – это сенсорные индикаторные устройства, которые могут быть присоединены к S7–200. С помощью этой сенсорной панели вы можете настроить свой интерфейс оператора в соответствии с требованиями пользователя.

Эти устройства могут отображать пользовательские графики, ползунковые индикаторы, переменные приложения, экранные кнопки и т.д. на удобной для пользователя сенсорной панели.

Дополнительное программное обеспечение для программирования сенсорных панелей TP070 и TP170 micro вы найдете в Приложении E.

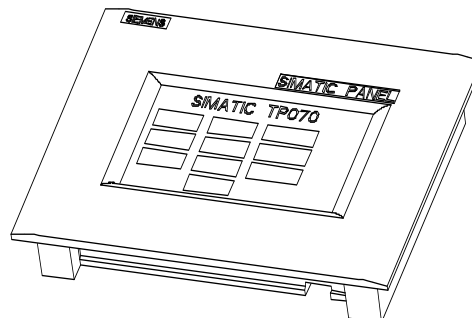


Рис. 1–4. Сенсорная панель

С помощью STEP 7-Micro/WIN вы можете легко программировать свой S7-200. Всего за несколько коротких шагов на простом примере вы узнаете, как подключать, программировать и эксплуатировать свой S7-200.

Для этого примера вам потребуется кабель PPI Multi-Master, CPU S7-200 и устройство программирования, на котором установлено программное обеспечение STEP 7-Micro/WIN.

В этой главе

Подключение CPU S7-200	8
Создание программы-примера	10
Загрузка программы-примера	14
Перевод S7-200 в режим RUN	14

Подключение CPU S7-200

Подключить ваш S7-200 совсем несложно. В этом примере вам нужно только присоединить к CPU S7-200 источник питания, а затем присоединить кабель связи к устройству программирования и CPU S7-200.

Подключение источника питания к CPU S7-200

Сначала подключите S7-200 к источнику питания. На рис. 2-1 показано присоединение проводов для моделей CPU S7-200 постоянного и переменного тока.

Перед монтажом или демонтажем любого электрического устройства вы должны обеспечить, чтобы питание этого устройства было выключено. Выполняйте все необходимые предписания по технике безопасности и убедитесь, что перед монтажом или демонтажем S7-200 его питание выключено.



Предупреждение

Попытки монтажа или присоединения проводов к S7-200 или подключенной к нему аппаратуре при включенном питающем напряжении может привести к поражению электрическим током или к сбоям в работе оборудования. Если питающее напряжение S7-200 и всех подключенных к нему устройств при монтаже или демонтаже устройств не выключено, то это может привести к гибели или тяжким телесным повреждениям персонала и/или к повреждению оборудования.

Выполняйте все необходимые предписания по технике безопасности и убеждайтесь перед выполнением монтажа или демонтажа S7-200 или подключенной к нему аппаратуры, что питание S7-200 выключено.

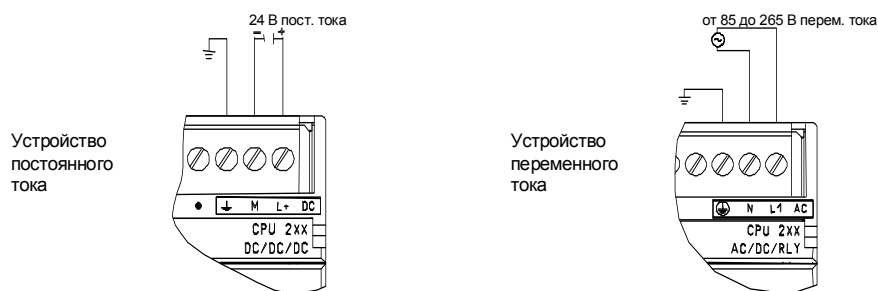


Рис. 2-1. Подключение питания к CPU S7-200

Присоединение кабеля RS-232/PPI Multi-Master

На рис. 2-2 показан кабель RS-232/PPI Multi-Master, соединяющий S7-200 с устройством программирования.

1. Присоедините штекер RS-232 (обозначенный «PC») кабеля RS-232/PPI Multi-Master к коммуникационному порту устройства программирования. (В этом примере используйте COM 1.)
2. Присоедините штекер RS-485 (обозначенный «PPI») кабеля RS-232/PPI Multi-Master к порту 0 или 1 S7-200.
3. Обратите внимание, чтобы DIP-переключатели кабеля RS-232/PPI Multi-Master были установлены, как показано на рис. 2-2.

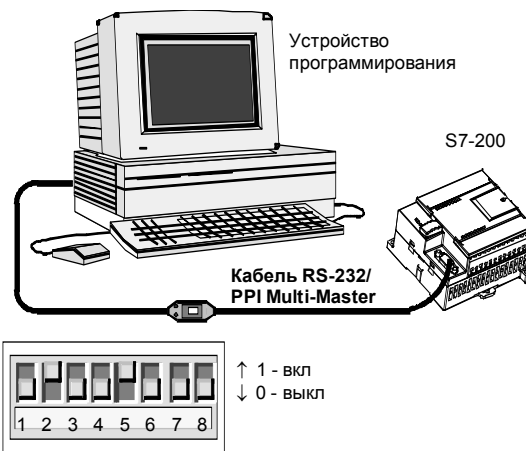


Рис. 2-2. Присоединение кабеля RS-232/PPI Multi-Master



Совет

В примерах, приведенных в этом руководстве, используется кабель RS-232/PPI Multi-Master. Кабель RS-232/PPI Multi-Master заменяет ранее использовавшийся кабель PC/PPI. В продаже имеется также кабель USB/PPI Multi-Master. Номера для заказа вы найдете в Приложении E.

Вызов STEP 7-Micro/WIN

Щелкните на символе STEP 7-Micro/WIN, чтобы открыть новый проект. На рис. 2-3 показан новый проект.

Обратите внимание на навигационную панель. С помощью символов на навигационной панели вы можете открывать отдельные элементы проекта STEP 7-Micro/WIN.

Щелкните на символе Communications на навигационной панели, чтобы вызвать диалоговое окно "Communications [Обмен данными]". Это диалоговое окно используется для установки связей для STEP 7-Micro/WIN.

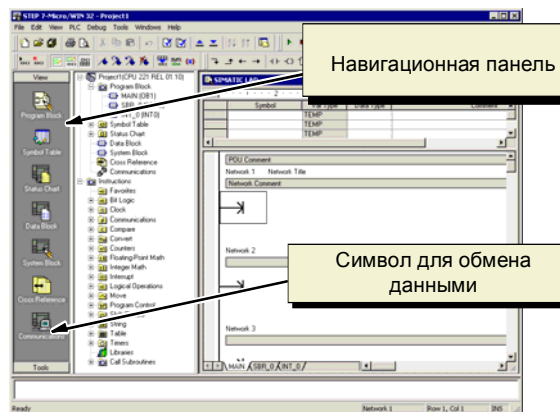


Рис. 2-3. Новый проект STEP 7-Micro/WIN

Проверка параметров обмена данными для STEP 7-Micro/WIN

Проект-пример использует для STEP 7-Micro/WIN и кабеля RS-232/PPI Multi-Master настройки по умолчанию. Эти настройки проверяются следующим образом:

1. Проверьте, чтобы адрес кабеля PC/PPI в диалоговом окне Communications был установлен на 0.
2. Проверьте, чтобы в качестве интерфейса для сетевых параметров был установлен кабель PC/PPI (COM1).
3. Проверьте, чтобы для скорости передачи (transmission rate) было установлено значение 9.6 Кбит/с.

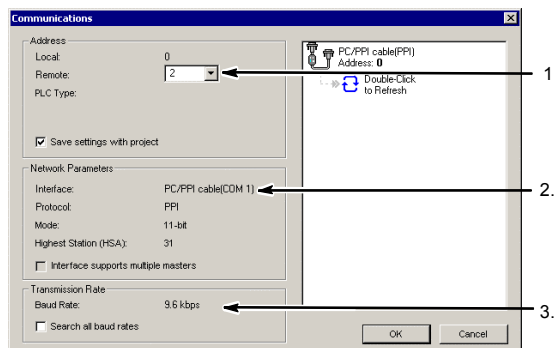


Рис. 2-4. Проверка параметров обмена данными

Если вам необходимо изменить настройки параметров для обмена данными, прочитайте главу 7.

Установка связи с S7-200

В диалоговом окне Communications [Обмен данными] установите связь с CPU S7-200:

1. В диалоговом окне Communications щелкните дважды на кнопке Double click to refresh [Дважды щелкните для обновления]. STEP 7-Micro/WIN ищет станцию S7-200 и отображает символ CPU для подключенной станции S7-200.
2. Выберите S7-200 и щелкните на ОК.

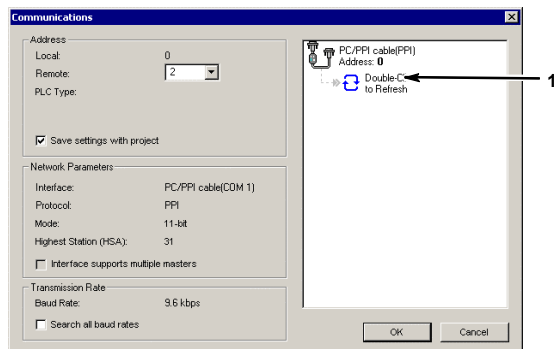


Рис. 2-5. Установление связи с S7-200

Если STEP 7-Micro/WIN не находит ваше CPU S7-200, проверьте настройки параметров для обмена данными и повторите эти шаги.

После установления связи с S7-200 вы готовы к созданию и загрузке программы-примера.

Создание программы-примера

Ввод этого примера программы управления поможет вам понять, как просто работать со STEP 7-Micro/WIN. Эта программа содержит шесть команд в тех сегментах (Network) и образует из них очень простой таймер, сам запускается и сам себя сбрасывает.

Команды для этого примера программы введите в редакторе LAD (KOP). Следующий пример показывает всю программу в виде контактного плана (LAD) и в виде списка команд (STL, AWL). Комментарии к сегменту в STL-программе объясняют логику для каждого сегмента. Импульсная диаграмма показывает, как программа работает.

Пример: Образец программы для начала работы со STEP 7-Micro/WIN

```

Network 1 //10-миллисекундный таймер T33 включается по
//истечении (100 x 10 мс = 1 с)
//Импульс M0.0 слишком короток, чтобы его можно
//было увидеть в отображении состояния
//программы.
LDN    M0.0
TON    T33, +100
Network 2 // Сравнение становится истинным со скоростью,
//позволяющей видеть его результат в отображении
//состояния программы. Включить Q0.0 по
//истечении (40 x 10 мс = 0,4 с), чтобы получить
//сигнал с соотношением времен
//выключения/включения 40% /60%.
LDW>= T33, +40
=      Q0.0
Network 3 //Импульс T33 (бит) слишком короток, чтобы его
//можно было увидеть в отображении состояния
//программы.
//Сбросить таймер через M0.0 по истечении
//(100 x 10 мс = 1 с).
LD     T33
=     M0.0
    
```

Импульсная диаграмма

Вызов редактора программ

Чтобы открыть редактор программ, щелкните на символе Program Block [Программный блок]. См. рис. 2–6.

Обратите внимание на дерево команд и редактор программ. Дерево команд используется для вставки команд контактного плана (LAD) в сегменты редактора программ путем буксировки команд с помощью мыши из дерева команд в сегменты.

Символы на панели инструментов предоставляют возможность быстрого вызова команд меню.

После ввода и сохранения программы вы можете загрузить ее в S7–200.

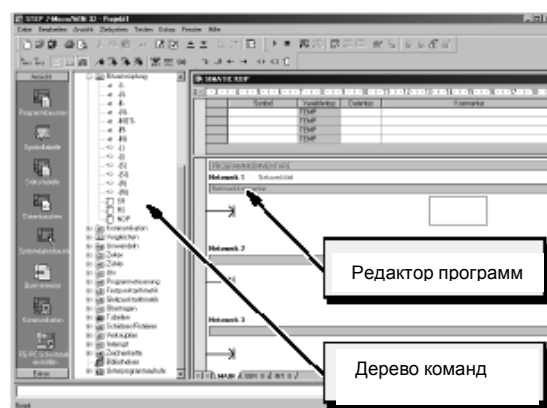


Рис. 2–6. Окно STEP 7-Micro/WIN

Ввод сегмента (Network) 1: Запуск таймера

Если M0.0 выключен (0), этот контакт включается и передает поток сигнала для запуска таймера. Для ввода контакта для M0.0:

1. Дважды щелкните на символе Bit Logic [Битовая логика] или один раз щелкните на знаке плюс (+) для отображения битовых логических операций.
2. Выберите размыкающий контакт.
3. Удерживая в нажатом состоянии левую кнопку мыши, перетащите этот контакт в первый сегмент.
4. Щелкните на «???» над контактом и введите следующий адрес: M0.0
5. Нажмите клавишу Return, чтобы ввести адрес для контакта.



Рис. 2–7. Сегмент 1

Чтобы ввести таймерную команду для T33:

1. Дважды щелкните на символе Timers [Таймеры], чтобы отобразить таймерные команды.
2. Выберите TON (On–Delay Timer – таймер с задержкой включения).
3. Удерживая в нажатом состоянии левую кнопку мыши, перетащите этот таймер в первый сегмент.
4. Щелкните на «???» над таймерным блоком и введите следующий номер таймера: T33
5. Нажмите клавишу Return, чтобы ввести номер таймера и перевести фокус на параметр (PT) для задания предустановленного времени.
6. Введите для предустановленного времени следующее значение: 100
7. Нажмите клавишу Return, чтобы ввести это значение.

Ввод сегмента 2: Включение выхода

Если значение таймера для T33 больше или равно 40 (40 раз по 10 миллисекунд, или 0,4 секунды), то контакт пропускает поток сигнала для включения выхода Q0.0 модуля S7–200. Для ввода команды сравнения:

1. Дважды щелкните на символе компаратора (Compare), чтобы отобразить команды сравнения. Выберите команду $\geq I$ (больше или равно для целых чисел).
2. Удерживая в нажатом состоянии левую кнопку мыши, перетащите эту команду сравнения во второй сегмент.
3. Щелкните на «???» над контактом и введите адрес для значения таймера: T33
4. Нажмите клавишу Return, чтобы ввести номер таймера и перевести фокус на другую величину, которая должна сравниваться со значением таймера.
5. Введите следующую величину для сравнения со значением таймера: 40
6. Нажмите клавишу Return, чтобы ввести это значение.

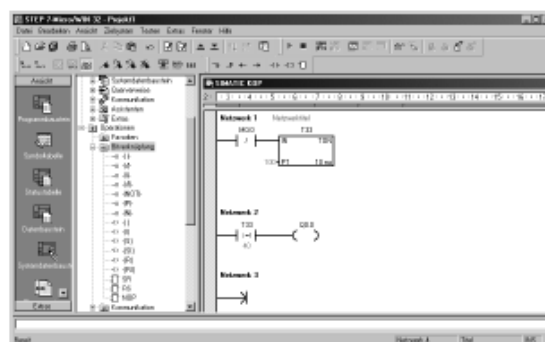


Рис. 2–8. Сегмент 2

Для ввода команды включения выхода Q0.0:

1. Дважды щелкните на символе Bit Logic [Битовая логика], чтобы отобразить битовые логические операции, и выберите выходную катушку.
2. Удерживая в нажатом состоянии левую кнопку мыши, перетащите эту катушку во второй сегмент.
3. Щелкните на «???» над катушкой и введите следующий адрес: Q0.0
4. Нажмите клавишу Return, чтобы ввести этот адрес для катушки.

Ввод сегмента 3: Сброс таймера

Когда таймер достигает предустановленного значения (100) и включает таймерный бит, контакт для T33 включается. Поток сигнала от этого контакта включает бит памяти M0.0. Так как таймер активизируется нормально замкнутым контактом для M0.0, то изменение состояния M0.0 с выключенного (0) на включенное (1) сбрасывает таймер.

Чтобы ввести контакт для таймерного бита T33:

1. Выберите из команд битовой логики замыкающий контакт.
2. Удерживая в нажатом состоянии левую кнопку мыши, перетащите этот контакт в третий сегмент.
3. Щелкните на «???» над контактом и введите адрес таймерного бита: T33
4. Нажмите клавишу Return, чтобы ввести этот адрес для контакта.

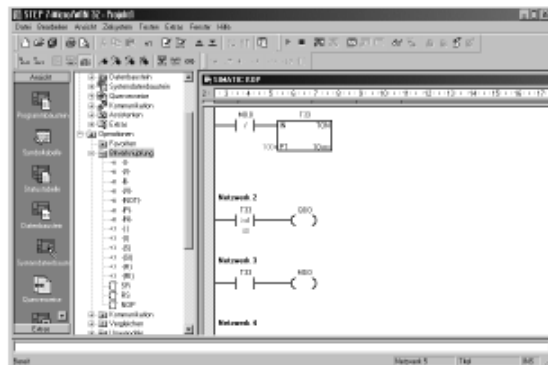


Рис. 2–9. Сегмент 3

Чтобы ввести катушку для включения M0.0:

1. Выберите из команд битовой логики выходную катушку.
2. Удерживая в нажатом состоянии левую кнопку мыши, перетащите эту выходную катушку в третий сегмент.
3. Дважды щелкните на «???» над катушкой и введите следующий адрес: M0.0
4. Нажмите клавишу Return, чтобы ввести этот адрес для катушки.

Сохранение примера проекта

После ввода трех сегментов с командами вы закончили ввод программы. Когда вы сохраняете эту программу, вы создаете проект, который включает в себя тип CPU S7–200 и другие параметры. Для сохранения проекта:

1. Выберите из строки меню команду **File > Save As [Файл > Сохранить как]**.
2. В диалоговом окне Save As [Сохранить как] введите имя для проекта.
3. Для сохранения проекта щелкните на ОК.

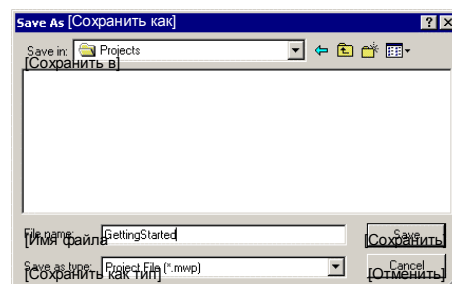



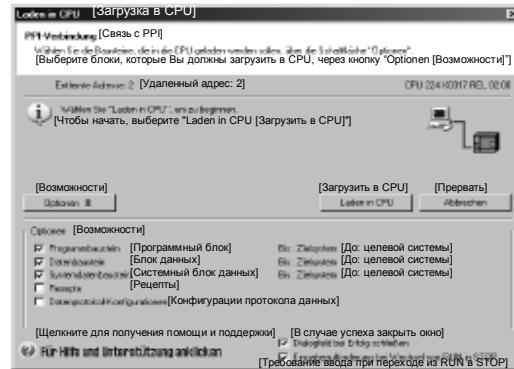
Рис. 2–10. Сохранение примера проекта

После сохранения проекта вы можете загрузить программу в S7–200.

Загрузка программы-примера

	<p>Совет Каждый проект STEP 7-Micro/WIN ставится в соответствие модели CPU (CPU 221, CPU 222, CPU 224, CPU 224XP или CPU 226). Если тип CPU, установленный в проекте, не соответствует подключенному CPU, то STEP 7-Micro/WIN указывает на это несоответствие и требует от вас проведения соответствующих мероприятий. Для этого примера выберите в этом случае «Continue Download [Продолжить загрузку]».</p>
---	--

1. Для загрузки программы щелкните на символе Download [Загрузить], находящемся на панели инструментов, или выберите команду меню **File > Download [Файл > Загрузить]**. См. рис. 2–11.
2. Для загрузки элементов программы в S7–200 щелкните на ОК.



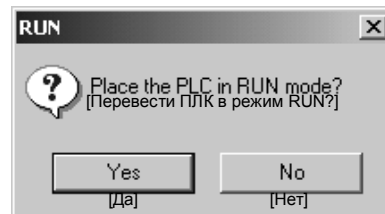
Если ваш S7–200 находится в режиме RUN, то появится сообщение с требованием перевести S7–200 в STOP. Для перевода S7–200 в STOP щелкните на Yes [Да].

Рис. 2–11. Загрузка программы

Перевод S7–200 в режим RUN

Чтобы STEP 7-Micro/WIN мог перевести CPU S7–200 в режим RUN, переключатель режимов S7–200 должен находиться в положении TERM или RUN. При переводе S7–200 в режим RUN S7–200 исполняет программу:

1. Щелкните на символе RUN, находящемся на панели инструментов, или выберите команду меню **PLC > RUN [ПЛК > RUN]**.
2. Щелкните на ОК, чтобы изменить режим работы S7–200.



Когда S7–200 переходит в режим RUN, светодиод для Q0.0 включается и выключается по мере исполнения программы в S7–200.

Рис. 2–12. Перевод S7–200 в режим RUN

Поздравляем! Вы только что создали свою первую программу для S7–200.

Вы можете наблюдать за работой программы, выбрав команду меню **Debug > Program Status [Отладка > Состояние программы]**. STEP 7-Micro/WIN отображает значения для команд. Для остановки программы переведите S7–200 в режим STOP, щелкнув на символе STOP или выбрав команду меню **PLC > STOP [ПЛК > STOP]**.

Монтаж S7–200

3

Оборудование S7-200 спроектировано так, чтобы его легко было устанавливать. Вы можете использовать монтажные отверстия, чтобы прикрепить модули к панели, или встроенные защелки, чтобы установить модули на стандартную (DIN) профильную шину. Малый размер S7-200 позволяет эффективно использовать пространство.

Эта глава дает указания по монтажу и подключению системы S7-200.

В этой главе

Указания по монтажу устройств S7–200	16
Монтаж и демонтаж модулей S7–200	17
Указания по заземлению и подключению	20

Указания по монтажу устройств S7-200

Вы можете монтировать S7-200 либо в щите управления, либо на стандартной профильной шине. Вы можете монтировать S7-200 горизонтально или вертикально.

Держите устройства S7-200 вдали от тепла, высокого напряжения и электрических помех

Общее правило размещения устройств в вашей системе состоит в том, что устройства, генерирующие высокое напряжение или сильные электрические помехи, следует держать вдали от таких низковольтных электронных устройств, как S7-200.

При проектировании размещения S7-200 внутри щита управления обращайтесь внимание на тепловыделяющие устройства и размещайте электронные устройства в более прохладных местах своего распределительного шкафа. Эксплуатация любого электронного устройства в среде с высокими температурами сокращает интервал времени до выхода из строя.

Обратите также внимание на прокладку проводов для устройств в щите управления. Избегайте размещения проводов с низковольтными сигналами и кабелей связи в одном кабельном канале с питающими линиями переменного тока и быстро переключаемыми линиями постоянного тока.

Оставляйте достаточный зазор для охлаждения и подключения проводов

Устройства S7-200 рассчитаны на естественное конвекционное охлаждение. Для надлежащего охлаждения вы должны обеспечить зазоры величиной не менее 25 мм выше и ниже модулей. Кроме того, позаботьтесь о минимальной монтажной глубине 75 мм.



Совет

В случае вертикального монтажа максимально допустимая температура окружающей среды понижается на 10° С. Монтируйте CPU S7-200 ниже модулей расширения.

При планировании компоновки для системы S7-200 оставляйте достаточный зазор для присоединения проводов и коммуникационных кабелей. Для придания дополнительной гибкости компоновке вашей системы S7-200 используйте кабели со штекерными разъемами.

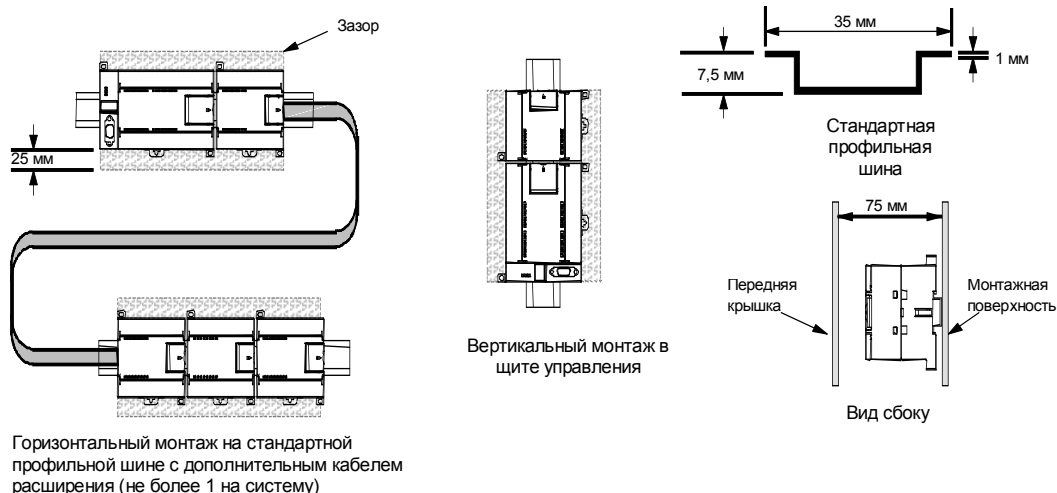


Рис. 3-1. Возможности монтажа, ориентация и зазор

Баланс мощностей

Все CPU S7-200 имеют внутренний источник питания, который обеспечивает энергией CPU, модули расширения и другие потребности пользователя в электроэнергии напряжением 24 В постоянного тока.

CPU S7-200 поставляет напряжение 5 В постоянного тока для всех расширений в вашей системе. Обращайте особое внимание на то, чтобы CPU в конфигурации вашей системы мог посылать напряжение 5 В для модулей расширения. Если ваша конфигурация требует больше мощности, чем может поставить CPU, вы должны удалить один модуль или выбрать CPU большей мощности. В Приложении А вы найдете информацию об источнике питания 5 В постоянного тока вашего CPU S7-200 и потребностях в мощности 5 В постоянного тока модулей расширения. С помощью информации, находящейся в Приложении В, вы можете рассчитать, какую мощность (или ток) может посылать CPU для вашей конфигурации.

Все CPU S7-200 имеют также в своем распоряжении источник питания датчиков 24 В постоянного тока, который поставляет 24 В постоянного тока для входов, для питания катушек реле на модулях расширения и других потребителей. Если ваши потребности в мощности превышают возможности источника питания датчиков, то вы должны подключить к своей системе внешний источник питания 24 В постоянного тока. В Приложении А вы найдете баланс мощностей для источника питания 24 В постоянного тока для своего конкретного CPU S7-200.

Если вам нужен внешний источник питания 24 В постоянного тока, то вы должны обратить внимание на то, чтобы этот источник питания не был подключен параллельно с источником питания датчиков CPU S7-200. Наилучшей помехоустойчивости вы достигнете, если соедините провода (М) соответствующих источников питания друг с другом.



Предупреждение

Подключение внешнего источника питания 24 В постоянного тока параллельно с источником питания 24 В постоянного тока S7-200 может вызвать конфликт между этими двумя источниками напряжения, когда каждый из них будет стремиться установить свой собственный уровень выходного напряжения.

Результатом этого конфликта может быть сокращение срока службы или немедленный выход из строя одного или обоих источников питания с последующей непредсказуемой работой системы автоматизации. Непредсказуемое функционирование может привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

Источник питания постоянного тока датчиков S7-200 и внешний источник питания должны подавать напряжение в разные точки.

Монтаж и демонтаж модулей S7-200

S7-200 может быть легко смонтирован на стандартной профильной шине или в щите управления.

Предпосылки

Перед установкой или снятием электрического устройства вы должны обеспечить, чтобы блок питания этого устройства был выключен. Обратите внимание, кроме того, на то, чтобы и все подключенное оборудование было выключено.



Предупреждение

Попытки монтажа или демонтажа S7-200 или других устройств во включенном состоянии могут привести к поражению людей электрическим током или к неверной работе оборудования.

Если питание S7-200 и всего подключенного к нему оборудования во время монтажа или демонтажа устройств не отключено, то это может привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

Всегда соблюдайте необходимые меры предосторожности и удостоверьтесь, что перед монтажом или демонтажом устройства питающее напряжение S7-200 отключено.

При монтаже или замене устройства S7-200 всегда обращайтесь внимание на то, чтобы применялся правильный модуль или устройство.



Предупреждение

При установке неправильного модуля программа в S7-200 может функционировать непредсказуемым образом.

Если устройство S7-200 заменяется другим вариантом, неправильно ориентировано или смонтировано в неправильном порядке, то это может привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

Всегда заменяйте устройство S7-200 такой же моделью и обеспечивайте его правильную ориентацию и расположение.

Монтажные размеры

CPU S7-200 и модули расширения имеют монтажные отверстия для упрощения монтажа в щитах управления. Монтажные размеры вы найдете в таблице 3-1.

Таблица 3-1. Монтажные размеры

Модуль S7-200	Ширина А	Ширина В
CPU 221 и CPU 222	90 мм	82 мм
CPU 224	120,5 мм	112,5 мм
CPU 224XP	140 мм	132 мм
CPU 226	196 мм	188 мм
Модули расширения: 4 или 8 входов/выходов пост. тока и реле (8I, 4Q, 8Q, 4I/4Q) и аналоговых выходов (2 AQ)	46 мм	38 мм
Модули расширения: 16 цифровых входов/выходов (16I, 8I/8Q), аналоговые входы/выходы (4AI, 4AI/1AQ), RTD, термopapa, PROFIBUS, Ethernet, Internet, AS-интерфейс, 8 входов/выходов перем. тока (8I и 8Q), позиционирование и модем	71,2 мм	63,2 мм
Модули расширения: 32 цифровых входа/выхода (16I/16Q)	137,3 мм	129,3 мм

Монтаж CPU или модуля расширения

Монтаж S7-200 очень прост! Просто выполните следующие шаги.

Монтаж в щите управления

1. Прорежьте отверстия (M4) в соответствии с размерами, указанными в таблице 3-1.
2. Закрепите модуль или модули в щите управления соответствующими винтами.
3. Если вы используете модуль расширения, подключите плоский кабель модуля расширения к порту расширения под передней дверцей.

Монтаж на стандартной профильной шине

1. Привинтите профильную шину к монтажной панели через каждые 75 мм.
2. Откройте стандартную (DIN) защелку (на нижней стороне модуля) и навесьте модуль задней стороной на профильную шину.
3. Если вы используете модуль расширения, подключите плоский кабель модуля расширения к порту расширения под передней дверцей.
4. Поверните модуль вниз по направлению к профильной шине и закройте защелку. Тщательно проверьте, что защелка правильно защелкнулась, и модуль надежно закреплен на шине. Во избежание повреждения модуля нажимайте на отверстие, а не непосредственно на переднюю стенку модуля.



Совет

В средах, в которых возникают сильные вибрации, или при вертикальном монтаже S7-200 может оказаться полезным закреплять S7-200 на профильной шине с помощью заземляющих клемм.

В средах, в которых возникают сильные вибрации, монтаж S7-200 в щите управления обеспечивает лучшую защиту от вибраций.

Демонтаж CPU или модуля расширения

Для демонтажа CPU S7-200 или модуля расширения действуйте следующим образом:

1. Выключите источник питания S7-200.
2. Отсоедините от устройства все провода и кабели. Это облегчается тем, что большинство CPU S7-200 и модулей расширения имеют съемные клеммные блоки.
3. Если к демонтируемому устройству присоединены модули расширения, то откройте переднюю дверцу и отсоедините плоский кабель соседнего модуля расширения.
4. Отверните монтажные винты или откройте стандартную защелку.
5. Снимите модуль с шины.

Демонтаж и обратная установка клеммного блока

Большинство модулей S7-200 снабжены штепсельными разъемами, которые облегчают монтаж и демонтаж модуля. Обратитесь к Приложению А, чтобы узнать, снабжен ли ваш модуль S7-200 съемным клеммным блоком. Для модулей, у которых нет штепсельных разъемов, вы можете заказать съемные клеммные блоки. Номера для заказа вы найдете в Приложении Е.

Демонтаж клеммного блока

1. Откройте переднюю дверцу, чтобы получить доступ к клеммному блоку.
2. Вставьте небольшую отвертку в углубление в середине клеммного блока.
3. Снимите клеммный блок, отжимая его отверткой от корпуса S7-200. См. рис. 3-2.

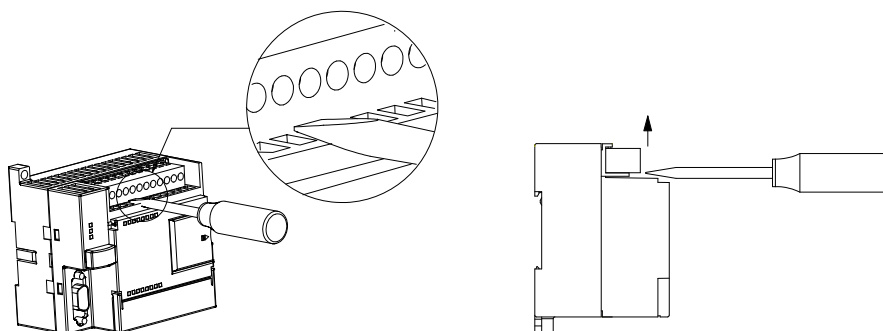


Рис. 3-2. Демонтаж клеммного блока

Обратная установка клеммного блока

1. Откройте переднюю дверцу.
2. Выровняйте клеммный блок с помощью штифтов на устройстве, а также проводку клеммного блока по краю основания блока.
3. Сильно надавите клеммный блок вращательным движением вниз, пока он не защелкнется. Тщательно проверьте, правильно ли клеммный блок выровнен и прочно ли он закреплен.

Указания по заземлению и подключению

Надлежащее заземление и подключение всех электрических устройств важно для оптимальной работы вашей системы и для обеспечения дополнительной помехоустойчивости вашего приложения и S7-200.

Предпосылки

Перед заземлением или подключением электрического устройства обеспечьте отключение питающего напряжения этого устройства. Обратите также внимание на то, чтобы и все присоединенные устройства были выключены.

При подключении S7-200 и всего присоединенного к нему оборудования обеспечьте выполнение всех действующих обязательных стандартов. При монтаже и эксплуатации оборудования учитывайте соответствующие национальные и региональные предписания. Узнайте у местных властей, каким стандартам и предписаниям необходимо следовать в вашем конкретном случае.



Предупреждение

Попытки монтажа или подключения S7-200 или присоединенных к нему устройств во включенном состоянии могут привести к поражению людей электрическим током или к неверной работе оборудования. Если питание S7-200 и всего подключенного к нему оборудования во время монтажа или демонтажа устройств не отключено, то это может привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

Всегда соблюдайте необходимые меры предосторожности и удостоверьтесь, что перед монтажом или демонтажем устройства питающее напряжение S7-200 отключено.

При проектировании заземления и подключения вашей системы S7-200 всегда учитывайте требования безопасности. Электронные устройства управления, подобные S7-200, могут выйти из строя и вызвать неожиданное поведение управляемого и контролируемого оборудования. Поэтому для защиты персонала и имущества вам следует реализовать меры безопасности, не зависящие от S7-200.



Предупреждение

Устройства управления могут выходить из строя в режимах, не обеспечивающих безопасность, вызывая неконтролируемое функционирование управляемых устройств. Результатом этого является непредвиденное поведение системы автоматизации, которое может привести к гибели или тяжким телесным повреждениям персонала и/или к повреждению оборудования.

Поэтому позаботьтесь о функции аварийного останова, об электромеханических или иных резервных устройствах обеспечения безопасности, не зависящих от S7-200.

Указания по разделению потенциалов

Граничные значения источника питания переменного тока и граничные значения входов/выходов цепей переменного тока у S7-200 имеют номинальное значение 1500 В. Эти границы для разделения потенциалов были проверены и обеспечивают надежную электрическую развязку между линиями переменного тока и цепями низкого напряжения.

Все цепи низкого напряжения, подключенные к S7-200, например, снабжение напряжением 24 В, должны получать питание от проверенного источника, обеспечивающего надежную потенциальную развязку с линией переменного тока и другими высокими напряжениями. Такие источники имеют двойную изоляцию в соответствии с международными стандартами электробезопасности и выходы, соответствующие стандартам с обозначением SELV, PELV, Class 2 или Limited Power [Ограничение мощности]..



Предупреждение

Использование источников питания без потенциальной развязки или с однократной изоляцией цепей низкого напряжения от линий переменного тока может привести к появлению опасных напряжений в цепях, которые считаются безопасными при прикосновении, например, линии связи или провода датчиков низкого напряжения. Такие неожиданные высокие напряжения могут привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

Применяйте только такие преобразователи высокого напряжения в низкое, которые имеют допуск к эксплуатации как источники для цепей с ограниченным напряжением, безопасным при прикосновении.

Указания по заземлению S7-200

Заземление вашего приложения будет наилучшим, если все общие клеммы и все клеммы заземления вашего S7-200 и всех подключенных устройств присоединены к земле в одной точке. Эта точка должна быть непосредственно соединена с системной землей.

Для достижения оптимальной помехоустойчивости рекомендуется, чтобы все общие обратные цепи постоянного тока были присоединены к земле в одной точке. Соедините общую клемму (M) источника питания датчиков 24 В постоянного тока с землей.

Все заземляющие провода должны быть возможно более короткими и иметь большое поперечное сечение, напр., 2 мм² (14 AWG).

При выборе точек заземления учитывайте соответствующие предписания по технике безопасности и обеспечивайте надлежащее функционирование защитных устройств, осуществляющих разрыв цепей тока.

Указания по подключению S7-200

При проектировании электропроводки для вашего S7-200 предусмотрите единое разъединяющее устройство, которое одновременно отключает напряжение источника питания CPU S7-200 и напряжение всех входных и выходных цепей. Предусмотрите защиту от перегрузки, например, с помощью плавкого предохранителя или защитного выключателя, чтобы ограничить аварийный ток в питающих проводах. Вы можете реализовать дополнительную защиту с помощью предохранителя или других ограничителей тока в отдельных выходных цепях.

Снабдите все провода, которые подвергаются опасности удара молнии, надлежащей защитой от перенапряжений.

Избегайте размещать линии сигналов низкого напряжения и кабели связи в одном кабельном канале с питающими линиями переменного тока и быстро переключаемыми линиями постоянного тока. Провода следует прокладывать парами: нейтральный или нулевой провод вместе с фазовым или сигнальным проводом.

Прокладывайте провода по возможно более короткому пути и обращайтесь внимание на то, чтобы поперечное сечение провода соответствовало требуемому току. Клеммный блок можно присоединять проводами с поперечным сечением от 2 мм² до 0,3 мм² (14 AWG – 22 AWG). Для оптимальной защиты от электрических помех используйте экранированные провода. Как правило, наилучшие результаты достигаются при заземлении экрана на S7-200.

При подключении входной цепи, получающей питание от внешнего источника, включайте в эту цепь защиту от перегрузки. Для цепей, получающих питание от источника питания датчиков напряжением 24 В постоянного тока из S7-200, внешняя защита не нужна, так как источник питания датчиков уже имеет ограничитель тока.

Большинство модулей S7-200 имеют съемные клеммные блоки для подключения пользовательской проводки. (В Приложении А вы узнаете, есть ли у вашего модуля съемный клеммный блок.) Во избежание обрыва соединений обратите внимание на то, чтобы клеммный блок был надежно закреплен и провода были прочно присоединены к клеммному блоку. Во избежание повреждения клеммного блока не заворачивайте винты слишком сильно. Для винтов клеммного блока максимальный крутящий момент составляет 0,56 Нм.

S7-200 работает с границами разделения потенциалов, что препятствует возникновению нежелательных токов в вашей установке. При планировании электропроводки для вашей системы следует учитывать эти границы разделения потенциалов. Значения предоставляемой в распоряжение потенциальной развязки и расположение границ разделения потенциалов вы найдете в Приложении А. Границы разделения потенциалов с номинальным напряжением ниже 1500 В переменного тока не должны использоваться как единственные границы безопасности.



Совет

В сети связи, при отсутствии шинного усилителя, максимальная длина кабеля связи составляет 50 м. Коммуникационный порт на S7-200 не имеет потенциальной развязки. Подробную информацию об этом вы найдете в главе 7.

Указания для индуктивных нагрузок

Снабжайте индуктивные нагрузки защитными схемами, чтобы ограничить нарастание напряжения при отключении выхода устройства управления. Защитные схемы предотвращают преждевременный износ выходов, вызываемый большими индуктивными коммутационными токами. Кроме того, защитные схемы ограничивают величину электрических помех, возникающих при выключении индуктивных нагрузок.



Совет

Эффективность защитной цепи зависит от соответствующего приложения и должна быть проверена в каждом конкретном случае. Все компоненты защитной схемы должны быть рассчитаны для использования в этом приложении.

Выходы постоянного тока и реле, управляющие нагрузками постоянного тока

Выходы постоянного тока имеют внутреннюю защиту, пригодную для большинства приложений. Так как реле могут использоваться для нагрузок как постоянного, так и переменного тока, то внутренняя защита отсутствует.

На рис. 3-3 показан пример защитной схемы для нагрузки постоянного тока. В большинстве приложений достаточно применения одного диода (А), включенного параллельно индуктивной нагрузке, но если ваше приложение требует быстрого отключения, то рекомендуется, кроме того, использовать стабилитрон (В). Рассчитывайте стабилитрон на ток в выходной цепи.

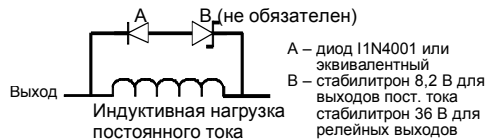


Рис. 3-3. Защитная схема для нагрузки постоянного тока

Выходы переменного тока и реле, управляющие нагрузками переменного тока

Выходы переменного тока имеют внутреннюю защиту, пригодную для большинства приложений. Так как реле могут использоваться для нагрузок как постоянного, так и переменного тока, то внутренняя защита отсутствует.

На рис. 3-4 показан пример защитной схемы для нагрузки переменного тока. При использовании реле или выхода переменного тока для включения нагрузок напряжением 115 /230 В перемен. тока поместите параллельно нагрузке переменного тока цепочку из резистора и конденсатора, как показано на этом рисунке. Вы можете использовать также металлооксидный варистор (MOV) для ограничения пикового напряжения. Обратите внимание на то, чтобы рабочее напряжение варистора было, по крайней мере, на 20 % больше, чем номинальное напряжение сети.

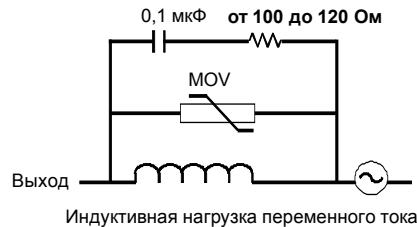


Рис. 3-4. Защитная схема для нагрузки переменного тока

Указание

Если для включения индуктивных нагрузок переменного тока 230 В используются релейные модули расширения, то параллельно нагрузке переменного тока должна быть помещена цепочка из резистора и конденсатора для подавления помех, как показано на рис. 3-4.

Указания для ламповых нагрузок

Ламповые нагрузки повреждают контакты реле из-за большого броска тока при включении. Этот бросок тока обычно в 10 – 15 раз превышает установившийся ток лампы с вольфрамовой нитью. Для ламповых нагрузок, которые очень часто включаются в течение срока службы приложения, рекомендуется использовать сменное промежуточное реле или ограничитель бросков тока.

Основы ПЛК

4

Основной функцией S7-200 является контроль полевых входов и, на основе логики управления, включение и выключение полевых выходных устройств. В этой главе объясняются основы выполнения программы, различные виды используемой памяти и способы сохранения.

В этой главе

Выполнение логики управления с помощью S7-200	24
Доступ к данным S7-200	26
Сохранение и извлечение данных с помощью S7-200	36
Установка режима работы CPU S7-200	41
Использование проводника S7-200	41
Функции S7-200	42

Выполнение логики управления с помощью S7-200

S7-200 обрабатывает логику управления в вашей программе циклически, считывая и записывая данные.

S7-200 ставит вашу программу в соответствие физическим входам и выходам

Основной принцип действия S7-200 очень прост:

- S7-200 считывает состояние входов.
- Программа, хранящаяся в S7-200, использует эти входы для анализа логики управления. Во время обработки программы S7-200 обновляет данные.
- S7-200 записывает данные на выходы.

На рис. 4-1 показана связь между простой коммутационной схемой и S7-200. В этом примере состояние выключателя для запуска двигателя логически связано с состояниями других входов. Оценки этих состояний определяют затем сигнальное состояние выхода для исполнительного устройства, которое запускает двигатель.

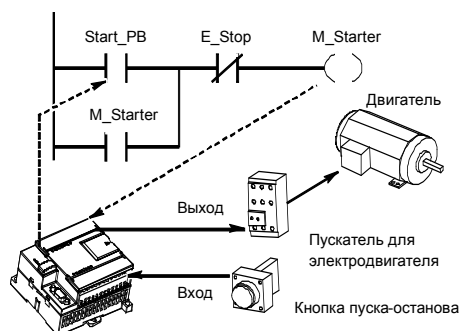


Рис. 4-1. Управление входами и выходами

S7-200 выполняет все задачи в цикле

S7-200 выполняет последовательность задач неоднократно. Эта регулярная обработка задач называется циклом. Как показано на рис. 4-2, S7-200 выполняет в цикле большинство или все из следующих задач:

- Чтение входов: S7-200 копирует состояние физических входов в регистр входов образа процесса.
- Выполнение логики управления в программе: S7-200 выполняет команды программы и сохраняет значения в различных областях памяти.
- Обработка запросов на обмен данными: S7-200 выполняет все задачи, необходимые для обмена данными.
- Самодиагностика CPU: S7-200 проверяет, чтобы встроенное программное обеспечение, программная память и все модули расширения работали надлежащим образом.
- Запись в выходы: Значения, хранящиеся в регистре выходов образа процесса, записываются в физические выходы.

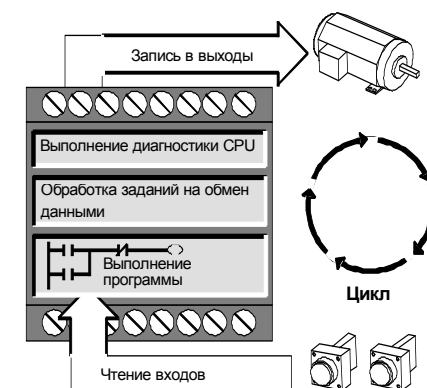


Рис. 4-2. Цикл S7-200

Выполнение программы пользователя зависит от того, находится ли S7-200 в состоянии STOP или в состоянии RUN. В состоянии RUN ваша программа выполняется; в состоянии STOP ваша программа не выполняется.

Чтение входов

Цифровые входы: В начале цикла текущие значения цифровых входов считываются, а затем записываются в регистр входов образа процесса.

Аналоговые входы: S7-200 не обновляет аналоговые входы модулей расширения автоматически как часть цикла, если вы не активизировали фильтрацию аналоговых входов. Аналоговый фильтр обеспечивает стабильность сигналов. Вы можете активизировать аналоговый фильтр для каждого входа.

Если фильтр для аналогового входа активизирован, то S7-200 обновляет этот аналоговый вход один раз за цикл, выполняет функцию фильтрации и сохраняет отфильтрованное значение внутри. Это отфильтрованное значение затем предоставляется в распоряжение всякий раз, когда ваша программа обращается к этому аналоговому входу.

Если фильтр аналогового входа выключен, то S7-200 считывает значение этого аналогового входа из модуля расширения всякий раз, когда ваша программа обращается к аналоговому входу.

Аналоговые входы AIW0 и AIW2 модуля CPU 224XP обновляются в каждом цикле самыми последними результатами аналого-цифрового преобразователя. Этот преобразователь работает со средними значениями (σ - δ), и эти значения обычно не нуждаются в программной фильтрации.



Совет

Фильтр аналогового входа обеспечивает стабильность аналоговых значений. Фильтр аналогового входа следует активизировать в приложениях, в которых входной сигнал медленно меняется с течением времени. Если речь идет о быстро меняющемся сигнале, то аналоговый фильтр активизировать не следует.

Не применяйте аналоговый фильтр у модулей, которые передают цифровые данные или сигналы тревоги в аналоговых словах. Всегда выключайте аналоговый фильтр для ведущих модулей с RTD, термодарами и AS-интерфейсом.

Исполнение программы

На этом участке цикла S7-200 обрабатывает программу с первой команды до последней. Вы можете непосредственно управлять входами и выходами и получать, таким образом, доступ к ним во время исполнения основной программы или программы обработки прерываний.

Если вы используете в своей программе прерывания, то программы обработки прерываний, которые ставятся в соответствие прерывающим событиям, хранятся как часть основной программы. Однако программы обработки прерываний исполняются не как составная часть нормального цикла, а только тогда, когда происходит прерывающее событие (оно возможно в любом месте цикла).

Обработка запросов на обмен данными

На участке цикла, выделенном для обработки коммуникаций, S7-200 обрабатывает все сообщения, полученные из коммуникационного порта или от интеллектуальных модулей ввода/вывода.

Самодиагностика CPU

На этом участке цикла S7-200 проверяет надлежащую работу CPU, области памяти и состояние модулей расширения.

Запись в цифровые выходы

В конце каждого цикла S7-200 записывает значения, хранящиеся в регистре выходов образа процесса, в цифровые выходы. (Аналоговые выходы обновляются немедленно, независимо от цикла.)

Доступ к данным S7–200

S7–200 хранит информацию в различных местах памяти, которые имеют однозначные адреса. Вы можете явно указать адрес в памяти, к которому вы хотите обратиться. Благодаря этому ваша программа имеет прямой доступ к информации. Таблица 4–1 показывает диапазон целых значений, которые могут быть представлены с помощью данных различной длины.

Таблица 4–1. Десятичные и шестнадцатеричные диапазоны для данных различной длины

Представление	Байт (B)	Слово (W)	Двойное слово (D)
Целое без знака	от 0 до 255 от 0 до FF	от 0 до 65 535 от 0 до FFFF	от 0 до 4 294 967 295 от 0 до FFFF FFFF
Целое со знаком	от -128 до +127 от 80 до 7F	от -32 768 до +32 767 от 8000 до 7FFF	от -2 147 483 648 до +2 147 483 647 от 8000 0000 до 7FFF FFFF
Вещественное IEEE 32–битовое с плавающей точкой	<i>Неприменимо</i>	<i>Неприменимо</i>	от +1.175495E-38 до +3.402823E+38 (положительное) от -1.175495E-38 до -3.402823E+38 (отрицательное)

Для обращения к биту в некоторой области памяти вы должны указать адрес бита. Этот адрес состоит из идентификатора области памяти, адреса байта и номера бита. На рис. 4–3 показан пример обращения к биту (адресация в формате «байт.бит»). В этом примере за область памяти и адресом байта (I = input [вход], 3 = байт 3) следует точка («.»), чтобы отделить адрес бита (бит 4).

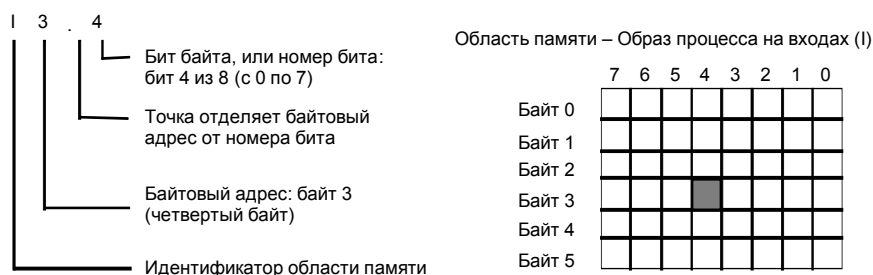


Рис. 4–3. Адресация байт.бит

Применяя формат байт.бит, вы можете обратиться к данным в большинстве областей памяти (V, I, Q, M, S, L и SM) как к байтам, словам или двойным словам. Если вы хотите обратиться к байту, слову или двойному слову данных в памяти, то вы должны указать эти адреса подобно адресу бита. Вы указываете идентификатор области, обозначение длины данных и начальный адрес байта, слова или двойного слова, как показано на рис. 4–4.

К данным в других областях памяти (напр., T, C, HC и аккумуляторы) вы обращаетесь, указывая в качестве адреса идентификатор области и номер элемента.

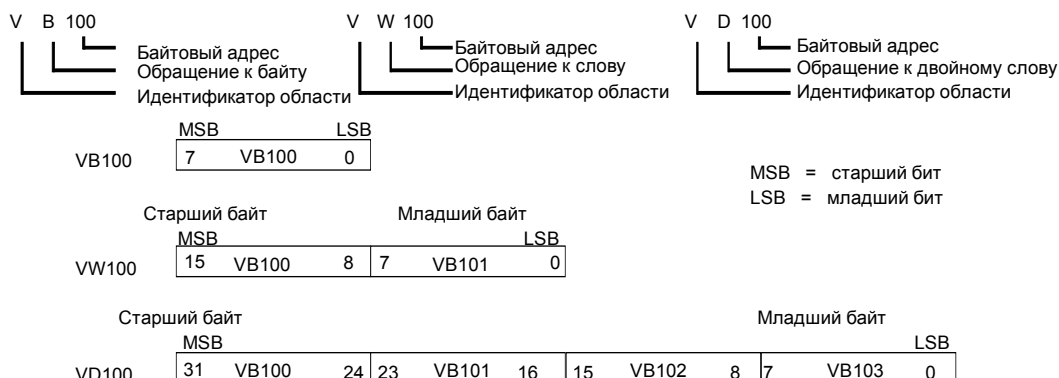


Рис. 4–4. Обращение к одному и тому же адресу в формате байта, слова и двойного слова

Обращение к данным в областях памяти

Регистр входов образа процесса: I

В начале каждого цикла S7–200 опрашивает физические входы и записывает полученные значения в регистр входов образа процесса. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $I[\text{адрес байта}].[\text{адрес бита}]$ I0.1
 Байт, слово или двойное слово: $I[\text{длина}][\text{начальный адрес байта}]$ IB4

Регистр выходов образа процесса: Q

В конце цикла S7–200 копирует значения, хранящиеся в регистре выходов образа процесса, в физические выходы. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $Q[\text{адрес байта}].[\text{адрес бита}]$ Q1.1
 Байт, слово или двойное слово: $Q[\text{длина}][\text{начальный адрес байта}]$ QB5

Область памяти переменных: V

Память переменных можно использовать для хранения промежуточных результатов операций, выполняемых в вашей программе. В памяти переменных вы можете хранить также другие данные, имеющие отношение к процессу или к решению вашей задачи автоматизации. К памяти переменных можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $V[\text{адрес байта}].[\text{адрес бита}]$ V10.2
 Байт, слово или двойное слово: $V[\text{длина}][\text{начальный адрес байта}]$ VW100

Область битовой памяти: M

Биты памяти (меркеры) можно использовать как управляющие реле для хранения промежуточных результатов операций или другой управляющей информации. К битам памяти можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $M[\text{адрес байта}].[\text{адрес бита}]$ M26.7
 Байт, слово или двойное слово: $M[\text{длина}][\text{начальный адрес байта}]$ MD20

Таймеры: T

S7-200 имеет в своем распоряжении таймеры, которые отсчитывают приращения времени с разрешениями (шагами базы времени) 1 мс, 10 мс или 100 мс. С таймером связаны две переменные:

- Текущее значение: это 16-битовое целое со знаком хранит количество времени, отсчитанное таймером.
- Бит таймера: этот бит устанавливается или сбрасывается, когда текущее значение становится равным предустановленному значению. Предустановленное значение вводится как часть таймерной команды.

Вы обращаетесь к обоим этим элементам данных через адрес таймера (T + номер таймера). Происходит ли обращение к биту таймера или к текущему значению, зависит от используемой команды: команды с операндами в битовом формате обращаются к биту таймера, тогда как команды с операндами в формате слова обращаются к текущему значению. Как показано на рис. 4–5, команда "Нормально открытый контакт" обращается к биту таймера, а команда "Передать слово" обращается к текущему значению таймера.



Рис. 4–5. Обращение к биту или к текущему значению таймера

Счетчики: C

S7-200 имеет в своем распоряжении три вида счетчиков, которые подсчитывают нарастающие фронты на счетных входах счетчика: один вид счетчиков ведет прямой счет, другой считает только в обратном направлении, а третий вид считает в обоих направлениях. Со счетчиком связаны две переменные:

- Текущее значение: это 16-битовое целое со знаком хранит счетное значение, накопленное счетчиком.
- Бит счетчика: этот бит устанавливается или сбрасывается, когда текущее значение становится равным предустановленному значению. Предустановленное значение вводится как часть команды счетчика.

Вы обращаетесь к обоим этим элементам данных через адрес счетчика (C + номер счетчика). Происходит ли обращение к биту счетчика или к текущему значению, зависит от используемой команды: команды с операндами в битовом формате обращаются к биту счетчика, тогда как команды с операндами в формате слова обращаются к текущему значению. Как показано на рис. 4–6, команда "Нормально открытый контакт" обращается к биту счетчика, а команда "Передать слово" обращается к текущему значению счетчика.

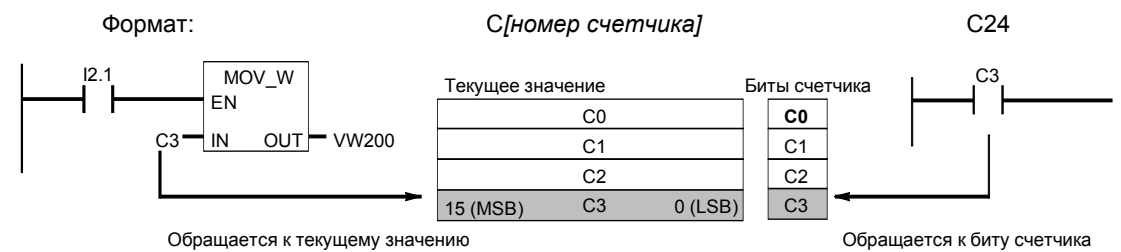


Рис. 4–6. Обращение к биту или к текущему значению счетчика

Скоростные счетчики: HC

Скоростные счетчики подсчитывают быстрые события независимо от цикла CPU. Скоростные счетчики имеют в своем распоряжении 32-битовое целое счетное значение (текущее значение). Для обращения к счетному значению скоростного счетчика введите его адрес, указав область памяти (HC) и номер счетчика (напр., HC0). Текущее значение скоростного счетчика защищено от записи и может быть адресовано только в формате двойного слова (32 бита).

Формат: HC[номер скоростного счетчика] HC1

Аккумуляторы: AC

Аккумуляторы – это элементы чтения/записи, которые могут использоваться как память. Например, вы можете использовать аккумуляторы для передачи параметров в подпрограммы и из них или для хранения промежуточных результатов расчетов. S7-200 имеет в своем распоряжении четыре 32-битовых аккумулятора (AC0, AC1, AC2 и AC3). К данным в аккумуляторах можно обратиться в формате бита, слова или двойного слова.

Длина данных, к которым производится обращение, зависит от команды, которая используется для обращения к аккумулятору. Как показано на рис. 4–7, при обращении к аккумулятору в формате бита или слова используются младшие 8 или 16 битов значения, хранящегося в аккумуляторе. При обращении к аккумулятору в формате двойного слова используются все 32 бита.

Информацию об использовании аккумуляторов в программах обработки прерываний вы найдете в разделе, посвященном прерываниям, главы 6.

Формат:

AC[номер аккумулятора] AC0

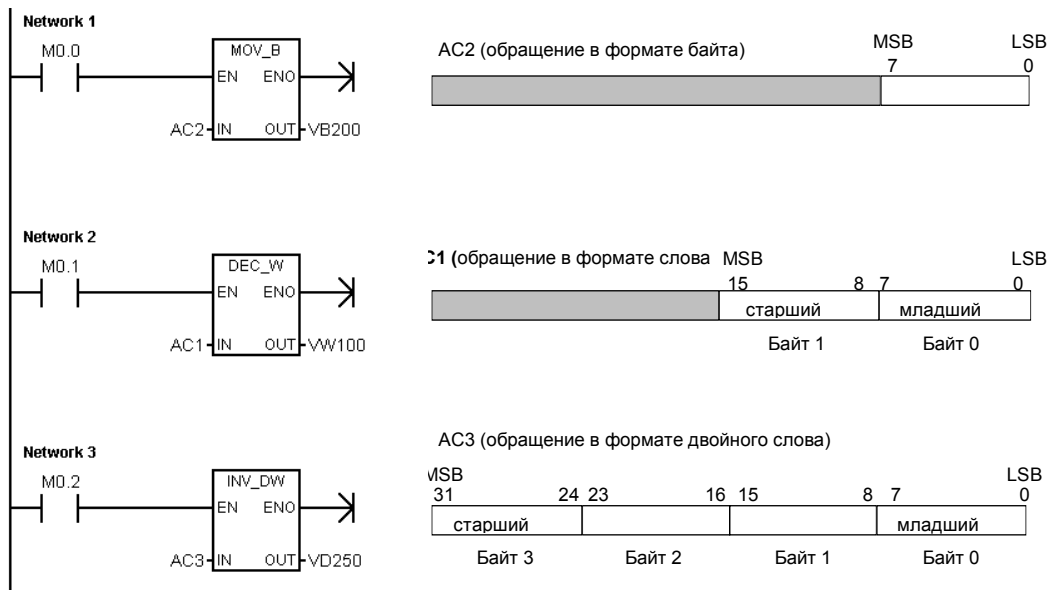


Рис. 4–7. Обращение к аккумуляторам


Специальные биты памяти: SM

Специальные биты памяти (SM) предоставляют средство для обмена данными между CPU и вашей программой. Вы можете использовать эти биты для выбора и управления некоторыми специальными функциями CPU S7-200, например: бит, который устанавливается только в первом цикле; бит, который устанавливается и сбрасывается с фиксированной частотой, или бит, который указывает на состояние арифметической или иной команды. (Подробную информацию о специальных битах памяти вы найдете в Приложении D.) К SM-битам можно обращаться в формате бита, слова или двойного слова:

Бит: $SM[\text{адрес байта}].[\text{адрес бита}]$ SM0.1
 Байт, слово или двойное слово: $SM[\text{длина}][\text{начальный адрес байта}]$ SMB86

Память локальных данных: L

S7-200 имеет в своем распоряжении 64 байта локальной памяти, из которых 60 могут быть использованы в качестве промежуточной памяти или для передачи формальных параметров в подпрограммы.

	<p>Совет При программировании в LAD или FBD последние четыре байта зарезервированы для STEP 7-Micro/WIN.</p>
---	--

Память локальных данных похожа на память переменных с одним существенным отличием. Память переменных доступна глобально, тогда как память локальных данных доступна локально. Глобальная доступность означает, что к адресу в этой области памяти можно обратиться из любой организационной единицы программы (из основной программы, подпрограммы или подпрограмм обработки прерываний). Локальная доступность означает, что эта область памяти ставится в соответствие определенной организационной единице программы. S7-200 выделяет 64 байта локальной памяти для главной программы, 64 байта для каждого уровня вложенности подпрограмм и 64 байта для программ обработки прерываний.

К области локальных данных, поставленной в соответствие основной программе, не имеют доступа подпрограммы и программы обработки прерываний. Подпрограмма не может обращаться к области локальных данных основной программы, программы обработки прерываний или другой подпрограммы. Аналогично, программа обработки прерываний не имеет доступа к области локальных данных основной программы или подпрограммы.

S7-200 выделяет область локальных данных по мере необходимости. Это значит, что при выполнении основной программы области локальных данных для подпрограмм и программ обработки прерываний не существуют. Если возникает прерывание или вызывается подпрограмма, то по потребности выделяется локальная память. Вновь выделенная локальная память может снова использовать те же адреса, которые использовались другой подпрограммой или программой обработки прерываний.

S7-200 не инициализирует область локальных данных к моменту ее назначения, поэтому она может содержать любые значения. Если при вызове подпрограммы передаются формальные параметры, то S7-200 сохраняет значения передаваемых параметров в соответствующих адресах области локальных данных, выделенной этой подпрограмме. Адреса в области локальных данных, которые не получили значений при передаче формальных параметров, не инициализируются и при выделении могут содержать произвольные значения.

Бит: $L[\text{адрес байта}].[\text{адрес бита}]$ L0.0
 Байт, слово или двойное слово: $L[\text{длина}][\text{начальный адрес байта}]$ LB33

Аналоговые входы: AI

S7-200 преобразует аналоговые величины (например, температуру или напряжение) в цифровые величины, имеющие длину слова (16 битов). Обращение к этим значениям производится через идентификатор области (AI), длину данных (W) и начальный адрес байта. Так как в случае аналоговых входов речь идет о словах, которые всегда начинаются на байтах с четными номерами (например, 0, 2, 4 и т.д.), то обращаются к этим значениям с помощью адресов четных байтов (например, AIW0, AIW2, AIW4). Аналоговые входы можно только считывать.

Формат: $AIW[\text{начальный адрес байта}] \quad AIW4$

Аналоговые выходы: AQ

S7-200 преобразует цифровые величины, имеющие длину слова (16 битов), в ток или напряжение пропорционально цифровой величине. Обращение к этим значениям производится через идентификатор области (AQ), длину данных (W) и начальный адрес байта. Так как в случае аналоговых выходов речь идет о словах, которые всегда начинаются на байтах с четными номерами (например, 0, 2, 4 и т.д.), то эти значения записываются с адресами четных байтов (например, AQW0, AQW2, AQW4). Аналоговые выходы можно только записывать.

Формат: $AQW[\text{начальный адрес байта}] \quad AQW4$

Реле управления очередностью (SCR): S

SCR или S-биты разделяют функционирование установки на отдельные шаги или эквивалентные части программы. С помощью реле управления очередностью программа управления представляется в виде структуры, состоящей из логических сегментов. К S-битам можно обращаться в формате бита, слова или двойного слова.

Бит: $S[\text{адрес байта}].[\text{адрес бита}] \quad S3.1$
 Байт, слово или двойное слово: $S[\text{длина}][\text{начальный адрес байта}] \quad SB4$

Формат вещественных чисел

Вещественные числа (или числа с плавающей точкой) представляются как 32-битовые числа однократной точности, формат которых описан в стандарте ANSI/IEEE 754-1985. См. рис. 4-8. Обращение к вещественным числам производится в формате двойного слова.

У S7-200 числа с плавающей точкой имеют точность до 6 десятичных разрядов. Поэтому при вводе константы с плавающей точкой можно указывать до 6 десятичных разрядов.

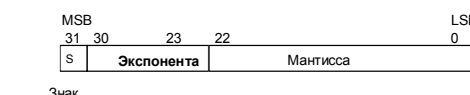


Рис. 4-8. Формат вещественного числа

Точность при вычислениях с вещественными числами

Расчеты, включающие в себя длинные последовательности значений, содержащие очень большие и очень малые числа, могут привести к неточным результатам. Это может произойти, если числа отличаются друг от друга в 10 в степени x раз, где x > 6.

Например: $100\ 000\ 000 + 1 = 100\ 000\ 000$

Формат для строк

Строка – это последовательность символов, причем каждый символ хранится как байт. Первый байт строки определяет ее длину, т.е. количество содержащихся в ней символов. На рис. 4-9 показан формат строки. Строка может включать в себя от 0 до 254 символов, плюс байт, содержащий информацию о длине, таким образом, максимальная длина строки равна 255 байтам. Строковая константа ограничена 126 байтами.

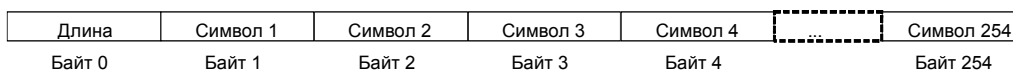


Рис. 4-9. Формат строк

Задание постоянного значения для команд S7–200

Во многих командах для S7–200 можно использовать константы. Константы могут быть байтами, словами или двойными словами. S7–200 хранит все константы в виде двоичных чисел, которые могут быть представлены в десятичном, шестнадцатеричном формате, в формате ASCII или в формате вещественных чисел (чисел с плавающей точкой). См. таблицу 4–2.

Таблица 4–2. Представление постоянных величин

Представление	Формат	Пример
Десятичное	[десятичное значение]	20047
Шестнадцатеричное	16#[шестнадцатеричное значение]	16#4E4F
Двоичное	2#[двоичное число]	2#1010_0101_1010_0101
ASCII	'[текст ASCII]'	'ABCD'
Вещественное	ANSI/IEEE 754-1985	+1.175495E-38 (положительное) –1.175495E-38 (отрицательное)
Строка	«[текст строки]»	«ABCDE»



Совет

У CPU S7–200 нельзя указывать конкретные типы данных (когда вы, например, хотите указать, что константа должна быть сохранена как целое число (16 битов), целое число со знаком или двойное целое (32 бита)). Например, команда сложения может использовать значение, хранящееся в VW100, как целое число со знаком, а команда "Исключающее ИЛИ" то же самое значение в VW100 может использовать как двоичное значение без знака.

Адресация встроенных входов/выходов и входов/выходов модулей расширения

Встроенные входы и выходы центрального устройства (CPU) имеют фиксированные адреса. Вы можете добавить входы и выходы к CPU S7–200, подключив с правой стороны CPU модули расширения. Адреса входов и выходов на модуле расширения определяются видом входов и выходов, а у нескольких модулей одного типа также их расположением. Например, модуль вывода не влияет на адреса модуля ввода и наоборот. Адреса входов и выходов аналоговых и цифровых модулей также не зависят друг от друга.



Совет

Для цифровых входов и выходов в образе процесса предусмотрены участки по восемь битов (одному байту) каждый. Если в модуле не для каждого бита зарезервированного байта имеется физический вход или выход, то свободные биты теряются и не могут быть поставлены в соответствие следующим модулям расширения этого CPU. У модулей ввода свободные биты в зарезервированных байтах в каждом цикле обновления устанавливаются в ноль.

Аналоговые входы и выходы всегда назначаются двойными шагами. Если в модуле не для каждого из этих входов и выходов имеется физический вход или выход, то эти входы и выходы теряются и не могут быть поставлены в соответствие следующим модулям расширения.

На рис. 4–10 показан пример нумерации входов и выходов для конкретной конфигурации аппаратуры. Пропуски в адресации (показаны серым курсивом) не могут использоваться вашей программой.



Рис. 4–10. Пример адресов встроенных входов/выходов и входов/выходов модулей расширения (CPU 224XP)

Косвенная адресация областей памяти S7–200 с помощью указателей

Косвенная адресация использует указатель для доступа к данным в памяти. Указатели – это ячейки памяти, имеющие размер двойного слова, которые содержат адрес другой ячейки памяти. В качестве указателей можно использовать только ячейки памяти переменных и локальных данных или аккумуляторные регистры (AC1, AC2 или AC3). Для создания указателя необходимо использовать команду "Переместить двойное слово". Эта команда передает адрес косвенно адресованной ячейки памяти в ячейку указателя. Указатели могут также передаваться в подпрограмму в качестве параметров.

S7–200 дает возможность использования указателей для косвенной адресации следующих областей памяти: I, Q, V, M, S, AI, AQ, SM, T (только текущее значение) и C (только текущее значение). Косвенную адресацию нельзя использовать для обращения к отдельному биту или к областям памяти HC или L.

Если вы хотите косвенно обратиться к данным, расположенным по некоторому адресу в памяти, вы можете создать указатель на этот адрес, введя амперсанд (&) и соответствующий адрес. Входному операнду команды должен предшествовать амперсанд (&), чтобы указать на необходимость перемещения в ячейку, обозначенную в выходном операнде команды (указателе), адреса ячейки памяти, а не ее содержимого.

Ввод астериска (*) перед операндом команды указывает, что этот операнд является указателем. Как показано на рис. 4–11, ввод *AC1 указывает, что AC1 является указателем на слово, на которое ссылается команда "Переместить слово" (MOVW). В этом примере значения, хранящиеся в VB200 и VB201, перемещаются в аккумулятор AC0.

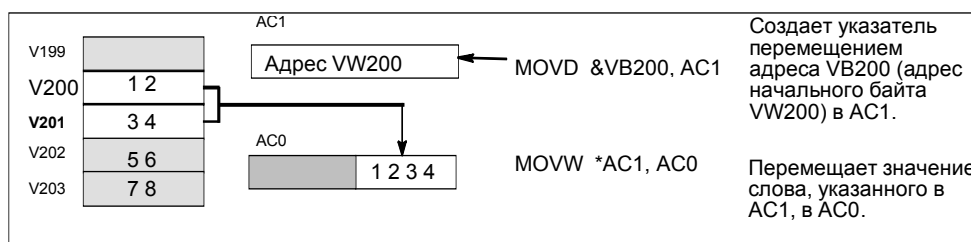


Рис. 4–11. Создание и использование указателя

Как показано на рис. 4–12, вы можете изменить значение указателя. Так как указатели имеют размер 32 бита, то для изменения значений указателей используйте операции над двойными словами. Для изменения значений указателей могут использоваться такие простые математические операции, как сложение или инкрементирование.

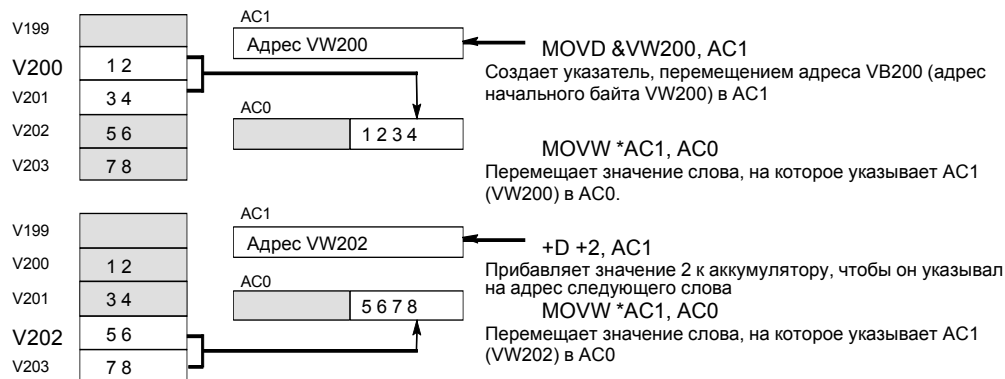


Рис. 4–12. Изменение указателя

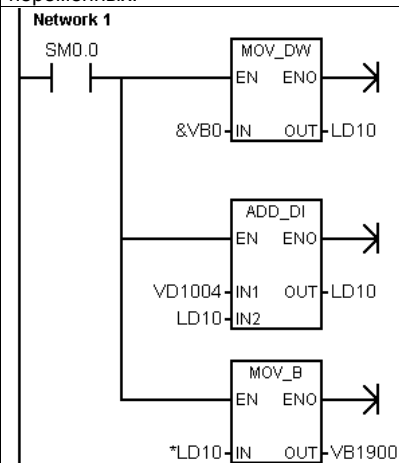


Совет

Не забывайте указывать длину данных, к которым вы хотите обратиться: для обращения к байту увеличьте значение указателя на 1; для обращения к слову или текущему значению таймера или счетчика, увеличьте значение указателя на 2, для обращения к двойному слову увеличьте значение указателя на 4.

Пример программы для обращения к данным в памяти переменных с использованием смещения

В этом примере используется LD10 как указатель на адрес VB0. Затем вы увеличиваете указатель на величину смещения, хранящуюся в VD1004. Теперь LD10 указывает на другой адрес в памяти переменных (VB0 + смещение). Значение, хранящееся в памяти переменных по адресу, на который указывает LD10, копируется в VB1900. Изменяя значение в VD1004, вы можете обратиться к любому адресу в памяти переменных.



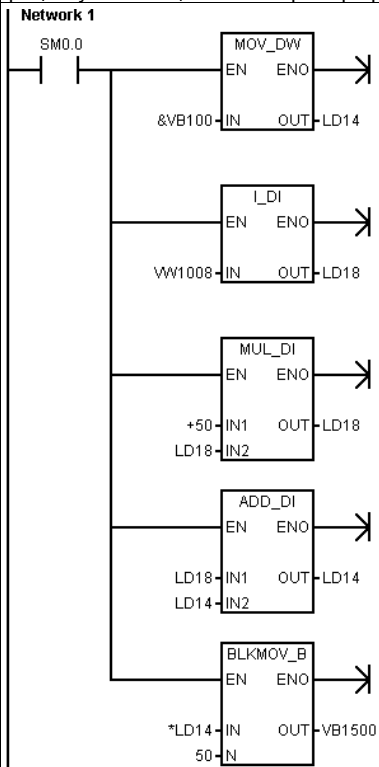
Сегмент 1 //Чтение значения из произвольного адреса VB //с помощью смещения:
 //1. Загрузить в указатель начальный адрес памяти переменных.
 //2. Прибавить к указателю величину смещения.
 //3. Скопировать значение из адреса в памяти переменных (смещение) в VB1900.

```

LD SM0.0
MOVW &VB0, LD10
+D VD1004, LD10
MOVW *LD10, VB1900
  
```


Пример программы для обращения к данным в таблице с использованием указателя

Этот пример использует LD14 как указатель на рецепт, хранящийся в таблице рецептов, которая начинается с VB100. В этом примере VW1008 хранит индекс места конкретного рецепта в таблице. Если каждый рецепт в таблице имеет длину 50 байтов, умножьте индекс на 50, чтобы получить смещение для начального адреса конкретного рецепта. Добавив смещение к указателю, вы можете получить доступ к каждому отдельному рецепту в таблице. В этом примере рецепт копируется в 50 байтов, которые начинаются с VB1500.



Сегмент 1 //Передача рецепта из таблицы с рецептами:
 // - каждый рецепт имеет длину 50 байтов.
 // - индексный параметр (VW1008) идентифицирует
 // рецепт, подлежащий загрузке.
 //
 //1. Создание указателя на начальный адрес таблицы
 // рецептов.
 //2. Преобразование индекса рецепта в значение
 // двойного слова.
 //3. Умножение смещения для учета длины рецепта.
 //4. Прибавление измененного смещения к
 // указателю.
 //5. Передача выбранного рецепта в ячейки с VB1500
 // по VB1549.

```
LD SM0.0
MOV &VB100, LD14
ITD VW1008, LD18
*D +50, LD18
+D LD18, LD14
BMB *LD14, VB1500, 50
```

Сохранение и извлечение данных с помощью S7–200

S7–200 предоставляет несколько методов, гарантирующих, что ваша программа и данные сохраняются в S7–200 надлежащим образом.

- Память сохраняемых (перманентных) данных – Области памяти данных, которые определяются пользователем и остаются неизменными при перерывах в подаче питающего напряжения, пока не разрядятся конденсатор большой емкости и необязательный батарейный модуль. Единственными областями в памяти данных, которые могут быть сконфигурированы как сохраняемые являются V и M, а также текущие значения таймеров и счетчиков.
- Постоянная память – Энергонезависимая память, используемая для хранения программного блока, блока данных, системного блока данных, принудительно присваиваемых значений, битов памяти, которые должны быть сохранены при потере питания, а также указанные значения, записываемые под управлением программы пользователя.
- Модуль памяти – Сменная энергонезависимая память, используемая для хранения программного блока, блока данных, системного блока данных, рецептов, протоколов данных и принудительно присваиваемых значений.

Для сохранения в модуле памяти файлов с документацией (*.doc, *.txt, *.pdf и т.д.) можно использовать проводник S7–200. С помощью проводника S7–200 можно также выполнять общее управление файлами в модуле памяти (копирование, удаление, открытие, создание каталогов).

Для установки модуля памяти снимите пластмассовую крышку с CPU S7–200 и вставьте модуль памяти в гнездо. Модуль памяти имеет такую форму, что он может быть вставлен в гнездо только надлежащим образом.

Осторожно

Электростатические разряды могут повредить модуль памяти или предназначенное для него гнездо в CPU S7–200.

При работе с модулем памяти необходимо стоять на хорошо проводящей заземленной площадке и/или носить заземленный браслет. Храните модуль в проводящем контейнере.

Загрузка компонентов проекта в CPU и из CPU

Ваш проект состоит из различных компонентов:

- программного блока
- блока данных (не обязателен)
- системного блока (не обязателен)
- рецептов (не обязательны)
- конфигураций протоколов данных (не обязательны)

При загрузке проекта программный блок, блок данных и системный блок данных для надежности сохраняются в постоянной памяти. Рецепты и конфигурации протоколов данных сохраняются в модуле памяти, заменяя при этом существующие рецепты и протоколы данных. Все элементы программы, не затронутые операцией загрузки, сохраняются неизменными в постоянной памяти и в модуле памяти.

Если при загрузке проекта загружаются также рецепты или конфигурации протоколов данных, то для надлежащего функционирования программы модуль памяти должен оставаться вставленным.

Для загрузки проекта в CPU S7–200 действуйте следующим образом:

1. Выберите команду меню **File > Download [Файл > Загрузить]**.
2. Щелкните на элементе проекта, который вы хотите загрузить.
3. Щелкните на кнопке **Download [Загрузить]**.

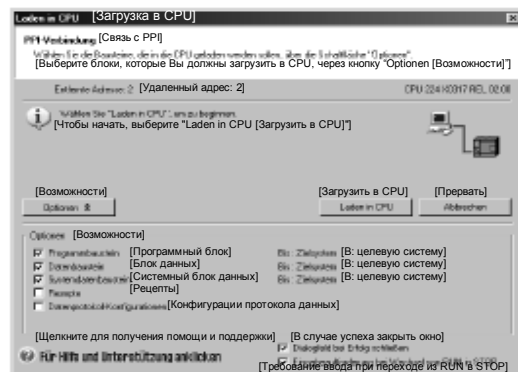


Рис. 4–13. Загрузка проекта в CPU S7–200

Когда вы загружаете проект из CPU в свой компьютер с помощью STEP 7-Micro/WIN, S7–200 загружает программный блок, блок данных и системный блок данных из постоянной памяти. Рецепты и конфигурации протоколов данных загружаются из модуля памяти. Данные из протоколов данных не загружаются в ваш компьютер с помощью STEP 7-Micro/WIN. Для загрузки данных из протоколов данных используется проводник S7–200 (см. главу 14).

Для загрузки вашего проекта из CPU S7–200 действуйте следующим образом:

1. Выберите команду меню **File > Upload [Файл > Загрузить из CPU]**.
2. Щелкните на каждом элементе проекта, который вы хотите загрузить.
3. Щелкните на кнопке **Upload [Загрузить из CPU]**.



Рис. 4–14. Загрузка проекта из CPU S7–200 в компьютер

Сохранение программы в модуле памяти

S7–200 дает возможность копировать программу пользователя из одного CPU в другой с помощью модуля памяти. Вы можете также распространять обновления для любого из следующих блоков в своем S7–200: программный блок, блок данных или системный блок данных.

Перед копирование элементов программы в модуль памяти STEP 7-Micro/WIN удаляет в модуле памяти все элементы программы (включая рецепты и протоколы данных), кроме файлов пользователя. Если ваша программа не помещается из-за размеров ваших файлов, то для создания достаточного места в памяти для хранения вашей программы вы можете сделать одну из двух вещей. Вы можете или очистить модуль памяти с помощью команды меню **PLC > Erase Memory Cartridge [ПЛК > Очистить модуль памяти]**. Или вы можете открыть проводник S7–200 и удалить не нужные более пользовательские файлы.

Для программирования модуля памяти ПЛК должен находиться в состоянии STOP.

Для сохранения программы в модуле памяти:

1. Выберите команду меню **PLC > Program Memory Cartridge [ПЛК > Программировать модуль памяти]**.
2. Щелкните на каждом элементе проекта, который вы хотите скопировать в модуль памяти (все элементы программы, имеющиеся в вашем проекте, выбираются по умолчанию). Если выбирается системный блок данных, то принудительно задаваемые значения тоже будут скопированы.
3. Щелкните на кнопке **Program [Программировать]**

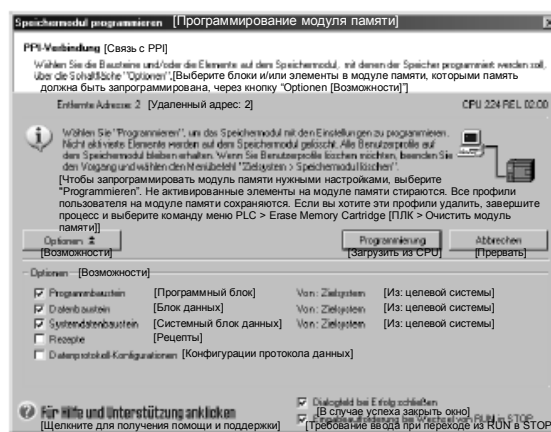


Рис. 4–15. Сохранение программы в модуле памяти

Программный блок, блок данных, системный блок данных и все принудительно устанавливаемые значения копируются из постоянной памяти S7–200 в модуль памяти. Рецепты и конфигурации протоколов данных копируются в модуль памяти из STEP 7-Micro/WIN.

Извлечение программы из модуля памяти

Для передачи программы из модуля памяти в S7-200 вы должны включить S7-с установленным модулем памяти. Если какие-либо блоки или принудительно установленные значения, находящиеся в модуле памяти, отличны от блоков или принудительно устанавливаемых значений в S7-200, то все блоки, находящиеся в модуле памяти, копируются в S7-200.

- Если из модуля памяти был передан программный блок, то программный блок в постоянной памяти заменяется.
- Если из модуля памяти был передан блок данных, то блок данных в постоянной памяти заменяется, вся память переменных стирается и инициализируется содержимым блока данных.
- Если из модуля памяти был передан системный блок данных, то системный блок данных и принудительно задаваемые значения в постоянной памяти заменяются и вся сохраняемая (реманентная) память стирается.

Как только передаваемая программа сохранена в постоянной памяти, вы можете удалить модуль памяти. Однако, если в модуле имеются рецепты или протоколы данных, то вы должны оставить модуль памяти установленным. Вставленные модуль памяти затягивает переход в режим RUN при следующем включении.

Примечание

Включение CPU S7-200 с установленным модулем памяти, запрограммированным в другой модели CPU S7-200, может вызвать ошибку. Модули памяти, запрограммированные в моделях CPU с меньшими номерами, могут читаться старшими моделями CPU. Противное, однако, неверно. Например, модули памяти, которые были запрограммированы в CPU 221 или CPU 222, могут быть прочитаны CPU 224, но модули памяти, запрограммированные в CPU 224, будут отвергнуты CPU 221 или CPU 222.

Подробный список ограничений при применении модулей памяти вы найдете в приложении А под заголовком "Дополнительные модули (модули памяти)".

Сохранение реманентной битовой памяти M при потере питания

Если первые 14 байтов битовой памяти (от M0 до M13) были определены при конфигурировании как реманентные (сохраняемые), то они сохраняются в постоянной памяти, когда S7-200 теряет питание. По умолчанию первые 14 байтов битовой памяти устанавливаются как не сохраняемые.

Извлечение данных после запуска

При запуске S7-200 восстанавливает программный блок и системный блок из постоянной памяти. Затем S7-200 проверяет конденсатор большой мощности и дополнительный батарейный модуль, если он установлен, относительно того, безошибочно ли производится буферизация данных в ОЗУ. Если эти данные были успешно буферизованы, то сохраняемые области в памяти пользователя остаются неизменными. Несохраняемые разделы памяти переменных восстанавливаются из соответствующего блока данных в постоянной памяти. Несохраняемые разделы других областей памяти стираются.

Если содержимое ОЗУ не удалось сохранить (например, после длительного перерыва в питании), S7-200 очищает все области данных пользователя, устанавливает специальный бит потери сохраняемых данных (SM0.2), извлекает память переменных из блока данных в постоянной памяти и восстанавливает первые 14 байтов битовой (M) памяти из постоянной памяти, если эти байты были ранее сконфигурированы как сохраняемые (реманентные).

Сохранение памяти переменных в постоянной памяти с помощью программы

Вы можете сохранить значение (байт, слово или двойное слово), находящееся в любом месте памяти переменных, в постоянной памяти. Операция сохранения в постоянной памяти обычно удлиняет время цикла не более чем на 5 мс. Значение, записанное операцией сохранения, заменяет предыдущее значение, хранящееся в области памяти переменных постоянной памяти.

Операция сохранения в постоянной памяти не обновляет данные в модуле памяти.



Совет

Так как число операций сохранения в постоянной памяти ограничено (минимум 100 000, обычно 1 000 000), вы должны обеспечить, чтобы сохранялись только необходимые значения. В противном случае постоянная память может изнашиваться, и CPU может выйти из строя. Обычно операции сохранения выполняются при возникновении определенных событий, которые встречаются относительно редко.

Например, если время обработки программы S7-200 составляет 50 мс, а значение сохранялось бы один раз за цикл, то ЭСППЗУ выдержало бы минимум 5 000 секунд, т.е. менее полутора часов. С другой стороны, если значение сохранялось бы один раз в час, то ЭСППЗУ прослужило бы минимум 11 лет.

Копирование V-памяти в постоянную память

Байт 31 специальной памяти (SMB31) дает S7-200 команду скопировать значение из V-памяти в область памяти переменных ЭСППЗУ. Слово 32 специальной памяти (SMW32) сохраняет адрес копируемой величины. На рис.4-16 показан формат SMB31 и SMW32.

Чтобы запрограммировать S7-200 на сохранение или запись определенного значения в V-памяти, выполните следующие шаги:

1. Загрузите адрес значения в V-памяти, которое вы хотите сохранить, в SMW32.
2. Загрузите длину данных в SM31.0 и SM31.1, как показано на рис. 4-16.
3. Установите SM31.7 в 1.

В конце каждого цикла выполнения программы S7-200 проверяет SM31.7; если SM31.7 равен 1, то указанное значение сохраняется в постоянной памяти. Операция завершается, когда S7-200 сбрасывает SM31.7 в 0.

Не изменяйте значение в V-памяти, пока операция сохранения не будет завершена.

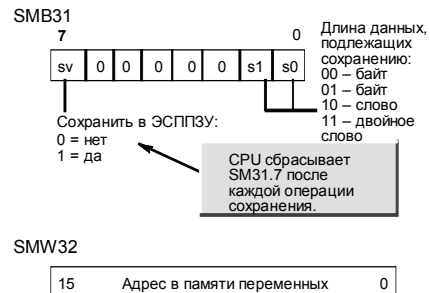
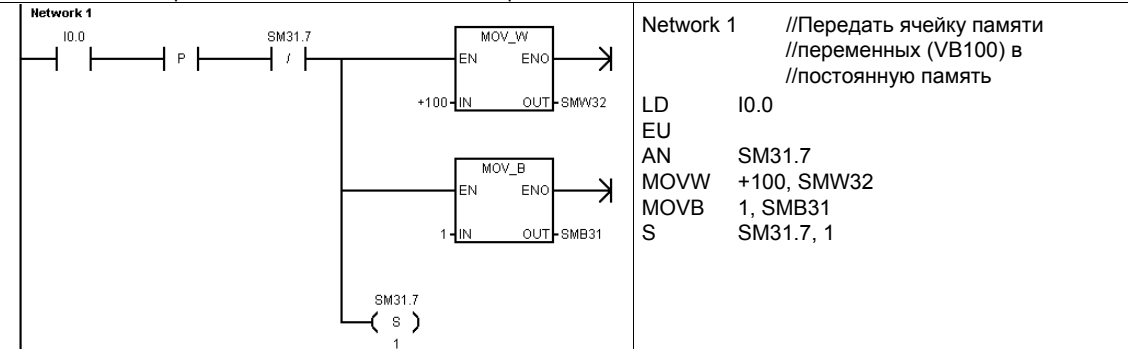


Рис. 4-16. SMB31 и SMW32

Пример программы: Копирование V-памяти в постоянную память

Этот пример передает VB100 в постоянную память. При нарастающем фронте на I0.0, если в это время не происходит другого переноса, происходит загрузка адреса места в памяти переменных, подлежащего передаче, в SMW32. Выбирается длина подлежащей передаче памяти переменных (1 = байт, 2 = слово, 3 = двойное слово или вещественное число). Затем устанавливается SM31.7, чтобы S7-200 передал данные в конце цикла.

По окончании передачи S7-200 автоматически сбрасывает SM31.7 в 0.



Установка режима работы CPU S7–200

S7–200 имеет два режима работы: STOP и RUN. Индикаторы состояния на передней панели CPU указывают на текущий режим работы. В состоянии STOP S7–200 не выполняет программы, и вы можете загрузить в CPU программу или конфигурацию CPU. В режиме RUN S7–200 исполняет программу.

- Для изменения режима работы S7–200 снабжен переключателем режимов. С помощью переключателя режимов (он находится под передней крышкой S7–200) вы можете установить режим работы вручную: установка переключателя режимов в STOP прекращает исполнение программы; установка переключателя режимов в RUN запускает исполнение программы, а установка переключателя режимов в режим TERM (терминал) не изменяет режима работы.

Если питание прерывается, когда переключатель режимов находится в положении STOP или TERM, S7–200 при восстановлении питания автоматически переходит в состояние STOP. Если питание прерывается, когда переключатель режимов находится в положении RUN, S7–200 при восстановлении питания переходит в режим RUN.

- STEP 7-Micro/WIN в режиме online дает возможность изменить режим работы S7–200. Чтобы это программное обеспечение могло управлять режимом работы, вы должны вручную перевести переключатель режимов работы на S7–200 в положение TERM или RUN. Для изменения режима работы вы можете использовать команды меню **PLC > STOP [ПЛК > STOP]** или **PLC > RUN [ПЛК > RUN]** или соответствующие кнопки на панели инструментов.
- Для перевода S7–200 в состояние STOP вы можете использовать в своей программе команду STOP. Это позволяет вам прекратить исполнение своей программы в зависимости от логики обработки программы. Подробную информацию о команде STOP вы найдете в главе 6.

Работа с проводником S7–200

Проводник S7–200 представляет собой расширение проводника Windows, предоставляющее доступ к ПЛК S7–200 и отображающее содержимое всех подключенных ПЛК. Могут быть определены различные блоки, которые могут находиться в ПЛК или в модуле памяти. Для каждого блока можно отобразить его свойства.

Так как проводник S7–200 является расширением проводника Windows, то поддерживаются обычный способ перемещения и поведение Windows.

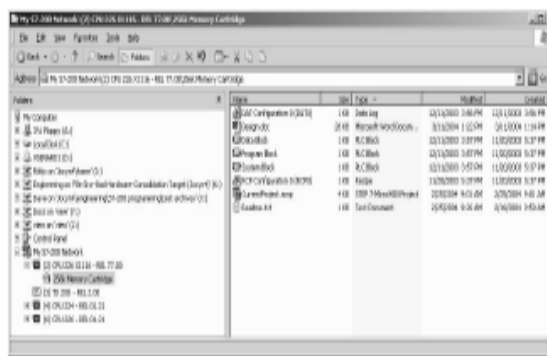


Рис. 4–17. Проводник S7–200

Проводник S7–200 – это механизм, используемый для чтения протоколов данных, хранящихся в модуле памяти. дополнительную информацию о протоколах данных вы найдете в главе 14.

Проводник S7–200 может также использоваться для чтения или записи файлов пользователя в модуль памяти. это могут быть файлы любых типов, документы Word, файлы битовых образов, файлы JPG или проекты STEP 7-Micro/WIN.

Функции S7–200

S7–200 предоставляет в распоряжение различные специальные функции, с помощью которых вы можете оптимально настроить S7–200 на свое приложение.

Программа S7–200 может непосредственно производить чтение и запись входов и выходов

Набор команд S7–200 содержит операции непосредственного чтения и записи физических входов/выходов. С помощью этих операций для прямого управления входами и выходами вы можете непосредственно обратиться к входу или выходу, хотя обычно источником или целью обращения к входам и выходам являются образы процесса.

При непосредственном обращении к входу соответствующая ячейка в регистре входов образа процесса не изменяется. При непосредственном обращении к выходу одновременно обновляется соответствующая ячейка в выходном регистре образа процесса.



Совет

S7–200 обрабатывает значения на аналоговых входах как непосредственные данные, если вы не активизировали фильтр на аналоговом входе. При записи значения на аналоговый выход, этот выход обновляется немедленно.

Обычно выгоднее работать с образами процесса и не обращаться во время обработки программы непосредственно к выходам и входам. Есть три существенных причины для использования образов процесса:

- В начале цикла система опрашивает входы. Благодаря этому значения этих входов на время обработки программы синхронизируются и замораживаются. Выходы обновляются после обработки программы через образ процесса. Это обеспечивает стабилизирующее воздействие на систему.
- Ваша программа может обратиться к образу процесса значительно быстрее, чем непосредственно к входам и выходам. Это ускоряет обработку программы.
- Входы и выходы являются битовыми объектами, к которым нужно обращаться в битовом или байтовом формате. Однако к образам процесса можно обращаться в формате бита, байта, слова или двойного слова. Поэтому образы процесса обеспечивают дополнительную гибкость.

Программа S7–200 может прерывать цикл

Если вы используете прерывания, то программы обработки прерываний, которые ставятся в соответствие прерывающим событиям, хранятся как часть основной программы. Однако они исполняются не как составная часть нормального цикла, а только тогда, когда происходит прерывающее событие (оно возможно в любом месте цикла).

Прерывания обслуживаются S7–200 в последовательности их появления с учетом соответствующих приоритетов. Подробную информацию о командах прерывания вы найдете в главе 6.

S7-200 позволяет выделить время для редактирования в режиме RUN и отображения состояния исполнения программы

Вы можете установить долю времени цикла (в процентах), предназначенную для обработки компиляции в режиме RUN или отображения состояния исполнения. (Редактирование в режиме RUN и отображение состояния исполнения – это возможности, предоставляемые STEP 7-Micro/WIN для облегчения отладки вашей программы.) Увеличивая долю времени для выполнения этих двух задач, вы увеличиваете время цикла, что делает протекание вашего процесса управления более медленным.

По умолчанию доля времени цикла, отводимая на обработку редактирования в режиме RUN и отображение состояния исполнения, составляет 10%. Эта установка является разумным компромиссом для обработки компиляций и состояния, минимизируя влияние на процесс управления. Вы можете настраивать это значение шагами по 5% максимум до 50%. Если вы хотите установить время для обмена данными в фоновом режиме, действуйте следующим образом:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Background Time [Фоновое время].
2. В закладке Background [Фон] выберите в выпадающем меню фоновое время для обмена данными.
3. Щелкните на ОК для подтверждения своего выбора.
4. Загрузите измененный системный блок данных в S7-200.

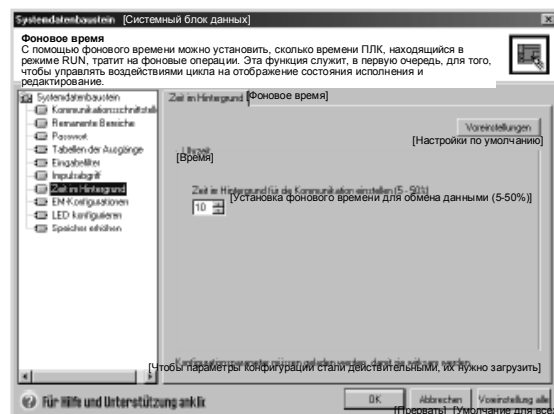


Рис. 4-18. Фоновое время для обмена данными

S7-200 дает возможность устанавливать состояния цифровых выходов в режиме STOP

С помощью таблицы выходов S7-200 вы можете установить сигнальные состояния цифровых выходов при переходе в режим STOP на определенные значения, или вы можете "заморозить" выходы точно в том состоянии, в котором они находились перед переходом в STOP. Таблица выходов – это часть системного блока данных, которая загружается и сохраняется в S7-200.

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Output Table [Таблица выходов]. Откройте закладку Digital [Цифровые].
2. Для замораживания выходов в их последнем состоянии активизируйте триггерную кнопку Freeze Outputs [Заморозить выходы].
3. Для копирования табличных значений в выходы введите эти значения в таблицу выходов, щелкая на триггерной кнопке для каждого выходного бита, который вы хотите установить в 1 после перехода из RUN в STOP. (По умолчанию все значения в таблице равны нулю.)
4. Подтвердите введенные значения, щелкнув на ОК.
5. Загрузите измененный системный блок данных в S7-200.

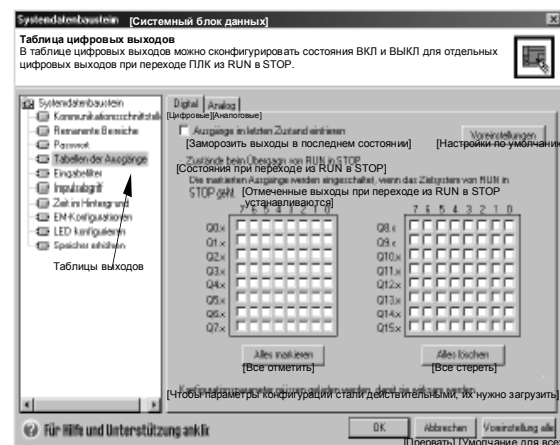


Рис. 4-19. Таблица цифровых выходов

S7-200 позволяет конфигурировать значения аналоговых выходов

В таблице аналоговых выходов вы можете установить аналоговые выходы на известные значения после перехода из RUN в STOP или сохранить значения выходов, существовавшие перед переходом в STOP. Таблица аналоговых выходов является частью системного блока данных, который загружается и сохраняется в CPU S7-200.

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Output Table [Таблица выходов]. Откройте закладку Analog [Аналоговые].
2. Для замораживания выходов в их последнем состоянии активизируйте триггерную кнопку Freeze Outputs [Заморозить выходы].
3. В таблице Freeze Values [Заморозить значения] вы можете установить аналоговые выходы на известное значение (от -32768 до 32762) после перехода из RUN в STOP.
4. Подтвердите введенные значения, щелкнув на ОК.
5. Загрузите измененный системный блок данных в S7-200.

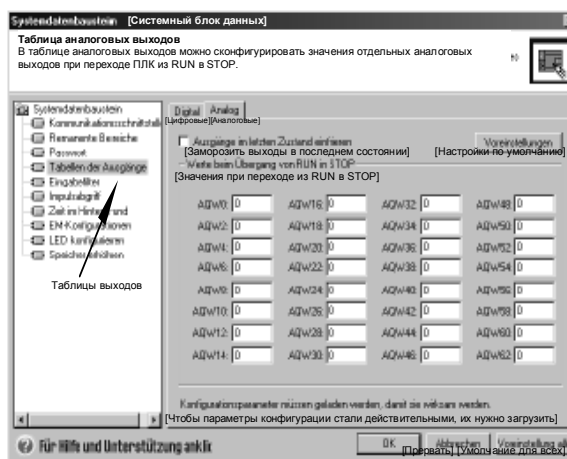


Рис. 4-20. Таблица аналоговых выходов

S7-200 позволяет определить память, которая сохраняется при потере питания

Вы можете определить в качестве сохраняемых до шести областей и выбрать области памяти, которые вы хотели бы буферизовать при потере питания. Вы можете определить диапазоны адресов, которые должны быть сохраняемыми, в следующих областях памяти: V, M, S и T. У таймеров могут быть буферизованы только сохраняемые таймеры (TONR). По умолчанию первые 14 байтов битовой (M) памяти не сохраняются.

У таймеров и счетчиков могут быть буферизованы только текущие значения: биты таймеров и счетчиков не сохраняются.



Совет

Если вы определите диапазон от M0 до M13 в качестве сохраняемого, то активизируется специальная функция, которая при потере питания автоматически сохраняет эти ячейки памяти в постоянной памяти.

Для определения сохраняемой памяти:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Retentive Ranges [Сохраняемые области].
2. Выберите области в памяти, которые должны быть буферизованы при потере питания, и щелкните на ОК.
3. Загрузите измененный системный блок в S7-200.

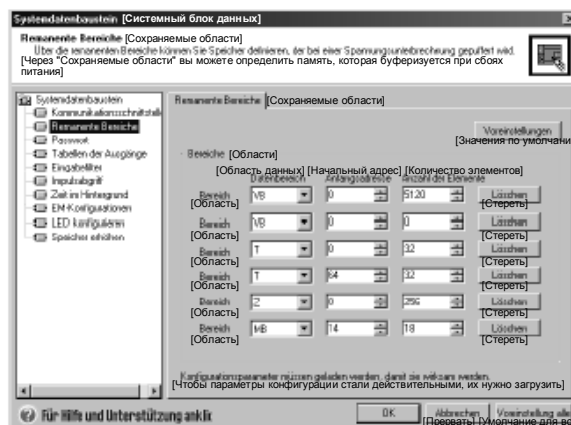


Рис. 4-21. Сохраняемая память

S7-200 дает возможность фильтровать цифровые входы

S7-200 позволяет выбрать входной фильтр, который определяет время задержки (выбираемое в пределах от 0,2 мс до 12,8 мс) для всех или некоторых встроенных цифровых входов. Эта задержка помогает отфильтровать шум во входной проводке, который может вызвать непреднамеренные изменения состояний входов.

Входной фильтр является частью системного блока данных, который загружается и хранится в S7-200. По умолчанию время фильтра равно 6,4 мс. Как показано на рис. 4-22, каждая данная задержка действительна для группы входов.

Для конфигурирования времен задержки для входного фильтра:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Input Filters [Входные фильтры]. Щелкните на закладке Digital [Цифровые].
2. Введите величину задержки для каждой группы входов и щелкните на OK.
3. Загрузите измененный системный блок в S7-200.

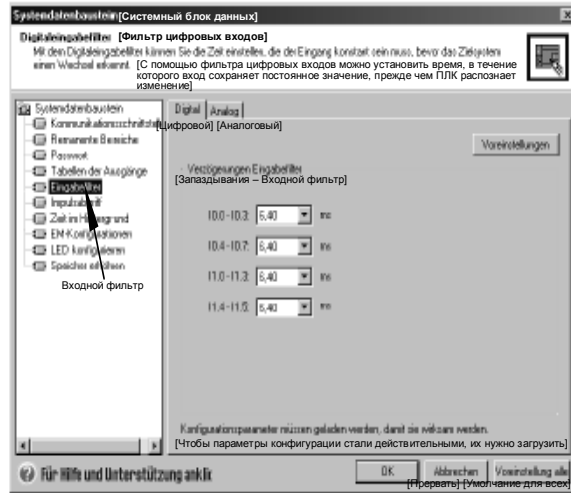


Рис. 4-22. Фильтр цифровых входов

Совет

Фильтр цифровых входов оказывает также влияние на входную величину с точки зрения команд чтения, прерываний по входам и регистраторов импульсов. В зависимости от настройки фильтра это может привести к тому, что ваша программа может пропустить прерывающее событие или импульс. Скоростные счетчики подсчитывают события на входах без фильтров.

S7-200 дает возможность фильтровать аналоговые входы

У S7-200 вы можете установить программный фильтр для отдельных аналоговых входов. Отфильтрованное значение является средним значением заранее установленного количества опросов аналоговых входов. Параметры фильтра (количество опросов и зона нечувствительности) одинаковы для всех аналоговых входов, для которых фильтр активизирован.

Фильтр обладает свойством быстрой реакции, что обеспечивает быстрое воздействие больших изменений на значение фильтра. Фильтр обеспечивает реакцию на последнее значение на аналоговом входе, как на ступенчатое воздействие, если изменение на этом входе по сравнению с текущим значением превышает определенную величину. Это изменение, называемое зоной нечувствительности, задается в отсчетах цифрового значения аналогового входа.

По умолчанию фильтр активизирован для всех аналоговых входов кроме AIW0 и AIW2 на CPU 224XP.

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Input Filters [Входные фильтры]. Щелкните на закладке Analog [Аналоговые].
2. Выберите аналоговые входы, которые вы хотите фильтровать, количество опросов и зону нечувствительности.
3. Щелкните на OK.
4. Загрузите измененный системный блок в S7-200.

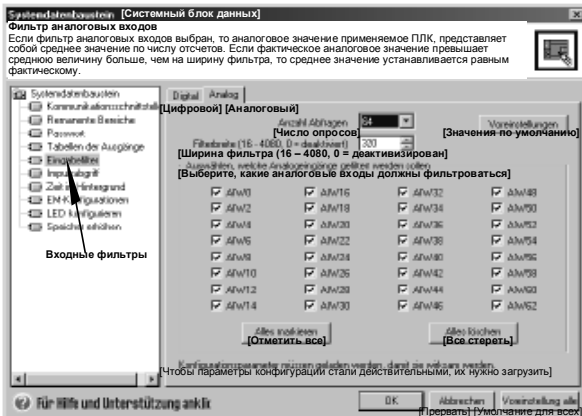


Рис. 4-23. Фильтр аналоговых входов

Совет
 Не используйте аналоговый фильтр с модулями, которые передают цифровую информацию или аварийные сигналы в аналоговых словах. Всегда выключайте аналоговый фильтр для модулей с RTD, термопарой и главных модулей AS-интерфейса.

Совет
 AIW0 и AIW2 на CPU 224XP фильтруются аналого-цифровым преобразователем и обычно не нуждаются в дополнительном программном фильтре.

S7-200 дает возможность регистрировать короткие импульсы

S7-200 имеет в своем распоряжении функцию "Регистратор импульсов", которая может быть использована для всех или некоторых встроенных цифровых входов. Функция "Регистратор импульсов" дает возможность регистрировать импульсы большой или малой амплитуды, имеющие столь малую продолжительность, что они легко могут быть пропущены модулем S7-200, который считывает цифровые входы в начале цикла. Если функция "Регистратор импульсов" активизирована для некоторого входа, то изменение сигнала на этом входе фиксируется и удерживается, пока не произойдет обновление данных в следующем цикле. Это гарантирует, что импульс, длящийся очень короткий интервал времени, будет зарегистрирован и сохранен, пока S7-200 не прочтет входы.

Функцию "Регистратор импульсов" можно активизировать для любого встроенного цифрового входа.

Чтобы вызвать диалоговое окно для конфигурирования регистрации импульсов:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите Pulse Catch Bits [Биты регистратора импульсов].
2. Активизируйте желаемую триггерную кнопку и щелкните на ОК.
3. Загрузите измененный системный блок в S7-200.

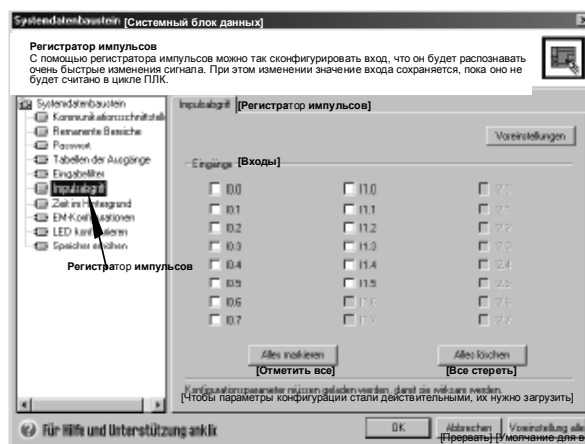


Рис. 4-24. Регистратор импульсов

На рис. 4-25 показан принцип действия S7-200 с активизированным и деактивизированным регистратором импульсов.



Рис. 4-25. Функционирование S7-200 с активизированным и деактивизированным регистратором импульсов

Так как функция регистрации импульсов работает на входе после того, как сигнал прошел через входной фильтр, вы должны так настроить время входного фильтра, чтобы импульс не был удален фильтром. На рис. 4–26 дано схематическое представление цепи цифрового входа.

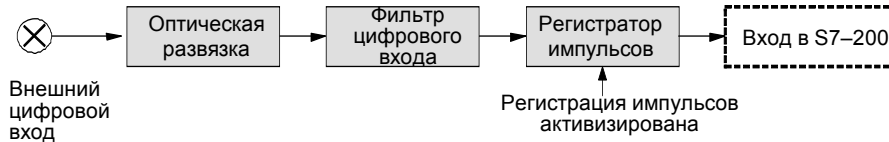


Рис. 4–26. Цепь цифрового входа

На рис. 4–27 показана реакция активизированного захвата импульсов на различные входные условия. Если в данном цикле имеется более одного импульса, то регистрируется только первый из них. При нескольких импульсах в одном цикле вам следует использовать прерывающие события для нарастающего и падающего фронтов. (Перечисление прерывающих событий вы найдете в таблице 6–46.)

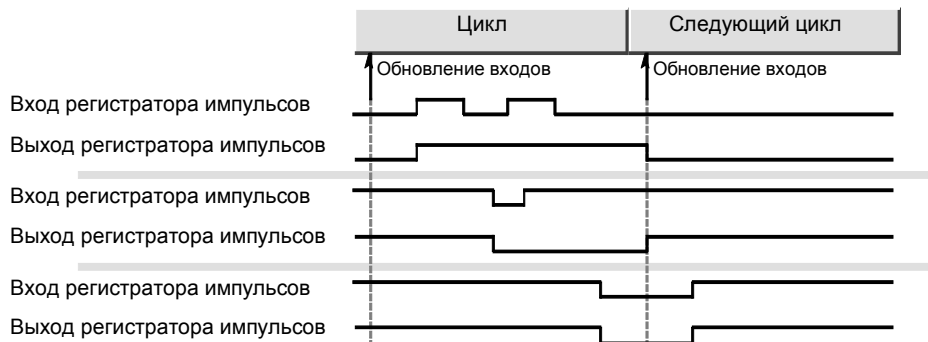


Рис. 4–27. Реакции регистратора импульсов на различные входные условия

У S7–200 имеются светодиод, которым может управлять пользователь

У S7–200 имеются светодиод (SF/DIAG), который может гореть красным (светодиод системной ошибки) или желтым (диагностический светодиод) светом. Диагностический светодиод может загораться под управлением программы или, при определенных условиях, загораться автоматически: когда вход или выход или значение данных устанавливается принудительно, или когда у модуля имеет место ошибка ввода-вывода.

Для конфигурирования автоматических настроек диагностического светодиода действуйте следующим образом:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и выберите **Configure LED [Конфигурировать светодиод]**.
2. Выберите для каждой записи, должен ли светодиод включаться или нет, когда значение для входа или выхода или элемента данных устанавливается принудительно, или когда на модуле произошла ошибка ввода-вывода.
3. Загрузите измененный системный блок в S7–200.

Для управления состоянием диагностического с помощью своей пользовательской программы используйте команду "Диагностический светодиод", описанную в главе 6.

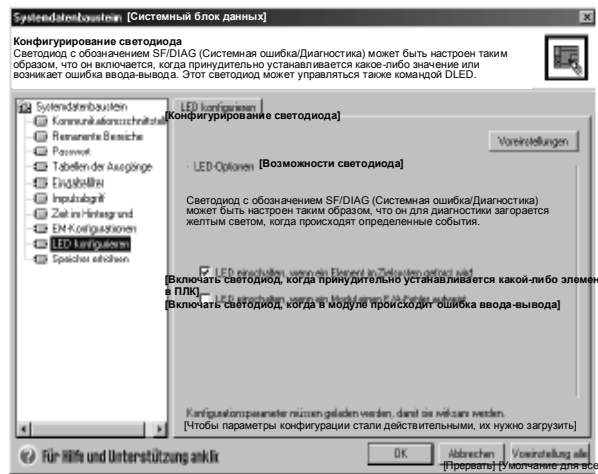


Рис. 4–28. Диагностический светодиод

S7-200 поддерживает протокол существенных событий CPU

S7-200 поддерживает протокол, содержащий историю существенных событий CPU с метками времени, например, когда включается напряжение, когда CPU переходит в режим RUN и когда происходят фатальные ошибки. Чтобы метка даты и времени для записей в протоколе была действительна, должны быть установлены часы реального времени. Для просмотра протокола выберите команду меню **PLC > Information [ПЛК > Информация]** и выберите Event History [История событий].

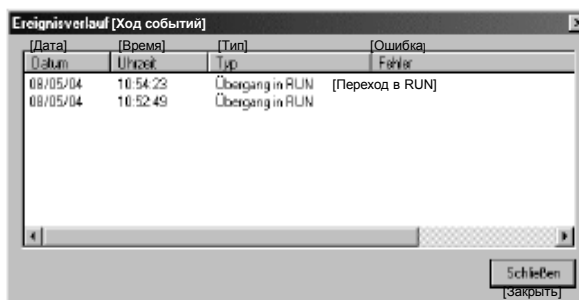


Рис. 4-29. Просмотр протокола истории событий

S7-200 позволяет увеличить доступную память для программы пользователя

S7-200 позволяет заблокировать свойство редактирования в режиме RUN в CPU 224, CPU 224XP и CPU 226, чтобы увеличить размер памяти программ, доступной для вашего использования. Размер памяти программ для каждой модели CPU вы найдете в таблице 1-2.

Чтобы заблокировать функцию редактирования в режиме RUN, действуйте следующим образом

1. Выберите команду меню **View > System Block [Вид > Системный блок]** и выберите Increase Program Memory [Увеличить память программ].
2. Выберите опцию Increase Memory [Увеличить память], чтобы заблокировать функцию редактирования в режиме RUN.
3. Загрузите измененный системный блок данных в S7-200.

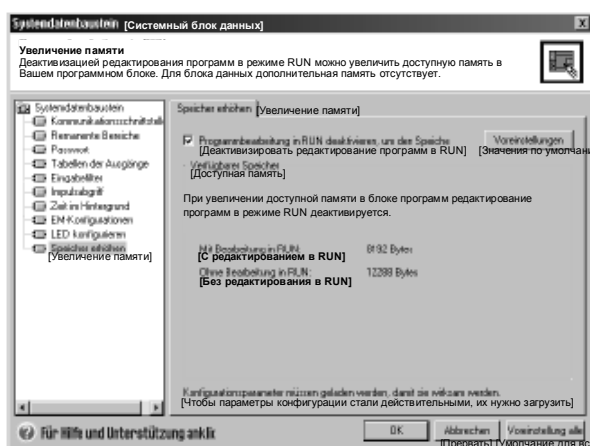


Рис. 4-30. Блокирование функции редактирования в режиме RUN

S7-200 предоставляет защиту с помощью пароля

Все модели S7-200 предоставляют защиту с помощью пароля для ограничения доступа к определенным функциям.


Благодаря паролю доступ к определенным функциям и памяти имеют только уполномоченные лица: без пароля возможен неограниченный доступ к S7-200. При наличии парольной защиты S7-200 ограничивает доступ к функциям в соответствии с конфигурацией пароля. Пароль не чувствителен к регистру символов.

Как показано в таблице 4-3, S7-200 предоставляет три уровня ограничения доступа. Каждый уровень предоставляет неограниченный доступ к определенным функциям без ввода пароля. Для всех трех уровней ввод правильного пароля предоставляет доступ ко всем функциям.

Таблица 4-3. Ограничение доступа к S7-200

Функция CPU	Уровень 1	Уровень 2	Уровень 3
Чтение и запись данных пользователя	Доступ разрешен	Доступ разрешен	Доступ разрешен
Запуск, останов и перезапуск CPU			
Чтение и установка часов реального времени			
Загрузка программы пользователя, данных и конфигурации CPU из CPU	Доступ разрешен	Доступ разрешен	Требуется пароль
Загрузка в CPU	Доступ разрешен	Требуется пароль	
Получение состояния выполнения			
Удаление программного блока, блока данных и системного блока			
Принудительное задание данных или исполнение одного или нескольких циклов			
Копирование в модуль памяти			
Запись в выходы в состоянии STOP			

По умолчанию для S7-200 установлен уровень 1 (без ограничений). Ввод пароля через сеть не оказывает влияния на парольную защиту S7-200. Если один пользователь имеет право доступа к защищенным функциям, то другие пользователи не имеют права доступа к этим функциям. В каждый данный момент времени неограниченный доступ к S7-200 имеет только один пользователь.



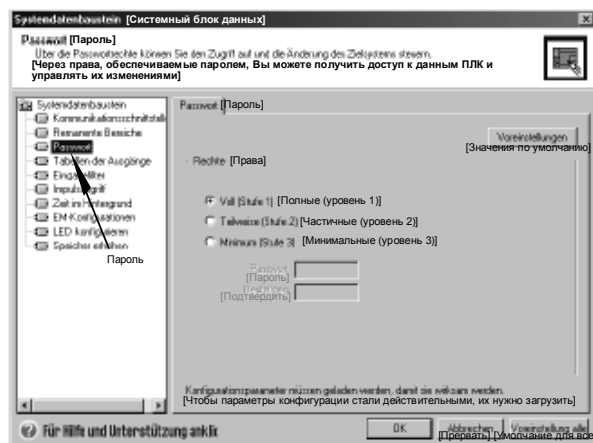
Совет

После того как вы ввели пароль, уровень защиты для этого пароля остается действительным в течение максимум одной минуты после отсоединения устройства программирования от S7-200. Всегда выходите из STEP 7-Micro/WIN перед отсоединением кабеля, чтобы другой пользователь не мог получить доступа к привилегиям этого устройства программирования.

Установка пароля для S7-200

Диалоговое окно System Block [Системный блок] (рис. 4-31) позволяет установить пароль для S7-200. По умолчанию для S7-200 установлен уровень 1 (полный доступ без ограничений).

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** для отображения диалогового окна System Block [Системный блок] и выберите Password [Пароль].
2. Выберите желаемый уровень доступа для S7-200.
3. Введите и подтвердите пароль для частичного (уровень 2) или минимального (уровень 3) доступа.
4. Щелкните на ОК.
5. Загрузите измененный системный блок в S7-200.



Последовательность действий при утере пароля

Если вы забыли пароль, то вы должны очистить память S7-200 и перезагрузить свою программу. Очистка памяти переводит S7-200 в режим STOP и восстанавливает в S7-200 значения заводской настройки, за исключением сетевого адреса, скорости передачи и часов реального времени. Для стирания программы S7-200:

1. Выберите команду меню **PLC > Clear [ПЛК → Очистить]**, чтобы отобразить диалоговое окно Clear [Очистка].
2. Выделите все три блока и подтвердите ваше действие щелчком на кнопке ОК.
3. Если пароль был создан, то STEP 7-Micro/WIN отображает диалоговое окно, в котором запрашивается пароль доступа. Для стирания пароля введите в этом диалоговом окне CLEARPLC, чтобы продолжить операцию общего стирания (Clear All). (Пароль CLEARPLC не чувствителен к регистру шрифта.)

При общем стирании программа в модуле памяти сохраняется. Так как модуль памяти наряду с программой хранит пароль, вы должны перепрограммировать также модуль памяти, чтобы удалить потерянный пароль.

Предупреждение

Очистка памяти S7-200 вызывает выключение выходов (или “замораживание” на определенном уровне в случае аналогового выхода).

Если во время очистки памяти S7-200 соединен с оборудованием, то изменения состояний выходов могут передаваться этому оборудованию. Если вы конфигурировали для выходов “безопасное состояние”, отличающееся от заводской настройки, то изменения выходов могут вызвать непредсказуемую реакцию вашего оборудования, которая может также вызвать гибель или тяжкие телесные повреждения персонала и/или повреждение оборудования.

Всегда соблюдайте соответствующие меры безопасности и перед очисткой памяти S7-200 обеспечьте, что ваш процесс находится в безопасном состоянии.

S7-200 имеет в своем распоряжении аналоговые потенциометры

Аналоговые потенциометры для настройки находятся под передней крышкой модуля. Вы можете настраивать эти потенциометры для увеличения или уменьшения значений, хранящихся в байтах в специальной памяти (SMB). Эти защищенные от записи величины могут использоваться программой для реализации ряда функций, например, актуализация текущего значения таймера или счетчика, ввод или изменение предустановленных значений или установка граничных значений. Для настройки нужна маленькая отвертка: поверните потенциометр по часовой стрелке (направо) для увеличения значения и против часовой стрелки (налево) для уменьшения значения.

SMB28 хранит цифровое значение, представляющее настройку аналогового потенциометра 0. SMB29 хранит цифровое значение, представляющее настройку аналогового потенциометра 1. Аналоговый потенциометр имеет номинальный диапазон от 0 до 255 и повторяемость ± 2 отсчета.

Пример программы обращения к величине, введенной с помощью аналогового потенциометра	
	<pre> Network 1 //Прочитать аналоговый потенциометр 0 //(SMB28). //Сохранить значение как целое в слове //VW100. LD I0.0 BTI SMB28, VW100 Network 2 //Использовать целое значение (VW100) в //качестве уставки для таймера. LDN Q0.0 TON T33, VW100 Network 3 //Включить Q0.0, когда T33 достигнет величины //уставки. LD T33 = Q0.0 </pre>

У S7-200 имеются скоростные входы и выходы

Скоростные счетчики

S7-200 предоставляют в распоряжение встроенные скоростные счетчики, которые считают быстро протекающие внешние события без ухудшения производительности S7-200. Скорости, поддерживаемые вашей моделью CPU, вы найдете в Приложении А. У каждого счетчика имеются входы, предназначенные для синхронизации, управления направлением, сброса и запуска, где эти функции поддерживаются. Вы можете варьировать скорость счета установкой различных А/В-счетчиков. За дополнительной информацией об использовании скоростных счетчиков обратитесь к главе 6.

Скоростные импульсные выходы

S7-200 поддерживает скоростные импульсные выходы, причем выходы Q0.0 и Q0.1 могут генерировать последовательности скоростных импульсов (РТО) или выполнять управление с помощью широтно-импульсной модуляции (PWM).

Функция «Последовательность скоростных импульсов» дает выход в виде прямоугольных импульсов (с относительной длительностью 50 %) для заданного количества импульсов (от 1 до 4 294 967 295) и заданного времени цикла (микро- или миллисекундными шагами). Функция «Последовательность скоростных импульсов» (РТО) может быть запрограммирована так, чтобы реализовать одну последовательность импульсов или конфигурацию, состоящую из нескольких последовательностей импульсов. Например, для управления шаговым двигателем вы можете использовать конфигурацию импульсов, состоящую из линейно нарастающего участка, рабочего участка и линейно убывающего участка, или более сложные последовательности.

Функция «Широтно-импульсная модуляция» обеспечивает фиксированное время цикла с переменной относительной длительностью импульсов, причем время цикла и ширина импульсов задаются микро- или миллисекундными шагами. Когда ширина импульса равна времени цикла, относительная длительность импульсов равна 100 процентам, и выход включен постоянно. Когда ширина импульсов равна нулю, относительная длительность импульсов равна 0 процентов, и выход выключен.

За дополнительной информацией о скоростных импульсных выходах обратитесь к главе 6. За дополнительной информацией об использовании РТО в управлении перемещением без обратной связи обратитесь к главе 9.

Основы программирования, соглашения и функции

5

S7–200 последовательно обрабатывает вашу программу для управления задачей или процессом. Программа создается с помощью STEP 7-Micro/WIN, а затем загружается в S7–200. STEP 7-Micro/WIN предоставляет в ваше распоряжение различные инструментальные средства и функции для разработки, реализации и тестирования вашей программы.

В этой главе

Рекомендации по проектированию системы автоматизации с микроконтроллером	52
Основные элементы программы	53
Использование STEP 7-Micro/WIN для создания вашей программы	55
Наборы команд SIMATIC и IEC 1131-3	57
Соглашения, используемые в редакторах программ	58
Создание программы управления с помощью мастеров	60
Устранение ошибок в S7–200	60
Назначение адресов и начальных значений в редакторе блоков данных	62
Использование таблицы символов для символической адресации переменных	63
Использование локальных переменных	64
Контроль над программой с помощью таблицы состояний	64
Создание библиотеки команд	65
Функции тестирования программы	65

Рекомендации по проектированию системы автоматизации с микроконтроллером

Есть много методов проектирования систем с микроконтроллером. Следующие общие рекомендации применимы ко многим проектам. Конечно, вы должны следовать предписанным процедурам вашей собственной компании и учитывать свой собственный опыт.

Расчленение процесса или машины

Разделите ваш процесс или машину на участки, не зависящие друг от друга. Эти участки определяют границы между несколькими системами автоматизации и влияют на описания функциональных областей и назначение ресурсов.

Описание функциональных областей

Запишите описания работы каждого участка процесса или машины. Включите следующие пункты: входы и выходы, описание функционирования, состояния, которые должны достигаться, перед тем как станет возможным управление исполнительными механизмами (например, выключателями с соленоидным приводом, двигателями и приводами), описание интерфейса оператора и интерфейсов с другими участками процесса или машинами.

Проектирование схем защиты

Определите оборудование, требующее для обеспечения безопасности аппаратно реализованной логики. Устройства управления могут выходить из строя опасным образом, вызывая неожиданный запуск или изменение в работе машинного оборудования. Там, где неожиданная или неправильная работа машинного оборудования может привести к физической травме людей или значительному материальному ущербу, нужно уделить внимание использованию электромеханических блокировок, которые работают независимо от S7-200, чтобы предотвратить опасные операции. В проектирование схем защиты должны включаться следующие задачи:

- Выявление ненадлежащей или неожиданной работы исполнительных механизмов, которая может оказаться опасной.
- Определение состояний, которые гарантировали бы, что работа не опасна, и выяснение того, как обнаруживать эти состояния независимо от S7-200.
- Определение влияния CPU S7-200 и входов/выходов на процесс при подаче и выключении питания и обнаружении ошибок. Эта информация должна использоваться только для проектирования нормального и ожидаемого аварийного режимов работы и не должна использоваться для целей безопасности.
- Проектирование ручных или электромеханических блокировок, которые блокируют опасную операцию независимо от S7-200.
- Предоставление в S7-200 надлежащей информации о состоянии от независимых цепей тока, чтобы программа и любые интерфейсы оператора имели необходимую информацию.
- Определение любых других связанных с безопасностью требований для безопасного протекания процесса.

Определение станций оператора

Основываясь на требованиях из описаний функциональных областей, разработайте чертежи станций оператора. Включите следующие пункты:

- Обзор, показывающий местоположение каждой станции оператора относительно процесса или машины
- Механическая компоновка устройств станции оператора, например, дисплеев, переключателей и ламп
- Электрические чертежи CPU S7-200 или модулей расширения с соответствующими входами-выходами

Разработка чертежей конфигурации

Основываясь на требованиях из описаний функциональных областей, разработайте чертежи конфигурации аппаратуры управления. Включите следующие пункты:

- Обзор, показывающий местоположение каждого S7-200 относительно процесса или машины
- Механическая компоновка S7-200 и модулей расширения входов-выходов (включая шкафы и другое оборудование)
- Электрические чертежи для каждого S7-200 и модуля расширения входов-выходов (включая номера моделей устройств, коммуникационные адреса и адреса входов-выходов)

Создание списка символических имен (не обязателен)

Если вы выбираете для адресации использование символических имен, то составьте список символических имен для абсолютных адресов. Включите не только сигналы физических входов-выходов, но также и другие элементы, которые нужно использовать в вашей программе.

Основные элементы программы

Программный блок состоит из исполняемого кода и комментариев. Исполняемый код состоит из основной программы (OB1), а также подпрограмм и программ обработки прерываний. Код компилируется и загружается в S7-200. Комментарии не компилируются и не загружаются. С помощью этих организационных элементов (основной программы, подпрограмм и программ обработки прерываний) вы можете структурировать свою управляющую программу.

В следующем примере показана программа, включающая в себя подпрограмму и программу обработки прерываний. Эта программа-пример с помощью прерывания, управляемого временем, считывает значение с аналогового входа каждые 100 мс.

Пример: Основные элементы программы		
M A I N		<pre> Network 1 // В 1-ом цикле вызвать // подпрограмму 0. LD SM0.1 CALL SBR_0 </pre>
S B R 0		<pre> Network 1 // Установить для прерывания, // управляемого временем, // интервал 100 мс. // Разблокировать прерывание 0. LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI </pre>
I N T 0		<pre> Network 1 // Опросить аналоговый вход 4. LD SM0.0 MOVW AIW4, VW100 </pre>

Основная программа

Эта основная часть программы содержит команды, управляющие вашим приложением. S7-200 выполняет эти команды последовательно и однократно в каждом цикле. Основная программа называется также OB1.

Подпрограммы

Эти необязательные элементы программы выполняются только тогда, когда они вызываются: основной программой, программой обработки прерываний или другой подпрограммой. Подпрограммы полезны, если вы хотите какую-нибудь функцию выполнять многократно. Чтобы не переписывать логику в каждом месте основной программы, где вы хотите выполнить эту функцию, вы можете записать логику функции один раз в подпрограмме, а затем вызывать эту подпрограмму столько раз, сколько необходимо при выполнении основной программы. Подпрограммы имеют много преимуществ:

- Использование подпрограмм уменьшает общую величину программы.
- Использование подпрограмм уменьшает время цикла, так как вы удалили соответствующий код из основной программы. S7-200 в каждом цикле анализирует код в основной программе независимо от того, исполняется этот код или нет, но код в подпрограмме анализируется только тогда, когда вы вызываете подпрограмму, и не анализируется в циклах, в которых подпрограмма не вызывается.
- С помощью подпрограмм создается мобильный код. Вы можете отграничить код для функции в подпрограмме, а затем копировать эту подпрограмму в другие программы без больших затрат.



Совет

Использование адресов памяти переменных может ограничить мобильность подпрограмм, так как назначение адресов в памяти переменных одной программы может привести к конфликту с назначением адресов в другой программе. Подпрограммы, которые используют для назначения всех адресов таблицу локальных переменных (локальные данные), напротив, хорошо переносятся, так как при применении локальных переменных невозможен конфликт адресов между подпрограммой и другими частями программы.

Программы обработки прерываний

Эти необязательные элементы программы реагируют на определенные прерывающие события. Программа обработки прерываний проектируется для обработки заранее определенных прерывающих событий. S7-200 исполняет программу обработки прерываний, когда возникает соответствующее событие.

Программы обработки прерываний не вызываются основной программой. Вы ставите программу обработки прерываний в соответствие прерывающему событию. S7-200 выполняет команды, находящиеся в программе обработки прерываний, только при возникновении прерывающего события.



Совет

Так как невозможно предсказать, когда S7-200 сгенерирует прерывание, то желательно ограничить количество переменных, используемых как в программе обработки прерываний, так и в других местах программы.

Используйте таблицу локальных переменных программы обработки прерываний, чтобы гарантировать, что ваша программа обработки прерываний будет использовать только временную память и не заменит данные, используемые где-нибудь еще в вашей программе.

Существует ряд методов программирования, которые гарантированно позволяют избежать ошибок при совместном использовании данных в основной программе и в программе обработки прерываний. Эти методы описаны в главе 6 вместе с командами прерываний.

Другие элементы программы

Другие блоки содержат информацию для S7-200. Вы можете загрузить эти блоки одновременно с загрузкой своей программы.



Системный блок

В системном блоке данных вы можете конфигурировать различные аппаратные возможности для S7-200.



Блок данных

Блок данных хранит значения для различных переменных (память переменных), используемых вашей программой. В блок данных можно вводить начальные значения для данных.

Использование STEP 7-Micro/WIN для создания вашей программы

Для открытия STEP 7-Micro/WIN дважды щелкните на символе STEP 7-Micro/WIN или выберите команду меню **Start > SIMATIC > STEP 7 MicroWIN 32 V4.0 [Пуск > SIMATIC > STEP 7 MicroWIN 32 V4.0]**. Как показано на рис. 5–1, окно проекта в STEP 7-Micro/WIN предоставляет удобную рабочую область для создания программы управления.

На панелях инструментов имеются кнопки для часто используемых команд меню. Вы можете эти панели инструментов по отдельности показывать или скрывать.

Навигационная панель предлагает группы символов для доступа к различным функциям программирования STEP 7-Micro/WIN.

Дерево команд отображает все объекты проекта и команды, необходимые для создания программы управления. Вы можете буксировать отдельные команды из этого дерева в свою программу или вставлять команду двойным щелчком в текущее положение курсора в редакторе программ.

Редактор программ содержит логику программы и таблицу локальных переменных, в которой вы можете назначить символические имена для временных локальных переменных.

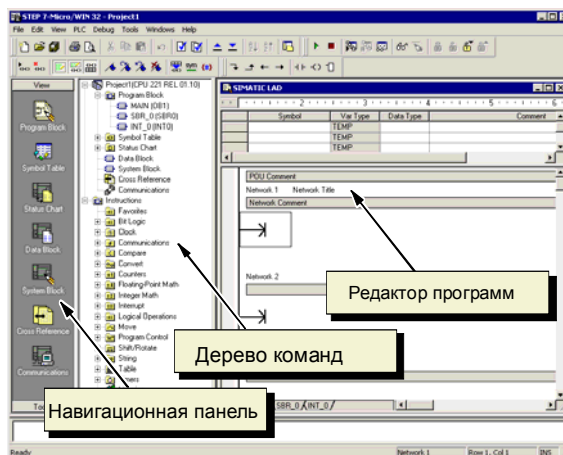


Рис. 5–1. STEP 7-Micro/WIN

Подпрограммы и программы обработки прерываний появляются как закладки в нижней части окна редактора программ. Для перемещения между подпрограммами, программами обработки прерываний и основной программой щелкайте по этим закладкам.



Редактор программ

STEP 7-Micro/WIN имеет в своем распоряжении три редактора для создания программ: цепная логическая схема (LAD), называемая также контактным планом (КОП), список команд (STL или AWL) и функциональная блок-схема (FBD), называемая также функциональным планом (FUP). С некоторыми ограничениями, программы, написанные в любом из этих редакторов программ, могут отображаться и редактироваться с помощью других редакторов программ.

Функции редактора STL

Редактор STL отображает программу на языке, имеющем текстовую основу. Редактор STL дает возможность создавать программы управления, вводя мнемонические обозначения команд. В редакторе STL можно создавать программы, которые невозможно создать в редакторе LAD или FBD. Это объясняется тем, что, используя STL, вы программируете на "родном языке" S7–200, а не в графическом редакторе, в котором имеются некоторые ограничения, чтобы можно было правильно изображать схемы соединений. Как показано на рис. 5–2, программирование в текстовом редакторе очень похоже на программирование на языках ассемблера.

S7–200 выполняет команды в порядке, определяемом программой, сверху вниз, а затем начинает сначала.

```
LD I0.0 //Прочитать вход
A I0.1 //Выполнить логическое И-
//сопряжение с другим входом
= Q1.0 //Записать значение на выход 1
```

Рис. 5–2. Пример программы на STL

В STL логика управления реализуется с помощью логического стека. В STL вы должны вводить команды для обработки стековых операций.

При выборе редактора STL примите во внимание следующее:

- STL лучше всего подходит опытным программистам.
- STL иногда позволяет решать проблемы, которые вы не можете достаточно легко решить при помощи редактора LAD или FBD.
- Вы можете использовать редактор STL только с системой команд SIMATIC.
- Тогда как вы всегда можете использовать редактор STL для просмотра или редактирования программы, созданной с помощью редактора LAD или FBD, обратное не всегда возможно. Вы не всегда можете использовать редактор LAD или FBD для отображения программы, написанной при помощи редактора STL.

Функции редактора LAD

Редактор цепных логических схем LAD отображает программу в виде графического представления, имеющего сходство с электрической монтажной схемой. Цепные логические схемы позволяют программе имитировать протекание электрического тока от источника напряжения через ряд логических условий на входах, которые, в свою очередь, активизируют логические условия на выходах. LAD-программа включает в себя находящуюся слева шину, находящуюся под напряжением, которая является источником потока сигнала. Замкнутые контакты позволяют потоку сигнала протекать через эти контакты к следующему элементу, а разомкнутые контакты препятствуют протеканию потока сигнала.

Логика подразделяется на сегменты. Программа выполняется сегмент за сегментом слева направо и сверху вниз. На рис. 5–3 показан пример программы в виде цепной логической схемы. Различные команды представляются графическими символами, имеющими три основные формы.

Контакты представляют логические состояния входов, например, выключателей, кнопок или внутренних условий.

Катушки обычно представляют логические результаты выходов, например, ламп, пускателей электродвигателей, промежуточных реле или внутренних выходных условий.

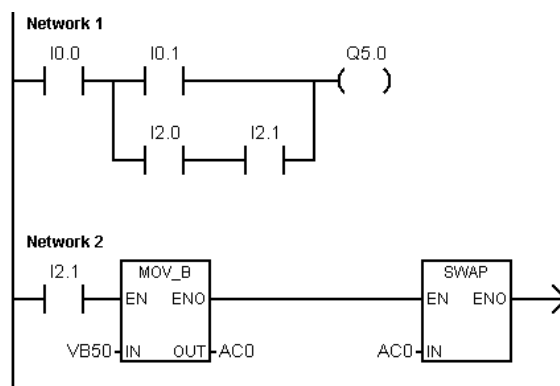


Рис. 5–3. Пример программы в виде цепной логической схемы

Блоки представляют дополнительные команды, например, таймеры, счетчики или математические команды.

При выборе редактора LAD примите во внимание следующее:

- Цепная логическая схема проста в использовании для начинающих программистов.
- Графическое представление легко понимается и популярно во всем мире.
- Редактор LAD можно использовать и с системой команд SIMATIC, и с системой команд IEC 1131–3.
- Для отображения программы, созданной при помощи редактора SIMATIC LAD, всегда можно использовать редактор STL.

Функции редактора FBD

Редактор функционального плана FBD отображает программу в виде графического представления, напоминающего обычные логические схемы. Нет никаких контактов и катушек, как в редакторе LAD, но имеются эквивалентные команды, представленные в виде блоков.

На рис. 5–4 показан пример программы в виде функционального плана.

FBD не использует понятия левой и правой токовой шины; поэтому понятие «поток сигнала» выражает аналогичное понятие потока управления через логические блоки FBD.

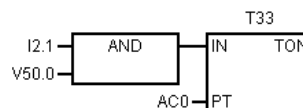


Рис. 5–4. Пример программы в виде функционального плана

По этой причине путь состояния «1» через элементы FBD называется потоком сигнала. Происхождение потока сигнала и место назначения его выхода ставятся в соответствие непосредственно операнду.

Логика программы вытекает из связей между функциональными блоками, обозначающими команды. Т.е. выход одной команды (например, блок логического И (AND)) может быть использован для разблокирования другой команды (например, таймера), формируя необходимую логику управления. Эта концепция позволяет решать широкий спектр задач управления.

При выборе редактора FBD примите во внимание следующее:

- Графическое представление функционального плана хорошо отражает процесс выполнения программы.
- Редактор FBD можно использовать и с системой команд SIMATIC, и с системой команд IEC 1131–3.
- Для отображения программы, созданной при помощи редактора SIMATIC FBD, всегда можно использовать редактор STL.

Наборы команд SIMATIC и IEC 1131-3

Большинство ПЛК предоставляют похожие основные команды, но обычно имеются незначительные различия в их внешнем виде, действии и т.д. в зависимости от поставщика. В течение последних нескольких лет Международная электротехническая комиссия (IEC) разработала всеобщий стандарт, который относится ко многим аспектам программирования ПЛК. Этот стандарт поощряет различных изготовителей ПЛК предлагать команды, являющиеся одинаковыми и по внешнему виду, и по действию.

Ваш S7–200 предлагает два набора команд, позволяющих решать широкий спектр задач автоматизации: набор команд IEC, соответствующий стандарту IEC 1131-3 для программирования ПЛК, и набор команд SIMATIC, разработанный специально для S7–200.



Совет

Когда STEP 7-Micro/WIN установлен в режим IEC, он отображает красный ромб (♦) в дереве команд рядом с командами, которые не определены стандартом IEC 1131-3.

Есть несколько ключевых различий между системой команд SIMATIC и системой команд IEC:

- Набор команд IEC ограничивается командами, которые являются стандартными среди поставщиков ПЛК. Некоторые команды, обычно включаемые в систему команд SIMATIC, не являются стандартными командами в спецификации IEC 1131–3. Они все еще доступны для использования как нестандартные команды, но если вы используете их, то программа больше не является строго совместимой с IEC 1131-3.
- У некоторых команд IEC в форме блоков возможна работа с несколькими форматами данных. Это свойство часто называют "перегрузкой". Например, вместо того, чтобы иметь отдельные математические блоки ADD_I (сложение целых чисел) и ADD_R (сложение вещественных чисел), команда ADD стандарта IEC 1131–3 проверяет формат складываемых данных и автоматически выбирает правильную команду S7–200. Это может несколько сэкономить затраты времени на программирование.
- Когда вы используете команды IEC, параметры команды автоматически проверяются на правильность формата данных, например, целое со знаком вместо целого без знака. Например, если вы попытались ввести целочисленное значение для команды, которая ожидала битовое значение (вкл/выкл), то происходит ошибка. Это свойство помогает минимизировать синтаксические ошибки программирования.

Делая выбор в пользу набора команд SIMATIC или IEC, примите во внимание следующие особенности:

- Команды SIMATIC обычно исполняются быстрее. Некоторые команды IEC могут иметь более длительные времена выполнения.
- Некоторые команды IEC, например, таймеры, счетчики, умножение и деление, работают иначе, чем их аналоги в SIMATIC.
- С набором команд SIMATIC можно использовать все три редактора программ (LAD, STL, FBD). С набором команд IEC можно использовать только редакторы LAD и FBD.
- Принцип действия команд IEC стандартизован для различных марок ПЛК, т.е. программы, удовлетворяющие IEC, могут разрабатываться независимо от системы автоматизации.
- Набор команд SIMATIC содержит больше операций, чем определено в стандарте IEC. Поэтому вы всегда можете включить команды SIMATIC в свою программу с командами IEC.
- IEC 1131-3 устанавливает, что переменные должны описываться с указанием типа, и поддерживает проверку типа данных системой.

Соглашения, используемые в редакторах программ

В STEP 7-Micro/WIN для всех редакторов действуют следующие соглашения:

- Знак # перед символическим именем (#var1) указывает, что этот символ имеет локальную сферу действия.
- Для команд IEC символ % указывает на прямую адресацию.
- Символ операнда «?.?» или «????» указывает, что требуется конфигурирование операнда.

Программы, написанные в редакторе LAD, делятся на сегменты (network). Сегмент – это упорядоченное расположение контактов, катушек и блоков, которые соединены между собой, образуя замкнутую токовую цепь: без коротких замыканий, разомкнутых цепей и условий для протекания потока сигнала в обратном направлении. STEP 7-Micro/WIN позволяет создавать комментарии к сегментам вашей программы, написанной в редакторе LAD. Программирование в редакторе FBD использует концепцию сегментов для разделения и комментирования вашей программы.

Программы на STL не используют сегментов; однако, вы можете использовать ключевое слово NETWORK для разбиения своей программы на части.

Соглашения, относящиеся к редактору LAD

В редакторе LAD вы можете использовать на своей клавиатуре клавиши F4, F6 и F9 для обращения к командам «Контакт», «Блок» и «Катушка». В редакторе LAD используются следующие соглашения:

- Символ «--->» означает разомкнутую цепь или требование подключения потока сигнала.
- Символ «→» указывает, что выход представляет собой необязательный поток сигнала для команды, которая может быть включена каскадом или последовательно.
- Символ «>>» указывает, что вы можете использовать поток сигнала.

Соглашения, относящиеся к редактору FBD

В редакторе FBD вы можете использовать на своей клавиатуре клавиши F4, F6 и F9 для доступа к командам AND [И], OR [ИЛИ] и «Блок». Используются следующие соглашения:

- Символ «--->» на операнде EN – это поток сигнала или индикатор операнда. Он может также изображать разомкнутую цепь или требование подключения потока сигнала.
- Символ «→» указывает, что выход представляет необязательный поток сигнала для команды, которая может быть включена каскадом или последовательно.
- Символы «<<» и «>>» показывают, что вы можете использовать значение или поток сигнала.
- Обозначение отрицания: Логическое отрицание NOT [НЕ] или инверсия состояния операнда или потока сигнала изображается небольшим кружком на входе. На рис. 5–5 Q0.0 равно результату логической операции НЕ I0.0 И I0.1. Такое обозначение отрицания действительно только для булевых сигналов, которые могут быть заданы как параметры или поток сигнала.



Рис. 5–5. Соглашения для FBD

- Индикаторы непосредственного управления: Как показано на рис. 5–5, редактор FBD изображает условие непосредственного управления булевым операндом вертикальной чертой на входе в команду FBD. Индикаторы непосредственного управления вызывают непосредственное чтение с указанного физического входа. Непосредственно управляемые операторы действительно только для физических входов.
- Блок без входов или выходов: Блок без входа обозначает команду, не зависящую от потока сигнала.



Совет

Количество операндов может быть расширено до 32 входов для команд AND [И] и OR [ИЛИ]. Для добавления и удаления обозначений операндов используйте клавиши «+» и «-» на своей клавиатуре.

Общие соглашения по программированию для S7–200

Определение EN/ENO

EN (Enable IN = Разрешающий вход) – это булев вход для блоков в LAD и FBD. Чтобы команда, представленная в виде блока, исполнялась, на этом входе должен присутствовать поток сигнала. В STL команды не имеют входа EN, но вершина стека должна быть логической “1”, чтобы соответствующая команда STL исполнялась.

ENO (Enable Out = Разрешающий выход) – это булев выход для блоков в LAD и FBD. Если у блока имеется поток сигнала на входе EN, и блок выполняет свою функцию без ошибок, то выход ENO передает поток сигнала следующему элементу. Если при исполнении блока обнаруживается ошибка, то поток сигнала завершается на блоке, в котором произошла ошибка.

В STL нет выхода ENO, но команды STL, соответствующие командам LAD и FBD с выходами ENO, устанавливают специальный бит ENO. Это бит доступен с помощью команды STL AENO (AND ENO) и может быть использован для создания того же эффекта, что и бит ENO блока.



Совет

Операнды и типы данных EN/ENO в таблице действительных операндов для отдельных команд не показаны, так как эти операнды одинаковы для всех команд LAD и FBD. Таблица 5–1 перечисляет эти операнды и типы данных для LAD и FBD. Эти операнды применимы ко всем командам LAD и FBD, представленным в данном руководстве.

Таблица 5–1. Операнды и типы данных EN/ENO для LAD и FBD

Редактор программ	Входы/выходы	Операнды	Типы данных
LAD	EN, ENO	Поток сигнала	BOOL
FBD	EN, ENO	I, Q, V, M, SM, S, T, C, L	BOOL

Условные и безусловные входы

В LAD и FBD блок или катушка, зависящие от потока сигнала, изображаются присоединенными к какому-нибудь элементу с левой стороны. Катушка или блок, не зависящие от потока сигнала, изображаются непосредственно подключенными к левой шине. Условный и безусловный входы показаны в таблице 5–2.

Таблица 5–2. Представление условного и безусловного входов

Поток сигнала	LAD	FBD
Команда, зависящая от потока сигнала (условная)		
Команда, не зависящая от потока сигнала (безусловная)		

Команды без выходов

Блоки, не допускающие каскадного соединения, изображаются без булевых выходов. Сюда относятся команды вызова подпрограммы, перехода на метку и условного завершения подпрограммы. В LAD тоже имеются катушки, которые могут быть помещены только у левой шины. Сюда относятся команды определения метки перехода, конца программного цикла с NEXT, загрузки реле последовательного управления (SCR), условного завершения SCR и конца SCR. В FBD они изображаются как блоки и отличаются непомеченными входами и отсутствием выходов.

Команды сравнения

Команда сравнения выполняется независимо от состояния потока сигнала. Если поток сигнала отсутствует (ложь), то выход ложен. Если поток сигнала присутствует (истина), то выход устанавливается в зависимости от результата сравнения. Команды сравнения SIMATIC FBD, IEC LAD и IEC FBD представляются как блоки, хотя операция выполняется как контакт.

Создание программы управления с помощью мастеров

В STEP 7-Micro/WIN имеются мастера, которые автоматизируют некоторые аспекты программирования и делают его более легким. В главе 6 команды, у которых есть соответствующий мастер, обозначены следующим символом:



Устранение ошибок в S7-200

S7-200 разделяет ошибки на фатальные и не фатальные. Коды, сгенерированные ошибкой, можно посмотреть, выбрав команду меню **PLC > Information [ПЛК → Информация]**.

На рис. 5–6 показано диалоговое PLC Information [Информация ПЛК], содержащее и описание ошибки.

Поле Last Fatal [Последняя фатальная ошибка] показывает код предыдущей фатальной ошибки, сгенерированной питанием, если сохраняется ОЗУ. Эта ячейка очищается всякий раз, когда очищается вся память S7-200, или когда ОЗУ не сохраняется после длительного перерыва в подаче питания.

Поле Total Fatal [Всего фатальных ошибок] представляет собой количество фатальных ошибок, сформированных S7-200 начиная с момента последней очистки всех областей памяти S7-200. Это значение сохраняется при выключениях и включениях питания, если сохраняется ОЗУ. Эта ячейка очищается всякий раз, когда очищается вся память S7-200, или когда ОЗУ не сохраняется после длительного перерыва в подаче питания.

В Приложении С перечислены коды ошибок S7-200, а в Приложении D описаны биты специальной памяти (SM), которые могут быть использованы для контроля ошибок.

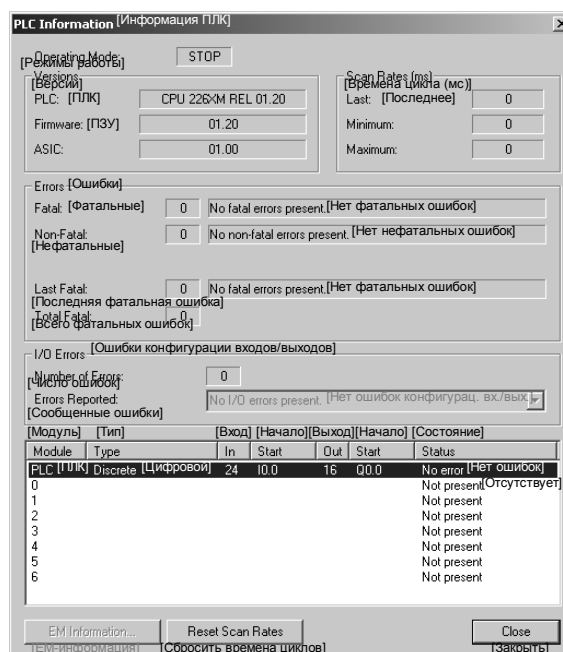


Рис. 5–6. Диалоговое окно с информацией ПЛК

Нефатальные ошибки

В случае нефатальных ошибок речь идет об ошибках в построении программы пользователя, об ошибке при выполнении команды в программе пользователя и об ошибках в модулях расширения. С помощью STEP 7-Micro/WIN можно отобразить коды нефатальных ошибок. Имеется три основных группы нефатальных ошибок.

Ошибки компиляции программы

S7-200 компилирует программу, когда он ее загружает. Если S7-200 обнаруживает, что программа нарушает правило компиляции, то загрузка прерывается и генерируется код ошибки. (Программа, которая уже была загружена в S7-200, по-прежнему будет существовать в постоянной памяти и не потеряется.) После исправления своей программы вы можете загрузить ее снова. Список нарушений правил компиляции вы найдете в Приложении С.

Ошибки конфигурации входов/выходов

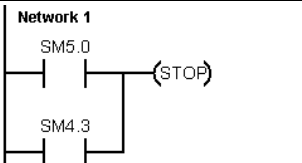
При запуске S7-200 считывает конфигурацию входов-выходов из каждого модуля. При нормальной работе S7-200 периодически проверяет состояние каждого модуля и сравнивает его с конфигурацией, полученной при запуске. Если S7-200 обнаруживает разницу, он устанавливает бит ошибки конфигурации в регистре ошибок модуля. S7-200 не считывает входные данные из этого модуля и не записывает выходные данные в этот модуль, пока конфигурация модуля снова не совпадет с конфигурацией, полученной при запуске.

Информация о состоянии модуля хранится в битах специальной памяти (SM). Ваша программа может контролировать и анализировать эти биты. Подробную информацию о битах специальной памяти, используемых для индикации ошибок конфигурации входов/выходов вы найдете в Приложении D. Бит SM5.0 является глобальным битом ошибок конфигурации входов/выходов, который остается установленным, пока в модуле расширения сохраняется сбойная ситуация.

Ошибки выполнения программы

Ваша программа может создавать состояния ошибки во время своего выполнения. Эти ошибки могут возникать из-за ненадлежащего использования команды или из-за обработки командой недопустимых данных. Например, указатель косвенного адреса, который был действительным, когда программа компилировалась, может быть изменен во время выполнения программы так, что станет указывать на адрес вне допустимого диапазона. Это пример ошибки программирования, проявляющейся при выполнении программы. При возникновении такой ошибки устанавливается бит SM4.3. Он остается установленным, пока S7-200 находится в режиме RUN. (Список ошибок программирования, проявляющихся при выполнении программы, вы найдете в Приложении С). Информация об ошибках выполнения программы хранится в битах специальной памяти (SM). Ваша программа может контролировать и анализировать эти биты. Подробную информацию о битах специальной памяти, используемых для индикации ошибок исполнения программы, вы найдете в Приложении D.

Когда S7-200 обнаруживает нефатальную ошибку, он не переключается в режим STOP. Он только регистрирует событие в памяти SM и продолжает выполнение вашей программы. Однако вы можете спроектировать свою программу так, чтобы она принуждала S7-200 к переходу в состояние STOP, когда обнаруживается нефатальная ошибка. Следующий пример показывает сегмент программы, которая контролирует два глобальных бита нефатальных ошибок и переводит S7-200 в STOP всякий раз, когда устанавливается любой из этих битов.

Пример программы: Логика обнаружения нефатальной ошибки	
	<pre> Network 1 //При возникновении ошибки конфигурации //входов/выходов или ошибки выполнения программы //перейти в STOP LD SM5.0 O SM4.3 STOP </pre>

Фатальные ошибки

Фатальные ошибки заставляют S7–200 прекратить выполнение программы. В зависимости от тяжести фатальной ошибки S7–200 может потерять способность к выполнению некоторых или всех функций. Целью обработки фатальных ошибок является перевод S7–200 в безопасное состояние, из которого S7–200 может реагировать на запросы о существующих сбойных состояниях. Когда S7–200 обнаруживает фатальную ошибку, он переключается в режим STOP, включает светодиоды SF/DIAG (красный) и STOP, заменяет таблицу выходов и выключает выходы. S7–200 остается в этом состоянии до исправления фатальной ошибки.

Как только вы сделали изменения для устранения фатальной ошибки, вы должны перезапустить S7–200, используя один из следующих методов:

- Выключите, а затем включите питание.
- Переведите переключатель режимов работы из RUN или TERM в STOP.
- Выберите из STEP 7-Micro/WIN команду меню **PLC > Power-Up Reset [ПЛК > Сброс при запуске]** для запуска S7–200. Это заставляет S7–200 перезапуститься и сбросить все фатальные ошибки.

Перезапуск S7–200 сбрасывает состояние фатальной ошибки и выполняет диагностический тест, связанный с включением питания, чтобы проверить, что фатальная ошибка была устранена. Если обнаруживается другая фатальная ошибка, то S7–200 снова устанавливает светодиод ошибки, показывая, что ошибка по-прежнему существует. В противном случае S7–200 начинает нормальную работу.

Имеется несколько возможных сбойных состояний, которые могут сделать S7–200 некоммуникабельным. В этих случаях вы не можете отобразить код ошибки S7–200. Эти типы ошибок указывают на аппаратные отказы, требующие ремонта S7–200; их невозможно устранить посредством изменений в программе или очистки памяти S7–200.

Назначение адресов и начальных значений в редакторе блоков данных



В редакторе блоков данных вы можете выполнить присваивание начальных значений в памяти переменных. Вы можете выполнять назначения байтам, словам или двойным словам памяти переменных. Комментарии не обязательны.

Редактор блоков данных – это текстовый редактор со свободно выбираемым форматом; это значит, что поля для тех или иных данных заранее не определяются. После того как вы напечатали строку и нажали клавишу Enter, редактор блоков данных форматирует эту строку (выравнивает столбцы адресов, данных, комментариев; представляет адреса в памяти переменных большими буквами) и вновь ее отображает. Если вы в конце строки нажмете CTRL-ENTER, то адрес автоматически пересчитывается на следующее доступное значение.

```

//BEGIN TD200_BLOCK -----
V80  TD0  //Set Language to English, set Update to as fast as possible
V81  16#10 //Set the display to 20 character mode; Up key V3.2; Down key V3.3;
V82  16#70 //Set the number of messages
V84  1 //Set the Function Keys notification bits to NO.0 - NO.7
V85  0 //Set the starting address for messages to V813
V86  13 //Set the starting address for message enable bits to V812
V88  12 //Global Password (if enabled)
//MESSAGE 1
//Message Enable Bit V12.7
V89  "Temperature" //No Edit;No Acknowledgment;No Password;
V89  16#0 //Signed Word; 1 Digits to the right of the decimal;
V89  16#0000 //Embedded Data Value; Move data for display here.
V89  "F"
//END TD200_BLOCK -----
  
```

Рис. 5–7. Редактор блоков данных

Редактор блоков данных выделяет соответствующее место в памяти переменных в соответствии с вашими предыдущими распределениями адресов и размерами (байт, слово или двойное слово) значений данных.

Первая строка блока данных должна содержать явно назначенный адрес. Адреса в следующих строках могут быть назначены явно или неявно. Неявное присвоение адресов выполняется редактором, когда вы вводите несколько значений данных после назначенного адреса или вводите строку, содержащую только значения данных.

Редактор блоков данных принимает большие и маленькие буквы и допускает использование запятых, табуляций и пробелов в качестве разделителей между адресами и значениями данных.

Использование таблицы символов для символической адресации переменных

Таблица символов дает возможность определять и редактировать символы, к которым можно обращаться во всей программе через символические имена. Вы можете создать несколько таблиц символов. В таблице символов имеется также закладка для символов, определенных системой, которые вы можете использовать в своей программе. Таблица символов называется также таблицей глобальных переменных.



Таблица символов

Вы можете указывать операнды команд в своей программе абсолютно или символически. При абсолютной адресации задается область памяти, а также бит или байт адреса. При символической адресации для указания адреса используются комбинации алфавитно-цифровых символов.

Для программ SIMATIC назначение глобальных символов производится в таблице символов. Для программ IEC назначение глобальных символов производится в таблице глобальных переменных.

Для присвоения адресу символического имени действуйте следующим образом:

	Symbol	Address	Comment
1	AlwaysOn	SM0.0	Always on contact
2	Pump1	Q2.3	Pump 1 on/off
3	Pump1Limit	I1.1	Pump 1 pressure limit switch
4	Pump1Pressure	VD100	Pump 1 current pressure (real)
5	Pump1Rpm	VW200	Pump1 PRMs (integer)
6			

Рис. 5–8. Таблица символов

1. Щелкните в навигационной панели на кнопке таблицы символов, чтобы вызвать таблицу.
2. Введите символическое имя (например, Input1) в столбце "Symbol Name [Символическое имя]". Максимальная длина символического имени составляет 23 символа.
3. В столбце Address [Адрес] введите адрес (например, I0.0).
4. Для таблицы глобальных переменных IEC введите значение в столбец Data Type [Тип данных] или выберите его из раскрывающегося окна списка.

Вы можете создать несколько таблиц символов, но вы не можете использовать одну и ту же строку более одного раза в качестве глобального символического имени ни в единственной таблице, ни в нескольких различных таблицах.

Использование локальных переменных

Таблицу локальных переменных редактора программ можно использовать для определения переменных, которые встречаются только в отдельной подпрограмме или программе обработки прерываний. См. рис. 5–9.

Локальные переменные могут использоваться как параметры, которые передаются в подпрограмму. Локальные переменные увеличивают мобильность и возможность повторного использования подпрограммы.

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	FinalPass	IN	BOOL	Final pass flag
LB1	Addr	IN	BYTE	Address of slave device
LW2	Data	IN	INT	Data to write to slaves
LB4	Status	IN_OUT	BYTE	Status of write
LB.0	Done	OUT	BOOL	Done flag
LWS	Error	OUT	WORD	Error number (if any)

Рис. 5–9. Таблица локальных переменных

Контроль над программой с помощью таблицы состояний



С помощью таблицы состояний можно наблюдать и изменять переменные процесса, когда ваш S7–200 исполняет программу управления. Вы можете отслеживать состояние входов, выходов или переменных программы, отображая их текущие значения. В таблице состояний можно также принудительно задавать или изменять значения переменных процесса.

Вы можете создать несколько таблиц состояний, чтобы иметь возможность просматривать элементы из различных частей своей программы.

Для вызова таблицы состояний выберите команду меню **View > Component > Status Chart [Вид > Компонент > Таблица состояний]** или щелкните на пиктограмме таблицы состояний на навигационной панели.

При создании таблицы состояний введите адреса переменных процесса, которые вы хотите наблюдать. Невозможно отобразить состояния констант, аккумуляторов и локальных переменных. Значения таймеров и счетчиков можно отображать в виде бита или слова. Если значение отображается в виде бита, то оно представляет состояние бита таймера или счетчика; если значение отображается в виде слова, то оно является значением таймера или счетчика.

	Address	Format	Current Value	New Value
1	Pump1	Bit	2#0	
2	PumpLimit	Bit	2#0	
3	PumpPressure	Signed	+0	
4	PumpFlow	Signed	+0	
5	M3.7	Bit	2#0	
6	W0.00	Hexadecimal	1000	
7	W0.00	Floating Point	0.0	
8		Signed		

Рис. 5–10. Таблица состояний

Для создания таблицы состояний и контроля переменных:

1. Введите в поле адресов адреса желаемых величин.
2. В столбце Format выберите тип данных.
3. Для отображения состояния переменных процесса в своем S7–200 выберите команду меню **Debug > Chart Status [Отладка > Состояние таблицы]**.
4. Если вы хотите опрашивать эти величины непрерывно или хотите однократно считать состояние, щелкните на соответствующем символе на панели инструментов. В таблице состояний можно также принудительно устанавливать или изменять значения различных переменных процесса.

В таблицу состояний можно вставлять дополнительные строки, выбрав команду меню **Edit > Insert > Row [Редактировать > Вставить > Строка]**.



Совет

Вы можете создать несколько таблиц состояний, чтобы разделить переменные на логические группы, чтобы каждую группу можно было наблюдать в собственной более короткой таблице.

Создание библиотеки команд

В STEP 7-Micro/WIN можно создать библиотеку команд для конкретного пользователя или использовать библиотеку, созданную другими лицами. См. рис. 5–11.

Для создания библиотеки команд создайте сначала в STEP 7-Micro/WIN подпрограммы и программы обработки прерываний и сгруппируйте их. Вы можете скрыть код в этих подпрограммах и программах обработки прерываний, чтобы предотвратить случайные изменения и защитить технологию или ноу-хау автора.

Для создания библиотеки команд действуйте следующим образом:

1. Напишите программу в виде стандартного проекта STEP 7-Micro/WIN и поместите функции, которые должны быть включены в библиотеку, в подпрограммы или программы обработки прерываний.
2. Обеспечьте, чтобы все адреса в памяти переменных в подпрограммах или программах обработки прерываний получили символические имена. В памяти переменных используйте адреса, следующие друг за другом, чтобы минимизировать размер памяти переменных, необходимой для библиотеки.
3. Переименуйте подпрограммы и программы обработки прерываний в соответствии с названиями, с которыми они должны находиться в библиотеке.
4. Выберите команду меню **File > Create Library [Файл > Создать библиотеку]**, чтобы скомпилировать новую библиотеку команд.

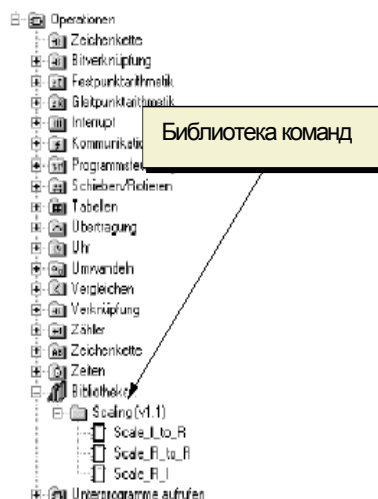


Рис. 5–11. Дерево команд с библиотеками

Дополнительную информацию о создании библиотек вы найдете в помощи для STEP 7-Micro/WIN в режиме online.

Чтобы получить доступ к командам из библиотеки команд, действуйте следующим образом:

1. Добавьте к дереву команд каталог "Libraries [Библиотеки]", выбрав команду меню **File > Add Libraries [Файл > Добавить библиотеки]**.
2. Выберите желаемую команду и вставьте ее в свою программу (как любую стандартную команду).

Если библиотечная программа нуждается в памяти переменных, то STEP 7-Micro/WIN после компиляции проекта потребует назначения области памяти. Области памяти назначаются в диалоговом окне "Library Memory Allocation [Выделение памяти для библиотеки]".

Функции тестирования программы

STEP 7-Micro/WIN предоставляет следующие функции для тестирования программы:

- Установка закладок в программе для облегчения поиска определенных строк программы.
- Таблица перекрестных ссылок, дающая возможность проверки ссылок, используемых в программе.
- Редактирование программы в режиме RUN, позволяющее выполнять небольшие изменения в пользовательской программе с минимальными помехами процессу, управляемому программой. При редактировании программы в режиме RUN вы можете также загрузить программный блок.

Дополнительную информацию о тестировании программы вы найдете в главе 8.

Набор команд S7–200

6

В этой главе описаны наборы команд SIMATIC и IEC 1131 для микро-ПЛК S7–200.

В этой главе

Соглашения, используемые для описания команд	69
Области памяти и функции S7–200	70
Битовые логические операции	72
Контакты	72
Катушки	75
Стековые операции	77
Функциональный блок с двумя устойчивыми состояниями: преимущество установки и преимущество сброса	79
Команды, связанные со временем	80
Команды для обмена данными	83
Команды чтения из сети и записи через сеть	83
Команды передачи и приема (свободно программируемый обмен данными)	88
Команды получения и установки адреса порта	97
Команды сравнения	98
Сравнение числовых величин	98
Сравнение строк	100
Команды преобразования	101
Стандартные команды преобразования	101
Команды преобразования ASCII	105
Команды преобразования строк	109
Команды кодирования и декодирования	114
Команды счета	115
Счетчик SIMATIC	115
Счетчик IEC	118
Скоростные счетчики	120
Команда вывода импульсов	135
Арифметические операции	142
Операции сложения, вычитания, умножения и деления	142
Умножение целых чисел с представлением результата в виде двойного целого числа и деление целых чисел с остатком	144
Числовые функции	145
Команды инкрементирования и декрементирования	146
Пропорционально-интегрально-дифференциальный (PID) регулятор	147
Команды прерывания	155
Логические операции	163
Операции инвертирования	163
Поразрядные логические операции И, ИЛИ и исключающее ИЛИ	164
Команды пересылки	166
Пересылка байта, слова, двойного слова или вещественного числа	166
Непосредственное чтение или запись и пересылка байта	167
Команды групповой пересылки	168
Команды управления программой	169
Условное завершение	169

Останов	169
Сброс контроля времени	169
Команды формирования программного цикла For-Next	171
Команды перехода	173
Команды для реле управления очередностью (SCR)	174
Команда управления диагностическим светодиодом	180
Команды сдвига и циклического сдвига	181
Команды сдвига вправо и сдвига влево	181
Команды циклического сдвига вправо и циклического сдвига влево	181
Команда перемещения значения бита в регистр сдвига	183
Команда перестановки байтов в слове	185
Строковые команды	186
Табличные команды	191
Ввод значения в таблицу	191
Удаление первой записи из таблицы и удаление последней записи из таблицы	192
Заполнение памяти комбинацией двоичных разрядов	194
Поиск значения в таблице	195
Таймерные команды	198
Таймерные команды SIMATIC	198
Таймерные команды IEC	203
Интервальные таймеры	205
Операции с подпрограммами	206

Соглашения, используемые для описания команд

На рис. 6–1 представлены типичное описание команды и указания на места в тексте, где описана эта команда и ее действие. На изображении команды показан ее формат в LAD, FBD и STL. В таблице операндов приведены операнды команды и показаны допустимые типы данных, области памяти и размеры для каждого операнда.

Операнды EN и ENO и их типы данных в таблице операндов команды не представлены, так как эти операнды одинаковы для всех команд LAD и FBD.

- Для LAD: EN и ENO являются потоками сигнала и имеют тип BOOL.
- Для FBD: EN и ENO являются потоками сигнала или I, Q, V, M, SM, S, T, C, L и имеют тип BOOL.

Описание команды и операндов

Список сбойных состояний, влияющих на ENO, и соответствующих битов специальной памяти

Операнды для команды

Допустимые типы данных

Допустимые области памяти и размеры операндов

Команда STL

Команды LAD и FBD

Transmit and Receive Instructions

The Transmit instruction (XMT) is used in Freeport mode to transmit data by means of the communication port(s).

The Receive instruction (RCV) initiates or terminates the Receive Message service. You must specify a start and an end condition for the Receive box to operate. Messages received through the specified port (PORT) are stored in the data buffer (TBL). The first entry in the data buffer specifies the number of bytes received.

Error conditions that set ENO = 0

- SM65.6 or SM186.6 error bit set (RCV parameter error)
- 0006 (indirect address)
- 0009 (simultaneous XMT/RCV on port 0)
- 000B (simultaneous XMT/RCV on port 1)
- S7-200 CPU not in Freeport mode

Table 6-11 Valid Operands for the Transmit and Receive Instructions

Inputs/Outputs	Data Type	Operands
TBL	BYTE	VB, IB, QB, MB, SB, SMB, *VD, *AC, *LD
PORT	BYTE	Constant for CPU 221, CPU 222, CPU 226: 0 for CPU 226 and CPU 226XM: 0 or 1

Using Freeport Mode to Control the Serial Communication Port

You can select the Freeport mode to control the serial communication port of the S7-200 by means of the program. When you select Freeport mode, your program controls the operation of the communication port. The use of the receive interrupts, the transmit interrupts, the Transmit instruction, and the Receive instruction. The communication protocol is entirely controlled by the ladder program while in Freeport mode. SMB30 (for port 0) and SMB130 (for port 1) if your S7-200 has two ports) are used to specify baud rate and parity.

The Freeport mode is disabled and normal communication is re-established (for example, programming device connection) when the S7-200 is in STOP mode.

Freeport communication is possible only when the S7-200 is in the RUN mode. Enable the Freeport mode by setting a value of 01 in the protocol select field of SMB30 (Port 0) or SMB130 (Port 1). While in Freeport mode, communication with the programming device is not possible.

Tip

Entering Freeport mode can be controlled using special memory bit SM0.7, which reflects the current position of the operating mode switch. When SM0.7 is equal to 0, the switch is in TERM position; when SM0.7 = 1, the operating mode switch is in RUN position. If you enable Freeport mode only when the switch is in RUN position, you can use the programming device to monitor or control the S7-200 operation by changing the switch to any other position.

Рис. 6–1. Описание команд

Области памяти и функции S7-200

Таблица 6-1. Области памяти и функции CPU S7-200

Описание	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Размер программы пользователя с редактированием в режиме RUN без редактирования в режиме RUN	4096 байт 4096 байт	4096 байт 4096 байт	8192 байта 12288 байт	12288 байт 16384 байта	16384 байта 24576 байт
Размер данных пользователя	2048 байт	2048 байт	8192 байта	10240 байт	10240 байт
Регистр входов образа процесса	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7
Регистр выходов образа процесса	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7
Аналоговые входы (только чтение)	AIW0 – AIW30	AIW0 – AIW30	AIW0 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	AQW0 – AQW30	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB8191	VB0 – VB10239	VB0 – VB10239
Локальная память (L) ¹	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63
Битовая память (M)	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7
Специальная память (SM) только чтение	SM0.0 – SM179.7 SM0.0 – SM29.7	SM0.0 – SM299.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7
Таймеры Задержка включения с запоминанием	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)
1 мс	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10 мс	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68
100 мс	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95
Задержка вкл/выкл					
1 мс	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10 мс	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100
100 мс	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255
Счетчики	C0 – C255	C0 – C255	C0 – C255	C0 – C255	C0 – C255
Скоростные счетчики	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5
Реле последовательного управления (S)	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7
Аккумуляторные регистры	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3
Переходы/Метки	0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
Вызовы/Подпрограмма	0 – 63	0 – 63	0 – 63	0 – 63	0 – 127
Программы обработки прерываний	0 – 127	0 – 127	0 – 127	0 – 127	0 – 127
Нарастающие/падающие фронты	256	256	256	256	256
PID-регуляторы	0 – 7	0 – 7	0 – 7	0 – 7	0 – 7
Порты	Порт 0	Порт 0	Порт 0	Порт 0, порт 1	Порт 0, порт 1

¹ LB60 – LB63 зарезервированы для STEP 7-Micro/WIN, версии 3.0 и выше.

Таблица 6-2. Диапазоны операндов CPU S7-200

Метод доступа	CPU 221	CPU 222	CPU 224	CPU 224 XP	CPU 226	
Доступ к биту (байт.бит)	I	0.0 – 15.7	0.0 – 15.7	0.0 – 15.7	0.0 – 15.7	
	Q	0.0 – 15.7	0.0 – 15.7	0.0 – 15.7	0.0 – 15.7	
	V	0.0 – 2047.7	0.0 – 2047.7	0.0 – 8191.7	0.0 – 10239.7	0.0 – 10239.7
	M	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7
	SM	0.0 – 165.7	0.0 – 299.7	0.0 – 549.7	0.0 – 549.7	0.0 – 549.7
	S	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7	0.0 – 31.7
	T	0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
	C	0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
	L	0.0 – 63.7	0.0 – 63.7	0.0 – 63.7	0.0 – 63.7	0.0 – 63.7
Доступ к байту	IB	0 – 15	0 – 15	0 – 15	0 – 15	0 – 15
	QB	0 – 15	0 – 15	0 – 15	0 – 15	0 – 15
	VB	0 – 2047	0 – 2047	0 – 8191	0 – 10239	0 – 10239
	MB	0 – 31	0 – 31	0 – 31	0 – 31	0 – 31
	SMB	0 – 165	0 – 299	0 – 549	0 – 549	0 – 549
	SB	0 – 31	0 – 31	0 – 31	0 – 31	0 – 31
	LB	0 – 63	0 – 63	0 – 63	0 – 63	0 – 63
	AC	0 – 3	0 – 3	0 – 3	0 – 255	0 – 255
	KB (константа)	KB (константа)	KB (константа)	KB (константа)	KB (константа)	KB (константа)
	Доступ к слову	IW	0 – 14	0 – 14	0 – 14	0 – 14
QW		0 – 14	0 – 14	0 – 14	0 – 14	0 – 14
VW		0 – 2046	0 – 2046	0 – 8190	0 – 10238	0 – 10238
MW		0 – 30	0 – 30	0 – 30	0 – 30	0 – 30
SMW		0 – 164	0 – 298	0 – 548	0 – 548	0 – 548
SW		0 – 30	0 – 30	0 – 30	0 – 30	0 – 30
T		0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
C		0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
LW		0 – 62	0 – 62	0 – 62	0 – 62	0 – 62
AC		0 – 3	0 – 3	0 – 3	0 – 3	0 – 3
AIW		0 – 30	0 – 30	0 – 62	0 – 62	0 – 62
AQW		0 – 30	0 – 30	0 – 62	0 – 62	0 – 62
KW (константа)		KW (константа)	KW (константа)	KW (константа)	KW (константа)	KW (константа)
Доступ к двойному слову	ID	0 – 12	0 – 12	0 – 12	0 – 12	0 – 12
	QD	0 – 12	0 – 12	0 – 12	0 – 12	0 – 12
	VD	0 – 2044	0 – 2044	0 – 8188	0 – 10236	0 – 10236
	MD	0 – 28	0 – 28	0 – 28	0 – 28	0 – 28
	SMD	0 – 162	0 – 296	0 – 546	0 – 546	0 – 546
	SD	0 – 28	0 – 28	0 – 28	0 – 28	0 – 28
	LD	0 – 60	0 – 60	0 – 60	0 – 60	0 – 60
	AC	0 – 3	0 – 3	0 – 3	0 – 3	0 – 3
	HC	0 – 5	0 – 5	0 – 5	0 – 5	0 – 5
	KD (константа)	KD (константа)	KD (константа)	KD (константа)	KD (константа)	KD (константа)

Битовые логические операции

Контакты

Стандартные контакты

Команды "Нормально открытый контакт" (LD, A и O) и "Нормально замкнутый контакт" (LDN, AN, ON) получают исходное значение из памяти или из регистра образа процесса. Стандартные контакты получают исходное значение из памяти (или из регистра образа процесса, если типом данных является I или Q).

Нормально открытый контакт замкнут (включен), когда бит равен 1, а нормально замкнутый контакт замкнут (включен), когда бит равен 0. В FBD к блокам И и ИЛИ может быть подключено не более 32 входов. В STL команды, представляющие нормально открытый контакт, загружают значение адресного бита в вершину стека или выполняют логическое сопряжение значения адресного бита со значением в вершине стека в соответствии с таблицей истинности логического И или ИЛИ, а команды, представляющие нормально замкнутый контакт, загружают логическое отрицание значения адресного бита в вершину стека или выполняют логическое сопряжение логического отрицания значения адресного бита со значением в вершине стека в соответствии с таблицей истинности логического И или ИЛИ.

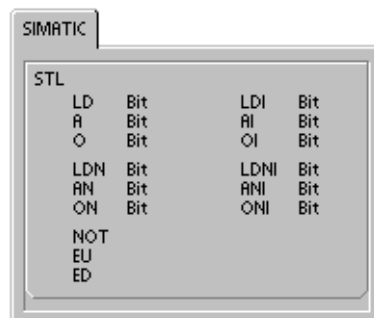
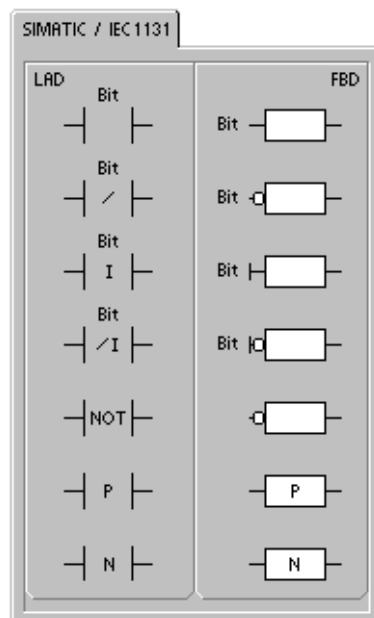
Непосредственно управляемые контакты

Непосредственно управляемый контакт при своей актуализации не зависит от цикла S7-200, его значение обновляется немедленно. Команды "Непосредственно управляемый нормально открытый контакт" (LDI, AI и OI) и "Непосредственно управляемый нормально замкнутый контакт" (LDNI, ANI и ONI) при выполнении команды получают значение физического входа, однако, регистр образа процесса не обновляется.

Непосредственно управляемый нормально открытый контакт замкнут (включен), когда физический вход (бит) находится в состоянии 1, а непосредственно управляемый нормально замкнутый контакт замкнут (включен), когда физический вход (бит) находится в состоянии 0. Команды, представляющие непосредственно управляемый нормально открытый контакт, непосредственно загружают значение физического входа в вершину стека или выполняют логическое сопряжение значения физического входа со значением в вершине стека в соответствии с таблицей истинности логического И или ИЛИ, а команды, представляющие непосредственно управляемый нормально замкнутый контакт, непосредственно загружают логическое отрицание значения физического входа в вершину стека или выполняют логическое сопряжение отрицания значения физического входа со значением в вершине стека в соответствии с таблицей истинности логического И или ИЛИ.

Команда NOT [NE]

Команда отрицания (NOT) изменяет состояние входа потока сигнала (т.е. она изменяет значение в вершине стека с 0 на 1 или с 1 на 0).



Положительный и отрицательный фронт

Контакт "Положительный фронт" (EU) пропускает поток сигнала в течение одного цикла при каждом появлении положительного фронта. Контакт "Отрицательный фронт" (ED) пропускает поток сигнала в течение одного цикла при каждом появлении отрицательного фронта. У команды "Положительный фронт" при обнаружении перехода значения в вершине стека с 0 на 1 значение в вершине стека устанавливается в 1; в противном случае оно устанавливается в 0. У команды "Отрицательный фронт" при обнаружении перехода значения в вершине стека с 1 на 0 значение в вершине стека устанавливается в 1; в противном случае оно устанавливается в 0.

При редактировании программы в режиме RUN необходимо ввести параметр для команд обнаружения положительного и отрицательного фронта. Подробную информацию о редактировании в режиме RUN вы найдете в главе 5.

Таблица 6-3. Допустимые операнды для битовых логических операций

Входы/выходы	Тип данных	Операнды
Бит	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Бит (непосредственно управляемый)	BOOL	I


Как показано на рис. 6-2, S7-200 использует логический стек для решения задач логики управления. В этих примерах «iv0» – «iv7» обозначают начальные значения логического стека, «nv» обозначает новое значение, поставяемое командой, а «S0» обозначает рассчитанное значение, сохраняемое в логическом стеке.



¹ S0 обозначает рассчитанное значение, сохраняемое в логическом стеке.

² После выполнения операции загрузки значение iv8 теряется.

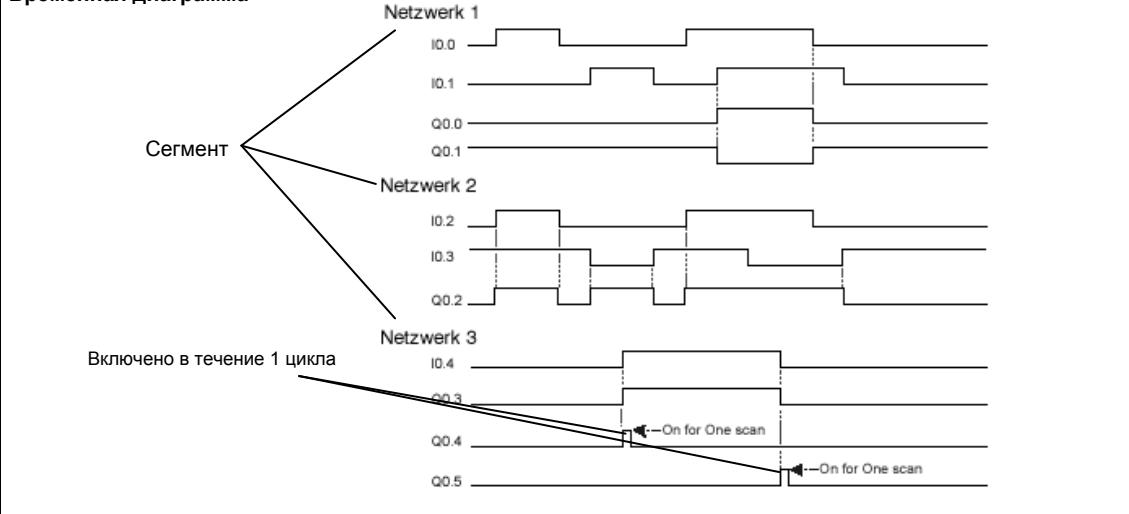
Рис. 6-2. Принцип действия контактов.

	<p>Совет</p> <p>Так как команды обнаружения положительного и отрицательного фронта требуют переключения, соответственно, с 0 на 1 или с 1 на 0, то нет возможности обнаружить нарастающий или падающий фронт в первом цикле. В первом цикле состояние указанного в этих командах бита устанавливает S7-200. В следующих циклах эти команды сами могут распознать фронты для указанного бита.</p>
---	---

Пример: Контакты

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p>	<p>Network 1 //Нормально открытые контакты I0.0 И I0.1 должны //быть замкнуты, чтобы активизировать Q0.0. //Команда NOT действует как инвертор. В режиме //RUN Q0.0 и Q0.1 имеют противоположные //логические состояния.</p> <pre>LD I0.0 A I0.1 = Q0.0 NOT = Q0.1</pre> <p>Network 2 //Чтобы активизировать Q0.2, должен быть включен //нормально открытый контакт I0.2 или выключен //нормально замкнутый контакт I0.3. Чтобы //активизировать выход, одна или более //параллельных ветвей цепной логической схемы //(LAD) (входов ИЛИ) должна принимать значение //"Истина".</p> <pre>LD I0.2 ON I0.3 = Q0.2</pre> <p>Network 3 //Положительный фронт на контакте P или //отрицательный фронт на контакте N приводят к //появлению импульса длительностью в 1 цикл. В // режиме RUN изменения импульсного состояния //Q0.4 и Q0.5 происходят слишком быстро, чтобы их //можно было увидеть в режиме просмотра статуса //программы. Выходы установки (S) и сброса (R) //сохраняют импульс в Q0.3 и делают изменение //состояния видимым в режиме просмотра статуса //программы.</p> <pre>LD I0.4 LPS EU S Q0.3, 1 = Q0.4 LPP ED R Q0.3, 1 = Q0.5</pre>
---	--

Временная диаграмма



Катушки

Присваивание

Команда присваивания (=) записывает новое значение для выходного бита в регистр образа процесса. При выполнении команды присваивания S7-200 устанавливает или сбрасывает выходной бит в регистре образа процесса. В LAD и FBD указанный бит устанавливается равным потоку сигнала. В STL значение, находящееся в вершине стека, копируется в указанный бит.

Непосредственное присваивание битового значения

Команда непосредственного присваивания битового значения (=I) при своем выполнении записывает новое значение как в физический выход, так и в образ процесса.

Когда выполняется команда непосредственного присваивания битового значения, физический выход (бит) немедленно устанавливается в соответствии с состоянием потока сигнала. В STL команда непосредственного присваивания битового значения непосредственно копирует значение, находящееся в вершине стека, в указанный физический выход. Символ "I" означает непосредственный доступ; при исполнении команды новое значение записывается в физический выход и в соответствующую ячейку регистра образа процесса. Здесь имеется отличие от других видов доступа, которые записывают новое значение только в регистр образа процесса.

Установка и сброс

Команды установки (S) и сброса (R) устанавливают (включают) или сбрасывают (выключают) указанное количество входов или выходов (N), начиная с указанного адреса (бита). Вы можете установить или сбросить от 1 до 255 входов и выходов.

Если команда сброса указывает на бит таймера (T) или счетчика (C), то команда сбрасывает бит таймера или счетчика и стирает текущее значение таймера или счетчика.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

Непосредственная установка и непосредственный сброс

Команды непосредственной установки и непосредственного сброса непосредственно устанавливают (включают) или непосредственно сбрасывают (выключают) указанное количество входов или выходов (N), начиная с указанного адреса (бита). Вы можете непосредственно и немедленно установить или сбросить от 1 до 128 входов и выходов. Символ "I" означает непосредственный доступ; при исполнении команды новое значение записывается в физический выход и в соответствующую ячейку регистра образа процесса. Здесь имеется отличие от других видов доступа, которые записывают новое значение только в регистр образа процесса.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

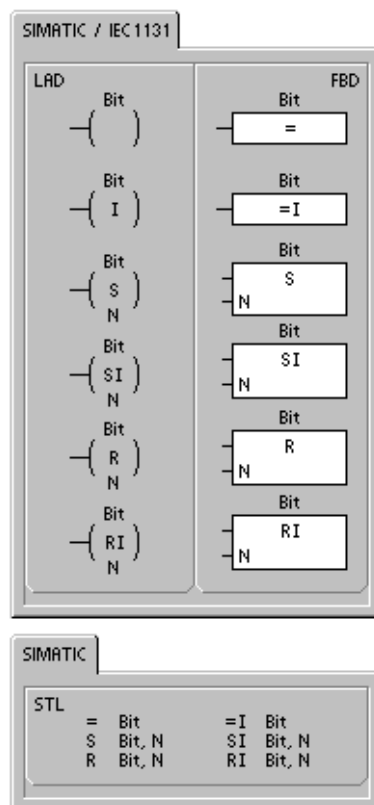
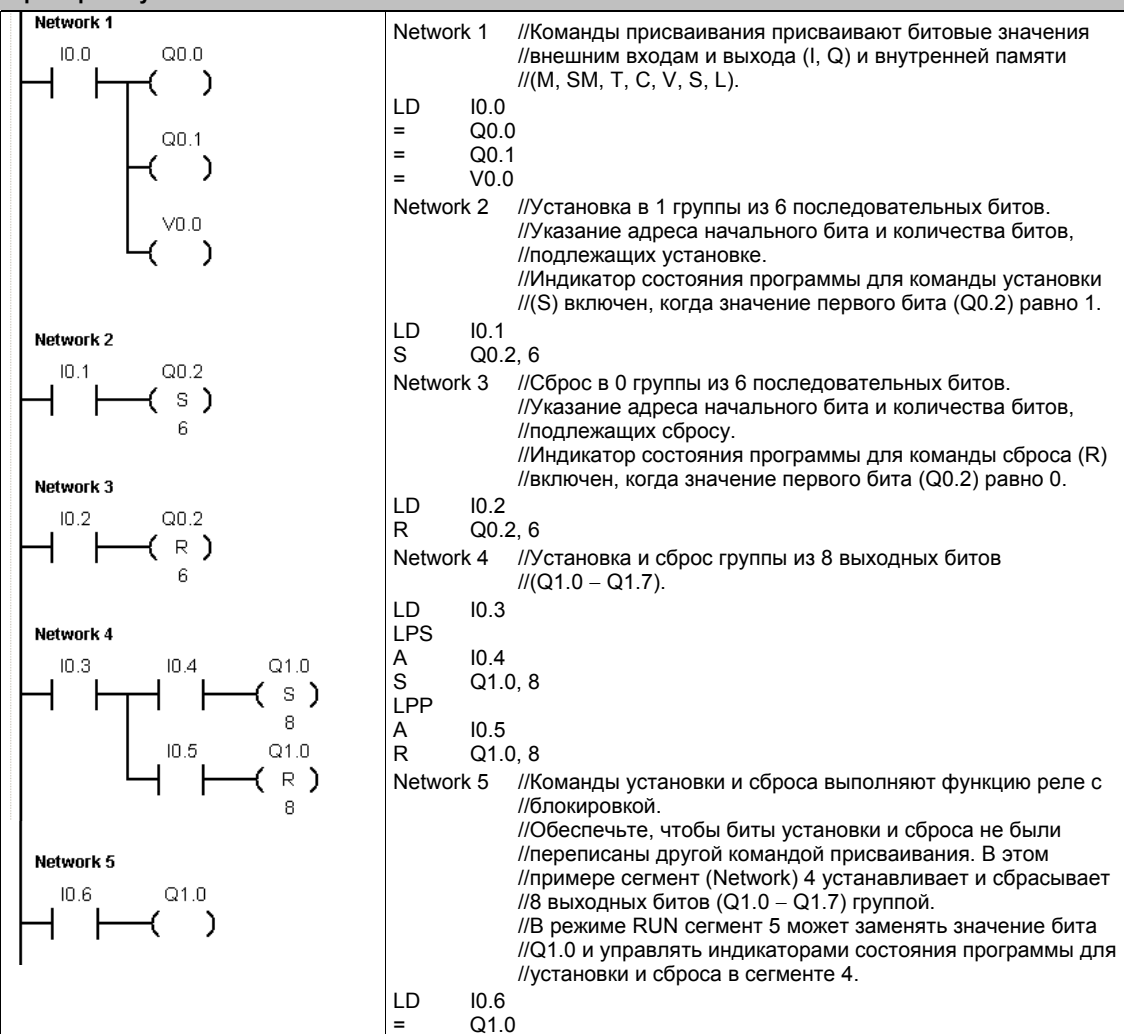


Таблица 6-4. Допустимые операнды для битовых логических операций

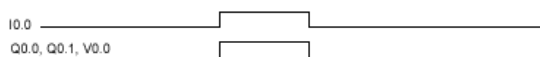
Входы/выходы	Тип данных	Операнды
Бит	BOOL	I, Q, V, M, SM, S, T, C, L
Бит (непосредственно управляемый)	BOOL	Q
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа

Пример: Катушки

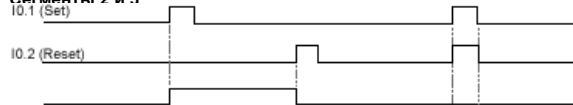


Временная диаграмма

Сегмент 1



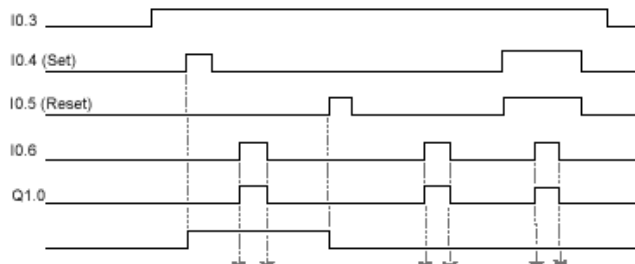
Сегменты 2 и 3



Set – установка
Reset - Сброс

Сброс в 0 перезаписывает установку в 1, так как программа выполняет сегмент 3, выполняющий сброс, после сегмента 2, выполняющего установку.

Сегменты 4 и 5

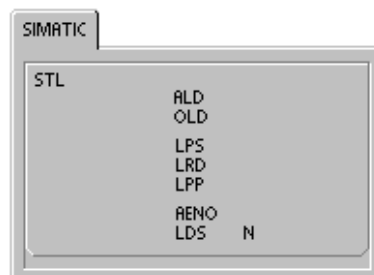


Команда присваивания (=) в сегменте 5 перезаписывает первый бит (Q1.0), устанавливаемый и сбрасываемый в сегменте 4, так как программа выполняет сегмент 5 последним.

Стековые операции

Логическое сопряжение 1-го и 2-го уровня по И

Команда логического сопряжения первого и второго уровня стека по И (ALD) логически сопрягает значения в первом и втором уровне стека, используя логическую операцию И. Результат загружается в вершину стека. После выполнения ALD глубина стека уменьшается на единицу.



Логическое сопряжение 1-го и 2-го уровня по ИЛИ

Команда логического сопряжения первого и второго уровня стека по ИЛИ (OLD) логически сопрягает значения в первом и втором уровне стека, используя логическую операцию ИЛИ. Результат загружается в вершину стека. После выполнения OLD глубина стека уменьшается на единицу.

Дублирование вершины логического стека

Команда дублирования вершины логического стека (LPS) дублирует значение вершины стека и помещает это значение в стек. Дно стека выталкивается и теряется.

Копирование второго уровня стека

Команда копирования второго уровня стека (LRD) копирует второй уровень стека в его вершину. В стек ничего не помещается и из него ничего не извлекается, но его вершина замещается копией.

Извлечение вершины стека

Команда извлечения вершины стека (LPP) извлекает одно значение из стека. Второй уровень становится новой вершиной стека.

Логическое И для бита ENO

Команда логического И для бита ENO (AENO) выполняет логическое сопряжение бита ENO и вершины стека с использованием операции И, производя такой же эффект, как и бит ENO в блоке LAD или FBD. Результатом операции И является новая вершина стека.

ENO – это булев выход для блоков в LAD и FBD. Если блок имеет поток сигнала на входе EN и выполняется без ошибок, то выход ENO передает поток сигнала следующему элементу. ENO может быть использован как деблокирующий бит, указывающий на успешное завершение команды. Бит ENO вместе с вершиной стека используется для воздействия на поток сигнала для выполнения последующих команд. У команд STL нет входа EN. Для выполнения условных операций вершина стека должна быть равна 1. В STL нет выхода ENO, но команды STL, соответствующие командам LAD и FBD с выходами ENO обязательно устанавливают специальный бит ENO. Доступ к этому биту осуществляется с помощью команды AENO.

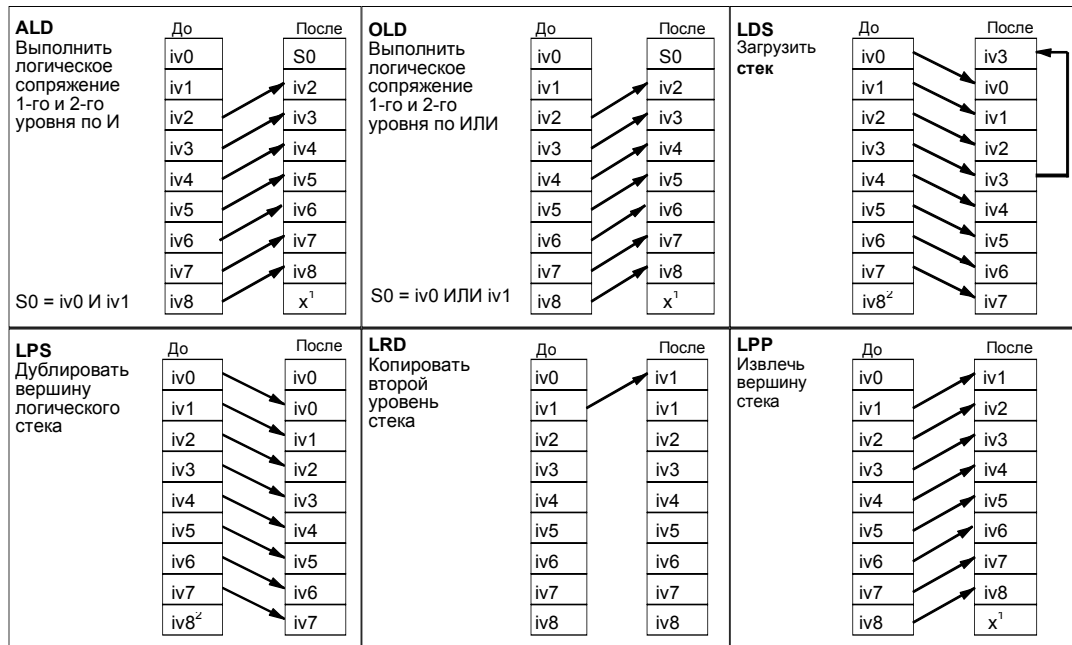
Загрузка стека

Команда загрузки стека (LDS) дублирует бит стека (N) и помещает это значение в вершину стека. Дно стека выталкивается и теряется.

Таблица 6–5. Допустимые операнды для команды загрузки стека

Входы/выходы	Тип данных	Операнды
N	BYTE	Константа (от 0 до 8)

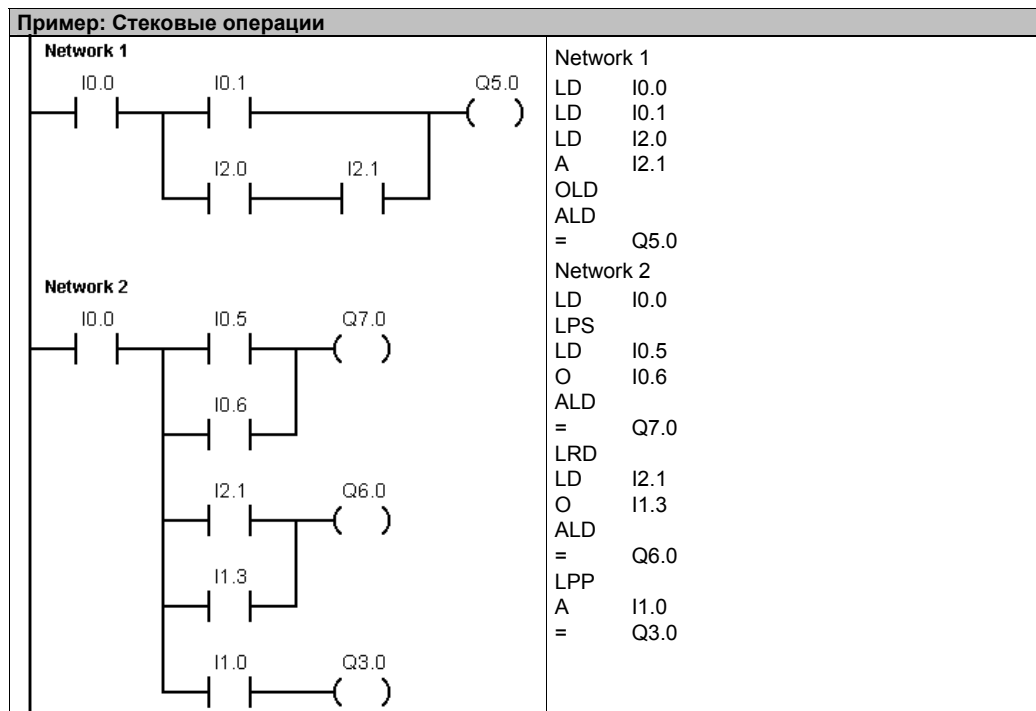
Как показано на рис. 6–3, S7–200 использует логический стек для решения задач логики управления. В этих примерах «iv0» «iv7» обозначают начальные значения логического стека, «nv» обозначает новое значение, поставляемое командой, а «S0» обозначает рассчитанное значение, сохраняемое в логическом стеке.



¹ Значение неизвестно (оно может быть равно 0 или

² После выполнения команды "Дублировать вершину логического стека" или "Загрузить стек" значение iv8 теряется.

Рис. 6–3. Принцип действия стековых операций



Функциональный блок с двумя устойчивыми состояниями: преимущество установки и преимущество сброса

Функциональный блок с двумя устойчивыми состояниями и преимуществом установки представляет собой триггер, у которого доминирует установка. Если сигнал установки (S1) и сигнал сброса (R) одновременно принимают значение истина, то выход (OUT) принимает значение истина.

Функциональный блок с двумя устойчивыми состояниями и преимуществом сброса представляет собой триггер, у которого доминирует сброс. Если сигнал установки (S) и сигнал сброса (R1) одновременно принимают значение истина, то выход (OUT) принимает значение ложь.

Параметр Bit представляет собой булев параметр, который устанавливается или сбрасывается. Дополнительный выход отражает сигнальное состояние параметра Bit.

В таблице 6–7 представлены состояния функциональных блоков для программы-примера.

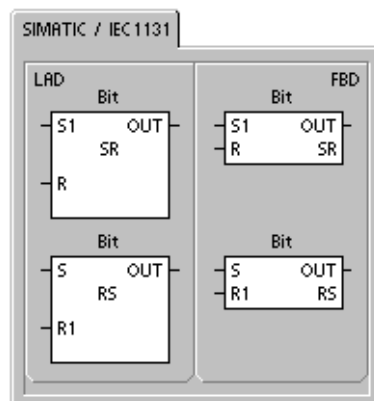


Таблица 6–6. Допустимые операнды для функционального блока с двумя устойчивыми состояниями

Входы/выходы	Типы данных	Операнды
S1, R	BOOL	I, Q, V, M, SM, S, T, C, поток сигнала
S, R1, OUT	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Bit	BOOL	I, Q, V, M, S

Пример: Функциональный блок с двумя устойчивыми состояниями: преимущество установки и преимущество сброса

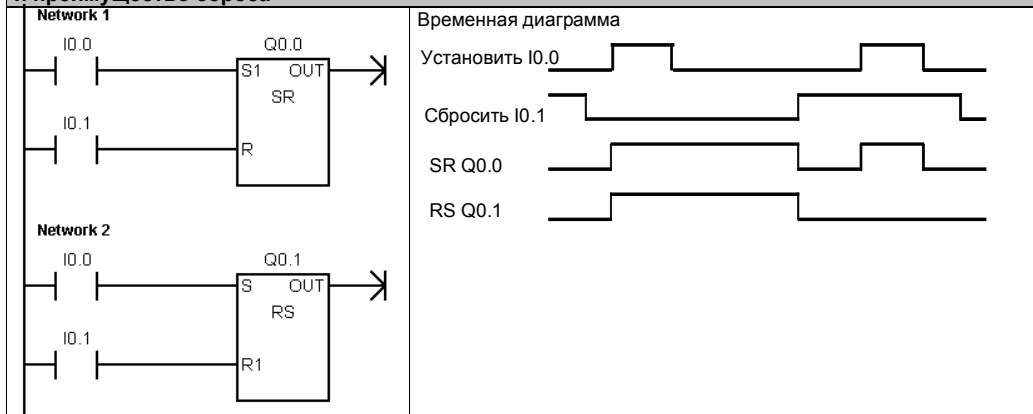


Таблица 6–7. Таблица состояний для функциональных блоков с двумя устойчивыми состояниями: преимущество установки и преимущество сброса

Команда	S1	R	Out (Bit)
Функциональный блок с двумя устойчивыми состояниями и преимуществом установки (SR)	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	1
Команда	S	R1	Out (Bit)
Функциональный блок с двумя устойчивыми состояниями и преимуществом сброса (RS)	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	0

Команды, связанные со временем

Чтение и установка часов реального времени

Команда чтения часов реального времени (TODR) считывает текущее время и дату из аппаратных часов и загружает их в 8-байтовый буфер времени, начиная с адреса T. Команда установки часов реального времени (TODW) записывает в аппаратные часы текущее время и дату, загруженные в 8-байтовый буфер, начинающийся с адреса, указанного параметром T.

Все значения даты и времени необходимо записывать в формате BCD (например, 16#97 для 1997 года). На рис. 6-4 показан формат 8-байтового буфера времени (T).

После продолжительного отключения питания или потери памяти часы реального времени (TOD) инициализируют следующую дату и время:

Дата: 01-Янв-90
 Время: 00:00:00
 День недели: Воскресенье

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0007 (ошибка даты TOD) *Только для установки часов реального времени*
- 000C (часы отсутствуют)

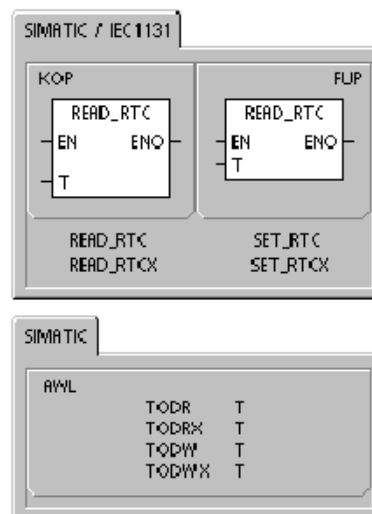
Таблица 6-8. Допустимые операнды для команд, связанных со временем

Входы/выходы	Типы данных	Операнды
T	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Год: 00 – 99	Месяц: 01 – 12	День: 01 – 31	Часы: 00 – 23	Минуты: 00 – 59	Секунды: 00 – 59	0	День недели: 0 – 7*

*T+7 1=воскресенье, 7=суббота
 0 деактивирует день недели.

Рис. 6-4. Формат 8-байтового буфера времени (T)



**Совет**

CPU S7-200 не проверяет соответствие дня недели дате. Могут быть восприняты неверные даты, например, 30 февраля. Правильность даты должны обеспечивать вы сами.

Не используйте команду TODR/TODW одновременно в главной программе и программе обработки прерывания. Эта команда не будет исполнена в программе обработки прерывания, которая пытается ее выполнить, когда действует другая команда TODR/TODW. Если делается попытка одновременно двух обращений к часам, то устанавливается SM4.3 (нефатальная ошибка 0007).

Часы реального времени в S7-200 используют только две младшие значащие цифры для года, так что 2000-й год представлен как 00. ПЛК S7-200 никак не использует информацию о годе. Однако пользовательские программы, которые используют арифметические операции или сравнения со значением года, должны учитывать двухзначное представление и изменение столетия.

Високосные годы обрабатываются правильно до 2096 года.

Чтение расширенных часов реального времени

Команда чтения расширенных часов реального времени (TODRX) считывает текущую конфигурацию времени, даты и летнего времени из ПЛК и загружает их в 19-байтовый буфер, начинающийся по адресу, указанному параметром T.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 000C (отсутствует модуль часов)
- 0091 (ошибка диапазона)

Установка расширенных часов реального времени

Команда установки расширенных часов реального времени (TODWX) записывает текущую конфигурацию времени, даты и летнего времени в ПЛК из 19-байтового буфера, находящегося по адресу, указанному в параметре T.

Все значения даты и времени необходимо записывать в формате BCD (например, 16#02 для 2002 года). В таблице 6-9 показан формат 19-байтового буфера времени (T).

После продолжительного отключения питания или потери памяти часы реального времени инициализируют следующую дату и время:

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0007 (ошибка данных TOD)
- 000C (отсутствует модуль часов)
- 0091 (ошибка диапазона)

Дата: 01-Jan-90
 Время: 00:00:00
 День недели: воскресенье

Таблица 6–9. Формат 19–байтового буфера времени (TI)

Т-байт	Описание	Данные, содержащиеся в байте
0	год (0-99)	текущий год (в формате BCD)
1	месяц (1-12)	текущий месяц (в формате BCD)
2	день (1-31)	текущий день (в формате BCD)
3	час (0-23)	текущий час (в формате BCD)
4	минута (0-59)	текущая минута (в формате BCD)
5	секунда (0-59)	текущая секунда (в формате BCD)
6	00	резерв – всегда устанавливается в 00
7	день недели (1-7)	текущий день недели, 1=воскресенье (в формате BCD)
8	режим (00H-03H, 08H, 10H-13H, FFH)	Режим корректировки: 00H = корректировка заблокирована 01H = Европа (смещение часового пояса от UTC = 0 часов) ¹ 02H = Европа (смещение часового пояса от UTC = +1 час) ¹ 03H = Европа (смещение часового пояса от UTC = +2 часа) ¹ 04H-07H = резерв 08H = Европа (смещение часового пояса от UTC = -1 час) ¹ 09H-0FH = резерв 10H = США ² 11H = Австралия ³ 12H = Австралия (Тасмания) ⁴ 13H = Новая Зеландия ⁵ 14H-FEH = резерв FFH = определяется пользователем (с использованием значений в байтах 9-18)
9	часы корректировки (0-23)	размер корректировки, часы (в формате BCD)
10	минуты корректировки (0-59)	размер корректировки, минуты (в формате BCD)
11	первый месяц (1-12)	первый месяц летнего времени (в формате BCD)
12	первый день (1-31)	первый день летнего времени (в формате BCD)
13	первый час (0-23)	первый час летнего времени (в формате BCD)
14	первая минута (0-59)	первая минута летнего времени (в формате BCD)
15	последний месяц (1-12)	последний месяц летнего времени (в формате BCD)
16	последний день (1-31)	последний день летнего времени (в формате BCD)
17	последний час (0-23)	последний час летнего времени (в формате BCD)
18	последняя минута (0-59)	последняя минута летнего времени (в формате BCD)

¹ Европейская конвенция: Переводить часы на один час вперед в последнее воскресенье марта в 1:00 час ночи UTC (Universal Time Coordinated = универсальное синхронизированное время). Переводить часы на один час назад в последнее воскресенье октября в 2:00 часа ночи UTC. (Местное время выполнения корректировки зависит от смещения часового пояса от UTC).

² Конвенция США: Переводить часы на один час вперед в первое воскресенье апреля в 2:00 часа ночи местного времени. Переводить часы на один час назад в последнее воскресенье октября в 2:00 часа ночи местного времени.

³ Австралийская конвенция: Переводить часы на один час вперед в последнее воскресенье октября в 2:00 часа ночи местного времени. Переводить часы на один час назад в последнее воскресенье марта в 3:00 часа ночи местного времени.

⁴ Австралийская конвенция (Тасмания): Переводить часы на один час вперед в первое воскресенье октября в 2:00 часа ночи местного времени. Переводить часы на один час назад в последнее воскресенье марта в 3:00 часа ночи местного времени

⁵ Новозеландская конвенция: Переводить часы на один час вперед в первое воскресенье октября в 2:00 часа ночи местного времени. Переводить часы на один час назад 15 марта, если этот день приходится на воскресенье, или в первое воскресенье после 15 марта в 3:00 часа ночи местного времени

Команды для обмена данными

Команды чтения из сети и записи через сеть

Команда чтения из сети (NETR) инициирует коммуникационную операцию для получения данных из удаленного устройства через указанный порт (PORT), как указано в таблице (TBL). Команда записи через сеть (NETW) инициирует коммуникационную операцию для записи данных в удаленное устройство через указанный порт (PORT), как указано в таблице (TBL).

Сбойные состояния, устанавливающие ENO = 0 :

- 0006 (косвенный адрес)
- Функция возвращает ошибку и устанавливает бит E бита состояния таблицы (см. рис. 6-5)

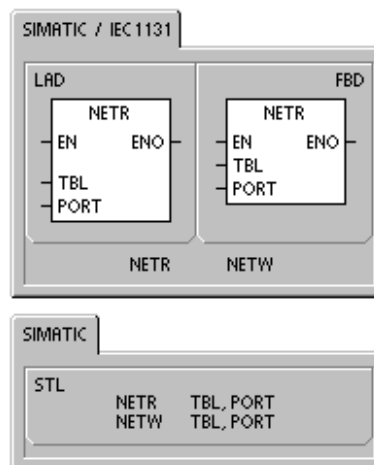
Команда NETR может прочитать до 16 байтов информации из удаленной станции, а команда NETW может записать до 16 байтов информации в удаленную станцию.

В программе можно иметь любое количество команд NETR/NETW, но одновременно можно активизировать не более восьми команд NETR и NETW. Например, в данном S7-200 в одно и то же время у вас могут активными четыре NETR и четыре NETW или два NETR и шесть NETW.

С помощью мастера команд для операций чтения из сети и записи через сеть вы можете создать счетчик. Для запуска этого мастера команд выберите команду меню **Tools > Instruction Wizard [Инструменты > Мастер команд]**, а затем в окне мастера команд выберите опцию Network Read/Network Write [Чтение из сети/Запись через сеть].

Таблица 6-10. Допустимые операнды для команд чтения из сети и записи через сеть

Входы/выходы	Тип данных	Операнды
TBL	BYTE	VB, MB, *VD, *LD, *AC
PORT	BYTE	константа для CPU 221, CPU 222, CPU 224: 0 для CPU 224XP, CPU 226: 0 или 1



Мастер команд

На рис. 6–5 описана таблица, к которой обращается параметр TBL, а в таблице 6–11 приведены коды ошибок.

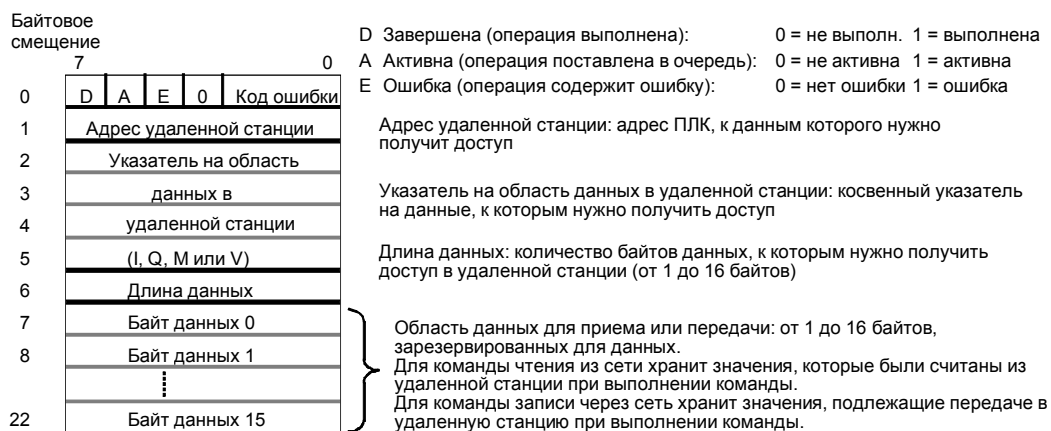


Рис. 6–5. Параметр TBL для команд чтения из сети и записи через сеть

Таблица 6–11. Коды ошибок для параметра TBL

Код	Определение
0	Нет ошибки.
1	Ошибка истечения времени ожидания: удаленная станция не отвечает.
2	Ошибка приема: ошибка четности, кадрирования или контрольной суммы в ответе.
3	Ошибка с переходом в автономный режим: конфликт, вызванный двойным адресом станции или неисправностью аппаратуры.
4	Ошибка переполнения очереди: активизировано более 8 команд NETR/NETW.
5	Нарушение протокола: попытка выполнить команду NETR или NETW, без разблокировки привилегированного режима PPI в SMB30 или SMB130.
6	Недопустимый параметр: параметр TBL содержит недопустимое или ошибочное значение.
7	Нет ресурса: удаленная станция занята (Идет процесс загрузки в станцию или из станции.)
8	Ошибка 7-го уровня: нарушение протокола приложения
9	Ошибка сообщения: неверный адрес данных или неправильная длина данных
A – F	Не используются. (Зарезервированы)

На рис. 6–6 показан пример, иллюстрирующий использование команд NETR и NETW. В этом примере речь идет о производственной линии, где стаканчики заполняются маслом и передаются одной из четырех упаковочных машин. Упаковочная машина пакует по восемь стаканчиков с маслом в одну картонную коробку. Направляющее устройство управляет потоком стаканчиков с маслом, направляя их к той или иной упаковочной машине. Для управления упаковочными машинами используются четыре модуля S7–200, а для управления направляющим устройством используется один S7–200, оборудованный интерфейсом оператора TD 200.

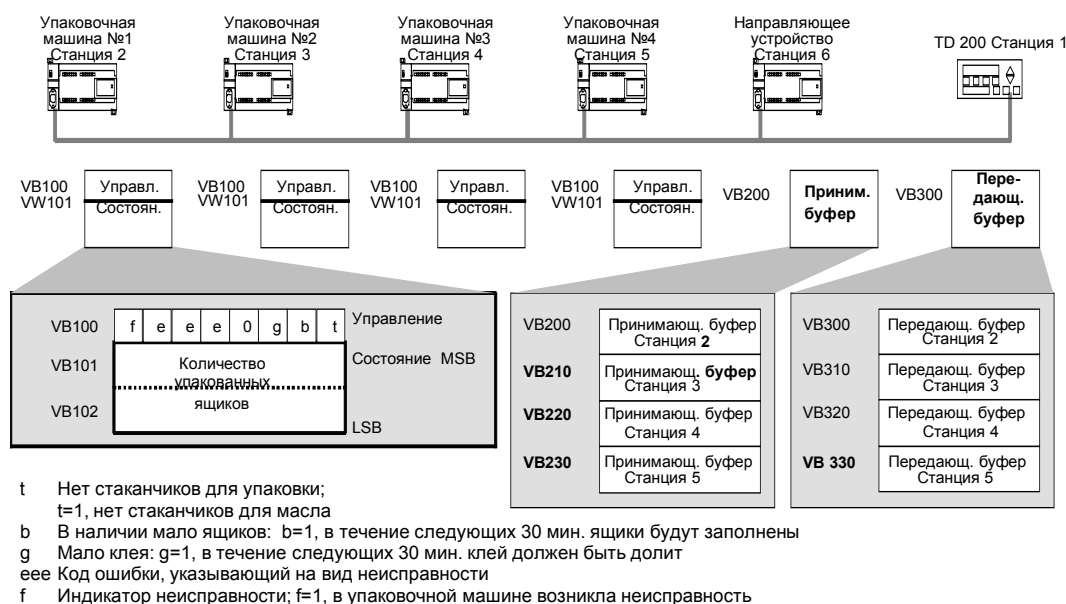
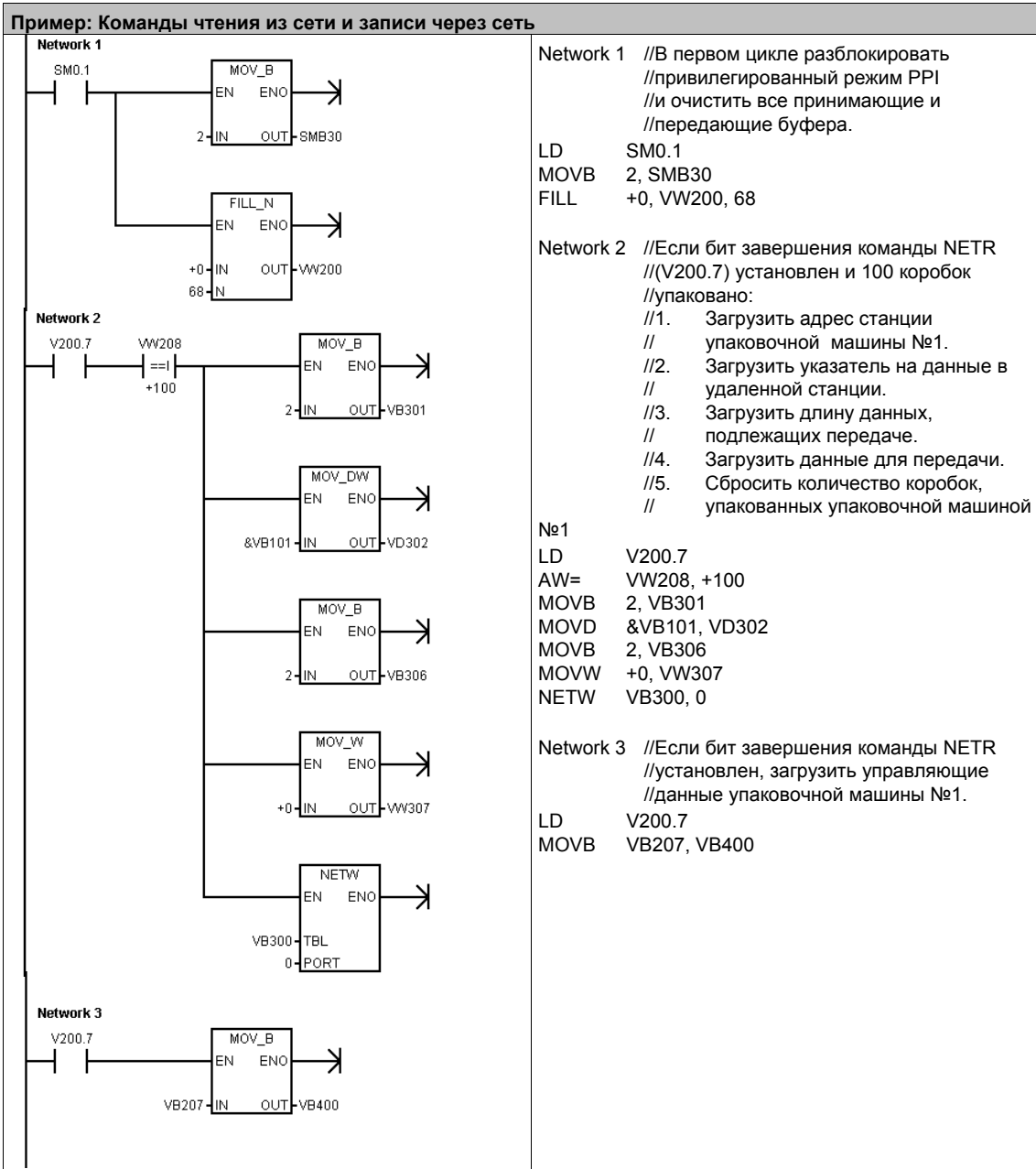


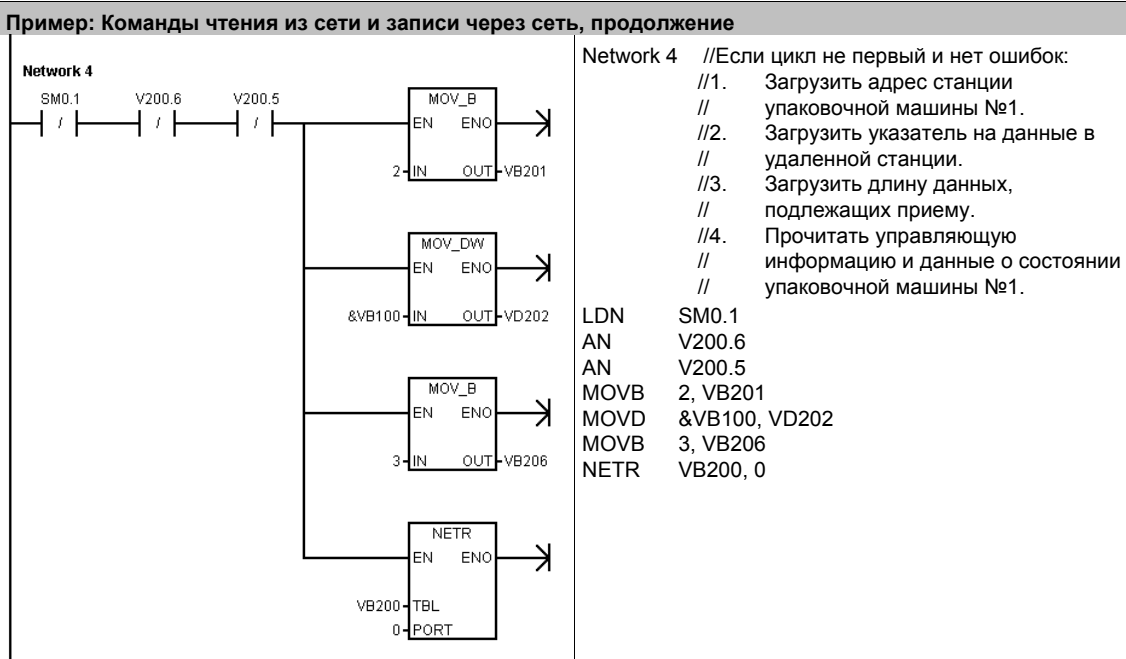
Рис. 6–6. Пример команд чтения из сети и записи через сеть

На рис. 6–7 показан принимающий буфер (VB200) и передающий буфер (VB300) для доступа к данным в станции 2. S7-200 использует команду NETR для регулярного чтения управляющей информации и информации о состоянии из каждой упаковочной машины. Каждый раз, когда упаковочная машина упаковывает 100 коробок, направляющее устройство замечает это и с помощью команды NETW передает сообщение для сброса слова состояния.

Принимающий буфер для чтения из упаковочной машины №1					Передающий буфер для сброса счетчика упаковочной машины №1						
	7			0		7			0		
VB200	D	A	E	0	Код ошибки	VB300	D	A	E	0	Код ошибки
VB201	Адрес удаленной станции = 2					VB301	Адрес удаленной станции = 2				
VB202	Указатель на					VB302	Указатель на				
VB203	область данных					VB303	область данных				
VB204	в					VB304	в				
VB205	удаленной станции = (&VB100)					VB305	удаленной станции = (&VB101)				
VB206	Длина данных = 3 байта					VB306	Длина данных = 2 байта				
VB207	Control					VB307	0				
VB208	Состояние (старший байт)					VB308	0				
VB209	Состояние (младший байт)										

Рис. 6–7. Образец данных TBL для примера команд чтения из сети и записи через сеть





Команды передачи и приема (свободно программируемый обмен данными)

Команда передачи (XMT) используется в режиме свободно программируемого обмена данными для передачи данных через коммуникационный порт(ы).

Команда приема (RCV) инициирует или завершает функцию приема сообщения. Чтобы блок приема мог работать, вы должны указать условия начала и конца сообщения. Сообщения, получаемые через указанный порт (PORT), сохраняются в буфере данных (TBL). Первая запись в буфере данных указывает количество принятых байтов.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0009 (одновременная передача и прием через порт 0)
- 000B (одновременная передача и прием через порт 1)
- Ошибка параметра приема устанавливает SM86.6 или SM186.6
- S7-200 CPU не находится в режиме свободно программируемого обмена данными

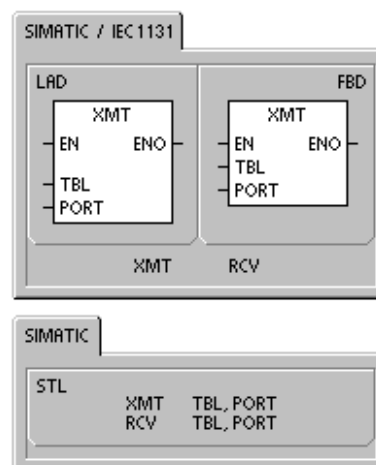


Таблица 6–12. Допустимые операнды для команд передачи и приема

Входы/выходы	Тип данных	Операнды
TBL	BYTE	IB, QB, VB, MB, SMB, SB, *VD, *LD, *AC
PORT	BYTE	константа для CPU 221, CPU 222, CPU 224: 0 для CPU 224XP, CPU 226: 0 или 1

Дополнительную информацию об использовании режима свободно программируемого обмена данными вы найдете в разделе "Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными" на стр. 226 в главе 7.

Использование режима свободно программируемого обмена данными для управления последовательным коммуникационным портом

Режим свободно программируемого обмена данными можно выбрать для управления последовательным коммуникационным портом S7-200 с помощью программы пользователя. Если вы выбираете режим свободно программируемого обмена данными, то ваша программа управляет работой коммуникационного порта путем использования прерываний приема, прерываний передачи, команды передачи и команды приема. В режиме свободно программируемого обмена данными протокол связи полностью управляется программой, представленной в виде цепной логической схемы (LAD). Для выбора скорости передачи и способа контроля четности используются SMB30 (для порта 0) и SMB130 (для порта 1, если ваш S7-200 имеет два порта).

Режим свободно программируемого обмена данными блокируется, и восстанавливается обычная связь (например, доступ через устройство программирования), когда S7-200 находится в состоянии STOP.

В простейшем случае вы можете послать сообщение на принтер или дисплей, используя только команду передачи (XMT). К другим примерам относятся связь с устройством для считывания штрихового кода, весами или сварочным аппаратом. В каждом случае вы должны написать программу для поддержки протокола, используемого устройством, с которым S7-200 поддерживает связь при нахождении в режиме свободно программируемого обмена данными.

Свободно программируемый обмен данными возможен только тогда, когда S7-200 находится в режиме RUN. Разблокируйте режим свободно программируемого обмена данными установкой значения 01 в поле выбора протокола SMB30 (порт 0) или SMB130 (порт 1). В режиме свободно программируемого обмена данными связь с устройством программирования невозможна.



Совет

Режимом свободно программируемого обмена данными можно управлять с помощью бита специальной памяти SM0.7, который отражает текущее положение переключателя режимов работы. Если SM0.7 равен 0, то переключатель находится в положении TERM; если SM0.7 = 1, то переключатель режимов работы находится в положении RUN. Если вы разблокируете режим свободно программируемого обмена данными только тогда, когда переключатель находится в положении RUN, то вы можете использовать устройство программирования для контроля и управления работой S7-200 путем перевода переключателя в любое другое положение.

Переход от обмена данными через PPI к свободно программируемому обмену данными

SMB30 и SMB130 конфигурируют коммуникационные порты 0 и 1 соответственно для свободно программируемого обмена данными, предоставляя возможность выбора скорости передачи, способа контроля четности и количества битов данных. На рис. 6–8 описан управляющий байт для свободно программируемого обмена данными. Для всех конфигураций генерируется один стоповый бит.

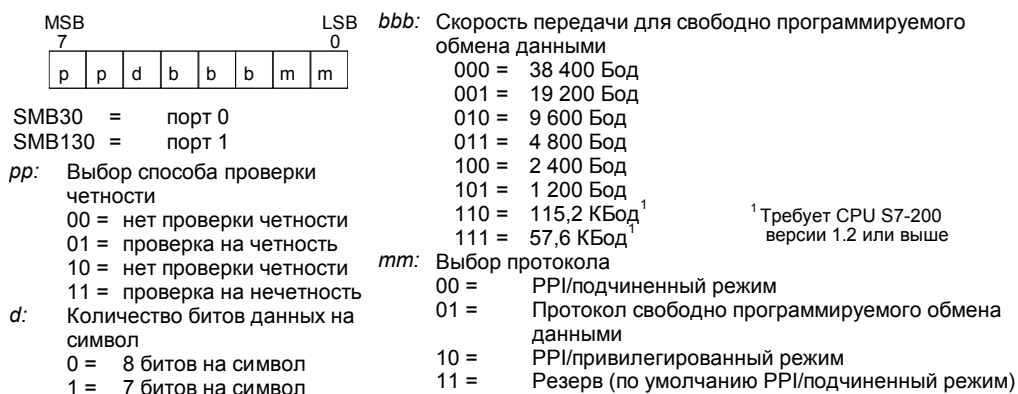


Рис. 6–8. Управляющий байт для свободно программируемого обмена данными (SMB30 или SMB130)

Передача данных

Команда передачи дает возможность передать буфер, состоящий из одного или нескольких символов (не более 255).

На рис. 6–9 показан формат буфера передачи.

После того как передан последний символ буфера, S7-200 генерирует прерывание (прерывающее событие 9 для порта 0 и прерывающее событие 26 для порта 1), если событию «Передача завершена» поставлена в соответствие программа обработки прерывания.

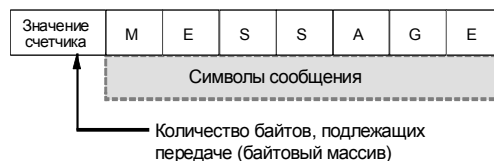


Рис. 6–9. Формат буфера передачи

Вы можете производить передачу и без использования прерываний (например, посылая сообщение на принтер), контролируя SM4.5 или SM4.6, сигнализирующих о завершении передачи.

Команда передачи может быть использована для генерирования условия паузы (BREAK) путем установки количества символов в ноль, а затем исполнения команды передачи. Это генерирует условие паузы в строке для передачи 16 битов с текущей скоростью передачи. Передача паузы обрабатывается так же, как передача любого другого сообщения, причем при завершении передачи паузы генерируется прерывание, а SM4.5 или SM4.6 отображают текущее состояние передачи.

Прием данных

Команда приема дает возможность принять буфер, состоящий из одного или нескольких символов (не более 255).

На рис. 6–10 показан формат буфера приема.

После того как принят последний символ буфера, генерируется прерывание (прерывающее событие 23 для порта 0 и прерывающее событие 24 для порта 1), если событию «Прием сообщения завершен» поставлена в соответствие программа обработки прерывания.

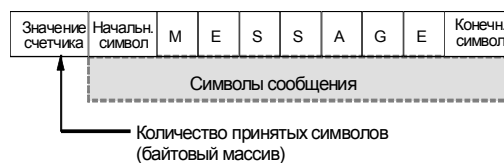


Рис. 6–10. Формат буфера приема

Вы можете принимать сообщения и без использования прерываний, контролируя SMB86 (порт 0) или SMB186 (порт 1). Этот байт не равен нулю, если приема не активен или завершен. Он равен нулю, когда происходит прием.

Как показано в таблице 6–12, команда приема позволяет выбирать условия начала и окончания сообщения с помощью SMB86 – SMB94 для порта 0 и SMB186 – SMB194 для порта 1.



	<p>Совет</p> <p>Функция приема сообщений автоматически завершается при превышении количества символов или ошибке четности. Вы должны определить условие начала и условие конца (максимальное количество символов), чтобы функция приема сообщений могла работать.</p>
---	--

Таблица 6–13. Байты буфера приема (от SMB86 до SMB94 и от SM1B86 до SMB194)

Порт 0	Порт 1	Описание				
SMB86	SMB186	<p>Байт состояния приема сообщения</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB 7</td> <td style="text-align: center;">LSB 0</td> </tr> <tr> <td style="text-align: center;">n</td> <td style="text-align: center;">p</td> </tr> </table> <p>n: 1 = Функция приема сообщений завершена: пользователь дал команду на деактивизацию. r: 1 = Функция приема сообщений завершена: ошибка во входном параметре или отсутствует условие начала или конца e: 1 = Принят символ конца t: 1 = Функция приема сообщений завершена: истекло время c: 1 = Функция приема сообщений завершена: достигнуто максимальное количество символов p: 1 = Функция приема сообщений завершена: ошибка четности</p>	MSB 7	LSB 0	n	p
MSB 7	LSB 0					
n	p					
SMB87	SMB187	<p>Байт управления приемом сообщения</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB 7</td> <td style="text-align: center;">LSB 0</td> </tr> <tr> <td style="text-align: center;">en</td> <td style="text-align: center;">0</td> </tr> </table> <p>en: 0 = функция приема сообщений заблокирована. 1 = функция приема сообщений разблокирована. Бит разблокировки/блокировки приема сообщений проверяется при каждом исполнении команды RCV. sc: 0 = игнорировать SMB88 или SMB188. 1 = использовать значение SMB88 или SMB188 для обнаружения начала сообщения. ec: 0 = игнорировать SMB89 или SMB189. 1 = использовать значение SMB89 или SMB189 для обнаружения конца сообщения. il: 0 = игнорировать SMW90 или SMW190. 1 = использовать значение SMW90 или SMW190 для обнаружения бездействия линии c/m: 0 = таймер измеряет время между символами 1 = таймер измеряет время сообщения. tmr: 0 = игнорировать SMW92 или SMW192. 1 = завершить прием, если превышен интервал времени, указанный в SMW92 или SMW192. bk: 0 = игнорировать условия паузы 1 = использовать условие паузы как начало обнаружения сообщения.</p>	MSB 7	LSB 0	en	0
MSB 7	LSB 0					
en	0					
SMB88	SMB188	Символ начала сообщения.				
SMB89	SMB189	Символ конца сообщения.				
SMW90	SMW190	Время бездействия линии в миллисекундах. Первый символ, принятый по истечении времени бездействия линии, является началом нового сообщения.				
SMW92	SMW192	Значение контроля времени при измерении времени между символами и времени сообщения в миллисекундах. Если этот интервал времени истек, то прием сообщения завершается.				
SMB94	SMB194	Максимальное количество символов, которое должно быть принято (от 1 до 255 байтов). Этот диапазон должен быть установлен на ожидаемый максимальный размер буфера, даже если завершение сообщения с помощью подсчета символов не используется.				

Условия начала и окончания сообщения для команды приема

Для определения условий начала и окончания сообщения команда приема использует биты байта управления приемом сообщений (SMB87 или SMB187).

	<p>Совет</p> <p>Если во время выполнения команды приема коммуникационный порт занят другими устройствами, то функция приема сообщений может начать прием символа в середине этого символа, что может привести к ошибке при контроле четности и завершению приема сообщения. Если проверка четности не активизирована, то принятое сообщение может содержать неправильные символы. Эта ситуация может возникнуть, если в качестве начального условия назначен определенный начальный символ или любой символ, как это описано ниже в пунктах 2 и 6.</p> <p>Команда приема поддерживает несколько условий начала сообщения. Если указанное вами условие начала приема содержит паузу или распознавание бездействия линия, то описанной выше проблемы удастся избежать, принуждая функцию приема сообщений синхронизировать начало сообщения с началом символа, прежде чем символ будет помещен в буфер сообщений.</p>
---	--

Команда приема поддерживает несколько условий начала сообщения:

1. *Распознавание бездействия линии:* Условие бездействия линии определяется как время простоя или перерыва в работе линии передачи. Прием начинается, если линия связи находилась в простое или имела перерыв в работе в течение некоторого количества миллисекунд, указанного в SMW90 или SMW190. Когда в вашей программе выполняется команда приема, то функция приема сообщений иницирует поиск условия бездействия линии. Если прием каких-либо символов происходит до истечения времени простоя линии, то функция приема сообщений игнорирует эти символы и вновь запускает таймер простоя линии с временем, полученным из SMW90 или SMW190. См. рис. 6–11. По истечении времени простоя линии функция приема сообщений сохраняет все последующие полученные символы в буфере сообщений.

Время простоя линии всегда должно быть больше, чем время, необходимое для передачи одного символа (начальный бит, биты данных, биты контроля четности и завершения) при заданной скорости передачи. Типичным значением для времени простоя линии является время, необходимое для передачи трех символов при заданной скорости передачи.

Распознавание бездействия линии используется в качестве начального условия для двоичных протоколов, для протоколов, не имеющих определенного начального символа, или в случае, когда протокол требует минимального времени между сообщениями.

Настройка: $il = 1, sc = 0, bk = 0, SMW90/SMW190 = \text{простой линии в мс}$

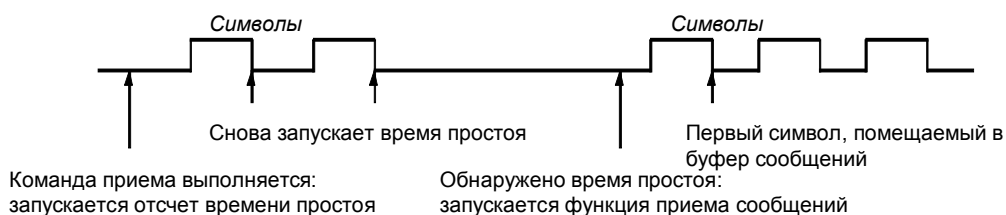


Рис. 6–11. Запуск команды приема с распознаванием бездействия линии

2. *Распознавание начального символа:* начальный символ – это любой символ, используемый в качестве первого символа сообщения. Сообщение начинается, когда принимается начальный символ, указанный в SMB88 или SMB188. Функция приема сообщений хранит начальный символ в буфере приема в качестве первого символа сообщения. Функция приема сообщений игнорирует все символы, принятые ранее начального символа. Начальный символ и все принятые после него символы сохраняются в буфере сообщения.

Обычно распознавание начального символа используется в протоколах ASCII, в которых все сообщения начинаются с одного и того же символа.

Настройка: $il = 0, sc = 1, bk = 0, SMW90/SMW190 =$ не имеют значения,
 $SMB88/SMB188 =$ начальный символ

3. *Простой линии и начальный символ:* Команда приема может начинать сообщение комбинацией простоя линии и начального символа. При выполнении команды приема функция приема сообщений ищет условие простоя линии. После нахождения условия простоя линия функция приема сообщений ожидает заданного начального символа. Если принимается любой символ, отличный от начального, функция приема сообщений снова начинает поиск условия простоя линии. Все символы, принятые до обнаружения условия простоя линии и до приема начального символа игнорируются. Начальный символ помещается в буфер сообщения вместе со всеми последующими символами.

Время простоя линии всегда должно быть больше, чем время, необходимое для передачи одного символа (начальный бит, биты данных, биты контроля четности и завершения) при заданной скорости передачи. Типичным значением для времени простоя линии является время, необходимое для передачи трех символов при заданной скорости передачи.

Обычно этот вид начальных условий используется с протоколом, который задает минимальное и максимальное время между сообщениями, и первым символом сообщения является адрес или что-либо иное, определяющее конкретное устройство. Это особенно полезно, если реализуется протокол, в котором в обмене данными участвует несколько устройств. В этом случае команда приема запускает прерывание только в том случае, когда сообщение принимается для конкретного адреса или конкретного устройства, указанного начальным символом.

Настройка: $il = 1, sc = 1, bk = 0, SMW90/SMW190 > 0, SMB88/SMB188 =$
 начальный символ

4. *Обнаружение паузы:* Признаком паузы является прием данных с нулевым значением в течение времени, превышающего длительность передачи полного символа. Время передачи полного символа определяется как общее время для передачи начального бита, битов данных и битов контроля четности и завершения. Если команда приема была сконфигурирована так, что началом сообщения является прием условия паузы, то все символы, принятые после обнаружения условия паузы, помещаются в буфер сообщения. Все символы, принятые до обнаружения условия паузы, игнорируются.

Обычно обнаружение паузы используется в качестве начального условия только в том случае, если этого требует протокол.

Настройка: $il = 0, sc = 0, bk = 1, SMW90/SMW190 =$ не имеют значения,
 $SMB88/SMB188 =$ не имеют значения

5. *Пауза и начальный символ:* Команда приема может быть сконфигурирована так, чтобы начинать прием символов после обнаружения условия паузы, за которым следует начальный символ. После обнаружения условия паузы функция приема сообщений ищет заданный начальный символ. Если принимается любой символ, кроме начального, то функция приема сообщений снова начинает искать условие паузы. Все символы, принятые до обнаружения условия паузы и до получения начального символа, игнорируются. Начальный символ помещается в буфер сообщений вместе со всеми последующими символами.

Настройка: $il = 0, sc = 1, bk = 1, SMW90/SMW190 =$ не имеют значения,
 $SMB88/SMB188 =$ начальный символ

6. *Любой символ*: Команда приема может быть сконфигурирована так, чтобы немедленно принимать все символы и помещать их в буфер сообщения. Это особый случай обнаружения бездействия линии. В этом случае время простоя линии (SMW90 или SMW190) устанавливается в ноль. Это заставляет команду приема начинать прием символов немедленно после начала исполнения.

Настройка: $il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = \text{не имеют значения}$

Если в качестве начала сообщения используется любой символ, то можно использовать таймер сообщений для контроля времени приема сообщения. Это полезно в случаях, когда для реализации основной части протокола используется свободно программируемый обмен данными, и требуется контроль времени на случай, если в течение определенного промежутка времени не будет получено ответа от slave-устройства. Таймер сообщения запускается, когда выполняется команда приема, так как время простоя линии было установлено в ноль. Таймер сообщения заканчивает отсчет времени и завершает функцию приема сообщений, если не выполнено какое-нибудь другое условие окончания приема.

Настройка: $il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = \text{не имеют значения}, c/m = 1, tmr = 1, SMW92 = \text{выдержка времени таймера в мс}$

Команда приема поддерживает несколько способов завершения сообщения. Сообщение может быть завершено одним или комбинацией нескольких из следующих условий:

1. *Обнаружение конечного символа*: Конечный символ – это любой символ, используемый для обозначения конца сообщения. После обнаружения начального условия команда приема проверяет каждый полученный символ на совпадение с конечным символом. Если получен конечный символ, то он помещается в буфер сообщения, и прием завершается.

Обычно обнаружение конечного символа используется в протоколах ASCII, где каждое сообщение завершается определенным символом. Для завершения сообщения обнаружение конечного символа можно использовать в сочетании с межсимвольным таймером, таймером сообщения или обнаружением достижения максимального количества символов.

Настройка: $ec = 1, SMB89/SMB189 = \text{конечный символ}$

2. *Межсимвольный таймер*: Время между символами – это промежуток времени от конца одного символа (стопового бита) до конца следующего символа (стопового бита). Если время между символами (включая второй символ) превышает количество миллисекунд, указанное в SMW92 или SMW192, то функция приема сообщений завершает работу. Межсимвольный таймер перезапускается после приема каждого символа. См. рис. 6–12.

Межсимвольный таймер можно использовать для завершения сообщения в протоколах, не имеющих специального символа конца сообщения. Этот таймер должен быть установлен на значение, большее, чем время передачи одного символа при выбранной скорости передачи, так как этот таймер всегда включает в себя время, необходимое для получения полного символа (начальный бит, биты данных, биты контроля четности и завершения).

Для завершения сообщения межсимвольный таймер можно использовать в сочетании с обнаружением конечного символа и обнаружением достижения максимального количества символов.

Настройка: $c/m = 0, tmr = 1, SMW92/SMW192 = \text{выдержка времени в мс}$

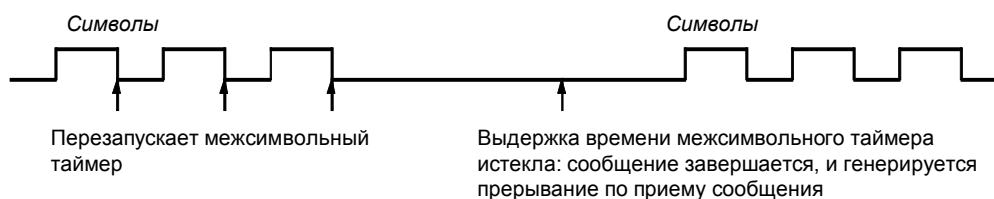


Рис. 6–12. Использование межсимвольного таймера для завершения команды приема

3. **Таймер сообщения:** Таймер сообщения завершает сообщение через определенное время после его начала. Таймер сообщения запускается, как только выполнено начальное условие или условия для функции приема сообщений. Таймер сообщения завершает отсчет времени по истечении количества миллисекунд, указанных в SMW92 или SMW192. См. рис. 6–13.

Обычно таймер сообщения используется, если устройства связи не могут гарантировать, что между символами не будет временных разрывов, или при работе через модемы. Для модемов можно использовать таймер сообщения, чтобы задать максимально допустимое время для приема сообщения после его начала. Типичным значением для таймера сообщения является полуторакратное время, необходимое для приема наиболее длинного возможного сообщения при выбранной скорости передачи.

Для завершения сообщения таймер сообщения можно использовать в сочетании с обнаружением конечного символа и обнаружением достижения максимального количества символов.

Настройка: $c/m = 1$, $tmr = 1$, SMW92/SMW192 = выдержка времени в мс

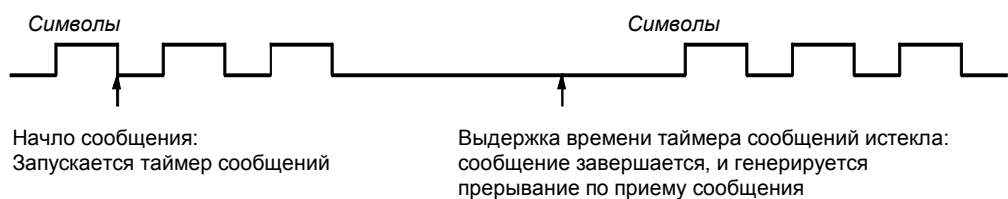


Рис. 6–13. Использование таймера сообщений для завершения команды приема

4. **Значение для наибольшего числа символов:** Команде приема можно сообщить максимальное количество символов, которое должно быть принято (SMB94 или SMB194). Если это значение достигнуто или превышено, то функция приема сообщений завершает работу. Команда приема требует, чтобы пользователь определил максимальное количество символов, даже если оно специально не используется в качестве условия завершения. Это необходимо потому, что команде приема необходимо знать максимальный размер принимаемого сообщения, чтобы данные пользователя, помещенные после буфера сообщений, не были перезаписаны.
Максимальное количество символов может использоваться для завершения сообщений в протоколах, в которых длина сообщений известна и всегда одинакова. Максимальное количество символов всегда используется в сочетании с обнаружением конечного символа, межсимвольного таймера или таймера сообщения.
5. **Ошибки, обнаруженные при проверке четности:** Функция приема автоматически завершает работу, если аппаратура сообщает об ошибке четности в принятом символе. Ошибки при проверке четности возможны только в том случае, если в SMB30 или SMB130 активизирована проверка четности. Эту функцию невозможно выключить.
6. **Завершение по инициативе пользователя:** Программа пользователя может завершить функцию приема сообщений, выполняя другую функцию приема сообщений и установив в ноль бит деблокировки (EN) в SMB87 или SMB187. Это приводит к немедленному завершению функции приема сообщений.

Прием данных, управляемый прерываниями

Для достижения полной гибкости в поддержке протоколов вы также можете принимать данные под управлением прерываний. Каждый принимаемый символ генерирует прерывание. Принятый символ помещается в SMB2, а результат контроля четности (если активизирован) помещается в SM3.0 непосредственно перед исполнением программы обработки прерывания, назначенной событию «Символ принят». SMB2 – это буфер для принятого символа при свободно программируемом обмене данными. Каждый символ, принятый в режиме свободно программируемого обмена данными, помещается по этому адресу для облегчения доступа к нему из программы пользователя. SMB3 используется для режима свободно программируемого обмена данными и содержит бит ошибки четности, который устанавливается, когда в принятом символе обнаруживается ошибка четности. Все остальные биты этого байта зарезервированы. Используйте этот бит для отклонения сообщения или для генерирования отрицательного квитирования этого сообщения.

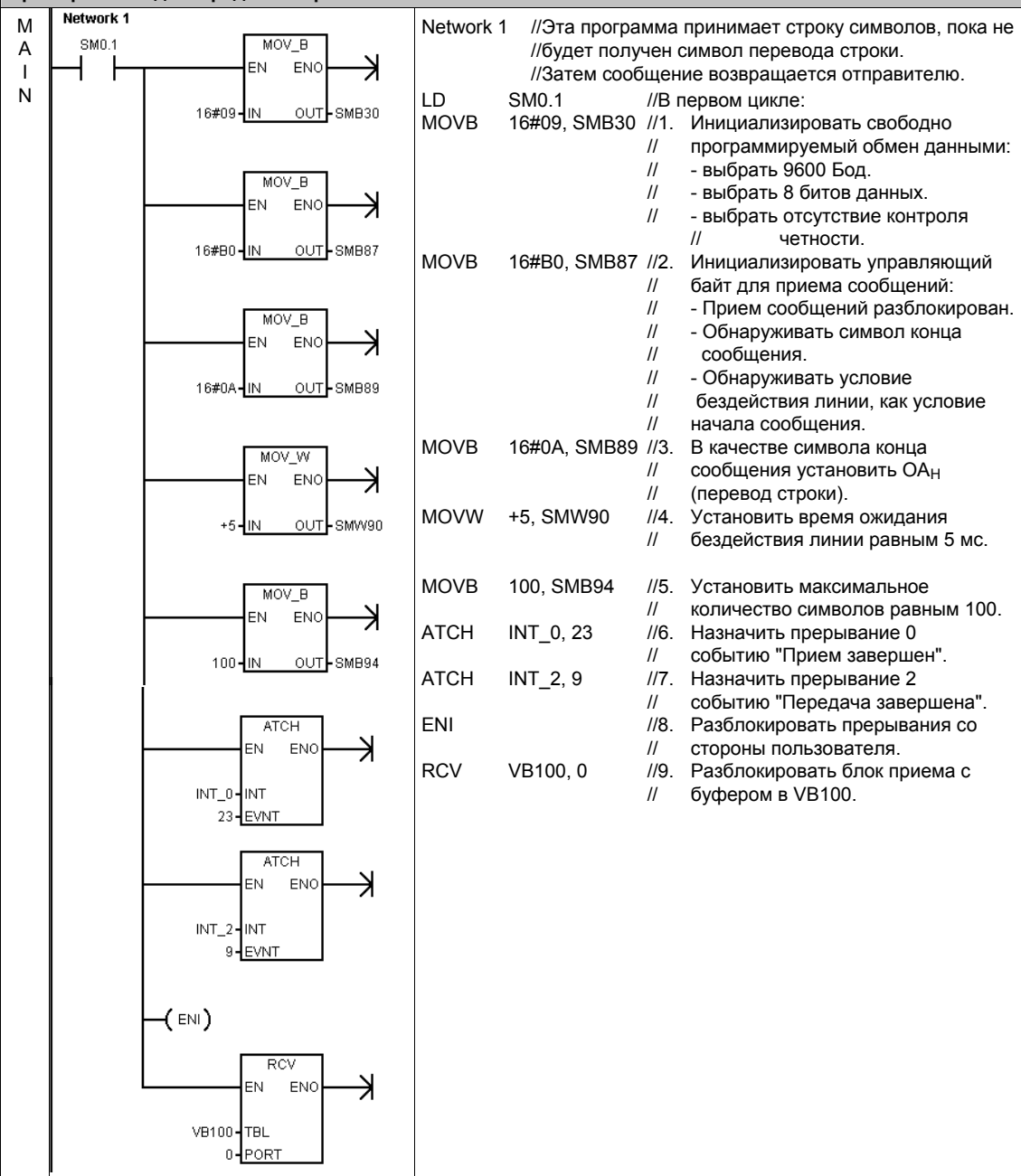
Если управление с помощью прерываний используется при высоких скоростях передачи (от 38,4 до 115,2 Кбод), то время между прерываниями очень мало. Например, прерывание для скорости 38,4 Кбод составляет 260 микросекунд, для скорости 57,6 Кбод – 173 микросекунды, а для 115,2 Кбод – 86 микросекунд. Во избежание потери символов обеспечьте краткость программ обработки прерываний или используйте команду приема.



Совет

SMB2 и SMB3 совместно используются портами 0 и 1. Когда прием символа в порт 0 приводит к исполнению программы обработки прерывания, назначенной этому событию (прерывающее событие 8), SMB2 содержит символ, принятый портом 0, а SMB3 содержит результат контроля четности этого символа. Когда прием символа в порт 1 приводит к исполнению программы обработки прерывания, назначенной этому событию (прерывающее событие 25), SMB2 содержит символ, принятый портом 1, а SMB3 содержит результат контроля четности этого символа.

Пример: Команды передачи и приема



Пример: Команды передачи и приема, продолжение		
I N T 0	<p>Network 1</p>	<p>Network 1 //Программа обработки прерываний для //завершения приема: //1. Если статус приема показывает, что принят // конечный символ, то назначить // 10-миллисекундный таймер для запуска // передачи и возврата. //2. Если прием завершен по любой другой // причине, то начать новый прием.</p> <p>LDB= SMB86, 16#20 MOVB 10, SMB34 ATCH INT_1, 10 CRETI NOT RCV VB100, 0</p>
I N T 1	<p>Network 1</p>	<p>Network 1 //Прерывание, управляемое 10-миллисекундным //таймером: //1. Отсоединить прерывание, управляемое //таймером. //2. Передать сообщение обратно пользователю.</p> <p>LD SM0.0 DTCH 10 XMT VB100, 0</p>
I N T 2	<p>Network 1</p>	<p>Network 1 //Прерывание при завершении передачи: //Разблокировать новый прием.</p> <p>LD SM0.0 RCV VB100, 0</p>

Команды получения и установки адреса порта

Команда получения адреса порта (GPA) считывает адрес станции из порта CPU S7-200, указанного в параметре PORT, и помещает значение по адресу, указанному в ADDR.

Команда установки адреса порта (SPA) устанавливает адрес порта станции (PORT) на значение, указанное в ADDR. Новый адрес не сохраняется постоянно. После нового пуска соответствующий порт возвращается к последнему адресу (к тому, который был загружен системным блоком).

Сбойные состояния, устанавливающие ENO = 0 :

- 0006 (косвенный адрес)
- 0004 (попытка выполнить команду установки адреса порта в программе обработки прерываний)

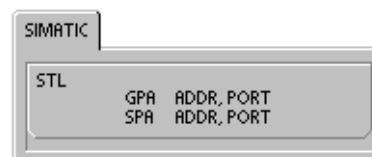
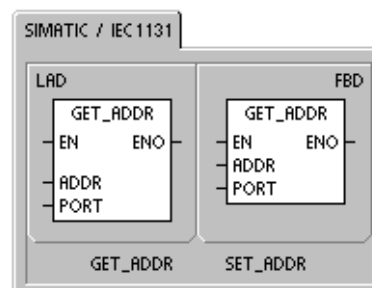


Таблица 6–14. Допустимые операнды для команд получения и установки адреса порта

Входы/выходы	Тип данных	Операнды
ADDR	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа (Постоянное значение действительно только для команды установки адреса порта.)
PORT	BYTE	константа для CPU 221, CPU 222, CPU 224: 0 для CPU 224XP, CPU 226: 0 или 1

Команды сравнения

Сравнение числовых величин

Команды сравнения используются для сравнения двух величин:

IN1 = IN2 IN1 >= IN2 IN1 <= IN2
 IN1 > IN2 IN1 < IN2 IN1 <> IN2

Операции сравнения байтов не учитывают знака.
 Операции сравнения целых учитывают знак.
 Операции сравнения двойных слов учитывают знак.
 Операции сравнения вещественных чисел учитывают знак.

Для LAD и FBD: Если сравнение истинно, команда сравнения включает контакт (LAD) или выход (FBD).

Для STL: Если сравнение истинно, то команда сравнения загружает «1» в вершину стека или выполняет логическое сопряжение значения «1» со значением в вершине стека в соответствии с таблицей истинности для И или ИЛИ (STL).

Если вы работаете с командами сравнения IEC, то вы можете использовать на входах различные типы данных. Однако значения на обоих входах должны быть одного типа.

Указание

При следующих условиях речь идет о фатальных ошибках, которые приводят к тому, что S7-200 немедленно останавливает обработку вашей программы:

- Обнаружен недопустимый косвенный адрес (в любой команде сравнения)
- Обнаружено недопустимое вещественное число (например, NAN) (сравнение вещественных чисел)

Во избежание появления таких состояний обратите внимание на правильную инициализацию всех указателей и величин, содержащих вещественные числа, перед выполнением команды сравнения.

Команды сравнения выполняются независимо от состояния потока сигнала.

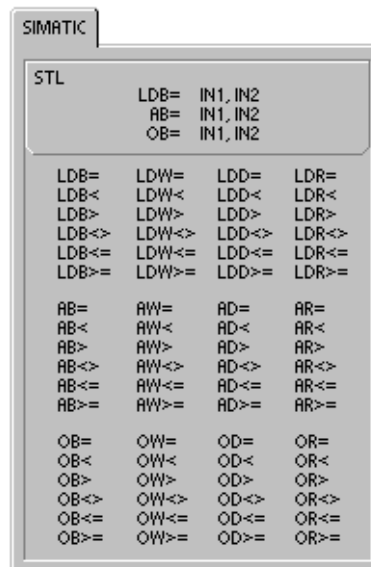
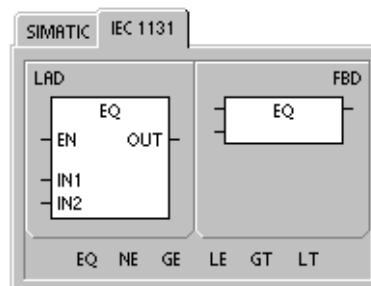
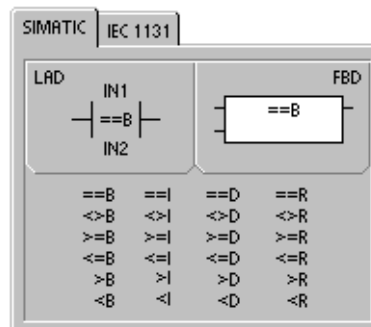


Таблица 6–15. Допустимые операнды для команд сравнения

Входы/выходы	Тип	Операнды
IN1, IN2	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
Выход (или OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала

Пример: Команды сравнения

<p>Network 1</p>	<p>Network 1 //Повернуть аналоговый потенциометр 0, чтобы //изменить значение байта SMB28. //Q0.0 активен, если значение SMB28 меньше //или равно 50. //Q0.1 активен, если значение SMB28 больше //или равно 150. //Индикатор состояния включен, если //сравнение истинно.</p>
<p>Network 2</p>	<p>LD I0.0 LPS AB<= SMB28, 50 = Q0.0 LPP AB>= SMB28, 150 = Q0.1</p>
<p>Network 3</p>	<p>Network 2 //Загрузить адреса из памяти переменных с //небольшими значениями, чтобы сравнения //были ложными и индикаторы состояния //выключились.</p> <p>LD I0.1 MOVW -30000, VW0 MOVD -200000000, VD2 MOVR 1.012E-006, VD6</p>
<p>Network 4</p>	<p>Network 3 //Загрузить адреса из памяти переменных с //большими значениями, чтобы сравнения были //истинными и индикаторы состояния //включились.</p> <p>LD I0.2 MOVW +30000, VW0 MOVD -100000000, VD2 MOVR 3.141593, VD6</p>
	<p>Network 4 //Сравнение целых чисел проверяет, истинно // ли, что VW0 > +10000. //с помощью констант программы показываются //различные типы данных. Вы можете также // сравнить две величины, хранящиеся в //программируемой памяти, напр., VW0 ></p> <p>VW100 LD I0.3 LPS AW> VW0, +10000 = Q0.2 LRD AD< -150000000, VD2 = Q0.3 LPP AR> VD6, 5.001E-006 = Q0.4</p>

Сравнение строк

Команда сравнения строк сравнивает между собой две строки символов ASCII:

IN1 = IN2 IN1 <> IN2

Если сравнение истинно, команда сравнения включает контакт (LAD) или выход (FBD) или загружает «1» в вершину стека или выполняет логическое сопряжение значения «1» со значением в вершине стека в соответствии с таблицей истинности для И или ИЛИ (STL).

Указание

При следующих условиях речь идет о фатальных ошибках, которые приводят к тому, что S7-200 немедленно останавливает обработку вашей программы:

- Обнаружен недопустимый косвенный адрес (в любой команде сравнения)
- Встретилась строка, содержащая более 254 символов (команда сравнения строк)
- Строка, начальный адрес которой и длина таковы, что она не помещается в указанной области памяти (команда сравнения строк)

Во избежание появления таких состояний обратите внимание на правильную инициализацию всех указателей и адресов в памяти, которые должны содержать строки символов ASCII, перед выполнением команды сравнения, использующей эти значения. Обеспечьте, чтобы буфер, зарезервированный для строки символов ASCII, полностью помещался в указанной области памяти.

Команды сравнения выполняются независимо от состояния потока сигнала.

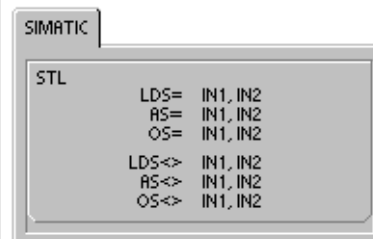
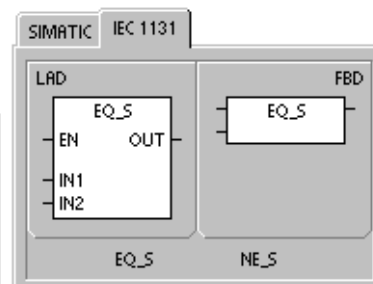
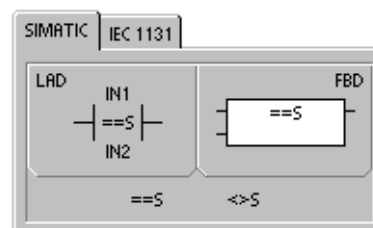


Таблица 6–16. Допустимые операнды для команд сравнения строк

Входы/выходы	Тип	Операнды
IN1	STRING	VB, LB, *VD, *LD, *AC, константа
IN2	STRING	VB, LB, *VD, *LD, *AC
Выход (OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала

Команды преобразования

Стандартные команды преобразования

Команды преобразования чисел

Команды преобразования байта в целое (BTI), целого в байт (ITB), целого в двойное целое (ITD), двойного целого в целое (DTI), двойного целого в вещественное (DTR), двоично-десятичного в целое (BCDI) и целого в двоично-десятичное (IBCD) преобразуют входную величину IN в указанный формат и сохраняют выходное значение по адресу, указанному в OUT. Например, вы можете преобразовать двойное целое в вещественное число. Вы можете также осуществлять преобразования между целым двоично-десятичным (BCD) форматом.

Округление и округление отбрасыванием

Команда округления (ROUND) преобразует вещественное число IN в двойное целое число и помещает округленный результат в переменную, указанную в OUT.

Команда округления отбрасыванием дробной части (TRUNC) преобразует вещественное число IN в двойное целое число и помещает целую часть результата в переменную, указанную в OUT.

Генерирование комбинации двоичных разрядов для семисегментного индикатора

Команда генерирования комбинации двоичных разрядов для семисегментного индикатора (SEG) дает возможность создать битовую комбинацию, с помощью которой высвечиваются сегменты семисегментного индикатора.

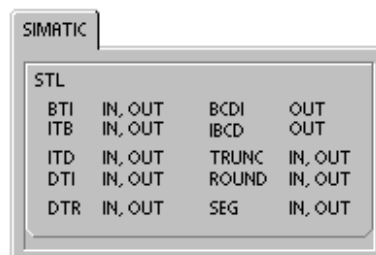
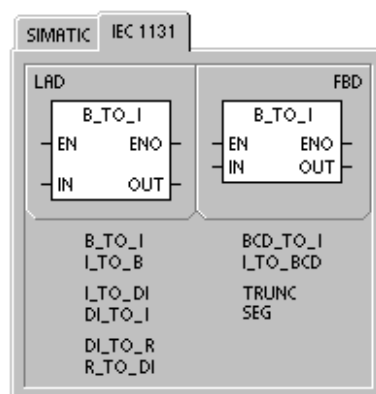
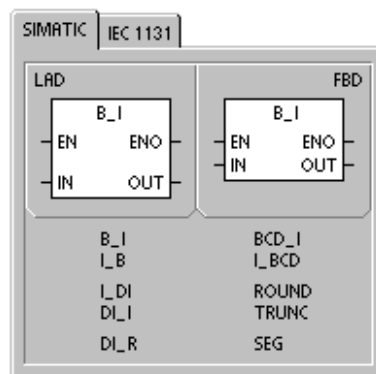


Таблица 6–17. Допустимые операнды для стандартных команд преобразования

Входы/выходы	Тип данных	Операнды
IN	BYTE WORD, INT DINT REAL	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, AC, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
OUT	BYTE WORD, INT DINT, REAL	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Принцип действия команды преобразования BCD в целое и целого в BCD

<p>Команда преобразования BCD в целое число (BCDI) преобразует входное число, представленное в двоично-десятичном формате, IN в целое число и загружает результат в переменную, указанную в OUT. Допустимый диапазон для IN: от 0 до 9999 (BCD).</p> <p>Команда преобразования целого числа в BCD (IBCD) преобразует входное целое число IN в двоично-десятичный формат и загружает результат в переменную, указанную в OUT. Допустимый диапазон для IN: от 0 до 9999 (целое).</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ SM1.6 (недопустимое значение BCD) ■ 0006 (косвенный адрес) <p>Биты специальной памяти (SM), на которые оказывается воздействие:</p> <ul style="list-style-type: none"> ■ SM1.6 (недопустимое значение BCD)
--	--

Принцип действия команды преобразования двойного целого в вещественное

<p>Команда преобразования двойного целого числа в вещественное (DTR) преобразует 32-битовое целое со знаком IN в 32-битовое вещественное число и помещает результат в переменную, указанную в OUT.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес)
--	---

Принцип действия команды преобразования двойного целого числа в целое

<p>Команда преобразования двойного целого числа в целое (DTI) преобразует двойное целое число IN в целое число и помещает результат в переменную, указанную в OUT.</p> <p>Если преобразованное вами значение слишком велико, чтобы быть представленным на выходе, то устанавливается бит переполнения, а выход не изменяется.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) <p>Биты специальной памяти, на которые действует команда:</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение)
---	---

Принцип действия команды преобразования целого числа в двойное целое

<p>Команда преобразования целого числа в двойное целое (ITD) преобразует целое число IN в двойное целое число и помещает результат в переменную, указанную в OUT. Знак распространяется.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес)
--	---

Принцип действия команды преобразования байта в целое число

<p>Команда преобразования байта в целое число (BTI) преобразует байт IN в целое число и помещает результат в переменную, указанную в OUT. Байт не имеет знака, поэтому распространение знака не происходит.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес)
---	---

Принцип действия команды преобразования целого числа в байт

<p>Команда преобразования целого числа в байт (ITB) преобразует слово IN в байт и помещает результат в переменную, указанную в OUT. Преобразуются значения от 0 до 255. Все остальные значения приводят к переполнению и не влияют на выход.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) <p>Биты специальной памяти, на которые действует команда:</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение)
--	---



Совет

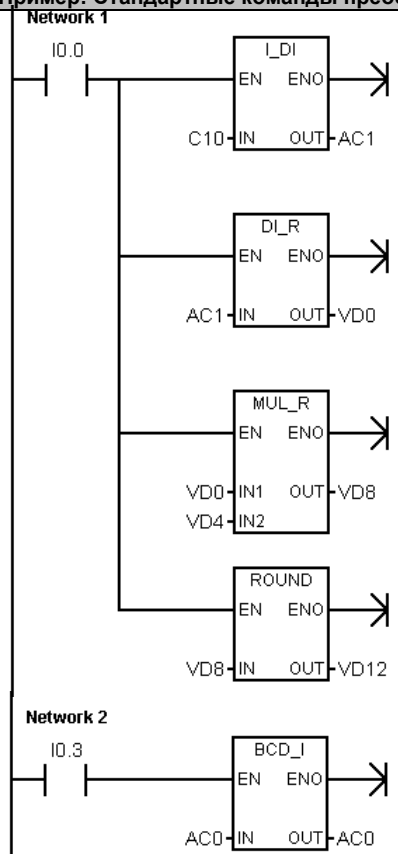
Для преобразования целого числа в вещественное используйте команду преобразования целого числа в двойное целое, а затем команду преобразования двойного целого числа в вещественное.

Принцип действия команд округления и округления отбрасыванием

<p>Команда округления (ROUND) преобразует вещественное число IN в двойное целое число и помещает результат в переменную, указанную в OUT. Если дробная часть равна 0,5 или больше, то число округляется в большую сторону.</p> <p>Команда округления отбрасыванием (TRUNC) преобразует вещественное число IN в двойное целое и помещает результат в переменную, указанную в OUT. Преобразуется только целая часть вещественного числа, а дробная часть отбрасывается.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) <p>Биты специальной памяти, на которые действует команда:</p> <ul style="list-style-type: none"> ■ SM1.1 (переполнение)
---	---

Если преобразуемое вами значение не является допустимым вещественным числом или слишком велико, чтобы быть представленным на выходе, то устанавливается бит переполнения, а выход не меняется.

Пример: Стандартные команды преобразования



```

Network 1 //Преобразовать дюймы в сантиметры:
//1. Загрузить значение счетчика (дюймы) в
// аккумулятор 1.
//2. Преобразовать это значение в
// вещественное число.
//3. Умножить на 2,54 (преобразовать в
// сантиметры).
//4. Преобразовать значение обратно в целое
// число.
LD I0.0
ITD C10, AC1
DTR AC1, VD0
MOVR VD0, VD8
*R VD4, VD8
ROUND VD8, VD12

Network 2 //Преобразовать BCD в целое число
LD I0.3
BCDI AC0
                    
```

Преобразование двойного целого в вещественное и округление		BCD в целое
C10	<input type="text" value="101"/>	Значение счетчика = 101 дюйм
VD0	<input type="text" value="101.0"/>	Значение счетчика (как вещественное число)
VD4	<input type="text" value="2.54"/>	Константа 2,54 (дюймы в сантиметры)
VD8	<input type="text" value="256.54"/>	256,54 сантиметра как вещественное число
VD12	<input type="text" value="257"/>	257 сантиметров как двойное целое
AC0	<input type="text" value="1234"/>	BCDI
AC0	<input type="text" value="04D2"/>	

Принцип действия команды генерирования комбинации двоичных разрядов для семисегментного индикатора

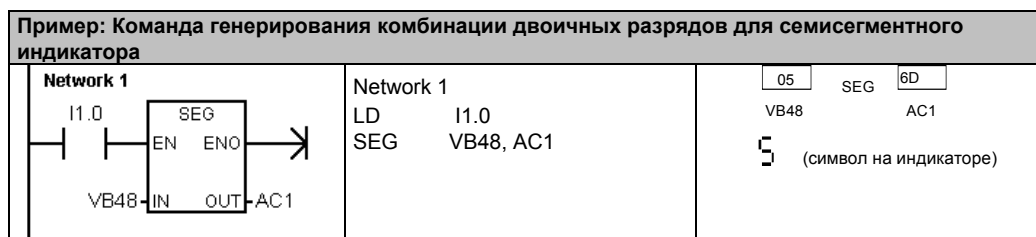
Для высвечивания сегментов семисегментного индикатора команда генерирования комбинации двоичных разрядов для семисегментного индикатора (SEG) преобразует символ (байт), указанный в IN, чтобы сгенерировать битовую комбинацию (байт) по адресу, указанному в OUT.

Высвечиваемые сегменты представляют собой символ в младшей цифре входного байта. На рис. 6–14 показано кодирование семисегментного дисплея, используемое командой SEG.	Сбойные состояния, устанавливающие ENO = 0 ■ 0006 (косвенный адрес)
--	---

(IN) LSD	Отображение сегментов	(OUT)						
		-	g	f	e	d	c	b
0	0	0	0	1	1	1	1	1
1	1	0	0	0	0	0	0	1
2	2	0	1	0	1	1	0	1
3	3	0	1	0	0	1	1	1
4	4	0	1	1	0	0	1	1
5	5	0	1	1	0	1	1	0
6	6	0	1	1	1	1	0	1
7	7	0	0	0	0	0	0	1

(IN) LSD	Отображение сегментов	(OUT)						
		-	g	f	e	d	c	b
8	8	0	1	1	1	1	1	1
9	9	0	1	1	0	0	1	1
A	A	0	1	1	1	0	0	1
B	B	0	1	1	1	1	1	0
C	C	0	0	1	1	1	0	0
D	D	0	1	0	1	1	1	0
E	E	0	1	1	1	1	0	0
F	F	0	1	1	1	0	0	0

Рис. 6–14. Кодирование семисегментного индикатора



Команды преобразования ASCII

Допустимые символы ASCII представляются шестнадцатеричными числами от 30 до 39 и от 41 до 46.

Преобразования между ASCII и шестнадцатеричными числами

Команда преобразования строки символов ASCII в 16-ричное число (ATH) преобразует строку символов ASCII длиной LEN, начинающуюся с адреса IN, в шестнадцатеричные цифры, начиная с адреса OUT. Команда преобразования 16-ричного числа в строку символов ASCII (HTA) преобразует шестнадцатеричные цифры, начинающиеся с входного байта IN, в строку символов ASCII, которая начинается по адресу OUT. Количество шестнадцатеричных цифр, подлежащих преобразованию, задается длиной LEN.

Максимальное количество шестнадцатеричных цифр, которое может быть преобразовано, равно 255.

Допустимыми входными символами ASCII являются алфавитно-цифровые символы от 0 до 9 с шестнадцатеричными значениями от 30 до 39 и большие буквы от A до F с шестнадцатеричными значениями от 41 до 46.

Сбойные состояния, устанавливающие ENO = 0

- SM1.7 (недопустимый код ASCII) *только для ASCII в 16-ричный код*
- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

Биты специальной памяти, на которые действует команда:

- SM1.7 (недопустимый код ASCII)

Преобразование числовых значений в ASCII

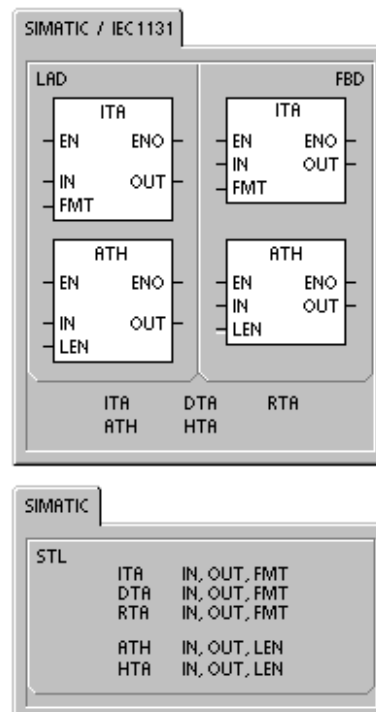
Команды преобразования целого числа в строку символов ASCII (ITA), преобразования двойного целого в строку символов ASCII (DTA) и преобразования вещественного числа в строку символов ASCII (RTA) преобразуют целое, двойное целое или вещественное число в символы ASCII.

Таблица 6–18. Допустимые операнды для команд преобразования ASCII

Входы/выходы	Тип данных	Операнды
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
LEN, FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

Принцип действия команды преобразования целого числа в строку символов ASCII

Команда преобразования целого числа в строку символов ASCII (ITA) преобразует целое число, содержащееся в слове IN, в массив символов ASCII. Операнд FMT (формат) определяет точность преобразования справа от десятичной точки, а также форму представления разделителя целой и дробной части – в виде запятой или точки. Результат преобразования помещается в 8 последовательных байтов, начиная с адреса OUT.



Массив символов ASCII всегда содержит 8 элементов.

На рис. 6–15 описан операнд формата для команды преобразования целого числа в строку символов ASCII. Размер выходного буфера всегда равен 8 байтам. Количество цифр справа от десятичной точки в выходном буфере определяется полем *nnn*. Допустимый диапазон поля *nnn* составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Для значений *nnn*, больших 5, выходной буфер заполняется пробелами ASCII. Бит *c* определяет использование запятой (*c*=1) или десятичной точки (*c*=0) в качестве разделителя целой и дробной части. Старшие 4 бита формата должны быть нулями.

На рис. 6–15 показаны примеры чисел, форматированных с помощью десятичной точки (*c* = 0), с тремя цифрами справа от десятичной точки (*nnn*=011). Выходной буфер форматировается в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Числа в выходном буфере выравниваются вправо.

FMT

MSB	7	6	5	4	3	2	1	0	LSB
	0	0	0	0	c	n	n	n	

c = запятая (1) или десятичная точка (0)
nnn = цифры справа от десятичной точки

in=12

in=-123

in=1234

in = -12345

Out	Out	Out	Out	Out	Out	Out	Out
+1	+2	+3	+4	+5	+6	+7	
		0	.	0	1	2	
	-	0	.	1	2	3	
		1	.	2	3	4	
-	1	2	.	3	4	5	

Рис. 6–15. Операнд FMT для команды преобразования целого в строку символов ASCII (ITA)

Принцип действия команды преобразования двойного целого в строку символов ASCII

Команда преобразования двойного целого в строку символов ASCII (DTA) преобразует двойное слово IN в массив символов ASCII. Операнд формат FMT определяет точность преобразования справа от десятичной точки. Результат преобразования помещается в 12 последовательных байтов, начиная с адреса OUT.	Сбойные состояния, устанавливающие ENO = 0 <ul style="list-style-type: none"> ■ 0006 (косвенный адрес) ■ Недопустимый формат ■ <i>nnn</i> > 5
--	--

Размер выходного буфера всегда равен 12 байтам.

На рис. 6–16 описан операнд формата для команды преобразования двойного целого в строку символов ASCII. Количество цифр справа от десятичной точки в выходном буфере определяется полем *nnn*. Допустимый диапазон поля *nnn* составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Для значений *nnn*, больших 5, выходной буфер заполняется символами пробела ASCII. Бит *c* определяет использование запятой (*c*=1) или десятичной точки (*c*=0) в качестве разделителя целой и дробной части. Старшие 4 бита формата должны быть нулями.

На рис. 6–16 показаны примеры чисел, форматированных с помощью десятичной точки (*c* = 0), с четырьмя цифрами справа от десятичной точки (*nnn*=100). Выходной буфер форматировается в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Числа в выходном буфере выравниваются вправо.

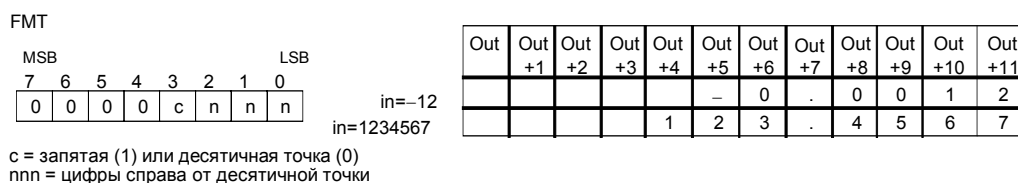


Рис. 6-16. Операнд FMT для команды преобразования целого числа в строку символов ASCII (DTA)

Принцип действия команды преобразования вещественного числа в строку символов ASCII

<p>Команда преобразования вещественного числа в строку символов ASCII (RTA) преобразует число с плавающей точкой IN в символы ASCII. Формат FMT определяет точность преобразования справа от десятичной точки, а также форму представления разделителя целой и дробной части – в виде запятой или точки и размер выходного буфера.</p> <p>Результат преобразования помещается в выходном буфере, начиная с адреса OUT.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес) ■ nnn > 5 ■ ssss < 3 ■ ssss < число символов в OUT
--	--

Длина (количество символов) результирующей строки символов ASCII соответствует размеру выходного буфера и может быть задана в диапазоне от 3 до 15 символов или байтов.

Формат чисел с плавающей точкой, используемый S7-200, поддерживает не более 7 значащих цифр. Попытка отобразить более 7 значащих цифр приводит к ошибке округления.

На рис. 6-17 описан операнд формата (FMT) для команды RTA. Размер выходного буфера определяется полем ssss. Размеры 0, 1 или 2 байта недопустимы. Количество цифр справа от десятичной точки в выходном буфере определяется полем nnn. Допустимый диапазон поля nnn составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Выходной буфер заполняется символами пробела ASCII для значений nnn, больших 5, или если заданный выходной буфер слишком мал для хранения преобразованного значения. Бит с определяет использование запятой (с=1) или десятичной точки (с=0) в качестве разделителя целой и дробной части.

На рис. 6-17 также приведены примеры чисел, форматированных с помощью десятичной точки (с = 0), с одной цифрой справа от десятичной точки (nnn=001) и размером буфера, равным 6 байтам (ssss=0110). Выходной буфер форматруется в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Значение справа от десятичной точки округляется в соответствии с заданным количеством цифр справа от десятичной точки.
- Размер выходного буфера должен, по крайней мере, на три байта превышать количество цифр справа от десятичной точки.
- Числа в выходном буфере выравниваются вправо.

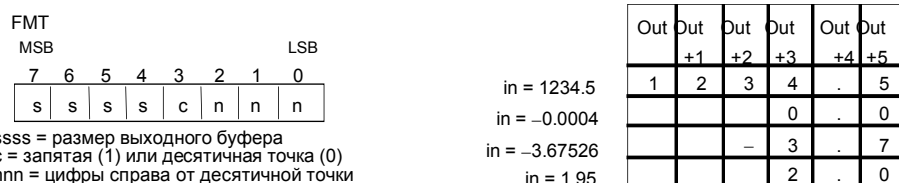


Рис. 6-17. Операнд FMT для команды преобразования вещественного числа в строку символов ASCII (RTA)

Пример: Команда преобразования строки символов ASCII в 16-ричное число

<p>Network 1</p>	<p>Network 1</p> <pre>LD I3.2 ATH VB30, VB40, 3</pre>															
<table border="0"> <tr> <td>'3'</td><td>'E'</td><td>'A'</td><td></td><td></td></tr> <tr> <td>33</td><td>45</td><td>41</td><td>ATH</td><td>3E Ax</td></tr> <tr> <td colspan="3">VB30</td><td></td><td>VB40</td></tr> </table>	'3'	'E'	'A'			33	45	41	ATH	3E Ax	VB30				VB40	<p>Указание: X показывает, что половина байта не меняется</p>
'3'	'E'	'A'														
33	45	41	ATH	3E Ax												
VB30				VB40												

Пример: Команда преобразования целого числа в строку символов ASCII

<p>Network 1</p>	<p>Network 1</p> <pre>//Преобразовать целое число из VW2 //в 8 символов ASCII, начиная с VB10, //с использованием формата 16#0B //(запятая вместо десятичной точки, //за которой следуют 3 цифры). LD I2.3 ITA VW2, VB10, 16#0B</pre>																											
<table border="0"> <tr> <td></td><td></td><td>'1'</td><td>'2'</td><td>'3'</td><td>'4'</td><td>'5'</td></tr> <tr> <td>12345</td><td>ITA</td><td>20</td><td>20</td><td>31</td><td>32</td><td>2C</td><td>33</td><td>34</td><td>35</td></tr> <tr> <td>VW2</td><td></td><td>VB10</td><td>VB11</td><td>...</td><td></td><td></td><td></td><td></td><td></td></tr> </table>			'1'	'2'	'3'	'4'	'5'	12345	ITA	20	20	31	32	2C	33	34	35	VW2		VB10	VB11	...						
		'1'	'2'	'3'	'4'	'5'																						
12345	ITA	20	20	31	32	2C	33	34	35																			
VW2		VB10	VB11	...																								

Пример: Команда преобразования вещественного числа в строку символов ASCII

<p>Network 1</p>	<p>Network 1</p> <pre>//Преобразовать вещественное число из VD2 //в 10 символов ASCII, начиная с адреса // VB10, с помощью формата 16#A3 //(запятая вместо десятичной точки, // за которой следуют 3 цифры). LD I2.3 RTA VD2, VB10, 16#A3</pre>																																
<table border="0"> <tr> <td></td><td></td><td>'1'</td><td>'2'</td><td>'3'</td><td>'4'</td><td>'5'</td><td>'0'</td></tr> <tr> <td>123.45</td><td>RTA</td><td>20</td><td>20</td><td>20</td><td>31</td><td>32</td><td>33</td><td>2E</td><td>34</td><td>35</td><td>30</td></tr> <tr> <td>VD2</td><td></td><td>VB10</td><td>VB11</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>			'1'	'2'	'3'	'4'	'5'	'0'	123.45	RTA	20	20	20	31	32	33	2E	34	35	30	VD2		VB10	VB11	...								
		'1'	'2'	'3'	'4'	'5'	'0'																										
123.45	RTA	20	20	20	31	32	33	2E	34	35	30																						
VD2		VB10	VB11	...																													

Команды преобразования строк

Преобразование числовых значений в строки

Команды преобразования целого числа в строку (ITS), преобразования двойного целого числа в строку (DTS), преобразования вещественного числа в строку (RTS) преобразуют целое, двойное целое или вещественное число (IN) в строку символов ASCII (OUT).

Принцип действия команды преобразования целого числа в строку символов

Команда преобразования целого числа в строку символов (ITS) преобразует целое число IN в строку символов ASCII длиной 8 символов. Формат (FMT) определяет точность преобразования справа от десятичной точки, а также форму представления разделителя целой и дробной части – в виде запятой или точки. Результирующая строка записывается в 9 последовательных байтов, начиная с OUT. Подробную информацию о форматах строк символов вы найдете в главе 4.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)
- Недопустимый формат ($nnn > 5$)

На рис. 6–18 описан операнд формата для команды преобразования целого числа в строку. Длина выходной строки всегда равна 8 символам. Количество цифр справа от десятичной точки в выходном буфере определяется полем *nnn*. Допустимый диапазон поля *nnn* составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Для значений *nnn*, больших 5, выходом является строка из 8 пробелов ASCII. Бит *s* определяет использование запятой ($s=1$) или десятичной точки ($s=0$) в качестве разделителя целой и дробной части. Старшие 4 бита формата должны быть нулями.

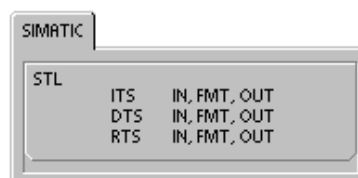
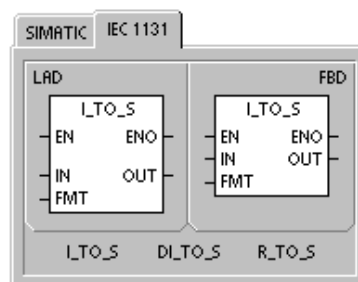
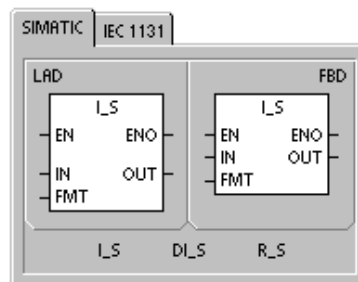
На рис. 6–18 показаны примеры чисел, форматированных с использованием десятичной точки ($s = 0$), с тремя цифрами справа от десятичной точки ($nnn = 011$). Значение OUT представляет собой длину строки.

Выходная строка форматируется в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Числа в выходной строке выравниваются вправо.

Таблица 6–19. Допустимые операнды для команд преобразования числовых значений в строки

Входы/выходы	Тип данных	Операнды
IN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC, константа
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	STRING	VB, LB, *VD, *LD, *AC



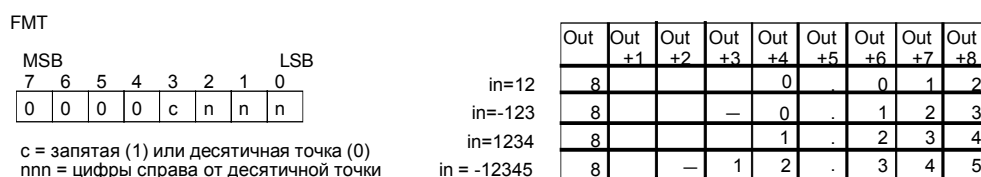


Рис. 6–18. Операнд FMT для команды преобразования целого числа в строку

Принцип действия команды преобразования двойного целого числа в строку

<p>Команда преобразования двойного целого числа в строку (DTS) преобразует двойное целое число IN в строку ASCII длиной 12 символов. Формат (FMT) определяет точность преобразования справа от десятичной точки, а также форму представления разделителя целой и дробной части – в виде запятой или точки. Результирующая строка записывается в 13 последовательных байтов, начиная с адреса OUT. Дополнительную информацию вы найдете в главе 4, где описан формат строк.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес) ■ 0091 (операнд вышел за пределы допустимого диапазона) ■ Недопустимый формат (nnn > 5)
--	--

На рис. 6–19 описан операнд формата для команды преобразования двойного целого числа в строку. Длина выходной строки всегда равна 12 символам. Количество цифр справа от десятичной точки в выходном буфере определяется полем *nnn*. Допустимый диапазон поля *nnn* составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Для значений *nnn*, больших 5, выходом является строка из 12 символов пробела ASCII. Бит *c* определяет использование запятой (*c*=1) или десятичной точки (*c*=0) в качестве разделителя целой и дробной части. Старшие 4 бита формата должны быть нулями.

На рис. 6–19 также показаны примеры чисел, форматированных с помощью десятичной точки (*c* = 0), с четырьмя цифрами справа от десятичной точки (*nnn* = 100). Значение OUT представляет собой длину строки. Выходная строка форматировается в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Числа в выходной строке выравниваются вправо.

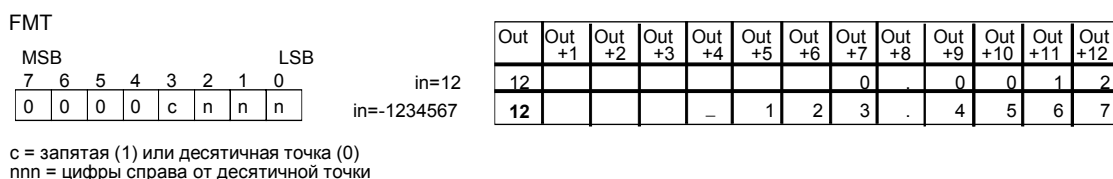


Рис. 6–19. Операнд FMT для команды преобразования двойного целого в строку

Принцип действия команды преобразования вещественного числа в строку

<p>Команда преобразования вещественного числа в строку (RTS) преобразует вещественное число IN в строку символов ASCII. Формат (FMT) определяет точность преобразования справа от десятичной точки, форму представления разделителя целой и дробной части – в виде запятой или точки, а также длину выходной строки.</p> <p>Результат преобразования помещается в строку, начинающуюся с OUT. Длина результирующей строки определяется в формате и может составлять от 3 до 15 символов. Дополнительную информацию вы найдете в главе 4, где описан формат строк.</p>	<p>Сбойные состояния, устанавливающие ENO = 0</p> <ul style="list-style-type: none"> ■ 0006 (косвенный адрес) ■ 0091 (операнд вышел за пределы допустимого диапазона) ■ Недопустимый формат: <ul style="list-style-type: none"> nnn > 5 ssss < 3 ssss < число требуемых символов
---	---

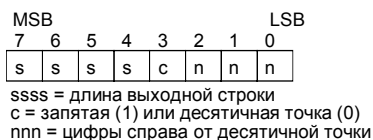
Формат вещественных чисел, используемый S7-200, поддерживает не более 7 значащих цифр. Попытка отобразить более 7 значащих цифр приводит к ошибке округления.

На рис. 6-20 описан операнд формата для команды преобразования вещественного числа в строку. Длина выходной строки определяется полем ssss. Размеры 0, 1 или 2 байта недопустимы. Количество цифр справа от десятичной точки в выходном буфере определяется полем nnn. Допустимый диапазон поля nnn составляет от 0 до 5. Задание 0 цифр справа от десятичной точки приводит к тому, что число отображается без десятичной точки. Выходная строка заполняется символами пробела ASCII для значений nnn, больших 5, или если указанная длина выходной строки слишком мала для хранения преобразованного значения. Бит c определяет использование запятой (c=1) или десятичной точки (c=0) в качестве разделителя целой и дробной части.

На рис. 6-20 приведены примеры чисел, форматированных с помощью десятичной точки (c = 0), с одной цифрой справа от десятичной точки (nnn = 001) и выходной строкой, имеющей длину 6 символов (ssss = 0110). Значение OUT представляет собой длину строки. Выходная строка форматируется в соответствии со следующими правилами:

- Положительные числа записываются в выходной буфер без знака.
- Отрицательные числа записываются в выходной буфер с ведущим знаком минус (-).
- Нули в старших разрядах слева от десятичной точки (кроме цифры, смежной с десятичной точкой) подавляются.
- Значение справа от десятичной точки округляется в соответствии с заданным количеством цифр справа от десятичной точки.
- Размер выходной строки должен, по крайней мере, на три байта превышать количество цифр справа от десятичной точки.
- Числа в выходной строке выравниваются вправо.

FMT



in=1234.5
 in= -0.0004
 in= -3.67526
 in = 1.95

Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6
6	1	2	3	4	.	5
6				0	.	0
6			-	3	.	7
6				2	.	0

Рис. 6-20. Операнд FMT для команды преобразования вещественного числа в строку

Преобразование подстроки в числовые значения

Команды преобразования подстроки в целое число (STI), преобразования подстроки в двойное целое число (STD) и преобразования подстроки в вещественное число (STR) преобразуют строку символов IN, начинающуюся со смещения INDX, в целое, двойное целое или вещественное число OUT.

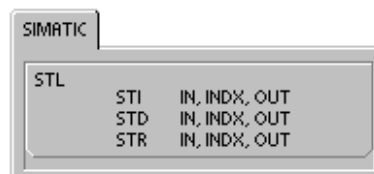
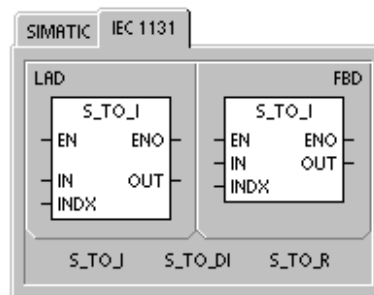
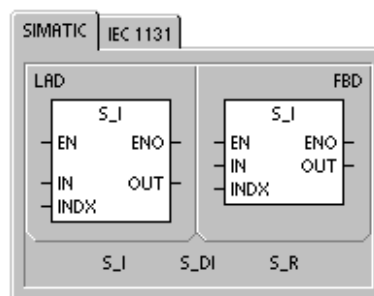
Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)
- 009B (индекс = 0)
- SM1.1 (переполнение)

Команды преобразования подстроки в целое и двойное целое число преобразуют строки, имеющие следующий формат: [пробелы] [+ или -] [цифры 0 - 9]

Команда преобразования подстроки в вещественное число преобразует строки, имеющие следующий формат: [пробелы] [+ или -] [цифры 0 - 9] [. или ,] [цифры 0 - 9]

Значение INDX обычно устанавливается в 1, так что преобразование начинается с первого символа строки. Значение INDX может быть и другим, чтобы начать преобразование с иной точки внутри строки. Этим можно воспользоваться, если входная строка содержит текст, не являющийся частью преобразуемого числа. Например, если строка имеет вид «Температура: 77.8», то установите INDX на значение 13, чтобы пропустить слово «Температура: » в начале строки.



Команда преобразования подстроки в вещественное число не преобразует строки, использующие научную нотацию или экспоненциальные формы вещественных чисел. Команда не генерирует ошибку переполнения (SM1.1), а преобразует строку в вещественное число до экспоненциальной функции, а затем завершает преобразование. Например, строка '1.234E6' преобразуется без ошибки в вещественное число 1.234.

Преобразование завершается при достижении конца строки или обнаружения первого недопустимого символа. Недопустимым является любой символ, отличный от цифры (0 - 9).

Ошибка переполнения (SM1.1) устанавливается, когда преобразование дает целое число, слишком большое для выходной величины. Например, команда преобразования подстроки в вещественное устанавливает ошибку переполнения, если входная строка дает значение, большее, чем 32767, или меньшее, чем -32768.

Ошибка переполнения (SM1.1) устанавливается также, если преобразование невозможно из-за отсутствия в строке допустимого значения. Например, если входная строка содержит 'A123', то команда преобразования устанавливает SM1.1 (переполнение), а выходная величина остается неизменной.

Таблица 6–20. Допустимые операнды для команд преобразования подстрок в числовые значения

Входы/выходы	Тип данных	Операнды
IN	STRING	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC, константа
INDX	BYTE	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	INT DINT, REAL	VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Допустимые входные строки для целых и двойных целых чисел

Входная строка	Выходное целое
'123'	123
'-00456'	-456
'123.45'	123
'+2345'	2345
'000000123ABCD'	123

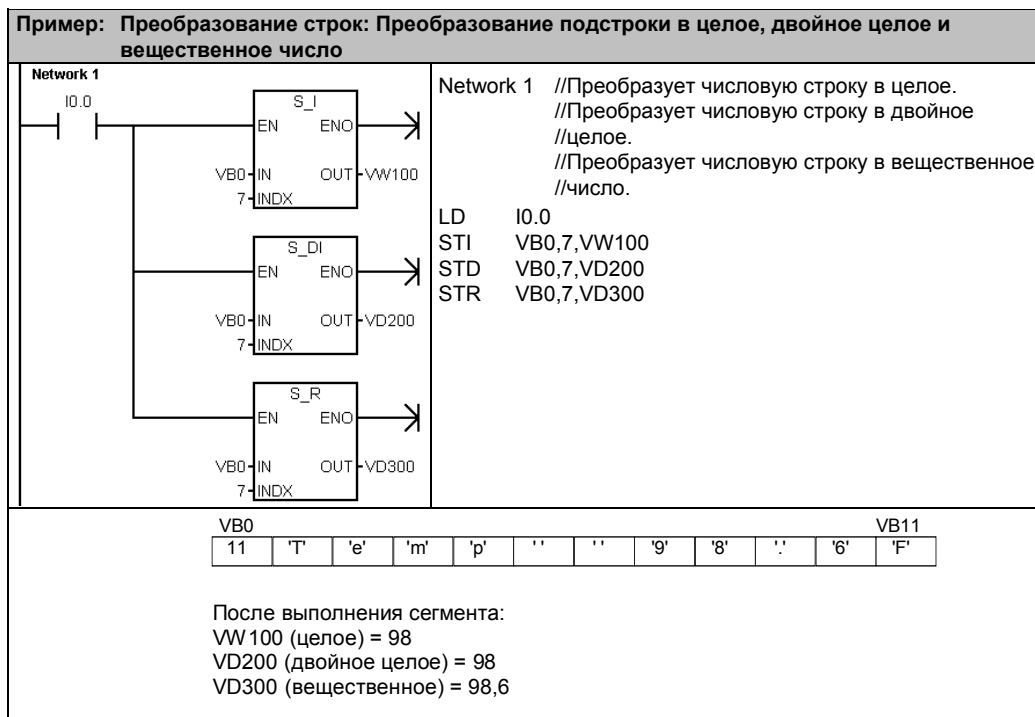
Допустимые входные строки для вещественных чисел

Входная строка	Выходное веществ. число
'123'	123.0
'-00456'	-456.0
'123.45'	123.45
'+2345'	2345.0
'00.000000123'	0.000000123

Недопустимые входные строки

Входная строка
'A123'
' '
'++123'
'+-123'
'+ 123'

Рис. 6-21. Примеры допустимых и недопустимых входных строк



Команды кодирования и декодирования

Кодирование

Команда кодирования (ENCO) записывает номер младшего установленного бита входного слова IN в младший полубайт (4 бита) выходного байта OUT.

Декодирование

Команда декодирования (DECO) устанавливает в выходном слове OUT бит, соответствующий номеру бита, представленному младшим полубайтом (4 бита) входного байта IN. Все остальные биты выходного слова устанавливаются в 0.

Биты специальной памяти и ENO

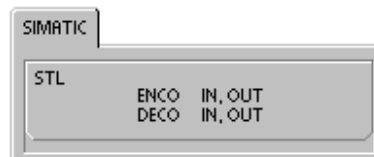
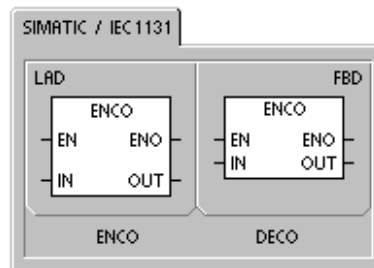
Для команд кодирования и декодирования следующие условия влияют на ENO.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Таблица 6–21. Допустимые операнды для команд кодирования и декодирования

Входы/выходы	Типы данных	Операнды
IN	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
OUT	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC



Пример: Команды декодирования и кодирования	
<p>Network 1</p>	<p>Network 1 //Аккумулятор 2 содержит биты ошибок. //1. Команда DECO устанавливает в VW40 // бит, соответствующий этому коду // ошибки. //2. Команда ENCO преобразует младший // установленный бит в код ошибки, // который сохраняется в VB50.</p> <p>LD I3.1 DECO AC2, VW40 ENCO AC3, VB50</p>
<p>AC2 3</p>	<p>AC3 15 9 0 1000 0010 0000 0000</p>
<p>DECO 15 3 0 0000 0000 0000 1000</p>	<p>ENCO 9</p>
<p>VW40</p>	<p>VB50</p>

Команды счета

Счетчик SIMATIC

Прямой счет

Команда прямого счета (CTU) увеличивает текущее значение счетчика при появлении нарастающего фронта на входе (CU). Когда текущее значение Sxx больше или равно предустановленному значению PV, бит счетчика Sxx устанавливается. Счетчик сбрасывается, когда включается вход сброса (R), или когда выполняется команда сброса. Счетчик прекращает счет при достижении максимального значения (32767).

Принцип действия в STL:

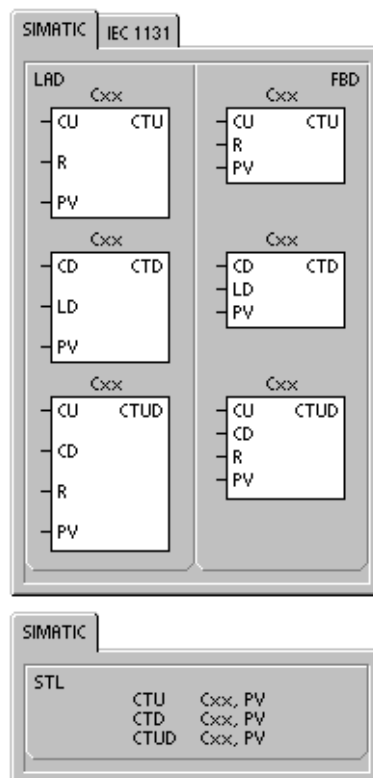
- Вход сброса: Вершина стека
- Вход прямого счета: Значение, загруженное на второй уровень стека

Обратный счет

Команда обратного счета (CTD) уменьшает текущее значение счетчика при появлении нарастающего фронта на входе (CD). Когда текущее значение Sxx равно нулю, бит счетчика Sxx включается. Счетчик сбрасывает свой бит Sxx и загружает текущее значение предустановленным значением PV, когда включается вход загрузки LD. Счетчик останавливается, когда он достигает нуля, и бит счетчика Sxx включается.

Принцип действия в STL:

- Вход загрузки: Вершина стека
- Вход обратного счета: Значение, загруженное на второй уровень стека.



Реверсивный счетчик

Команда реверсивного счета (CTUD) увеличивает значение счетчика при появлении нарастающего фронта на входе (CU) и уменьшает значение счетчика при появлении нарастающего фронта на входе (CD). Sxx содержит текущее счетное значение. Предустановленное значение PV сравнивается с текущим значением при каждом выполнении команды.

При достижении максимального значения (32 767) следующий нарастающий фронт на входе прямого счета вызывает переход к минимальному значению (-32 768). При достижении минимального значения (-32 768) следующий нарастающий фронт на входе обратного счета вызывает переход к максимальному значению (32 767).

Если текущее значение Sxx больше или равно предустановленному значению PV, то бит счетчика Sxx устанавливается. В противном случае бит счетчика выключается. Счетчик сбрасывается, когда включается вход сброса (R) или когда выполняется команда сброса. Счетчик CTUD прекращает счет при достижении PV.

Принцип действия в STL:

- Вход сброса: Вершина стека
- Вход обратного счета: Значение, загруженное на второй уровень стека
- Вход прямого счета: Значение, загруженное на третий уровень стека

Таблица 6–22. Допустимые операнды для счетчиков SIMATIC

Входы/выходы	Типы данных	Операнды
Sxx	WORD	константа (от C0 до C255)
CU, CD, LD, R	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
PV	INT	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, AIW, *VD, *LD, *AC, константа



Совет

Так как для каждого счетчика имеется только одно текущее значение, не назначайте один и тот же номер более чем одному счетчику (прямые, реверсивные и обратные счетчики с одним и тем же номером обращаются к одному и тому же текущему значению).

Если вы сбрасываете счетчик с помощью команды Сброс, то бит счетчика сбрасывается, а текущее значение устанавливается в ноль. Номер счетчика используется для обращения как к текущему значению, так и к биту счетчика.

Таблица 6–23. Принцип действия счетчиков

Тип	Режим	Бит счетчика	Выключение-включение питания/Первый цикл
STU	CU увеличивает текущее значение. Текущее значение продолжает увеличиваться до достижения 32 767.	Бит счетчика устанавливается, когда: Текущее значение >= предустановленному	Бит счетчика сброшен. Текущее значение может быть сохранено. ¹
CTUD	CU увеличивает текущее значение. CD уменьшает текущее значение. Текущее значение продолжает увеличиваться или уменьшаться, пока счетчик не сброшен.	Бит счетчика устанавливается, когда: Текущее значение >= предустановленному	Бит счетчика сброшен. Текущее значение может быть сохранено. ¹
CTD	CD уменьшает текущее значение, пока текущее значение не достигнет 0.	Бит счетчика устанавливается, когда: Текущее значение = 0	Бит счетчика сброшен. Текущее значение может быть сохранено. ¹

¹ Вы можете определить текущее значение счетчика как сохраняемую величину. Информацию о сохраняемой памяти CPU S7–200 вы найдете в главе 4.

Пример: Команда обратного счета SIMATIC

Network 1

Network 2

Network 1 //Текущее значение счетчика обратного счета C1
//уменьшается с 3 до 0, когда I0.1 выключен.
//Положительный фронт на I0.0 уменьшает текущее
//значение C1
//Включение I0.1 загружает предустановленное
//значение 3 для обратного счета

LD I0.0
LD I0.1
CTD C1, +3

Network 2 //Бит C1 установлен, когда текущее значение
//счетчика C1 = 0

LD C1
= Q0.0

Временная диаграмма

Пример: Реверсивный счетчик SIMATIC

Network 1

Network 2

Network 1 //I0.0 считает вперед
//I0.1 считает назад
//I0.2 сбрасывает текущее значение в 0

LD I0.0
LD I0.1
LD I0.2
CTUD C48, +4

Network 2 //Реверсивный счетчик C48 устанавливает
//бит C48, когда текущее значение >= 4

LD C48
= Q0.0

Временная диаграмма

Счетчик IEC

Суммирующий счетчик

Суммирующий счетчик (CTU) ведет счет вверх от текущего значения к предустановленному значению (PV) при нарастающем фронте на входе прямого счета (CU). Когда текущее значение (CV) больше или равно предварительно установленному значению (PV), выходной бит счетчика (Q) включается. Счетчик сбрасывается, когда включается вход сброса (R). Суммирующий счетчик прекращает счет, когда он достигает предварительно установленного значения.

Вычитающий счетчик

Вычитающий счетчик (CTD) ведет счет вниз от предустановленного значения (PV) при положительном фронте на входе обратного счета (CD). Когда текущее значение (CV) равно нулю, выходной бит счетчика (Q) включается. Счетчик сбрасывается и загружает в качестве текущего значения (CV) предустановленное значение (PV), когда включается вход загрузки (LD). Вычитающий счетчик прекращает счет, когда он достигает нуля.

Реверсивный счетчик

Реверсивный счетчик (CTUD) ведет счет вверх или вниз от предустановленного значения (CV) при положительном фронте на входе прямого счета (CU) или обратного счета (CD). Когда текущее значение равно предварительно установленному значению, включается выход прямого счета (QU). Когда текущее значение равно нулю, включается выход обратного счета (QD). Счетчик загружает в качестве текущего значения (CV) предварительно установленное значение (PV), когда включается вход загрузки (LD). Аналогично, счетчик сбрасывается и загружает в качестве текущего значения (CV) нуль, когда включается сброс (R). Счетчик прекращает счет, когда он достигает предварительно установленного значения или нуля.

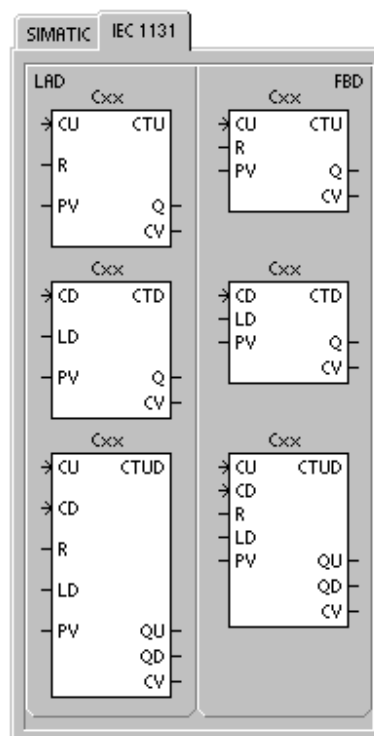


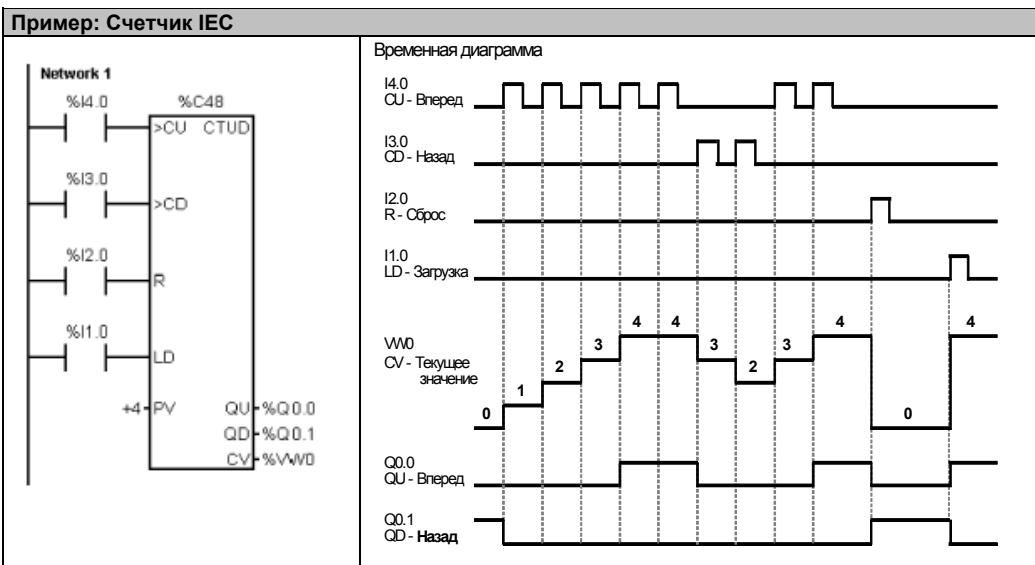
Таблица 6–24. Допустимые операнды для счетчиков IEC

Входы/выходы	Типы данных	Операнды
Cxx	CTU, CTD, CTUD	константа (от C0 до C255)
CU, CD, LD, R	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
PV	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, константа
Q, QU, QD	BOOL	I, Q, V, M, SM, S, L
CV	INT	IW, QW, VW, MW, SW, LW, AC, *VD, *LD, *AC



Совет

Так как для каждого счетчика имеется одно текущее значение, не назначайте один и тот же номер более чем одному счетчику. (Суммирующие, вычитающие и реверсивные счетчики обращаются к одному и тому же текущему значению.)



Скоростные счетчики

Определение режима работы скоростного счетчика

Команда определения режима работы скоростного счетчика (HDEF) устанавливает режим работы для определенного скоростного счетчика (HSCx). Выбором режима определяются датчик тактовых импульсов, направление и функции запуска и сброса скоростного счетчика.

Для каждого скоростного счетчика используется одна команда определения режима работы.

Сбойные состояния, устанавливающие ENO = 0

- 0003 (конфликт входов)
- 0004 (недопустимая команда в прерывании)
- 000A (повторное определение HSC)

Скоростной счетчик

Команда активизации скоростного счетчика (HSC) конфигурирует и управляет режимом работы скоростного счетчика через сигнальные состояния битов специальной памяти HSC. Параметр N определяет номер скоростного счетчика.

Скоростные счетчики могут быть сконфигурированы на двенадцать различных режимов работы. См. таблицу 6–26.

Каждый счетчик имеет специализированные входы, которые поддерживают такие функции, как датчик тактовых импульсов, управление направлением, сброс и запуск. Для двухфазных счетчиков оба датчика тактовых импульсов могут работать со своей максимальной скоростью. В квадратурных режимах (A/B-счетчики) предоставляется возможность выбора однократной (1x) или четырехкратной (4x) скорости счета. Все счетчики работают с максимальной скоростью, не создавая помех друг другу.

Сбойные состояния, устанавливающие ENO = 0

- 0001 (HSC перед HDEF)
- 0005 (одновременно HSC/PLS)

Таблица 6–25. Допустимые операнды для скоростных счетчиков

Входы/выходы	Типы данных	Операнды
HSC, MODE	BYTE	константа
N	WORD	константа



Советы по программированию

За программами, использующими скоростные счетчики, обратитесь к советам по программированию на компакт-диске с документацией. См. советы 4 и 29.

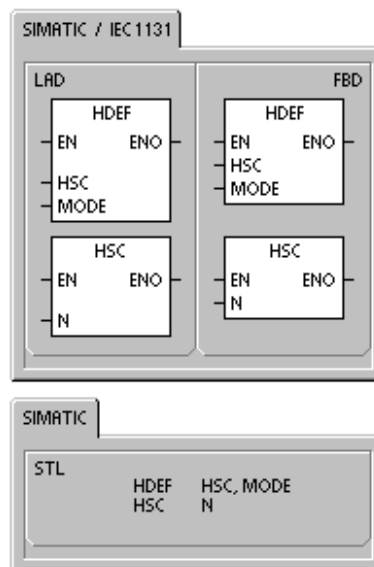
Скоростные счетчики ведут счет происходящих с высокой скоростью событий, которые не могут управляться при скоростях сканирования S7–200. Максимальная частота счета скоростного счетчика зависит от типа вашего CPU S7–200. Дальнейшую информацию вы найдете в Приложении А.



Совет

CPU 221 и CPU 222 поддерживают четыре скоростных счетчика: HSC0, HSC3, HSC4 и HSC5. Эти CPU не поддерживают HSC1 и HSC2.

CPU 224, CPU 224XP и CPU 226 поддерживают шесть скоростных счетчиков: от HSC0 до HSC5.



Обычно скоростные счетчики используются в качестве привода для счетных механизмов, в которых вал, вращающийся с постоянной скоростью, снабжен угловым шаговым датчиком. Угловой шаговый датчик дает определенное количество отсчетов на оборот, а также импульс сброса один раз за оборот. Датчик (датчики) тактовых импульсов и импульс сброса от углового шагового датчика обеспечивают входы для скоростного счетчика.

Скоростной счетчик загружается первым из нескольких предустановленных значений, и желаемые выходы активизируются на интервал времени, в течение которого текущее значение счетчика меньше текущего предустановленного значения. Счетчик настроен таким образом, что, когда текущее значение счетчика становится равным предустановленному значению, или при появлении сброса происходит прерывание.

Когда при равенстве текущего значения счетчика и предустановленного значения происходит прерывающее событие, загружается новое предустановленное значение, и устанавливается следующее состояние для выходов. Когда происходит событие, вызывающее прерывание по сбросу, то устанавливаются первое предустановленное значение и первые состояния выходов, и цикл повторяется.

Так как прерывания происходят со значительно меньшей частотой, чем считает скоростной счетчик, то может быть реализовано точное управление быстрыми операциями при относительно малом воздействии на общий цикл обработки программы ПЛК. Метод подключения прерываний позволяет выполнять каждую загрузку нового предустановленного значения в отдельной программе обработки прерывания, что упрощает управление состоянием. (В качестве альтернативы, все события, вызывающие прерывания, могут быть обработаны и в одной единственной программе обработки прерываний.)

Описание различных скоростных счетчиков

Все счетчики в одном и том же режиме работают одинаково. Имеется четыре основных вида счетчиков: однофазный счетчик с внутренним управлением направлением, однофазный счетчик с внешним управлением направлением, двухфазный счетчик с 2 тактовыми входами и квадратурный счетчик с фазами А и В. Обратите внимание, что не каждый счетчик поддерживает все режимы. Каждый счетчик можно использовать: без входов сброса и пуска, со сбросом, но без пуска, или с входами пуска и сброса.

- Когда вы активизируете вход сброса, он сбрасывает текущее значение и сохраняет его сброшенным, пока вы не деактивизируете сброс.
- Когда вы активизируете вход пуска, он разрешает счетчику считать. Если вход пуска деактивизирован, текущее значение счетчика остается постоянным, а тактовые события игнорируются.
- Если сброс активизируется, когда пуск неактивен, то сброс игнорируется, а текущее значение не изменяется. Если вход пуска становится активным, когда активен вход сброса, текущее значение сбрасывается.

Перед использованием скоростного счетчика вы должны с помощью команды HDEF (**High-Speed Counter Definition** = Определение скоростного счетчика) выбрать его режим. С помощью бита памяти первого цикла SM0.1 (этот бит включен в течение первого цикла обработки программы, а затем выключается) вызовите подпрограмму, которая содержит команду HDEF.

Программирование скоростного счетчика



Мастер команд

Для проектирования скоростного счетчика можно использовать мастер команд HSC. Этот мастер использует следующую информацию: тип и режим работы счетчика, предустановленное значение счетчика, текущее значение счетчика и начальное направление счета. Для вызова мастера команд HSC выберите команду меню **Tools > Instruction Wizard [Инструменты > Мастер команд]**, а затем из окна мастера команд выберите вариант HSC.

Для программирования скоростного счетчика вы должны выполнить следующие основные задачи:

- Определить счетчик и режим.
- Настроить управляющий байт.
- Установить текущее (начальное) значение.
- Задать предустановленное (целевое) значение.
- Назначить и разблокировать программу обработки прерываний.
- Активизировать скоростной счетчик.

Определение режимов и входов счетчика

Для определения режимов и входов счетчика используется команда определения скоростного счетчика HDEF.

В таблице 6–26 показаны входы для таких функций скоростных счетчиков, как генератор тактовых импульсов, управление направлением, сброс и запуск. Один и тот же вход не может быть использован для двух разных функций, но любой вход, не используемый текущим режимом скоростного счетчика, может быть использован для другой цели. Например, если HSC0 используется в режиме 1, который использует I0.0 и I0.2, то I0.1 может быть использован для прерываний по фронту сигнала или для HSC3.


	<p>Совет</p> <p>Примите во внимание, что все режимы HSC0 (кроме режима 12) всегда используют I0.0, а все режимы HSC4 всегда используют I0.3, так что эти входы не бывают доступными для других целей, когда используются данные счетчики.</p>
---	--

Таблица 6–26. Входы для скоростных счетчиков

Режим	Описание	Входы			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I1.0	I1.1
	HSC2	I1.2	I1.3	I1.4	I1.5
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	Однофазный счетчик с внутренним управлением направлением	Датчик тактовых импульсов			
1		Датчик тактовых импульсов		Сброс	
2		Датчик тактовых импульсов		Сброс	Пуск
3	Однофазный счетчик с внешним управлением направлением	Датчик тактовых импульсов	Направление		
4		Датчик тактовых импульсов	Направление	Сброс	
5		Датчик тактовых импульсов	Направление	Сброс	Пуск
6	Двухфазный счетчик с 2 тактовыми входами	Датчик тактовых импульсов для прямого направления	Датчик тактовых импульсов для обратного направления		
7		Датчик тактовых импульсов для прямого направления	Датчик тактовых импульсов для обратного направления	Сброс	
8		Датчик тактовых импульсов для прямого направления	Датчик тактовых импульсов для обратного направления	Сброс	Пуск
9	Квадратурный счетчик с фазами А и В	Датчик тактовых импульсов А	Датчик тактовых импульсов В		
10		Датчик тактовых импульсов А	Датчик тактовых импульсов В	Сброс	
11		Датчик тактовых импульсов А	Датчик тактовых импульсов В	Сброс	Пуск
12	Режим счета 12 поддерживают только HSC0 и HSC3. HSC0 считает количество импульсов, выдаваемых Q0.0. HSC3 считает количество импульсов, выдаваемых Q0.1.				

Примеры режимов HSC

Временные диаграммы на рисунках 6-22 - 6-26 показывают, как работает каждый счетчик в соответствии с режимом.

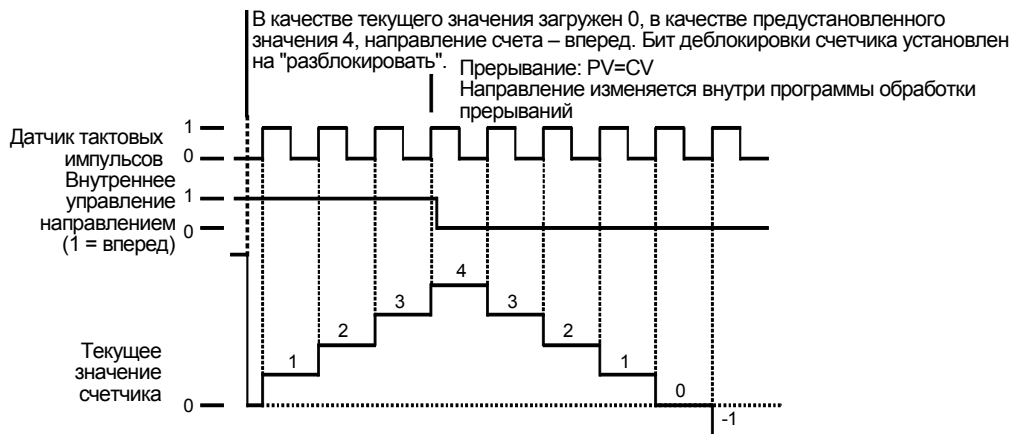


Рис. 6-22. Пример работы в режимах 0, 1 или 2

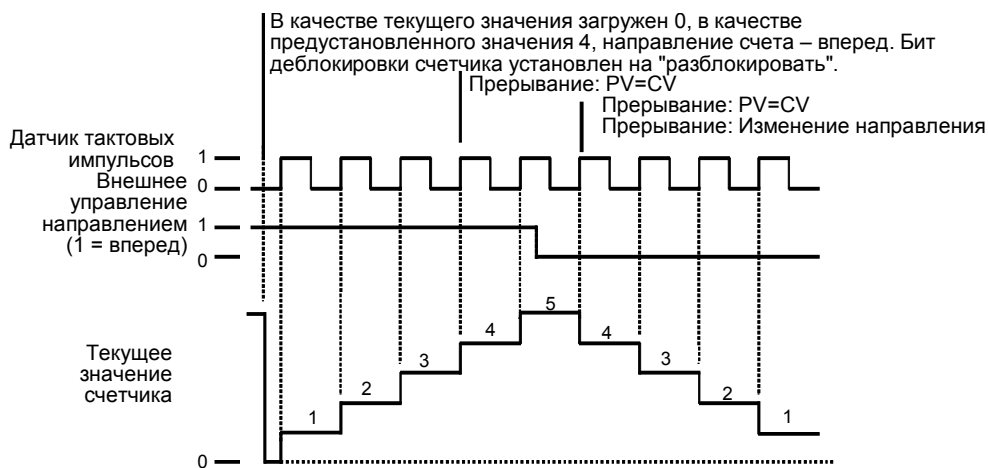


Рис. 6-23. Пример работы в режимах 3, 4 или 5

Когда используются режимы счета 6, 7 или 8, и в течение 0,3 микросекунды друг за другом появляется нарастающий фронт на тактовых входах счета вперед и счета назад, скоростной счетчик может рассматривать эти события как происходящие одновременно. Если это происходит, то текущее значение не меняется и не отображается изменение в направлении счета. Если между поступлениями нарастающих фронтов на тактовые входы счета вперед и счета назад проходит больше 0,3 микросекунды, то скоростной счетчик воспринимает эти события отдельно. В этом случае ошибки не происходит, и счетчик сохраняет правильное счетное значение.

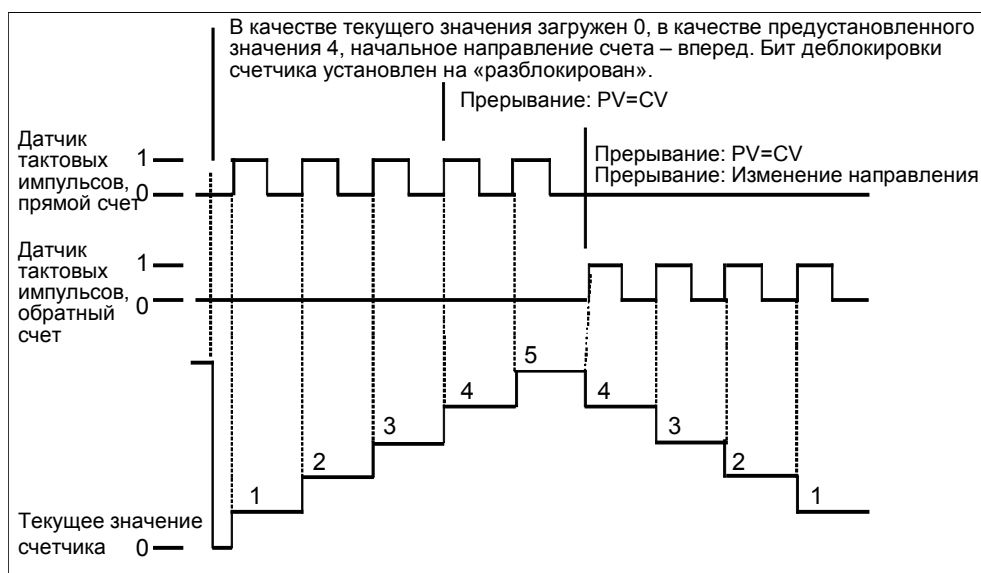


Рис. 6–24. Пример работы в режимах 6, 7 или 8

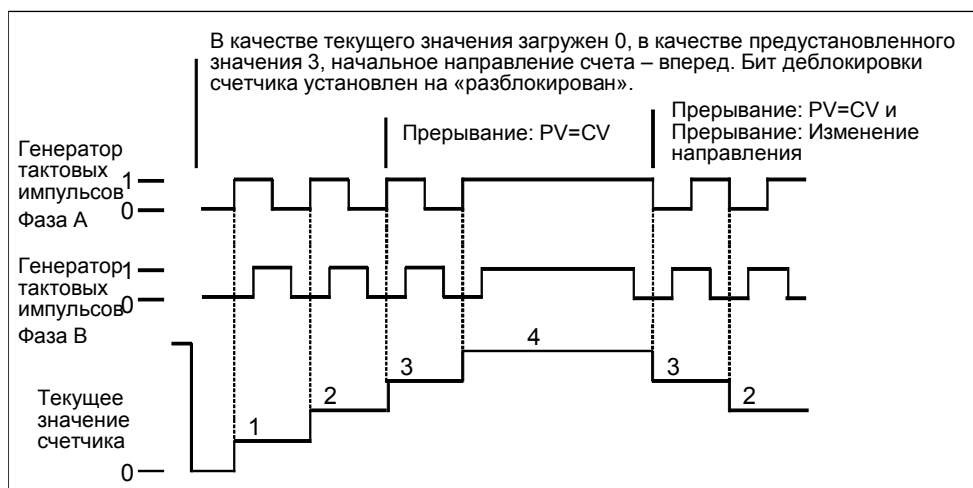


Рис. 6–25. Пример работы в режимах 9, 10 или 11 (квадратурный режим, однократная скорость)

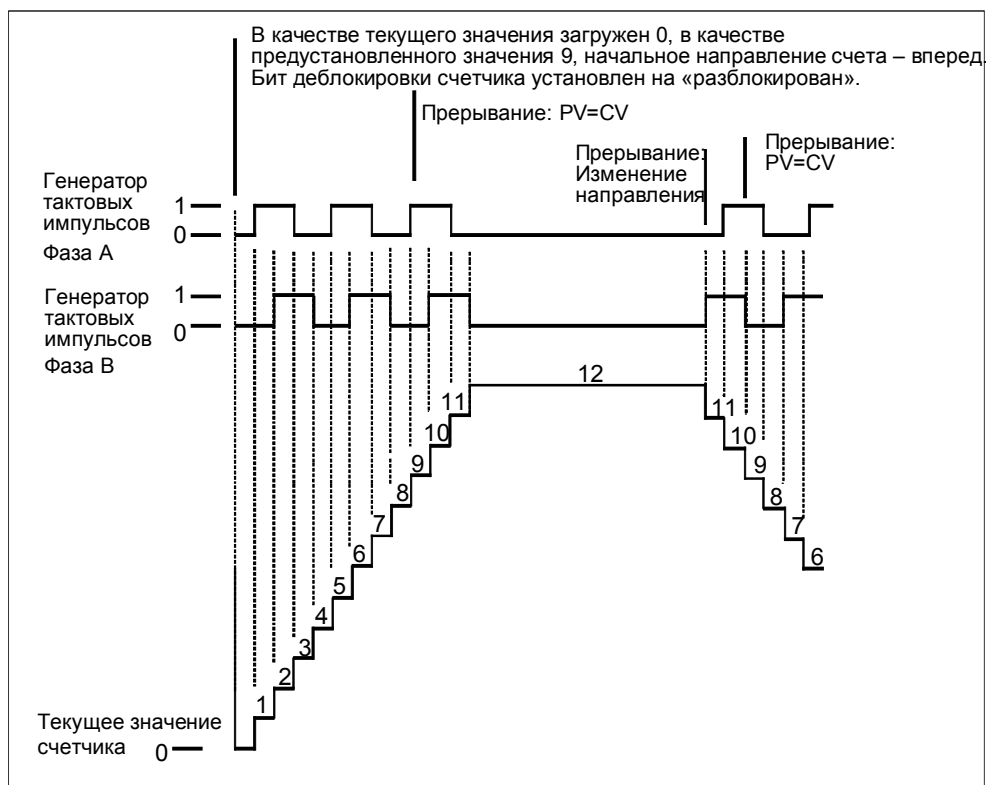


Рис. 6–26. Пример работы в режимах 9, 10 или 11 (квадратурный режим, четырехкратная скорость)

Принцип действия входов сброса и пуска

Работа входов сброса и запуска показана на рис. 6–27 и действительна для всех режимов, использующих входы сброса и запуска. На диаграммах для входов сброса и запуска активность обоих входов запрограммирована для высокого уровня сигнала.

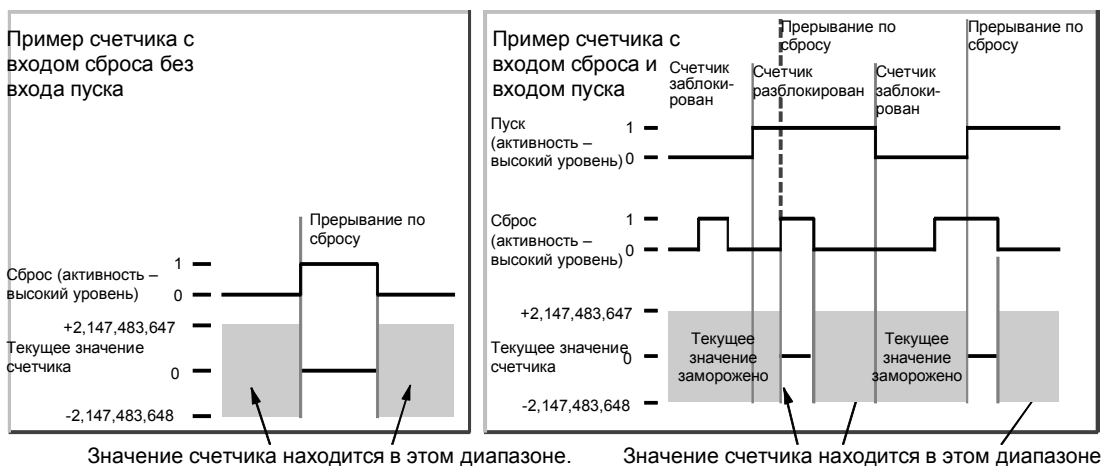


Рис. 6–27. Примеры работы счетчика, использующего вход сброса с входом и без входа пуска

Четыре счетчика имеют три управляющих бита, которые используются для конфигурирования активного состояния входов сброса и пуска и для выбора односкоростного или четырехскоростного режима счета (только для квадратурных счетчиков). Эти биты находятся в управляющем байте соответствующего счетчика и используются только тогда, когда выполняется команда HDEF. Эти биты определены в таблице 6–27.


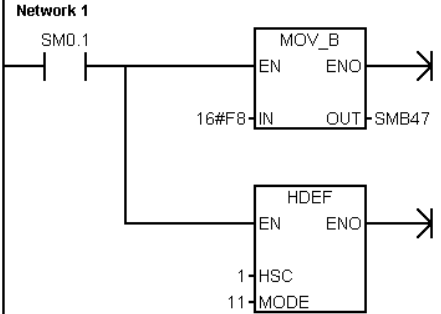
	<p>Совет</p> <p>Вы должны установить эти управляющие биты в соответствии с желаемым состоянием до исполнения команды HDEF. В противном случае счетчик принимает конфигурацию, определенную по умолчанию для выбранного режима работы счетчика.</p> <p>Если команда HDEF была выполнена, вы не можете изменить настройку счетчика, не переведя сначала S7–200 в состояние STOP.</p>
---	---

Таблица 6–27. Активный уровень для управляющих битов сброса, пуска и выбора однократной или четырехкратной скорости

HSC0	HSC1	HSC2	HSC4	Описание (используются только при исполнении HDEF)
SM37.0	SM47.0	SM57.0	SM147.0	Активный уровень управляющего бита для сброса ¹ : 0 = сброс активен при высоком уровне; 1 = сброс активен при низком уровне
---	SM47.1	SM57.1	---	Активный уровень управляющего бита для пуска ¹ : 0 = пуск активен при высоком уровне; 1 = пуск активен при низком уровне
SM37.2	SM47.2	SM57.2	SM147.2	Выбор скорости счета для квадратурных счетчиков: 0 = 4-кратная скорость 1 = 1-кратная скорость

¹ По умолчанию входы сброса и пуска активны при высоком уровне сигнала, а в квадратурных счетчиках скорость счета установлена четырехкратной (по отношению к частоте входного датчика тактовых импульсов).

Пример: Команда определения режима скоростного счетчика	
<p>M A I N</p>	
	<p>Network 1 //В первом цикле: //1. Выбрать для входов пуска и сброса в качестве активного высокого уровня и 4-кратную скорость счета. //2. Настроить HSC1 на квадратурный режим (A/B-счетчик) с входами сброса и пуска.</p> <pre>LD SM0.1 MOV_B 16#F8, SMB47 HDEF 1, 11</pre>

Настройка управляющего бита

Определив счетчик и режим его работы, вы можете программировать динамические параметры счетчика. Каждый скоростной счетчик имеет управляющий байт, который позволяет выполнить следующие действия:

- разблокировать или заблокировать счетчик
- управлять направлением (только для режимов 0, 1 и 2) или устанавливать начальное направление счета для всех остальных режимов
- загружать текущее значение
- загружать предустановленное значение

Проверка управляющего байта и соответствующих текущего и предустановленного значений производится при выполнении команды HSC. В таблице 6–28 описан каждый из этих управляющих битов.

Таблица 6–28. Управляющие биты для HSC0, HSC1, HSC2, HSC3, HSC4 и HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Описание
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Бит управления направлением счета: 0 = обратный счет 1 = прямой счет
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Записать направление счета в HSC: 0 = не актуализировать 1 = актуализировать направление
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Записать новое предустановленное значение в HSC: 0 = не актуализировать; 1 = актуализировать предустановленное значение
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Записать новое текущее значение в HSC: 0 = не актуализировать; 1 = актуализировать текущее значение
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Разблокировка HSC: 0 = заблокировать HSC; 1 = разблокировать HSC

Установка текущего и предустановленного значений

Каждый скоростной счетчик имеет 32–битное текущее значение и 32–битное предустановленное значение. Оба значения являются целыми числами со знаком. Чтобы загрузить новое текущее или предустановленное значение, вы должны настроить управляющий байт и байты специальной памяти, содержащие текущее и/или предустановленное значение, а также выполнить команду HSC, чтобы новые значения были переданы в скоростной счетчик. Таблица 6–29 описывает байты специальной памяти, используемые для хранения новых текущих и предустановленных значений.

В дополнение к управляющим байтам и байтам, содержащим новые текущие и предустановленные значения, текущее значение каждого скоростного счетчика может быть прочитано путем задания типа данных HC (текущее значение скоростного счетчика), за которым следует номер (0, 1, 2, 3, 4 или 5) счетчика, как показано в таблице 6–29. Текущее значение непосредственно доступно для операций чтения, но оно может быть записано только с помощью команды HSC.

Таблица 6–29. Новое текущее и новое предустановленное значение HSC0, HSC1, HSC2, HSC3, HSC4 и HSC5

Загружаемое значение	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Новое текущее значение	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Новое предустановленное значение	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Таблица 6–30. Текущие значения HSC0, HSC1, HSC2, HSC3, HSC4 и HSC5

Значение	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Текущее значение	HC0	HC1	HC2	HC3	HC4	HC5

Адресация скоростных счетчиков (НС)

Для доступа к счетному значению скоростного счетчика указывается адрес этого счетчика с помощью типа памяти (НС) и номера счетчика (например, НС0). Текущее значение скоростного счетчика доступно только для чтения и может быть адресовано только как двойное слово (32 бита), как показано на рис. 6–28.

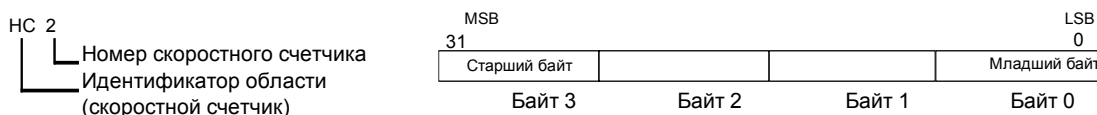


Рис. 6–28. Доступ к текущему значению скоростного счетчика

Назначение прерываний

Все режимы счетчиков поддерживают прерывание по равенству текущего значения HSC заданному предустановленному значению. Режимы счетчиков, использующие вход внешнего сброса, поддерживают прерывание по активизации внешнего сброса. Все режимы счетчиков, кроме режимов 0, 1 и 2, поддерживают прерывание по изменению направления счета. Каждое из этих условий возникновения прерываний может быть заблокировано или разблокировано по отдельности. Полностью использование прерываний обсуждается в разделе о командах обмена данными и прерывания.

Замечание

Если вы попытаетесь загрузить новое текущее значение или заблокировать, а затем снова разблокировать скоростной счетчик из программы обработки прерываний для внешнего сброса, то может произойти фатальная ошибка.

Байт состояния

Каждому скоростному счетчику поставлен в соответствие байт состояния, предоставляющий в распоряжение биты памяти, указывающие текущее направление счета, а также информацию о том, действительно ли текущее значение больше или равно предустановленному. Таблица 6–31 определяет эти биты состояния для каждого скоростного счетчика.



Совет

Биты состояния действительны только во время исполнения программы обработки прерывания скоростного счетчика. Цель контроля состояния скоростного счетчика состоит в том, чтобы разблокировать прерывания для событий, оказывающих воздействие на выполняемую операцию.

Таблица 6–31. Биты состояния для HSC0, HSC1, HSC2, HSC3, HSC4 и HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Описание
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	Не используются
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	Не используются
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	Не используются
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	Не используются
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	Не используются
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Бит состояния текущего направления счета: 0 = обратный счет 1 = прямой счет
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	Бит состояния, указывающий, равно ли текущее значение предустановленному: 0 = не равно 1 = равно
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	Бит состояния, указывающий, больше ли текущее значение, чем предустановленное: 0 = меньше или равно 1 = больше

Примеры инициализирующих последовательностей для скоростных счетчиков

В следующих описаниях инициализации и последовательности обработки в качестве примера используется HSC1. При описании инициализаций предполагается, что S7-200 только что переведен в режим RUN, и поэтому бит памяти первого цикла установлен. Если это не так, помните, что команда HDEF может быть выполнена только один раз для каждого скоростного счетчика после вхождения в режим RUN. Выполнение HDEF для скоростного счетчика во второй раз приводит к ошибке выполнения и не изменяет настройку счетчика по сравнению с тем, как она была выполнена для данного счетчика при первом выполнении HDEF.



Совет

Хотя приведенные далее последовательности показывают, как изменить направление, текущее и предустановленное значение по отдельности, вы можете изменить все эти настройки или любую их комбинацию в той же последовательности, устанавливая надлежащим образом SMB47, а затем выполняя команду HSC.

Инициализация режимов 0, 1 и 2

Следующие шаги описывают, как инициализировать HSC1 в качестве однофазного реверсивного счетчика с внутренним управлением направлением счета (режим 0, 1 или 2).

- Используйте бит памяти первого цикла для вызова подпрограммы, в которой будет выполняться операция по инициализации. Когда вы используете вызов подпрограммы, следующие циклы эту подпрограмму не вызывают, что сокращает время цикла и делает программу более структурированной.
- В подпрограмме инициализации загрузите SMB47 в соответствии с желаемой операцией управления. Например:

SMB47 = 16#F8	<i>дает следующие результаты:</i> Разблокирует счетчик Записывает новое текущее значение Записывает новое предустановленное значение Устанавливает прямое направление счета Настраивает входы пуска и сброса на активность при высоком уровне сигнала
---------------	--
- Выполните команду HDEF с входом HSC, установленным в 1, и входом MODE [режим], установленным в 0 при отсутствии внешнего сброса и пуска, 1 для внешнего сброса без пуска или 2 для внешнего сброса и пуска.
- Загрузите SMD48 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).
- Загрузите SMD52 (двойное слово) желаемым предустановленным значением.
- Чтобы распознавать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствии программе обработки прерывания прерывающее событие CV = PV (событие 13). Подробную информацию об обработке прерываний вы найдете в разделе, посвященном командам прерывания.
- Чтобы распознавать внешний сброс, запрограммируйте прерывание, поставив в соответствии программе обработки прерывания прерывающее событие «внешний сброс» (external reset) (событие 15).
- Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
- Выполните команду HSC, чтобы S7-200 запрограммировал HSC1.
- Выйдите из подпрограммы.

Инициализация режимов 3, 4 и 5

Следующие шаги описывают, как инициализировать HSC1 в качестве однофазного реверсивного счетчика с внешним управлением направлением счета (режим 3, 4 или 5):

1. Используйте бит памяти первого цикла для вызова подпрограммы, в которой будет выполняться операция по инициализации. Когда вы используете вызов подпрограммы, следующие циклы эту подпрограмму не вызывают, что сокращает время цикла и делает программу более структурированной.
2. В подпрограмме инициализации загрузите SMB47 в соответствии с желаемой операцией управления. Например:
 $SMB47 = 16\#F8$ *дает следующие результаты:*
 Разблокирует счетчик
 Записывает новое текущее значение
 Записывает новое предустановленное значение
 Устанавливает начальное направление счета HSC на прямой счет
 Настраивает входы пуска и сброса на активность при высоком уровне сигнала
3. Выполните команду HDEF с входом HSC, установленным в 1, и входом MODE [режим], установленным одним из следующих способов: 3 при отсутствии внешнего сброса и пуска, 4 для внешнего сброса без пуска или 5 для внешнего сброса и пуска.
4. Загрузите SMD48 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).
5. Загрузите SMD52 (двойное слово) желаемым предустановленным значением.
6. Чтобы распознать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствии с программой обработки прерывания прерывающее событие CV = PV (событие 13). Подробную информацию об обработке прерываний вы найдете в разделе, посвященном командам прерывания.
7. Чтобы распознавать изменения направления счета, запрограммируйте прерывание, поставив в соответствии с программой обработки прерывания прерывающее событие «изменение направления» (direction changed) (событие 14).
8. Чтобы распознавать внешний сброс, запрограммируйте прерывание, поставив в соответствии с программой обработки прерывания прерывающее событие «внешний сброс» (external reset) (событие 15).
9. Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
10. Выполните команду HSC, чтобы S7-200 запрограммировал HSC1.
11. Выйдите из подпрограммы.

Инициализация режимов 6, 7 и 8

Следующие шаги описывают, как инициализировать HSC1 в качестве двухфазного реверсивного счетчика с датчиками тактовых импульсов прямого и обратного направления счета (режим 6, 7 или 8):

1. Используйте бит памяти первого цикла для вызова подпрограммы, в которой будет выполняться операция по инициализации. Когда вы используете вызов подпрограммы, следующие циклы эту подпрограмму не вызывают, что сокращает время цикла и делает программу более структурированной.
2. В подпрограмме инициализации загрузите SMB47 в соответствии с желаемой операцией управления. Например:
 $SMB47 = 16\#F8$ *дает следующие результаты:*
 Разблокирует счетчик
 Записывает новое текущее значение
 Записывает новое предустановленное значение
 Устанавливает начальное направление счета HSC на прямой счет
 Настраивает входы пуска и сброса на активность при высоком уровне сигнала
3. Выполните команду HDEF с входом HSC, установленным в 1, и входом MODE [режим], установленным в одно из следующих состояний: 6 при отсутствии внешнего сброса и пуска, 7 для внешнего сброса без пуска или 8 для внешнего сброса и пуска.
4. Загрузите SMD48 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).

5. Загрузите SMD52 (двойное слово) желаемым предустановленным значением.
6. Чтобы распознать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие CV = PV (событие 13). См. раздел о прерываниях.
7. Чтобы распознавать изменения направления счета, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие «изменение направления» (direction changed) (событие 14).
8. Чтобы распознавать внешний сброс, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие «внешний сброс» (external reset) (событие 15).
9. Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
10. Выполните команду HSC, чтобы S7–200 запрограммировал HSC1.
11. Выйдите из подпрограммы.

Инициализация режимов 9, 10 и 11

Следующие шаги описывают, как инициализировать HSC1 в качестве квадратурного счетчика с фазами А и В (режим 9, 10 или 11):

1. Используйте бит памяти первого цикла для вызова подпрограммы, в которой будет выполняться операция по инициализации. Когда вы используете вызов подпрограммы, следующие циклы эту подпрограмму не вызывают, что сокращает время цикла и делает программу более структурированной.
2. В подпрограмме инициализации загрузите SMB47 в соответствии с желаемой операцией управления.

Пример (однократная скорость счета):

SMB47 = 16#FC *даёт следующие результаты:*

Разблокирует счетчик
 Записывает новое текущее значение
 Записывает новое предустановленное значение
 Устанавливает начальное направление счета HSC на прямой счет
 Настраивает входы пуска и сброса на активность при высоком уровне сигнала

Пример (четырёхкратная скорость счета):

SMB47 = 16#F8 *даёт следующие результаты:*

Разблокирует счетчик
 Записывает новое текущее значение
 Записывает новое предустановленное значение
 Устанавливает начальное направление счета HSC на прямой счет
 Настраивает входы пуска и сброса на активность при высоком уровне сигнала

3. Выполните команду HDEF с входом HSC, установленным в 1, и входом MODE [режим], установленным одним из следующих способов: 9 при отсутствии внешнего сброса и пуска, 10 для внешнего сброса без пуска или 11 для внешнего сброса и пуска.
4. Загрузите SMD48 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).
5. Загрузите SMD52 (двойное слово) желаемым предустановленным значением.
6. Чтобы распознать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие CV = PV (событие 13). Подробную информацию о разблокировании прерываний (ENI) вы найдете в разделе о командах прерывания.
7. Чтобы распознавать изменения направления счета, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие «изменение направления» (direction changed) (событие 14).
8. Чтобы распознавать внешний сброс, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие «внешний сброс» (external reset) (событие 15).
9. Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
10. Выполните команду HSC, чтобы S7–200 запрограммировал HSC1.
11. Выйдите из подпрограммы.

Инициализация режима 12

Следующие шаги описывают, как инициализировать HSC0 для счета импульсов, сгенерированных PTO0 (режим 12).

1. Используйте бит памяти первого цикла для вызова подпрограммы, в которой будет выполняться операция по инициализации. Когда вы используете вызов подпрограммы, следующие циклы эту подпрограмму не вызывают, что сокращает время цикла и делает программу более структурированной.
2. В подпрограмме инициализации загрузите SMB47 в соответствии с желаемой операцией управления. Например:

SMB37 = 16#F8	<i>дает следующие результаты:</i>
	Разблокирует счетчик
	Записывает новое текущее значение
	Записывает новое предустановленное значение
	Устанавливает направление счета на прямой счет
	Настраивает входы пуска и сброса на активность при высоком уровне сигнала
3. Выполните команду HDEF с входом HSC, установленным в 0, и входом MODE [режим], установленным в 12.
4. Загрузите SMD38 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).
5. Загрузите SMD42 (двойное слово) желаемым предустановленным значением.
6. Чтобы распознать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствии программе обработки прерывания прерывающее событие CV = PV (событие 13). Подробную информацию об обработке прерываний вы найдете в разделе, посвященном командам прерывания.
7. Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
8. Выполните команду HSC, чтобы S7-200 запрограммировал HSC1.
9. Выйдите из подпрограммы.

Изменение направления в режиме 0, 1, 2 или 12

Следующие шаги описывают, как сконфигурировать изменение направления для HSC1 как однофазного счетчика с внутренним управлением направлением счета (режим 0, 1, 2 или 12):

1. Загрузите SMB47, чтобы записать желаемое направление:

SMB47 = 16#90	Разблокирует счетчик
	Устанавливает направление счета HSC на обратный счет
SMB47 = 16#98	Разблокирует счетчик
	Устанавливает направление счета HSC на прямой счет
2. Выполните команду HSC, чтобы S7-200 запрограммировал HSC1.

Пример: Скоростной счетчик		
M A I N	<p>Network 1</p>	<p>Network 1 //В первом цикле вызвать SBR_0.</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 // В первом цикле настроить HSC1:</p> <p>//1. Разблокировать счетчик. // - Записать новое текущее значение. // - Записать новое предустановленное значение. // - Установить в качестве начального направления счета прямой счет. // - Сделать входы пуска и сброса активными при высоком уровне сигнала. // - Установить 4-кратную скорость.</p> <p>//2. Настроить HSC1 на квадратурный режим с входами сброса и пуска.</p> <p>//3. Сбросить текущее значение HSC1.</p> <p>//4. Ввести предустановленное значение HSC1, равное 50.</p> <p>//5. Программе обработки прерываний INT_0 поставить в соответствие событие 13 (текущее значение HSC1 = предустановленному)</p> <p>//6. Разблокировать все прерывания.</p> <p>//7. Запрограммировать HSC1.</p> <pre>LD SM0.1 MOVB 16#F8, SMB47 HDEF 1, 11 MOVD +0, SMD48 MOVD +50, SMD52 ATCH INT_0, 13 ENI HSC 1</pre>
I N T 0	<p>Network 1</p>	<p>Network 1 // Запрограммировать HSC1:</p> <p>//1. Сбросить текущее значение HSC1.</p> <p>//2. Только записать новое текущее значение и оставить HSC1 активным.</p> <pre>LD SM0.0 MOVD +0, SMD48 MOVB 16#C0, SMB47 HSC 1</pre>

Команда вывода импульсов



Управление позиционированием

Команда вывода импульсов (PLS) используется для управления функциями "Вывод последовательности импульсов" (PTO) и "Широтно-импульсная модуляция" (PWM), имеющимися в распоряжении скоростных выходов (Q0.0 и Q0.1).

Усовершенствованный Мастер управления позиционированием создает команды, настроенные на ваше приложение, что упрощает программирование ваших задач и позволяет использовать преимущества дополнительных функций CPU S7-200. Дополнительную информацию о мастере управления позиционированием вы найдете в главе 9.

Для создания своего собственного приложения для управления перемещением вы можете по-прежнему использовать старую команду PLS, однако линейное нарастание сигнала в PTO поддерживается только командами, создаваемыми усовершенствованным мастером управления позиционированием.

PTO предоставляет в распоряжение выходной сигнал в виде прямоугольных импульсов (с относительной длительностью импульсов 50%), период следования которых и количество определяется пользователем.

PWM предоставляет в распоряжение выходной импульсный сигнал с постоянным периодом следования и переменной относительной длительностью импульсов, причем период следования импульсов и их ширину определяет пользователь.

S7-200 имеет два генератора PTO/PWM, которые формируют или быструю последовательность импульсов, или широтно-модулированный импульсный сигнал. Один генератор поставлен в соответствие цифровому выходу Q0.0, другой генератор – цифровому выходу Q0.1. Определенный адрес специальной памяти (SM) хранит для каждого генератора следующие данные: управляющий байт (8-битовое значение), количество импульсов (32-битовое значение без знака), а также период следования импульсов и значение ширины импульса (16-битовое значение без знака).

Генераторы PTO и PWM и регистр образа процесса совместно используют Q0.0 и Q0.1. Когда функция PTO или PWM активна на Q0.0 или Q0.1, то выходом управляет генератор PTO или PWM, а нормальное использование выхода заблокировано. На форму выходного сигнала не влияет ни состояние регистра образа процесса, ни принудительное присваивание значений выходам, ни выполнение команд непосредственного вывода. Когда генератор PTO/PWM не активен, управление выходом возвращается регистру образа процесса. Регистр образа процесса определяет начальное и конечное состояние импульсного выходного сигнала, вызывая его начало и завершение на высоком или низком уровне.

Таблица 6-32. Допустимые операнды для команды вывода импульсов

Входы/выходы	Типы данных	Операнды
Q0.X	WORD	константа: 0 (= Q0.0) или 1 (= Q0.1)



Совет

Перед активизацией функции PTO или PWM установите значение регистра образа процесса для Q0.0 и Q0.1 в 0.

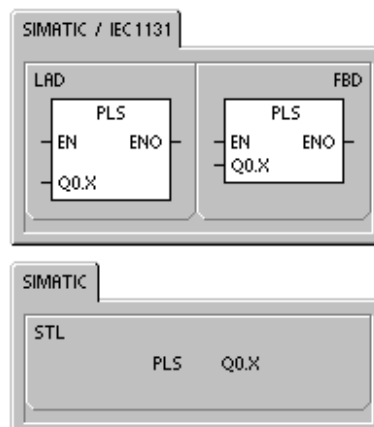
По умолчанию значения всех управляющих битов, периода следования импульсов, ширины и количества импульсов равны нулю.

Выходы PTO/PWM должны иметь минимальную нагрузку не менее 10% от номинальной нагрузки, чтобы обеспечить четкий переход от выключенного состояния к включенному и наоборот.



Советы по программированию

За программами, использующими команду PLS для функций PTO/PWM, обратитесь к советам по программированию на компакт-диске с документацией. См. советы 7, 22, 23, 30 и 50.



Последовательность импульсов (РТО)

Функция РТО обеспечивает генерирование последовательности импульсов прямоугольной формы (с относительной длительностью 50%) с заданным количеством импульсов и заданным периодом следования импульсов. (См. рис. 6–29.) РТО может генерировать отдельную последовательность импульсов или несколько последовательностей импульсов (используя профиль импульсов). Вы задаете количество импульсов и период следования импульсов (в микро- или миллисекундах):

- Количество импульсов: от 1 до 4 294 967 295
- Период следования импульсов:
 - от 10 мкс до 65 535 мкс
 - или
 - от 2 мс до 65 535 мс.

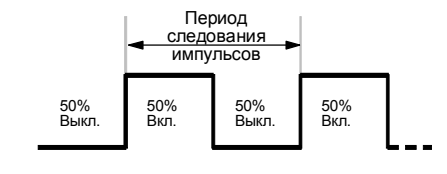


Рис. 6–29. Вывод последовательности импульсов (РТО)

Если при задании периода следования импульсов указывается нечетное число микросекунд или миллисекунд (напр., 75 мс), то это приводит к некоторому искажению относительной длительности импульсов.

Ограничения по количеству импульсов и периода следования импульсов приведены в таблице 6–33.

Таблица 6–33. Количество импульсов и период следования импульсов в функции РТО

Количество импульсов/Период следования импульсов	Реакция
Период следования импульсов < 2 единиц времени	Период следования импульсов устанавливается по умолчанию равным 2 единицам времени.
Количество импульсов = 0	Количество импульсов устанавливается по умолчанию равным 1 импульсу.

Функция РТО допускает сцепление, или конвейерную обработку последовательностей импульсов. Когда активная последовательность импульсов завершена, немедленно начинается вывод новой последовательности импульсов. Это обеспечивает непрерывность следующих друг за другом последовательностей импульсов.

Использование мастера управления позиционированием

Мастер управления позиционированием автоматически обрабатывает односегментные и многосегментные конвейерные последовательности импульсов РТО, широтно-импульсную модуляцию, конфигурацию адресов SM и создание таблицы профилей. Эта информация приведена здесь для вашего сведения. Мы рекомендуем вам использовать мастер управления позиционированием. Дополнительную информацию о мастере управления позиционированием вы найдете в главе 9.

Односегментная конвейерная обработка импульсов РТО

При односегментной конвейерной обработке вы несете ответственность за обновление ячеек области SM для следующей последовательности импульсов. Как только был запущен первый сегмент РТО, вы должны немедленно изменить ячейки SM в соответствии с требованиями второй последовательности и снова выполнить команду PLS. Атрибуты второй последовательности импульсов будут храниться в конвейере до завершения первой последовательности импульсов. В конвейере в каждый момент времени может храниться только одна запись. Как только завершится первая последовательность импульсов, начнется вывод второй последовательности, и конвейер становится доступным для задания характеристик новой последовательности импульсов. Вы можете затем повторить этот процесс, чтобы установить характеристики следующей последовательности импульсов.

Между последовательностями импульсов происходит плавный переход, если не меняется база времени и если активная последовательность импульсов не завершается раньше, чем настройка новой последовательности импульсов распознается при исполнении команды PLS.

Многосегментная конвейерная обработка импульсов РТО

При многосегментной конвейерной обработке S7-200 автоматически считывает характеристики каждого сегмента последовательности импульсов из таблицы профиля, расположенной в V-памяти. Используемыми в этом режиме адресами специальной памяти являются управляющий байт, байт состояния и начальное смещение в V-памяти таблицы профиля (SMW168 или SMW178). В качестве базы времени могут быть заданы микросекунды или миллисекунды, но этот выбор применяется ко всем значениям периода следования импульсов в таблице профиля и не может быть изменен, когда профиль активен. Многосегментный режим работы запускается исполнением команды PLS.

Запись для каждого сегмента имеет длину 8 байтов и состоит из 16-битового значения периода следования импульсов, 16-битового значения приращения периода и 32-битового значения количества импульсов. Формат таблицы профиля показан в 6-34. Вы можете автоматически увеличивать или уменьшать длительность периода следования импульсов, программируя для каждого импульса определенное значение. Положительное значение в поле приращения периода следования импульсов увеличивает период, отрицательное значение уменьшает период, а нулевое значение не меняет периода следования импульсов.

При обработке профиля РТО номер активного в данный момент сегмента находится в SMB166 (или SMB176).

Таблица 6-34. Формат таблицы профиля для многосегментных последовательностей импульсов

Смещение в байтах	Сегмент	Описание записей таблицы
0		Количество сегментов: от 1 до 255 ¹
1	#1	Начальный период следования импульсов (от 2 до 65 535 единиц базы времени)
3		Приращение периода следования импульсов на импульс (величина со знаком) (от -32768 до 32767 единиц базы времени)
5		Количество импульсов (от 1 до 4 294 967 295)
9	#2	Начальный период следования импульсов (от 2 до 65 535 единиц базы времени)
11		Приращение периода следования импульсов на импульс (величина со знаком) (от -32768 до 32767 единиц базы времени)
13		Количество импульсов (от 1 до 4 294 967 295)
(продолжение)	#3	(продолжение)

1 Ввод значения 0 для количества сегментов генерирует нефатальную ошибку. Вывод РТО не производится.

Широтно-импульсная модуляция (PWM)

Функция PWM обеспечивает фиксированный период следования импульсов при переменной относительной длительности импульсов. (См. рис. 6-30.) Период следования и ширину импульсов можно задавать в микро- или миллисекундах:

- Период следования импульсов: от 10 мкс до 65 535 мкс
или от 2 мс до 65 535 мс
- Ширина импульса: от 0 мкс до 65 535 мкс или от 0 мс до 65 535 мс

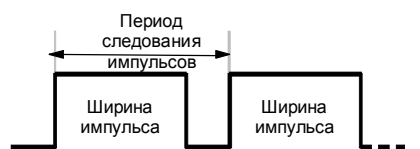


Рис. 6-30. Широтно-импульсная модуляция (PWM)


Как показано в таблице 6-35, при установке ширины импульса равной периоду следования импульсов (что делает относительную длительность импульсов равной 100 процентам) выход включен постоянно. При установке ширины импульса равной.

Таблица 6-35. Ширина и период следования импульсов и реакции функции PWM

Ширина импульса/период следования импульсов	Реакция
Ширина импульса \geq периоду следования импульсов	Относительная длительность импульса равна 100%: выход включен постоянно.
Ширина импульса = 0	Относительная длительность импульса равна 0%: выход выключен.
Период следования импульсов $<$ 2 единиц времени	Период следования импульсов устанавливается по умолчанию равным двум единицам времени.

Имеется два способа изменения характеристик импульсов PWM:

- Синхронное обновление: Если не требуется изменения базы времени, то может быть выполнено синхронное обновление. При синхронном обновлении изменение характеристик импульсов происходит на границе периода следования импульсов, обеспечивая плавный переход.
- Асинхронное обновление: Обычно при работе PWM меняется ширина импульсов при постоянном периоде следования, так что изменение базы времени не требуется. Однако, если требуется изменение базы времени генератора PTO/PWM, то используется асинхронное обновление. Асинхронное обновление вызывает на мгновение блокирование генератора PTO/PWM асинхронно со следованием импульсов PWM. Это может вызвать нежелательную неустойчивость работы управляемого устройства. Поэтому рекомендуется синхронное обновление PWM. Выбирайте базу времени, которая будет работать со всеми ожидаемыми вами значениями периода следования импульсов.

	<p>Совет</p> <p>Бит метода обновления PWM (SM67.4 или SM77.4) в управляющем байте определяет тип обновления, используемый при исполнении команды PLS для реализации изменений. Если изменяется база времени, то произойдет асинхронное обновление независимо от состояния бита метода обновления PWM.</p>
---	--

Конфигурирование и управление функциями PTO/PWM с помощью битов специальной памяти

Команда PLS считывает данные, хранящиеся в указанных ячейках специальной памяти (SM), и соответствующим образом программирует генератор PTO/PWM. SMB67 управляет PTO 0 или PWM 0, а SMB77 управляет PTO 1 или PWM 1. Таблица 6–36 описывает регистры, используемые для управления функционированием PTO/PWM. Вы можете использовать таблицу 6–37 в качестве быстрой справки, чтобы определить значение, которое следует поместить в управляющий регистр PTO/PWM, чтобы вызвать желаемую операцию.

Вы можете изменить характеристики сигнала PTO или PWM, изменяя биты памяти в области SM (включая управляющий байт), а затем исполняя команду PLS. Вы можете в любое время заблокировать генерирование импульсов PTO или PWM, записав 0 в бит деблокировки PTO/PWM управляющего бита (SM67.7 или SM77.7), а затем исполняя команду PLS.

Бит бездействия PTO в байте состояния (SM66.7 или SM76.7) предназначен для указания на завершение запрограммированной последовательности импульсов. Кроме того, по окончании последовательности импульсов может быть вызвана программа обработки прерываний. (По этому вопросу читайте описания команд прерывания и обмена данными.) Если вы используете многосегментный режим, то команда обработки прерываний вызывается после завершения таблицы профиля.

Следующие условия устанавливают SM66.4 (или SM76.4) и SM66.5 (или SM76.5):

- Если вы задаете приращение периода следования импульсов, которое через некоторое количество импульсов приведет к недопустимой длительности периода, то генерируется условие арифметического переполнения, которое завершает функцию PTO и устанавливает бит ошибки вычисления приращения (SM66.4 или SM76.4) в 1. Выход снова управляется через регистр образа процесса.
- Если вы обрабатываемый профиль PTO завершаете (деактивизируете) вручную, то в 1 устанавливается бит завершения по инициативе пользователя (SM66.5 или SM76.5).
- Если вы пытаетесь загрузить конвейерную обработку, когда конвейер полон, то в 1 устанавливается бит переполнения PTO (SM66.6 или SM76.6). Если вы хотите обнаруживать последующие переполнения, то при обнаружении переполнения вы должны сбросить этот бит вручную. Этот бит устанавливается в 0 при переходе в режим RUN.


	<p>Совет</p> <p>Если вы собираетесь загрузить новое количество импульсов (SMD72 или SMD82), ширину импульсов (SMW70 или SMW80) или период следования импульсов (SMW68 или SMW78), то установите также соответствующие биты обновления в управляющем регистре перед выполнением команды PLS. Если вы используете многосегментную последовательность импульсов, то перед выполнением команды PLS вы также должны загрузить начальное смещение (SMW168 или SMW178) таблицы профиля и значения таблицы профиля.</p>
---	--

Таблица 6–36. Биты специальной памяти для управляющих регистров PTO / PWM

Q0.0	Q0.1	Биты состояния
SM66.4	SM76.4	Профиль PTO прерван (ошибка расчета приращения): 0 = нет ошибки 1 = прерван
SM66.5	SM76.5	Профиль PTO прерван по команде пользователя: 0 = нет завершения 1 = прерван
SM66.6	SM76.6	Переполнение/потеря значимости в конвейере PTO: 0 = нет переполнения 1 = переполнение/потеря значимости
SM66.7	SM76.7	PTO не действует: 0 = действует 1 = PTO не действует
Q0.0	Q0.1	Управляющие биты
SM67.0	SM77.0	PTO/PWM: обновить период следования импульсов: 0 = не обновлять 1 = обновить период следования импульсов
SM67.1	SM77.1	PWM: обновить значение ширины импульсов: 0 = не обновлять 1 = обновить ширину импульсов
SM67.2	SM77.2	PTO: обновить количество импульсов: 0 = не обновлять 1 = обновить количество импульсов
SM67.3	SM77.3	PTO/PWM: выбрать базу времени: 0 = 1 мкс/такт 1 = 1 мс/такт
SM67.4	SM77.4	PWM: метод обновления: 0 = асинхронное обновление 1 = синхронное обновление
SM67.5	SM77.5	PTO: односегментный/многосегментный режим: 0 = односегментный 1 = многосегментный
SM67.6	SM77.6	Выбор PTO или PWM: 0 = PTO 1 = PWM
SM67.7	SM77.7	Активизация PTO/PWM: 0 = заблокировать 1 = активизировать
Q0.0	Q0.1	Другие регистры PTO/PWM
SMW68	SMW78	Значение периода следования импульсов PTO/PWM диапазон: от 2 до 65535
SMW70	SMW80	Значение ширины импульса PWM диапазон: от 0 до 65535
SMD72	SMD82	Значение количества импульсов PTO диапазон: от 1 до 4 294 967 295
SMB166	SMB176	Номер активного сегмента PTO только в многосегментном режиме
SMW168	SMW178	Начальный адрес таблицы профиля PTO (байтовое смещение от V0) только в многосегментном режиме
SMB170	SMB180	Байт состояния линейного профиля
SMB171	SMB181	Результирующий регистр линейного профиля
SMD172	SMD182	Регистр частот, ручной режим

Таблица 6–37. Справочные данные об управляющем байте PTO/PWM

Управляющий регистр (16-ричное значение)	Деблокировка	Результат выполнения операции PLS						
		Функция	Сегментирование PTO	Метод обновления PWM	База времени	Кол-во импульсов	Ширина импульса	Период следования импульсов
16#81	Да	PTO	Односегм.		1 мкс/цикл			Загружен
16#84	Да	PTO	Односегм.		1 мкс/цикл	Загружено		
16#85	Да	PTO	Односегм.		1 мкс/цикл	Загружено		Загружен
16#89	Да	PTO	Односегм.		1 мс/цикл			Загружен
16#8C	Да	PTO	Односегм.		1 мс/цикл	Загружено		
16#8D	Да	PTO	Односегм.		1 мс/цикл	Загружено		Загружен
16#A0	Да	PTO	Многосегм.		1 мкс/цикл			
16#A8	Да	PTO	Многосегм.		1 мс/цикл			
16#D1	Да	PWM		Синхрон.	1 мкс/цикл			Загружен
16#D2	Да	PWM		Синхрон.	1 мкс/цикл		Загруж.	
16#D3	Да	PWM		Синхрон.	1 мкс/цикл		Загруж.	Загружен
16#D9	Да	PWM		Синхрон.	1 мс/цикл			Загружен
16#DA	Да	PWM		Синхрон.	1 мс/цикл		Загруж.	
16#DB	Да	PWM		Синхрон.	1 мс/цикл		Загруж.	Загружен

Расчет значений таблицы профилей

Возможность многосегментного режима конвейерной обработки генераторов РТО/PWM может быть полезной во многих приложениях, в частности, при управлении шаговыми двигателями.

Например, вы можете использовать РТО с профилем импульсов для управления шаговым двигателем с помощью простой последовательности, состоящей из разгона, рабочего режима и торможения, или значительно более сложных последовательностей. Для этого вы определяете профиль импульсов, который может включать в себя до 255 сегментов, причем каждый сегмент соответствует некоторой функции, напр., разгон, рабочий режим и торможение.

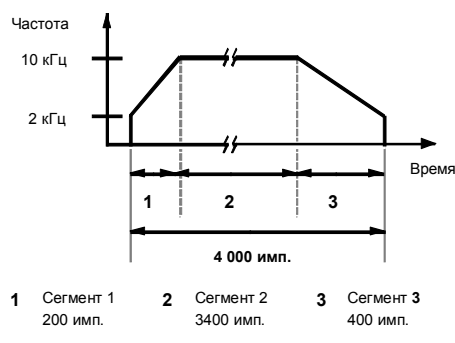


Рис. 6–31. Частотно-временная диаграмма

На рис. 6–31 показан пример значений таблицы профиля, необходимых для генерирования последовательности выходных импульсов, которая разгоняет шаговый двигатель (сегмент 1), обеспечивает его работу с постоянной скоростью (сегмент 2), а затем тормозит двигатель (сегмент 3).

В этом примере: Начальная и конечная частота следования импульсов равна 2 кГц, максимальная частота следования импульсов равна 10 кГц, и для достижения желаемого количества оборотов двигателя требуется 4 000 импульсов. Так как величины в таблице профиля выражаются в терминах периода следования импульсов, а не в частоте, то вы должны преобразовать заданные значения частоты в значения периода следования импульсов. Тогда начальное и конечное значение периода следования импульсов составит 500 мкс, а период следования импульсов, соответствующий максимальной частоте, равен 100 мкс. На интервале ускорения выходного профиля максимальная частота следования импульсов должна быть достигнута примерно через 200 импульсов. Интервал замедления профиля должен быть завершен примерно за 400 импульсов.

Для данного сегмента может быть использована следующая формула для определения приращения периода следования импульсов, которую генератор РТО/PWM использует для настройки периода каждого импульса:

Приращение периода следования импульсов для сегмента =

$$\frac{|\text{End_CT}_{\text{seg}} - \text{Init_CT}_{\text{seg}}|}{\text{Quantity}_{\text{seg}}}$$

где: End_CT_{seg} = конечное значение периода следования импульсов для данного сегмента
 Init_CT_{seg} = начальное значение периода следования импульсов для данного сегмента
 Quantity_{seg} = количество импульсов в данном сегменте

С помощью этой формулы можно рассчитать значения приращений периода следования импульсов для данного примера применения:

Сегмент 1 (ускорение):

Приращение периода следования импульсов = -2

Сегмент 2 (постоянная скорость):

Приращение периода следования импульсов = 0

Сегмент 3 (замедление):

Приращение периода следования импульсов = 1

В таблице 6–38 приведены значения для генерирования последовательности импульсов из примера (предполагается, что таблица профиля находится в памяти переменных, начиная с V500). Вы можете включить в свою программу команды для загрузки этих значений в память переменных, или вы можете определить значения профиля в блоке данных.

Таблица 6–38. Значения таблицы профиля

Адрес	Значение	Описание	
VB500	3	Общее количество сегментов	
VW501	500	Начальный период следования импульсов	Сегмент 1
VW503	-2	Приращение начального периода следования импульсов	
VD505	200	Количество импульсов	
VW509	100	Начальный период следования импульсов	Сегмент 2
VW511	0	Приращение периода следования импульсов	
VD513	3400	Количество импульсов	
VW517	100	Начальный период следования импульсов	Сегмент 3
VW519	1	Приращение периода следования импульсов	
VD521	400	Количество импульсов	

Чтобы определить, приемлемы ли переходы между сегментами последовательности импульсов, вам нужно знать период для последнего импульса сегмента. Если приращение периода следования импульсов не равно 0, вы должны рассчитать период для последнего импульса сегмента, так как это значение не указано в профиле. Для расчета периода для последнего импульса пользуйтесь следующей формулой:

$$\text{Период последнего импульса сегмента} = \text{Init_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$$

где: $\text{Init_CT}_{\text{seg}}$ = начальный период следования импульсов для этого сегмента

$\text{Delta}_{\text{seg}}$ = приращение периода следования импульсов для этого сегмента

$\text{Quantity}_{\text{seg}}$ = количество импульсов в этом сегменте

В то время как приведенный выше упрощенный пример полезен в качестве введения, реальные приложения могут потребовать более сложных профилей последовательностей импульсов. Помните, что приращение периода следования импульсов может быть задано только как целое количество микросекунд или миллисекунд, и изменение периода выполняется на каждом импульсе.

Влияние этих двух пунктов состоит в том, что расчет приращения периода следования импульсов для данного сегмента может потребовать итеративного подхода. Может потребоваться некоторая гибкость в значении конечного периода следования импульсов или количества импульсов для данного сегмента.

В процессе определения правильных значений таблицы профиля может быть полезна длительность данного сегмента профиля. Время, необходимое для завершения данного сегмента профиля, может быть рассчитано с помощью следующей формулы:

$$\text{Длительность сегмента} = \text{Quantity}_{\text{seg}} * (\text{Init_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$$

где: $\text{Quantity}_{\text{seg}}$ = количество импульсов в данном сегменте

$\text{Init_CT}_{\text{seg}}$ = начальный период следования импульсов для данного сегмента

$\text{Delta}_{\text{seg}}$ = приращение периода следования импульсов для данного сегмента

Арифметические операции

Операции сложения, вычитания, умножения и деления

Сложение

$IN1 + IN2 = OUT$
 $IN1 + OUT = OUT$

Вычитание

$IN1 - IN2 = OUT$ *LAD u FBD*
 $OUT - IN1 = OUT$ *STL*

Команды сложения целых чисел (+I) и вычитания целых чисел (-I) складывают или вычитают два 16-битовых целых числа и дают 16-битовый результат. Команды сложения двойных целых чисел (+D) и вычитания двойных целых чисел (-D) складывают или вычитают два 32-битовых целых числа и дают 32-битовый результат. Команды сложения вещественных чисел (+R) и вычитания вещественных чисел (-R) складывают или вычитают два 32-битовых вещественных числа и дают результат в виде 32-битового вещественного числа.

Умножение

$IN1 * IN2 = OUT$
 $IN1 * OUT = OUT$

Деление

$IN1 / IN2 = OUT$ *LAD u FBD*
 $OUT / IN1 = OUT$ *STL*

Команды умножения целых чисел (*I) и деления целых чисел (/I) перемножают или делят два 16-битовых целых числа и дают 16-битовый результат. (Для деления остаток не сохраняется.) Команды умножения двойных целых чисел (*D) и деления двойных целых чисел (/D) перемножают или делят два 32-битовых целых числа и дают 32-битовый результат. (Для деления остаток не сохраняется.) Команды умножения вещественных чисел (*R) и деления вещественных чисел (/R) перемножают или делят два 32-битовых вещественных числа и дают в результате 32-битовое вещественное число.

Биты специальной памяти и ENO

SM1.1 указывает на ошибки переполнения и недопустимые значения. Если бит SM1.1 установлен, то состояние SM1.0 и SM1.2 не имеет значения, и первоначальные входные операнды не меняются. Если SM1.1 и SM1.3 не установлены, то арифметическая операция завершилась с допустимым результатом, и SM1.0 и SM1.2 содержат допустимый статус. Если во время операции деления устанавливается SM1.3, то все остальные биты состояния арифметической операции остаются неизменными.

Сбойные состояния, устанавливающие ENO = 0

- SM1.1 (переполнение)
- SM1.3 (деление на ноль)
- 0006 (косвенный адрес)

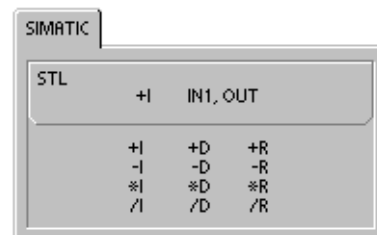
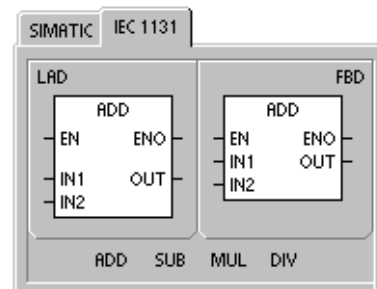
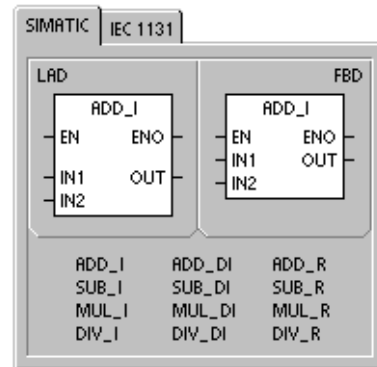
Биты специальной памяти, на которые действует команда

- SM1.0 (ноль)
- SM1.1 (переполнение, во время операции получено недопустимое значение или обнаружен недопустимый входной параметр)
- SM1.2 (отрицательное число)
- SM1.3 (деление на ноль)

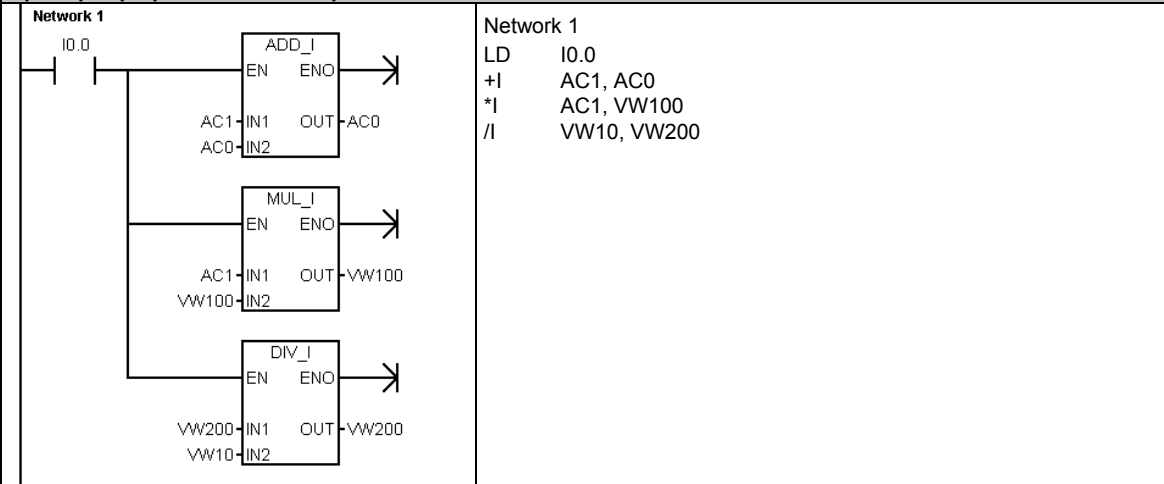
Таблица 6–39. Допустимые операнды для команд сложения, вычитания, умножения и деления

Входы/выходы	Типы данных	Операнды
IN1, IN2	INT DINT REAL	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, константа ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
OUT	INT DINT, REAL	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, *VD, *AC, *LD ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Вещественные числа (или числа с плавающей точкой) представляются в формате, описанном в стандарте ANSI/IEEE 754-1985 (одинарная точность). За дополнительной информацией обратитесь к этому стандарту.



Пример: Арифметические операции с целыми числами

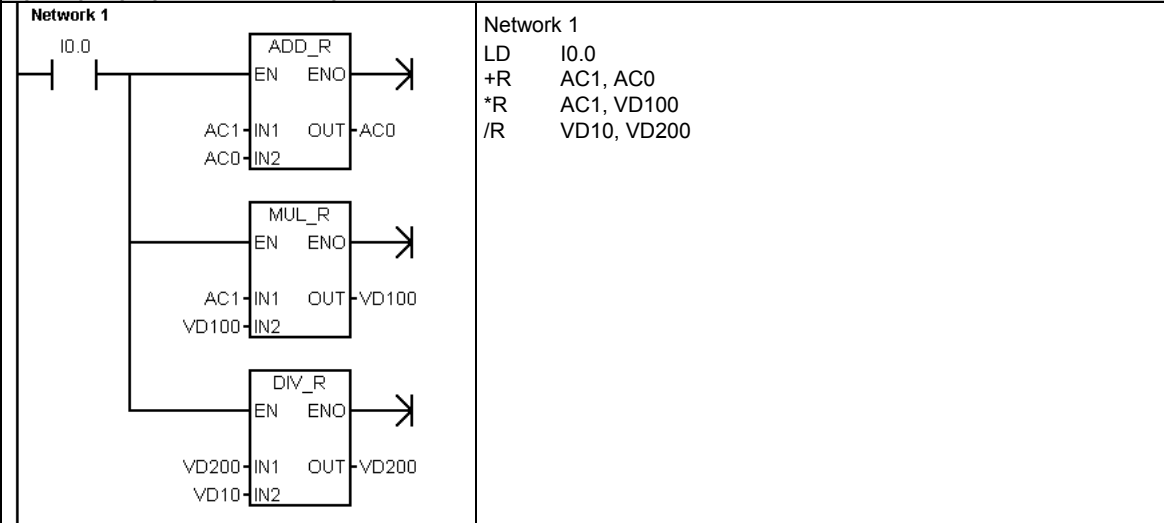


```

Network 1
LD I0.0
+I AC1, AC0
*I AC1, VW100
/I VW10, VW200
    
```

Сложение	Умножение	Деление
$40 + 60 = 100$ AC1 AC0 AC0	$40 * 20 = 800$ AC1 VW100 VW100	$4000 / 40 = 100$ VW200 VW10 VW200

Пример: Арифметические операции с вещественными числами



```

Network 1
LD I0.0
+R AC1, AC0
*R AC1, VD100
/R VD10, VD200
    
```

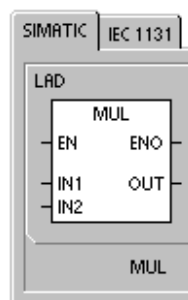
Сложение	Умножение	Деление
$4000.0 + 6000.0 = 10000.0$ AC1 AC0 AC0	$400.0 * 200.0 = 80000.0$ AC1 VD100 VD100	$4000.0 / 41.0 = 97.5609$ VD200 VD10 VD200

Умножение целых чисел с представлением результата в виде двойного целого числа и деление целых чисел с остатком

Умножение целых чисел с представлением результата в виде двойного целого числа

IN1 * IN2 = OUT LAD и FBD
 IN1 * OUT = OUT STL

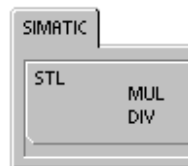
Команда умножения целых чисел с представлением результата в виде двойного целого числа (MUL) перемножает два 16-битовых целых числа и дает 32-битовое произведение. В команде умножения STL младшее слово (16 битов) 32-битового выхода OUT используется как один из сомножителей.



Деление целых чисел с остатком

IN1 / IN2 = OUT LAD и FBD
 OUT / IN1 = OUT STL

Команда деления целых чисел с остатком (DIV) делит два 16-битовых целых числа и дает 32-битовый результат, состоящий из 16-битового остатка (старшее слово) и 16-битового частного (младшее слово).



В команде деления STL младшее слово (16 битов) 32-битового выхода OUT используется как делимое.

Биты специальной памяти и ENO

Для обеих команд, представленных на этой странице, биты специальной памяти (SM) указывают на ошибки и недопустимые значения. Если во время операции деления устанавливается SM1.3 (деление на ноль), то все остальные биты состояния арифметической операции остаются неизменными. В противном случае все поддерживаемые биты состояния арифметической операции содержат после завершения арифметической операции допустимый статус.

Сбойные состояния, устанавливающие ENO = 0

- SM1.1 (переполнение)
- SM1.3 (деление на ноль)
- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда

- SM1.0 (ноль)
- SM1.1 (переполнение)
- SM1.2 (отрицательное число)
- SM1.3 (деление на ноль)

Таблица 6–40. Допустимые операнды для умножения целых чисел с представлением результата в виде двойного целого числа и деления целых чисел с остатком

Входы/выходы	Типы данных	Операнды
IN1, IN2	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
OUT	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Пример: Умножение целых чисел с представлением результата в виде двойного целого числа и деление целых чисел с остатком

Network 1

Network 1

```
LD IO.0
MUL AC1, VD100
DIV VW10, VD200
```

Умножение целых чисел с представлением результата в виде двойного целого числа	400	*	200	=	80000
	AC1		VW102		VD100

Деление целых чисел с остатком	4000	/	41	=	23	97
	VW202		VW10		остат. VW200	частн. VW202
						VD200

Примечание: VD100 содержит VW100 и VW102, а VD200 содержит VW200 и VW202.

Числовые функции

Синус, косинус и тангенс

Команды вычисления синуса (SIN), косинуса (COS) и тангенса (TAN) вычисляют тригонометрическую функцию угловой величины IN и помещают результат в OUT. Входной угол задается в радианах.

$$\text{SIN (IN)} = \text{OUT} \quad \text{COS (IN)} = \text{OUT} \quad \text{TAN (IN)} = \text{OUT}$$

Для преобразования угла из градусов в радианы: Используйте MUL_R (*R) для умножения угла в градусах на 1.745329E-2 (примерно на $\pi/180$).

Натуральный логарифм и натуральная экспонента

Команда вычисления натурального логарифма находит натуральный логарифм числа в IN и помещает результат в OUT.

Команда вычисления натуральной экспоненты выполняет экспоненциальную операцию по возведению e в степень, заданную значением в IN, и помещает результат в OUT.

$$\text{LN (IN)} = \text{OUT} \quad \text{EXP (IN)} = \text{OUT}$$

Для получения десятичного логарифма из натурального: Разделите натуральный логарифм на 2,302585 (примерно натуральный логарифм 10).

Для возведения любого вещественного числа в степень, заданную другим вещественным числом, включая дробные: Объедините команду вычисления натуральной экспоненты с командой вычисления натурального логарифма. Например, для возведения X в степень Y введите следующую команду: EXP (Y * LN (X)).

Квадратный корень

Команда вычисления квадратного корня (SQRT) извлекает квадратный корень из вещественного числа (IN) и дает результат в виде вещественного числа OUT.

$$\text{SQRT (IN)} = \text{OUT}$$

Для получения других корней:

$$5 \text{ в кубе} = 5^3 = \text{EXP}(3 * \text{LN}(5)) = 125$$

$$\text{Кубический корень из } 125 = 125^{(1/3)} = \text{EXP}((1/3) * \text{LN}(125)) = 5$$

$$\text{Квадратный корень из } 5 \text{ в кубе} = 5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11.18034$$

Биты специальной памяти и ENO для числовых операций

Для всех команд, описанных на этой странице, SM1.1 используется для указания на ошибки переполнения и недопустимые значения. Если SM1.1 установлен, то состояние SM1.0 и SM1.2 не имеет значения, первоначальные входные операнды не изменяются. Если SM1.1 не установлен, то арифметическая операция завершилась с допустимым результатом, а SM1.0 и SM1.2 содержат действительное состояние.

Сбойные состояния, устанавливающие ENO = 0

- SM1.1 (переполнение)
- 0006 (косвенный адрес)

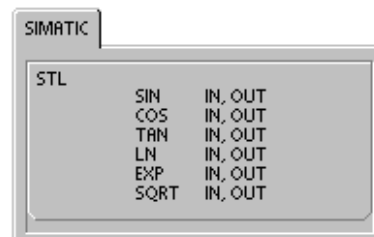
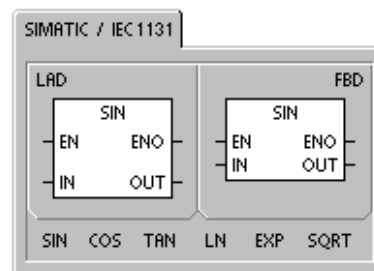
Биты специальной памяти, на которые действует команда

- SM1.0 (ноль)
- SM1.1 (переполнение)
- SM1.2 (отрицательное число)

Таблица 6–41. Допустимые операнды числовых функций

Входы/выходы	Типы данных	Операнды
IN	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
OUT	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Вещественные числа (или числа с плавающей точкой) представляются в формате, описанном в стандарте ANSI/IEEE 754-1985 (одинарная точность). За дополнительной информацией обратитесь к этому стандарту.



Команды инкрементирования и декрементирования Marker 22

Увеличение на 1

IN + 1 = OUT LAD u FBD
 OUT + 1 = OUT STL

Уменьшение на 1

IN - 1 = OUT LAD u FBD
 OUT - 1 = OUT STL

Команды инкрементирования и декрементирования прибавляют 1 к входу IN или вычитают из него 1 и помещают результат в переменную OUT.

Операции увеличения байта на 1 (INCB) и уменьшения байта на 1 (DECB) являются беззнаковыми.

Операции увеличения слова на 1 (INCW) и уменьшения слова на 1 (DECW) учитывают знак.

Операции увеличения двойного слова на 1 (INCD) и уменьшения двойного слова на 1 (DECD) учитывают знак.

Сбойные состояния, устанавливающие ENO = 0 :

- SM1.1 (переполнение)
- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.0 (ноль)
- SM1.1 (переполнение)
- SM1.2 (отрицательное число) для операций со словами и двойными словами

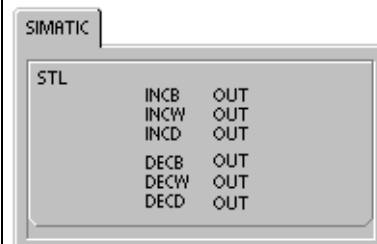
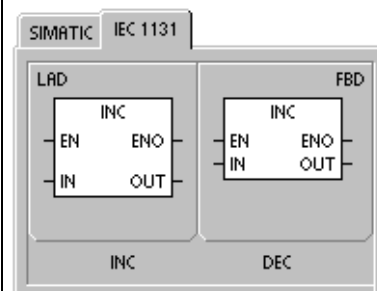
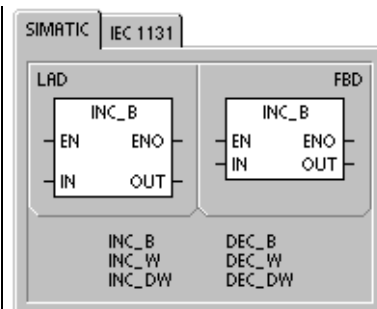


Таблица 6–42. Допустимые операнды для команд инкрементирования и декрементирования

Входы/выходы	Типы данных	Операнды
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Пример: Команды инкрементирования и декрементирования

Network 1

Network 1

```
LD I4.0
INCW AC0
DECD VD100
```

Увеличение слова на 1	125	+ 1 =	126
	AC0		AC0
Уменьшение двойного слова на 1	128000	- 1 =	127999
	VD100		VD100

Пропорционально-интегрально-дифференциальный (PID) регулятор

Команда "PID-регулятор" выполняет расчет контура PID-регулятора для заданного контура регулирования LOOP с помощью данных о входных величинах и конфигурации в таблице (TBL).

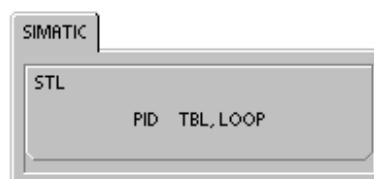
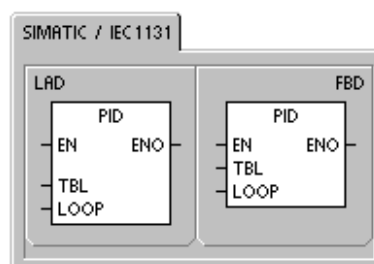
Сбойные состояния, устанавливающие ENO = 0 :

- SM1.1 (переполнение)
- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.1 (переполнение)

Команда PID-регулятор (пропорционально-интегрально-дифференциальный регулятор) предназначена для расчета PID-регуляторов. Чтобы эти расчеты можно было выполнять, вершина логического стека (TOS) должна быть активизирована (поток сигнала). Команда имеет два операнда: TABLE, являющийся начальным адресом таблицы с данными контура регулирования, и LOOP – номер контура регулирования, являющийся константой от 0 до 7.



В программе можно использовать до восьми команд PID. Если две или более команд PID используются с одним и тем же номером контура регулирования (даже если у них разные номера таблиц), то расчеты PID-регуляторов будут влиять друг на друга, и выход будет непредсказуемым.

Таблица контура регулирования хранит девять параметров, используемых для управления и контроля за работой контура регулирования. Сюда входят текущее и предыдущее значение регулируемой переменной (фактическое значение), заданное значение, регулирующее воздействие (выход), коэффициент усиления, период квантования, постоянная времени интегрирования (или время интегрирования), постоянная времени воздействия по производной (скорость) и интегральная сумма (смещение).

Для выполнения расчета PID-регулятора с желаемым временем квантования команда PID-регулятор должна выполняться или из программы обработки прерывания по времени, или из главной программы со скоростью, управляемой таймером. Время квантования должно подаваться как вход в команду PID-регулятор через таблицу контура регулирования.

В команду PID-регулятор встроена возможность автоматической настройки. В главе 15 вы найдете подробное описание автонастройки. Панель управления настройкой PID работает только с PID-регуляторами, созданными с помощью мастера PID.

Таблица 6-43. Допустимые операнды для PID-регулятора

Входы/выходы	Типы данных	Операнды
TBL	BYTE	VB
LOOP	BYTE	константа (от 0 до 7)



Мастер команд

STEP 7-Micro/WIN предоставляет в распоряжение мастер PID для руководства вами при определении PID-алгоритма для процесса с замкнутым контуром управления. Выберите команду меню **Tools** → **Instruction Wizard** [Инструментальные средства → **Мастер команд**], а затем выберите **PID** из окна Instruction Wizard [Мастер команд].



Совет

Заданное значение нижней границы и заданное значение верхней границы должны соответствовать верхней и нижней границе регулируемой переменной.

PID-алгоритм

В установленном режиме PID-регулятор управляет своим выходом (регулирующим воздействием) таким образом, чтобы свести ошибку регулирования (e) к нулю. Мерой ошибки является разность между заданным значением (setpoint, SP) и значением регулируемой переменной (process variable, PV) (фактическое значение). Принцип PID-регулятора основан на следующем уравнении, которое выражает регулирующее воздействие $M(t)$ как функцию пропорциональной составляющей, интегральной составляющей и дифференциальной составляющей:

Выход (регулирующее воздействие)	=	Пропорциональная составляющая	+	Интегральная составляющая	+	Дифференциальная составляющая
$M(t)$	=	$K_C * e$	+	$K_C \int_0^t e dt + M_{нач}$	+	$K_C * de/dt$
где:	$M(t)$	- регулирующее воздействие (выход регулятора) как функция времени				
	K_C	- коэффициент усиления контура регулирования				
	e	- ошибка регулирования (разность между заданным значением и регулируемой переменной)				
	$M_{нач}$	- начальное значение регулирующего воздействия				

Чтобы реализовать эту функцию управления в цифровой вычислительной машине, должно быть выполнено квантование непрерывной функции в соответствии с периодическими замерами значения ошибки с последующим расчетом регулирующего воздействия. Соответствующее уравнение, являющееся основой для решения на цифровой вычислительной машине, имеет вид:

M_n	=	$K_C * e_n$	+	$K_I * \sum_{x=1}^n e_x + M_{нач}$	+	$K_D * (e_n - e_{n-1})$
Выход (регулирующее воздействие)	=	Пропорциональная составляющая	+	Интегральная составляющая	+	Дифференциальная составляющая
где:	M_n	- расчетное значение регулирующего воздействия в момент квантования n				
	K_C	- коэффициент усиления контура регулирования				
	e_n	- значение ошибки регулирования в момент квантования n				
	e_{n-1}	- предыдущее значение ошибки регулирования (в момент квантования $n - 1$)				
	K_I	- коэффициент пропорциональности интегральной составляющей				
	$M_{нач}$	- начальное значение регулирующего воздействия				
	K_D	- коэффициент пропорциональности дифференциальной составляющей				

Из этого уравнения следует, что интегральная составляющая является функцией всех составляющих ошибки от первого до текущего отсчета. Дифференциальная составляющая является функцией текущего и предыдущего отсчета, тогда как пропорциональная составляющая является функцией только текущего отсчета. В цифровой вычислительной машине нецелесообразно хранить все отсчеты ошибки регулирования, да в этом и нет необходимости.

Так как компьютер должен вычислять регулирующее воздействие каждый раз, как опрашивается значение ошибки, начиная с первого отсчета, то необходимо сохранять только предыдущее значение ошибки и предыдущее значение интегральной составляющей. Как результат повторяющейся природы компьютерного решения, может быть получено упрощение уравнения, подлежащего решению в каждый момент квантования. Упрощенное уравнение имеет вид:

M_n	=	$K_C * e_n$	+	$K_I * e_n + MX$	+	$K_D * (e_n - e_{n-1})$
Выход (регулирующее воздействие)	=	Пропорциональная составляющая	+	Интегральная составляющая	+	Дифференциальная составляющая
где:	M_n	- расчетное значение регулирующего воздействия в момент квантования n				
	K_C	- коэффициент усиления контура регулирования				
	e_n	- значение ошибки регулирования в момент квантования n				
	e_{n-1}	- предыдущее значение ошибки регулирования (в момент квантования $n - 1$)				
	K_I	- коэффициент пропорциональности интегральной составляющей				
	MX	- предыдущее значение интегральной составляющей (в момент квантования $n - 1$)				
	K_D	- коэффициент пропорциональности дифференциальной составляющей				

S7-200 использует модифицированную форму упрощенного выше уравнения при расчете регулирующего воздействия. Это модифицированное уравнение имеет вид:

M_n	=	MP_n	+	MI_n	+	MD_n
Выход (регулирующее воздействие)	=	Пропорциональная составляющая	+	Интегральная составляющая	+	Дифференциальная составляющая

где: M_n - расчетное значение регулирующего воздействия в момент квантования n
 MP_n - значение пропорциональной составляющей регулирующего воздействия в момент квантования n
 MI_n - значение интегральной составляющей регулирующего воздействия в момент квантования n
 MD_n - значение дифференциальной составляющей регулирующего воздействия в момент квантования n

Пропорциональная составляющая в уравнении PID-регулятора

Пропорциональная составляющая MP – это произведение коэффициента усиления (K_C), определяющего точность расчета регулирующего воздействия, и ошибки регулирования (e), представляющей собой разность между заданным значением (SP) и регулируемой переменной (PV) в данный момент квантования. Уравнение для пропорциональной составляющей, решаемое S7-200, имеет вид:

MP_n	=	K_C	*	$(SP_n - PV_n)$
где: MP_n	-	значение пропорциональной составляющей регулирующего воздействия в момент квантования n		
K_C	-	коэффициент усиления контура регулирования		
SP_n	-	заданное значение регулируемой величины в момент квантования n		
PV_n	-	значение регулируемой переменной в момент квантования n		

Интегральная составляющая в уравнении PID-регулятора

Интегральная составляющая MI пропорциональна сумме ошибок за все время управления. Уравнение для интегральной составляющей, решаемое S7-200, имеет вид:

MI_n	=	K_C	*	T_S	/	T_I	*	$(SP_n - PV_n)$	+	MX
где: MI_n	-	значение интегральной составляющей регулирующего воздействия в момент квантования n								
K_C	-	коэффициент усиления контура регулирования								
T_S	-	период квантования контура регулирования								
T_I	-	постоянная времени интегрирования контура регулирования (называемая также временем интегрирования)								
SP_n	-	заданное значение регулируемой величины в момент квантования n								
PV_n	-	значение регулируемой переменной в момент квантования n								
MX	-	значение интегральной составляющей в момент квантования $n - 1$ (называемое также интегральной суммой или смещением)								

Интегральная сумма или смещение (MX) – это текущая сумма всех предыдущих значений интегральной составляющей. После каждого расчета MI_n смещение обновляется значением MI_n , которое может быть согласовано или ограничено (подробности см. в разделе «Переменные и диапазоны»). Начальное значение смещения обычно устанавливается равным значению регулирующего воздействия ($M_{нач}$) сразу перед его первым расчетом для контура регулирования. Частью интегральной составляющей являются также несколько констант: коэффициент усиления (K_C), период квантования (T_S), представляющий собой время цикла, с которым PID-регулятор пересчитывает регулирующее воздействие, и постоянная времени интегрирования (или сброс) (T_I), которая используется для управления влиянием интегральной составляющей на расчет регулирующего воздействия.

Дифференциальная составляющая в уравнении PID-регулятора

Дифференциальная составляющая MD пропорциональна изменению ошибки регулирования. S7-200 использует следующее уравнение для расчета дифференциальной составляющей:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

Во избежание ступенчатых изменений или скачков регулирующего воздействия при изменениях заданного значения это уравнение модифицировано в предположении, что заданное значение постоянно ($SP_n = SP_{n-1}$). В результате рассчитывается изменение регулируемой переменной, а не изменение ошибки регулирования. Это показывает следующее уравнение:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_{n-1} + PV_{n-1})$$

или:

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

где:	MD_n	- значение дифференциальной составляющей регулирующего воздействия в момент квантования n
	K_C	- коэффициент усиления контура регулирования
	T_S	- период квантования контура регулирования
	T_D	- постоянная времени воздействия по производной контура регулирования (называемая также временем упреждения)
	SP_n	- заданное значение регулируемой величины в момент квантования n
	SP_{n-1}	- заданное значение регулируемой величины в момент квантования n-1
	PV_n	- значение регулируемой переменной в момент квантования n
	PV_{n-1}	- значение регулируемой переменной в момент квантования n-1

Для использования в следующих расчетах дифференциальной составляющей должна сохраняться регулируемая переменная, а не ошибка регулирования. При первом отсчете значение PV_{n-1} инициализируется равным PV_n .

Выбор типа регулятора

Во многих системах управления может оказаться необходимым использовать только один или два метода регулирования. Например, может потребоваться только пропорциональное или пропорционально-интегральное управление. Выбор желаемого типа регулятора выполняется установкой значений постоянных параметров.

Если вам не нужно интегральное воздействие (нет составляющей "I" в расчете PID-регулятора), то постоянная времени интегрирования должна быть задана равной бесконечности. Даже при отсутствии интегрального воздействия значение этой составляющей не может быть равно нулю из-за начального значения интегральной суммы MX.

Если вам не нужно дифференцирующее воздействие (нет составляющей "D" в расчете PID-регулятора), то постоянная времени воздействия по производной (упреждение) должна быть задана равной 0.0.

Если вам не нужно пропорциональное воздействие (нет составляющей "P" в расчете PID-регулятора), а вы хотите иметь интегральный или интегрально-дифференциальный регулятор, то значение 0.0 должно быть задано для коэффициента усиления. Так как усиление является коэффициентом в уравнениях для расчета интегральной и дифференциальной составляющей, установка значения 0.0 для коэффициента усиления контура регулирования приведет к тому, что при расчете интегральной и дифференциальной составляющей для коэффициента усиления будет использовано значение 1.0.

Преобразование и нормализация входов контура регулирования

Контур регулирования имеет две входных переменных – заданное значение и регулируемую переменную. Заданное значение – это обычно фиксированная величина, например, уставка скорости для регулятора скорости движения в вашем автомобиле. Регулируемая переменная – это величина, связанная с регулирующим воздействием, поэтому она измеряет влияние, оказываемое регулирующим воздействием на управляемую систему. В примере с регулятором скорости движения автомобиля регулируемой переменной является вход тахометра, измеряющего скорость вращения колес.

Заданное значение и регулируемая переменная – это реальные физические величины, диапазон значений которых и единицы измерения могут быть различными. Прежде чем эти физические величины могут быть использованы командой PID-регулятор, они должны быть преобразованы в нормализованные представления с плавающей точкой.

Первый шаг состоит в преобразовании физической величины, представленной в виде 16-битового целого числа в вещественное число, или число с плавающей точкой. Следующая последовательность команд показывает, как можно преобразовать целое число в вещественное.

ITD	AIW0, AC0	// Преобразовать входную величину в двойное слово
DTR	AC0, AC0	// Преобразовать 32-битовое целое в вещественное число

Следующий шаг состоит в преобразовании вещественного представления реальной физической величины в нормализованное значение между 0.0 и 1.0. Для нормализации заданного значения или регулируемой переменной используется следующее уравнение:

R_{Norm}	=	((R_{Raw} / Span) + Offset)
где:	R _{Norm}	- нормализованное вещественное число, представляющее реальную физическую величину
	R _{Raw}	- ненормализованное, или необработанное (raw), вещественное представление реальной физической величины
	Offset	- смещение, равное 0.0 для униполярных величин, и 0,5 для биполярных величин
	Span	- диапазон, равный разности между максимально возможным и минимально возможным значением: = 32 000 для униполярных величин (обычно) = 64 000 для биполярных величин (обычно)

Следующая последовательность команд, являющаяся продолжением предыдущей последовательности, показывает, как нормализовать биполярную величину в AC0 (с диапазоном 64 000):

/R	64000.0, AC0	//Нормализовать значение в аккумуляторе
+R	0.5, AC0	//Сместить значение в диапазон от 0.0 до 1.0
MOVR	AC0, VD100	//Сохранить нормализованное значение в таблице контура регулирования
		//TABLE

Преобразование регулирующего воздействия в масштабированную целую величину

Регулирующее воздействие – это управляющая переменная, например, установка дроссельной заслонки в примере с регулятором скорости движения автомобиля. Регулирующее воздействие – это нормализованное вещественное значение между 0.0 и 1.0. Прежде чем регулирующее воздействие может быть использовано для управления аналоговым выходом, оно должно быть преобразовано в 16-битовую масштабированную целую величину. Этот процесс противоположен преобразованию PV и SP в нормализованную величину. Первый шаг состоит в преобразовании регулирующего воздействия в масштабированное вещественное число с помощью следующей формулы:

R_{Scal}	=	(M_n – Offset) * Span
где:	R _{Scal}	- масштабированное (scaled) вещественное значение регулирующего воздействия
	M _n	- нормализованное вещественное значение регулирующего воздействия
	Offset	- смещение, равное 0.0 для униполярных величин, и 0,5 для биполярных величин
	Span	- диапазон, равный разности между максимально возможным и минимально возможным значением = 32 000 для униполярных величин (обычно) = 64 000 для биполярных величин (обычно)

Следующая последовательность команд показывает, как масштабировать регулирующее воздействие:

```

MOVR  VD108, AC0      //Передать регулирующее воздействие в аккумулятор
-R    0.5, AC0        //Этот оператор включить только для биполярного значения
*R    64000.0, AC0    //Масштабировать значение в аккумуляторе
    
```

Затем масштабированное вещественное представление регулирующего воздействия должно быть преобразовано в 16-битовое целое. Как выполнить это преобразование, показывает следующая последовательность команд:

```

ROUND AC0, AC0        //Преобразовать вещественное число в 32-битовое целое
DTI   AC0, LW0        //Преобразовать эту величину в 16-битовое целое число
MOVW  LW0, AQW0       //Записать это значение в аналоговый выход
    
```

Контуры регулирования с положительной и отрицательной обратной связью

Контур регулирования имеет положительную обратную связь, если его коэффициент усиления положителен, и отрицательную обратную связь, если его коэффициент усиления отрицателен. (Для интегрального и интегрально-дифференциального регулятора, где значение коэффициента усиления равно 0.0, задание положительных значений для постоянных времени интегрирования и воздействия по производной приведет к положительной обратной связи, а задание отрицательных значений – к отрицательной обратной связи.)

Переменные и диапазоны

Регулируемая переменная и заданное значение являются входами для расчета PID-регулятора. Поэтому поля таблицы контура регулирования для этих переменных могут считываться, но не могут быть изменены командой PID-регулятор.

Регулирующее воздействие генерируется как результат расчетов PID-регулятора, так что поле регулирующего воздействия в таблице контура регулирования обновляется после каждого расчета PID-регулятора. Регулирующее воздействие заключено между 0.0 и 1.0. Поле регулирующего воздействия может быть использовано пользователем для задания начального регулирующего воздействия при организации перехода от ручного управления к автоматическому с помощью команды PID-регулятор (см. обсуждение этого вопроса в нижеследующем разделе «Режимы»).

Если используется интегральный регулятор, то значение смещения обновляется в результате расчета PID-регулятора, и обновленное значение используется как вход в следующем расчете PID-регулятора. Если рассчитанное регулирующее воздействие выходит за пределы допустимого диапазона (меньше 0.0 или больше 1.0), то смещение корректируется в соответствии со следующими формулами:

$$MX = 1.0 - (MP_n + MD_n) \quad \text{если расчетное значение } M_n > 1.0$$

или

$$MX = - (MP_n + MD_n) \quad \text{если расчетное значение } M_n < 0.0$$

где:

MX	- значение скорректированного смещения
MP _n	- значение пропорциональной составляющей регулирующего воздействия в момент квантования p
MD _n	- значение дифференциальной составляющей регулирующего воздействия в момент квантования p
M _n	- значение регулирующего воздействия в момент квантования p

При корректировке смещения описанным выше способом достигается улучшение чувствительности системы, когда рассчитанное регулирующее воздействие возвращается в надлежащий диапазон. Рассчитанное смещение также устанавливается в диапазоне между 0.0 и 1.0, а затем записывается в поле смещения таблицы контура регулирования при завершении каждого расчета PID-регулятора. Значение, сохраняемое в таблице контура регулирования, используется в следующем цикле расчетов PID-регулятора.

Значение смещения в таблице контура регулирования может быть изменено пользователем перед выполнением команды PID-регулятор для оказания воздействия на определенные ситуации в некоторых приложениях. При ручной корректировке смещения необходимо учитывать, что любое смещение, записываемое в таблицу контура регулирования, должно быть вещественным числом в диапазоне между 0.0 и 1.0.

Значение регулируемой величины, используемое для сравнения при расчете дифференциальной составляющей PID-регулятора, сохраняется в таблице контура регулирования. Это значение не может быть изменено пользователем.

Режимы

Для PID-регуляторов S7-200 нет встроенного управления режимом работы. Расчет PID-регулятора выполняется только тогда, когда поток сигнала достигает блока PID. Поэтому «автоматический» режим имеет место, когда расчеты PID-регулятора выполняются циклически. «Ручной» режим имеет место, когда расчеты PID-регулятора не выполняются.

Команда PID-регулятор имеет бит истории потока сигнала, аналогичный биту, используемому в команде счета. Команда использует этот бит для обнаружения изменения потока сигнала с 0 на 1. Когда изменение потока сигнала обнаруживается, это заставляет команду выполнить ряд действий, обеспечивающих плавный переход от ручного управления к автоматическому. Для плавного перехода к автоматическому режиму управления значение регулирующего воздействия, установленное при ручном управлении, должно быть передано в качестве входа команде PID-регулятор (записано в таблицу контура регулирования для M_n) перед переключением в автоматический режим управления. Команда PID-регулятор выполняет следующие действия с величинами в таблице контура регулирования, чтобы обеспечить плавный переход от ручного режима управления к автоматическому при обнаружении перехода потока сигнала из 0 в 1:

- устанавливает заданное значение (SP_n) = регулируемой переменной (PV_n)
- устанавливает старое значение регулируемой переменной (PV_{n-1}) = регулируемой переменной (PV_n)
- устанавливает смещение (MX) = регулирующему воздействию (M_n)

По умолчанию бит истории PID-регулятора установлен, и это состояние формируется при запуске и при каждом переходе контроллера из STOP в RUN. Если сигнал достигает блока PID при первом его исполнении после перехода в режим RUN, то переход потока сигнала из 0 в 1 не обнаруживается, и действия, обеспечивающие плавный переход, не выполняются.

Аварийный контроль и специальные операции

PID-регулятор – это простая, но мощная команда, выполняющая расчеты, необходимые для PID-регулирования. Если требуются другие функции, например, контроль аварийных ситуаций или выполнение специальных расчетов с переменными контура регулирования, то они должны быть реализованы с помощью основных команд, поддерживаемых S7-200.

Сбойные ситуации

Если начальный адрес таблицы контура регулирования или операнды контура регулирования, указанные в команде, находятся вне допустимого диапазона, то при компиляции CPU выдаст ошибку компиляции (ошибка диапазона), и компиляция потерпит неудачу.

Некоторые входные значения таблицы контура регулирования не проверяются на соответствие допустимому диапазону командой PID-регулятор. Вы должны позаботиться о том, чтобы регулируемая переменная и заданное значение (а также смещение и предыдущее значение регулируемой переменной, если они используются в качестве входов) были вещественными числами в диапазоне между 0.0 и 1.0.

Если при выполнении арифметических операций в расчетах PID-регулятора встретится любая ошибка, то будет установлен бит SM1.1 (переполнение или недопустимое значение), и выполнение команды PID-регулятор будет завершено. (Обновление выходных значений в таблице контура регулирования может быть неполным, так что вы не должны принимать эти значения во внимание, а исправить входное значение, вызвавшее арифметическую ошибку перед следующим выполнением команды PID-регулятор.)

Таблица контура регулирования

Таблица контура регулирования имеет длину 80 байтов и формат, показанный в таблице 6–44.

Таблица 6–44. Таблица контура регулирования

Смещение	Поле	Формат	Тип	Описание
0	Регулируемая переменная (PV_n)	REAL	In	Содержит регулируемую переменную, которая должна быть масштабирована между 1.0.
4	Заданное значение (SP_n)	REAL	In	Содержит заданное значение, которое должно быть масштабировано между 0.0 и 1.0.
8	Регулирующее воздействие (M_n)	REAL	In/Out	Содержит расчетное регулирующее воздействие, масштабированное между 0.0 и 1.0.
12	Коэффициент усиления (K_C)	REAL	In	Содержит усиление, являющееся коэффициентом пропорциональности. Может быть положительным или отрицательным числом.
16	Период квантования (T_S)	REAL	In	Содержит период квантования в секундах. Должен быть положительным числом.
20	Постоянная времени интегрирования (T_I)	REAL	In	Содержит постоянную времени интегрирования в минутах. Должна быть положительным числом.
24	Постоянная времени воздействия по производной (T_D)	REAL	In	Содержит постоянную времени воздействия по производной в минутах. Должна быть положительным числом.
28	Смещение (MX)	REAL	In/Out	Содержит значение смещения или интегральной суммы в пределах от 0.0 до 1.0.
32	Предыдущее значение регулируемой переменной (PV_{n-1})	REAL	In/Out	Содержит предыдущее значение регулируемой переменной, сохраняемое после последнего выполнения команды PID-регулятор.
36 – 79	Зарезервировано для переменных автонастройки. Подробности см. в таблице 15–1.			

Команды прерывания

Разблокирование и блокирование прерываний

Команда разблокирования прерываний (ENI) разблокирует обработку всех назначенных прерывающих событий. Команда блокирования прерываний (DISI) блокирует обработку всех прерывающих событий.

Когда вы переходите в режим RUN, прерывания первоначально заблокированы. Находясь в режиме RUN, вы можете разблокировать все прерывания, выполнив команду разблокирования прерываний. Выполнение команды блокирования прерываний запрещает обработку прерываний, однако активные прерывающие события и далее будут ставиться в очередь.

Сбойные состояния, устанавливающие ENO = 0 :

- 0004 (попытка исполнения команд ENI, DISI или HDEF в программе обработки прерываний)

Условный возврат из программы обработки прерываний

Команда условного возврата из программы обработки прерываний (CRETI) может быть использована для возврата из программы обработки прерываний в зависимости от условия, задаваемого предшествующей логикой.

Назначение прерывания

Команда назначения прерывания (ATCH) связывает прерывающее событие EVNT с номером программы обработки прерываний INT и разблокирует прерывающее событие.

Сбойные состояния, устанавливающие ENO = 0 :

- 0002 (конфликт при назначении входов для HSC)

Отсоединение прерывания

Команда отсоединения прерывания (DTCH) разрывает связь прерывающего события EVNT со всеми программами обработки прерываний и блокирует прерывающее событие.

Очистка прерывающих событий

Команда очистки прерывающих событий удаляет все прерывающие события типа EVNT из очереди прерываний. Эта команда используется для очистки очереди прерываний от нежелательных прерывающих событий. Если эта команда используется для удаления ложных прерывающих событий, вы должны отсоединить это событие перед удалением событий из очереди. Иначе после выполнения команды очистки прерывающих событий к очереди будут добавлены новые события.

В примере показан скоростной счетчик в квадратном режиме, использующий команду CLR_EVNT для удаления прерываний. Если шаговый датчик оптического устройства остановился на границе перехода от света к тьме, то небольшие колебания машины могут вызвать нежелательные прерывания перед тем, как сможет быть загружена новое значение PV.

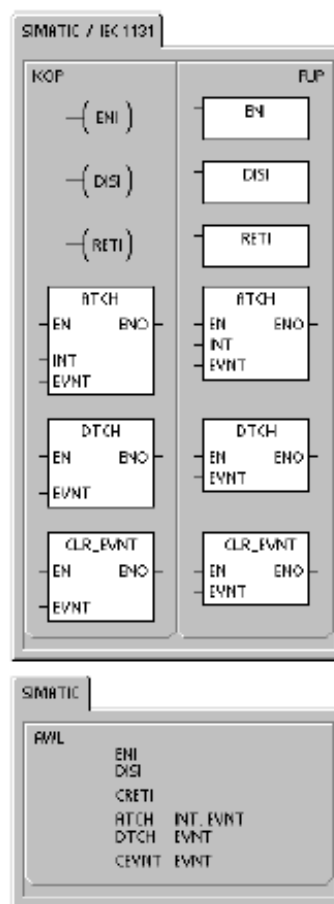


Таблица 6–45. Допустимые операнды для команд прерывания

Входы/выходы	Типы данных	Операнды
INT	BYTE	константа (от 0 до 127)
EVNT	BYTE	константа CPU 221 и CPU 222: от 0 до 12, от 19 до 23 и от 27 до 33 CPU 224: от 0 до 23 и от 27 до 33 CPU 224XP и CPU 226: от 0 до 33

Принцип действия команд назначения и отсоединения прерываний

Прежде чем программа обработки прерывания может быть вызвана, должно быть установлено соответствие между прерывающим событием и сегментом программы, который вы хотите выполнить, когда это событие происходит. Для организации связи между прерывающим событием (задаваемым номером прерывающего события) и сегментом программы (задаваемым номером программы обработки прерывания) используйте команду назначения прерывания (ATCH). Одной программе обработки прерываний можно поставить в соответствие несколько прерывающих событий, но одно событие не может быть одновременно поставлено в соответствие нескольким программам обработки прерываний.

Когда вы назначаете прерывающее событие программе обработки прерываний, это прерывание автоматически разблокируется. Если вы заблокировали все прерывания с помощью команды блокирования прерываний, то каждое возникновение прерывающего события ставится в очередь, пока прерывания не будут снова разблокированы с помощью команды разблокирования прерываний или не произойдет переполнение очереди прерываний.

Отдельные прерывающие события можно заблокировать разрывом связи между этим прерывающим событием и программой обработки прерывания с помощью команды отсоединения прерывания. Команда отсоединения возвращает прерывание в неактивное или игнорируемое состояние. Таблица 6–46 перечисляет различные типы прерывающих событий.

Таблица 6–46. Прерывающие события

Событие	Описание	CPU 221 CPU 222	CPU 224	CPU 224XP CPU 226
0	I0.0 Нарастающий фронт	да	да	да
1	I0.0 Падающий фронт	да	да	да
2	I0.1 Нарастающий фронт	да	да	да
3	I0.1 Падающий фронт	да	да	да
4	I0.2 Нарастающий фронт	да	да	да
5	I0.2 Падающий фронт	да	да	да
6	I0.3 Нарастающий фронт	да	да	да
7	I0.3 Падающий фронт	да	да	да
8	Порт 0 Символ принят	да	да	да
9	Порт 0 Передача завершена	да	да	да
10	Управляемое временем прерывание 0 SMB34	да	да	да
11	Управляемое временем прерывание 1 SMB35	да	да	да
12	HSC0 CV=PВ (текущее значение = предустановленному)	да	да	да
13	HSC1 CV=PВ (текущее значение = предустановленному)		да	да
14	HSC1 Изменение направления		да	да
15	HSC1 Внешний сброс		да	да
16	HSC2 CV=PВ (текущее значение = предустановленному)		да	да
17	HSC2 Изменение направления		да	да
18	HSC2 Внешний сброс		да	да
19	PLS0 Прерывание по завершению отсчета количества импульсов PTO	да	да	да
20	PLS1 Прерывание по завершению отсчета количества импульсов PTO	да	да	да
21	Таймер T32 Прерывание СТ=PT	да	да	да
22	Таймер T96 Прерывание СТ=PT	да	да	да
23	Порт 0 Прием сообщения завершен	да	да	да
24	Порт 1 Прием сообщения завершен			да
25	Порт 1 Символ принят			да
26	Порт 1 Передача завершена			да
27	HSC0 Изменение направления	да	да	да

Таблица 6–46. Прерывающие события, продолжение

28	HSC0	Внешний сброс	да	да	да
29	HSC4	CV=PV (текущее значение = предустановленному)	да	да	да
30	HSC4	Изменение направления	да	да	да
31	HSC4	Внешний сброс	да	да	да
32	HSC3	CV=PV (текущее значение = предустановленному)	да	да	да
33	HSC5	CV=PV (текущее значение = предустановленному)	да	да	да

Исполнение программ обработки прерываний в S7-200

Программа обработки прерывания выполняется в ответ на соответствующее внутреннее или внешнее событие. После выполнения последней команды программы обработки прерывания управление возвращается в главную программу. Вы можете покинуть эту программу, выполнив команду условного возврата из программы обработки прерываний (CRETI). В таблице 6–47 приведены некоторые указания и ограничения по использованию программ обработки прерываний в вашей программе.

Таблица 6–47. Рекомендации и ограничения по использованию программ обработки прерываний

Указания
Обработка прерываний обеспечивает быструю реакцию на определенные внутренние или внешние события. Вам следует оптимизировать программы обработки прерываний, чтобы выполнить конкретную задачу, а затем вернуть управление главной программе. Если программа обработки прерывания спроектирована короткой с точными спецификациями, то она будет быстро выполняться и не будет задерживать другие процессы на длительные промежутки времени. Если этого не сделать, то неожиданные условия могут вызвать ненормальную работу оборудования, управляемого главной программой. Для прерываний безусловно верна аксиома «чем короче, тем лучше».
Ограничения
В программе обработке прерывания нельзя использовать команды блокирования прерываний (DISI), разблокирования прерываний (ENI), определения режима работы скоростного счетчика (HDEF) и завершения обработки (END).

Системная поддержка прерываний

Так как прерывания могут оказывать влияние на контакты, катушки и аккумуляторы, то система сохраняет и перезагружает логический стек, аккумуляторные регистры и биты специальной памяти (SM), которые отображают состояние аккумуляторов и команд. Это позволяет избежать искажения главной программы пользователя из-за перехода в программу обработки прерывания и возвращения из нее.

Совместное использование данных главной программой и программами обработки прерываний

Данные могут совместно использоваться главной программой и одной или несколькими программами обработки прерываний. Так как невозможно предсказать, когда S7-200 может сгенерировать прерывание, то желательно ограничить количество переменных, которые применяются как в программе обработки прерываний, так и в других местах программы. В результате действий программы обработки прерываний могут возникнуть проблемы непротиворечивости совместно используемых данных, когда выполнение команд вашей главной программы прерывается событиями, вызывающими прерывания. Чтобы гарантировать, что ваша программа обработки прерываний будет использовать только временную память и не перезапишет данные, используемые еще в каком-либо месте вашей программы, пользуйтесь таблицей локальных переменных программы обработки прерываний.

Существует ряд методов программирования, которые вы можете использовать, чтобы обеспечить корректное разделение данных между вашей главной программой и программами обработки прерываний. Эти методы или ограничивают способ доступа к совместно используемым ячейкам памяти, или препятствуют прерыванию последовательностей команд, использующих разделяемые ячейки памяти.

- Для программы на STL, совместно использующей только одну переменную: Если совместно используемые данные представляют собой одну переменную в виде байта, слова или двойного слова, и ваша программа написана на STL, то корректный доступ к совместно используемым данным может быть обеспечен сохранением промежуточных результатов операций над совместно используемыми данными только в тех адресах памяти или аккумуляторах, которые совместно не используются.
- Для программы на LAD, которая совместно использует единственную переменную: Если разделяемые данные представляют собой единственную переменную в виде байта, слова или двойного слова и ваша программа написана на LAD, то корректный совместный доступ может быть обеспечен установлением соглашения, что доступ к разделяемым ячейкам памяти может осуществляться только с помощью команд пересылки (MOVB, MOVW, MOVD, MOVR). В то время как многие команды LAD составлены из непрерываемых последовательностей команд STL, команды пересылки состоят из единственной команды STL, на исполнение которой не могут влиять прерывающие события.
- Для программы на STL или LAD, совместно использующей несколько переменных: Если разделяемые данные составлены из ряда связанных байтов, слов или двойных слов, то для управления исполнением программ обработки прерываний могут быть использованы команды блокировки/разблокировки прерываний (DISI и ENI). В той точке вашей программы, где должны начаться операции с разделяемыми ячейками памяти, заблокируйте прерывания. Как только все действия, влияющие на совместно используемые ячейки памяти, завершены, вновь разблокируйте прерывания. В течение времени, когда прерывания заблокированы, программы обработки прерываний не могут выполняться и, следовательно, не имеют доступа к разделяемым ячейкам памяти; однако такой подход может привести к запаздыванию реакции на прерывающие события.

Вызов подпрограмм из программ обработки прерываний

Из программы обработки прерывания можно вызвать только один уровень вложенности подпрограмм. Аккумуляторы и логический стек совместно используются программой обработки прерывания и вызываемой подпрограммой.

Виды прерываний, поддерживаемых S7–200

S7–200 поддерживает следующие виды программ обработки прерываний:

- Прерывания коммуникационных портов: S7–200 генерирует события, которые позволяют вашей программе управлять коммуникационным портом.
- Прерывания по вводу/выводу: S7–200 генерирует события для различных изменений состояния различных входов-выходов. Эти события позволяют вашей программе реагировать на скоростные счетчики, вывод импульсов и на нарастающие или падающие фронты на входах.
- Прерывания, управляемые временем: S7–200 генерирует события, которые позволяют вашей программе реагировать через определенные интервалы времени.

Прерывания коммуникационных портов

Ваша программа может управлять последовательным коммуникационным портом S7–200. Этот режим работы коммуникационного порта называется свободно программируемым обменом данными (Freeport). В этом режиме ваша программа определяет скорость передачи, количество битов на символ, способ контроля четности и протокол. Для облегчения обмена данными, управляемого вашей программой, в вашем распоряжении имеются прерывания приема и передачи. За дополнительной информацией обратитесь к командам приема и передачи.

Прерывания по вводу/выводу

К прерываниям по вводу/выводу относятся прерывания при нарастающем/падающем фронте, прерывания от скоростных счетчиков и прерывания от последовательности импульсов. S7–200 может генерировать прерывание при нарастающем и/или падающем фронте на входе (I0.0, I0.1, I0.2 или I0.3). Появления нарастающего и падающего фронта могут быть распознаны для любой из этих точек ввода. Эти события могут использоваться для отображения условия, которое немедленно должно быть принято во внимание, когда это событие происходит.

Прерывания от скоростных счетчиков дают возможность реагировать на такие условия, как достижение текущим значением предустановленного значения, изменение направления счета, которое может соответствовать реверсированию направления, в котором вращается вал, или внешний сброс счетчика. Каждое из этих прерываний дает возможность предпринимать в реальном времени действия в ответ на быстрые события, которыми нельзя управлять при скоростях, определяемых временем цикла программируемого логического контроллера.

Прерывания от последовательности импульсов немедленно извещают о завершении вывода предписанного количества импульсов. Импульсные последовательности часто используются для управления шаговыми двигателями.

Каждое из вышеописанных прерываний может быть разблокировано назначением программы обработки прерывания соответствующему событию ввода/вывода.

Прерывания, управляемые временем

К прерываниям, управляемым временем, относятся циклические прерывания и прерывания, вызываемые таймерами T32 и T96. С помощью циклических прерываний вы можете задать действия, которые должны выполняться циклически. Время цикла устанавливается в пределах от 1 до 255 мс шагами по 1 мс. Вы должны записать время цикла в SMB34 для циклического прерывания 0 и в SMB35 для циклического прерывания 1. Событие, вызывающее циклические прерывания, передает управление соответствующей программе обработки прерываний каждый раз, как истекает время работы таймера. Обычно циклические прерывания используются для управления опросом аналоговых входов или для вызова PID-регулятора через регулярные интервалы времени.

Циклическое прерывание разблокируется, и начинается отсчет времени, когда вы назначаете программу обработки прерывания событию, вызывающему циклическое прерывание. При этом система воспринимает значение времени цикла, и последующие изменения в SMB34 и SMB35 на это время цикла влияния не оказывают. Чтобы изменить время цикла, вы должны задать для него новое значение, а затем снова назначить программу обработки прерывания событию, вызывающему циклическое прерывание. Когда происходит повторное назначение, функция циклического прерывания сбрасывает все накопленное время от предыдущего назначения и начинает отсчет времени с новым значением.

Будучи разблокированным, циклическое прерывание работает постоянно, выполняя назначенную программу обработки прерывания при каждом истечении заданного временного интервала. Если вы выйдете из режима RUN или отсоедините циклическое прерывание, то циклическое прерывание блокируется. Если выполняется глобальная команда блокирования прерываний, то циклические прерывания продолжают происходить. Каждое появление циклического прерывания ставится в очередь (пока прерывания не будут разблокированы или не заполнятся очередь).

Прерывания от таймера T32 или T96 позволяют своевременно реагировать на завершение заданного интервала времени. Эти прерывания поддерживаются только для таймеров T32 и T96 с задержкой включения (TON) и с задержкой выключения (TOF) с разрешением 1 мс. В противном случае таймеры T32 и T96 обладают обычными свойствами. Когда прерывание разблокировано, назначенная программа обработки прерывания выполняется, когда текущее значение активного таймера становится равным предустановленному значению во время нормального обновления 1-миллисекундного таймера, выполняемого в S7-200. Эти прерывания разблокируются назначением программы обработки прерывания событиям, вызывающим прерывание от таймера T32/T96.

Приоритет прерываний и постановки их в очередь

Прерывания обслуживаются S7-200 в порядке их возникновения с учетом соответствующей группы приоритета. В любой момент времени исполняется только одна программа обработки прерывания. Когда исполнение программы обработки прерывания начинается, программа исполняется до своего завершения. Она не может быть прервана другой программой обработки прерывания, даже если последняя имеет более высокий приоритет. Прерывания, возникающие во время обработки другого прерывания, ставятся в очередь для последующей обработки.

Три очереди прерываний и максимальное количество прерываний, которое они могут хранить, показаны в таблице 6-48.

Таблица 6-48. Максимальное количество записей на очередь прерываний

Очередь	CPU 221, CPU 222, CPU 224	CPU 224XP и CPU 226
Очередь коммуникационных прерываний	4	8
Очередь прерываний по вводу/выводу	16	16
Очередь прерываний, управляемых временем	8	8

В принципе может возникнуть больше прерываний, чем может содержать очередь. Поэтому системой поддерживаются биты переполнения очереди (обозначающие тип прерывающих событий, которые были потеряны). Биты переполнения очереди прерываний показаны в таблице 6-49. Вам следует использовать эти биты только в программе обработки прерывания, так как они сбрасываются, когда очередь опустошается и управление возвращается главной программе.

В табл. 6–50 представлены все прерывающие события, их приоритеты и соответствующие номера событий.

Таблица 6–49. Биты переполнения очереди прерываний

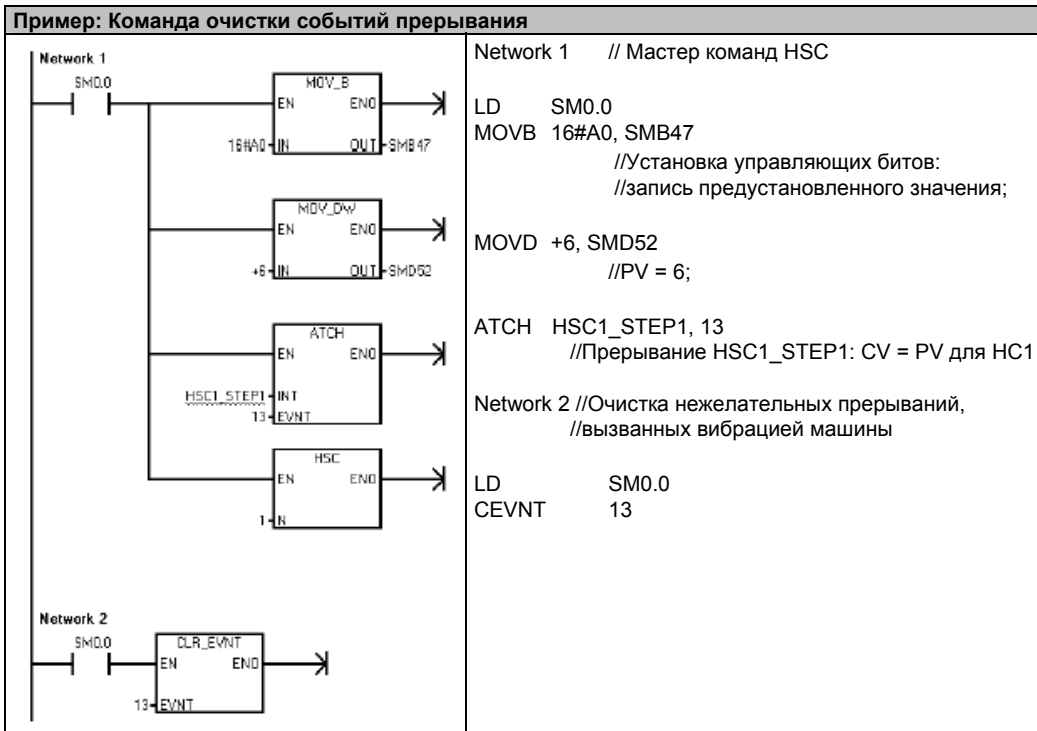
Описание (0 = нет переполнения, 1 = переполнение)	Бит специальной памяти
Очередь коммуникационных прерываний	SM4.0
Очередь прерываний по вводу/выводу	SM4.1
Очередь прерываний, управляемых временем	SM4.2

Таблица 6–50. Порядок приоритетов для прерывающих событий

Событие	Описание		Группа приоритета	Приоритет в группе
8	Порт 0	Символ принят	Коммуникации <i>Наивысший приоритет</i>	0
9	Порт 0	Передача завершена		0
23	Порт 0	Прием сообщения завершен		0
24	Порт 1	Прием сообщения завершен		1
25	Порт 1	Символ принят		1
26	Порт 1	Передача завершена		1
19	PLS0	Прерывание по завершению отсчета количества импульсов РТО	Дискретные прерывания <i>Средний приоритет</i>	0
20	PLS1	Прерывание по завершению отсчета количества импульсов РТО		1
0	I0.0	Нарастающий фронт		2
2	I0.1	Нарастающий фронт		3
4	I0.2	Нарастающий фронт		4
6	I0.3	Нарастающий фронт		5
1	I0.0	Падающий фронт		6
3	I0.1	Падающий фронт		7
5	I0.2	Падающий фронт		8
7	I0.3	Падающий фронт		9
12	HSC0	CV=PV (текущее значение = предустановленному)		10
27	HSC0	Изменение направления		11
28	HSC0	Внешний сброс		12
13	HSC1	CV=PV (текущее значение = предустановленному)		13
14	HSC1	Изменение направления		14
15	HSC1	Внешний сброс		15
16	HSC2	CV=PV (текущее значение = предустановленному)		16
17	HSC2	Изменение направления		17
18	HSC2	Внешний сброс		18
32	HSC3	CV=PV (текущее значение = предустановленному)		19
29	HSC4	CV=PV (текущее значение = предустановленному)		20
30	HSC4	Изменение направления		21
31	HSC4	Внешний сброс		22
33	HSC5	CV=PV (текущее значение = предустановленному)	23	
10	Управляемое временем прерывание 0	SMB34	Прерывания, управляемые временем <i>Низший</i>	0
11	Управляемое временем прерывание 1	SMB35		1
21	Таймер T32	Прерывание CT=PT		2
22	Таймер T96	Прерывание CT=PT		3

Пример: Команды прерывания		
M A I N	<p>Network 1</p> <p>Network 2</p> <p>Network 3</p>	<p>Network 1 //В первом цикле: //1. Определить программу обработки прерываний INT_0 // как программу обработки прерыванию по падающему // фронту на I0.0 //2. Разблокировать все прерывающие события.</p> <pre>LD SM0.1 ATCH INT_0, 1 ENI</pre> <p>Network 2 //Если обнаружена ошибка ввода/вывода, //заблокировать прерывание по падающему фронту на // I0.0. Этот сегмент не обязателен.</p> <pre>LD SM5.0 DTCH 1</pre> <p>Network 3 //Если M5.0 установлен, //заблокировать все прерывания.</p> <pre>LD M5.0 DISI</pre>
I N T 0	<p>Network 1</p>	<p>Network 1 //Программа обработки прерываний по падающему фронту //на I0.0: //Условный возврат при ошибке ввода/вывода.</p> <pre>LD SM5.0 CRETI</pre>

Пример: Прерывание, управляемое временем, для чтения значения аналогового входа		
M A I N	<p>Network 1</p>	<p>Network 1 // В первом цикле вызвать подпрограмму 0.</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 //1. Установить интервал для прерывания, // управляемого временем, от 0 до 100 мс. //2. Управляемое временем прерывание 0 поставить в // соответствие INT_0 (событие 10). //3. Разблокировать все прерывающие события.</p> <pre>LD SM0.0 MOV_B 100, SMB34 ATCH INT_0, 10 ENI</pre>
I N T 0	<p>Network 1</p>	<p>Network 1 //Считывать значение AIW4 каждые 100 мс</p> <pre>LD SM0.0 MOVW AIW4, VW100</pre>



Логические операции

Операции инвертирования

Инвертирование байта, слова и двойного слова

Команды инвертирования байта (INVB), слова (INW) и двойного слова (INVD) образуют дополнение входа IN до единицы и загружают результат по адресу OUT.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.0 (ноль)

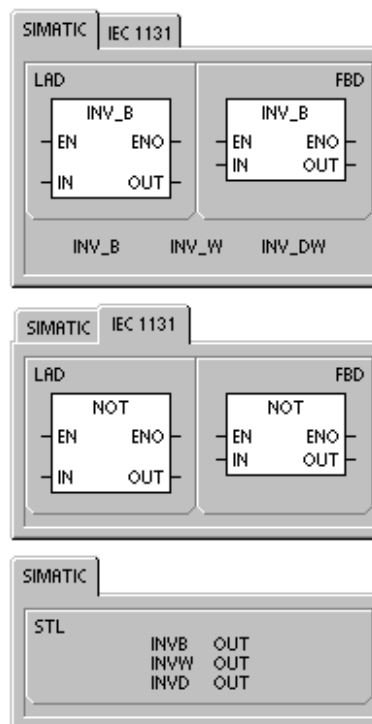


Таблица 6–51 Допустимые операнды для команд инвертирования

Входы/выходы	Типы данных	Операнды
IN	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
OUT	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Пример: Операция инвертирования

<p>Network 1</p>	<p>Network 1</p> <pre>LD I4.0 INW AC0</pre> <p>Инвертирование слова ACO 1101 0111 1001 0101</p> <p>Формирование дополнения до единицы ACO 0010 1000 0110 1010</p>
-------------------------	--

Поразрядные логические операции И, ИЛИ и исключающее ИЛИ

Поразрядное логическое И с байтами, словами и двойными словами

Поразрядные логические операции И с байтами (ANDB), словами (ANDW) и двойными словами (ANDD) логически сопрягают соответствующие биты двух входных величин IN1 и IN2 в соответствии с таблицей истинности логической операции И и загружают результат по адресу OUT.

Поразрядное логическое ИЛИ с байтами, словами и двойными словами

Поразрядные логические операции ИЛИ с байтами (ORB), словами (ORW) и двойными словами (ORD) логически сопрягают соответствующие биты двух входных величин IN1 и IN2 в соответствии с таблицей истинности логической операции ИЛИ и загружают результат по адресу OUT.

Поразрядное логическое исключающее ИЛИ с байтами, словами и двойными словами

Поразрядные логические операции Исключающее ИЛИ с байтами (XORB), словами (XORW) и двойными словами (XORD) логически сопрягают соответствующие биты двух входных величин IN1 и IN2 в соответствии с таблицей истинности логической операции Исключающее ИЛИ и загружают результат по адресу OUT.

Биты специальной памяти и ENO

Для всех команд, описанных на этой странице, следующие условия влияют на биты специальной памяти и ENO.

Сбойные состояния, устанавливающие ENO = 0

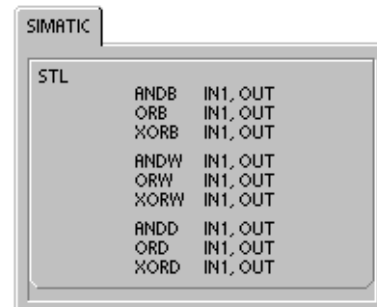
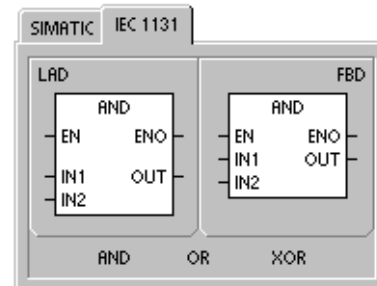
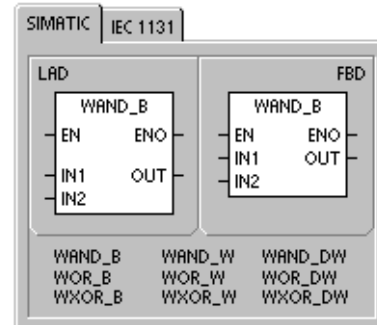
- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.0 (ноль)

Таблица 6–52. Допустимые операнды для операций И, ИЛИ и Исключающее ИЛИ

Входы/выходы	Типы данных	Операнды
IN1, IN2	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
OUT	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *AC, *LD ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD





Команды пересылки

Пересылка байта, слова, двойного слова или вещественного числа

Команды пересылки байта (MOVB), слова (MOVW), двойного слова (MOVD) и вещественного числа (MOVR) пересылают значение из адреса IN в адрес OUT, не изменяя исходной величины.

Используйте команду пересылки двойного слова для создания указателя. Подробную информацию вы найдете в разделе об указателях и косвенной адресации в главе 4.

У команды пересылки IEC на входе и выходе могут применяться различные типы данных, но они должны быть одного размера.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

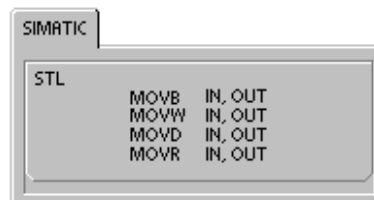
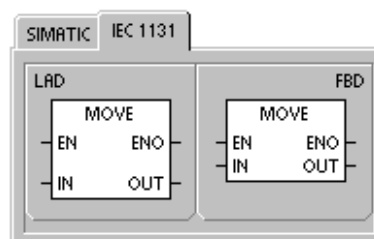
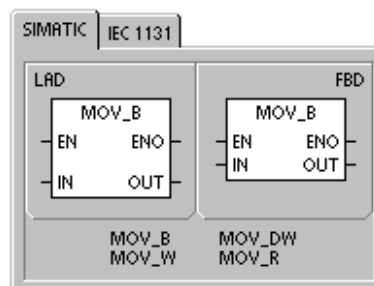


Таблица 6–53. Допустимые операнды для команд пересылки

Входы/ выходы	Типы данных	Операнды
IN	BYTE WORD, INT DWORD, DINT REAL	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, константа ID, QD, VD, MD, SMD, SD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, &SMB, &AIW, &AQW, AC, *VD, *LD, *AC, константа, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа
OUT	BYTE WORD, INT DWORD, DINT, REAL	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Непосредственное чтение или запись и пересылка байта

Команды непосредственной пересылки байта позволяют пересылать байт непосредственно между физическим входом или выходом и адресом в памяти.

Команда пересылки непосредственно считанного байта (BIR) считывает физический вход (IN) и записывает результат по адресу в памяти (OUT), не обновляя регистр образа процесса.

Команда пересылки байта для непосредственной записи (BIW) считывает данные из адреса в памяти (IN) и записывает их в физический выход (OUT) и в соответствующую ячейку образа процесса.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- Нет доступа к модулю расширения

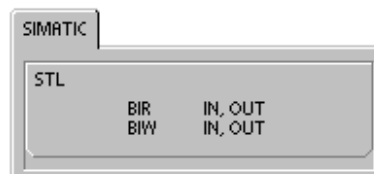
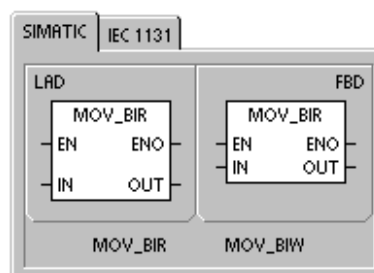


Таблица 6–54. Допустимые операнды для команды пересылки непосредственно считанного байта

Входы/выходы	Типы данных	Операнды
IN	BYTE	IB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Таблица 6–55. Допустимые операнды для команды пересылки байта для непосредственной записи

Входы/выходы	Типы данных	Операнды
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	BYTE	QB, *VD, *LD, *AC

Команды групповой пересылки

Групповая пересылка байтов, слов, двойных слов

Команды групповой пересылки байтов (BMB), слов (BMW) и двойных слов (BMD) пересылают указанное количество (N) байтов, слов или двойных слов, начиная с входного адреса IN, в новую область, начинающуюся с выходного адреса OUT.

N находится в диапазоне от 1 до 255.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

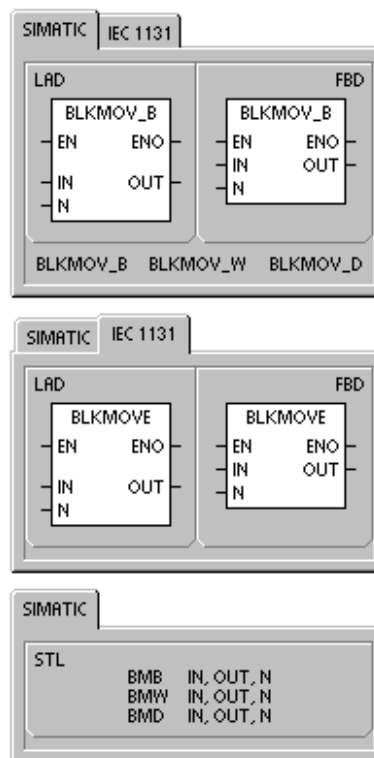
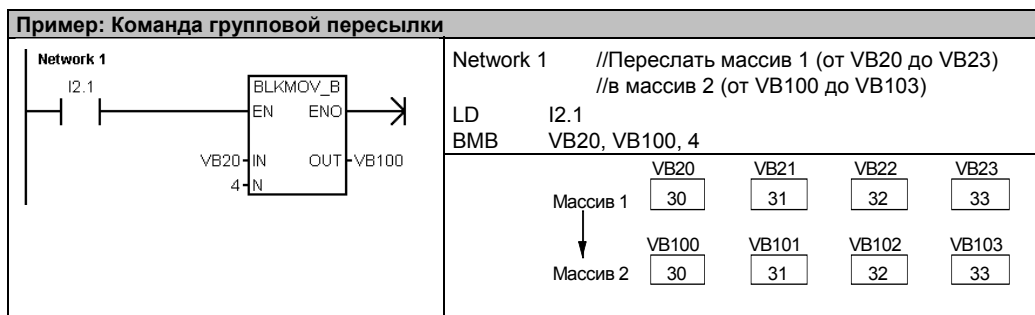


Таблица 6–56. Допустимые операнды для команд групповой пересылки

Входы/выходы	Типы данных	Операнды
IN	BYTE WORD, INT DWORD, DINT	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
OUT	BYTE WORD, INT DWORD, DINT	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AQW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, константа, *VD, *LD, *AC

Пример: Команда групповой пересылки



Команды управления программой

Условное завершение

Команда условного завершения (END) завершает текущий цикл в зависимости от результата предшествующей логической операции. Вы можете использовать команду условного завершения в главной программе, но ее нельзя использовать в подпрограммах и программах обработки прерываний.

Останов

Команда останова (STOP) завершает выполнение программы, вызывая переход CPU S7-200 из RUN в STOP.

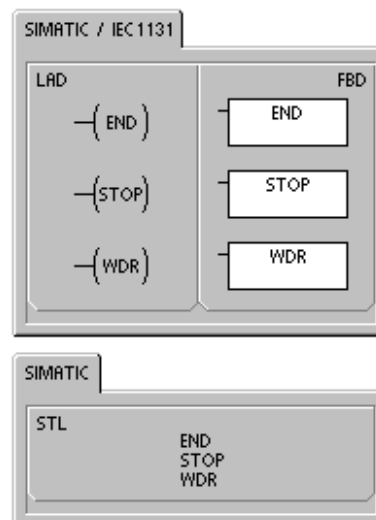
Если команда STOP выполняется в программе обработки прерывания, то эта программа завершается немедленно, а все прерывания, стоящие в очереди, игнорируются. Оставшиеся действия в текущем цикле обработки программы завершаются, включая выполнение главной программы пользователя, а переход из RUN в STOP производится в конце текущего цикла.

Сброс контроля времени

Команда сброса контроля времени (WDR) перезапускает системный таймер контроля времени CPU S7-200, увеличивая время, которое может занимать цикл обработки программы, не вызывая ошибки контроля времени.

Команду сброса контроля времени следует использовать с осторожностью. Если вы с помощью программных циклов препятствуете завершению цикла обработки программы или существенно задерживаете его завершение, то следует иметь в виду, что до завершения цикла обработки программы запрещены следующие процессы:

- связь (за исключением режима свободно программируемого обмена данными Freepoint)
- обновление входов и выходов (кроме входов и выходов с непосредственным доступом)
- обновление принудительно задаваемых значений
- обновление битов специальной памяти (не обновляются биты SM0, SM5 – SM29)
- диагностика в режиме реального времени
- 10-миллисекундные и 100-миллисекундные таймеры не накапливают время должным образом для циклов обработки программы, превышающих 25 секунд
- команда STOP при использовании в программе обработки прерывания
- модули расширения с цифровыми выходами также имеют таймер контроля времени, который выключает выходы, если S7-200 не производит запись в этот модуль. Чтобы эти выходы оставались включенными во время продленного цикла, пользуйтесь командой непосредственной записи для каждого модуля расширения. См. также пример, следующий за этим описанием.





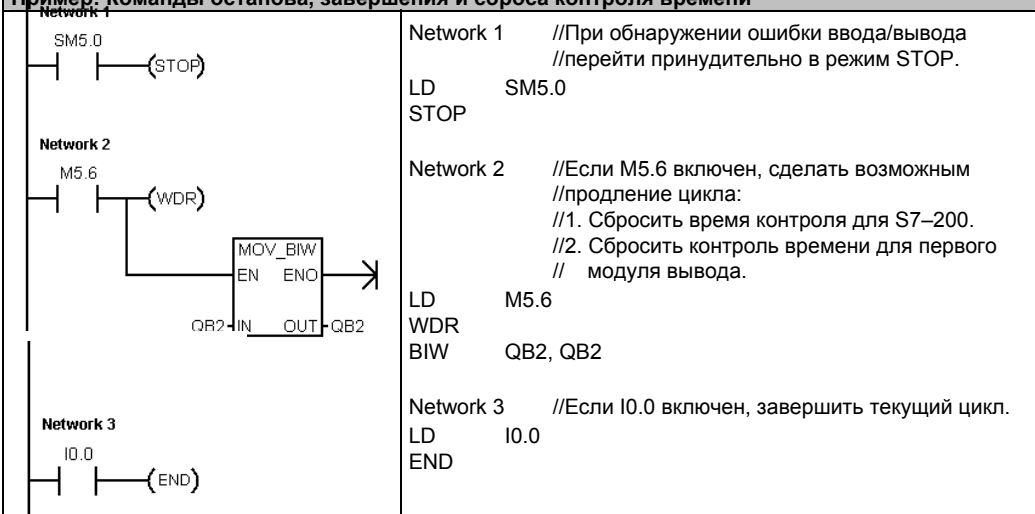
Совет

Команду сброса контроля времени следует использовать для перезапуска таймера контроля времени, если вы ожидаете, что время цикла обработки программы превысит 500 мс, или вы ожидаете увеличения активности прерываний, что может воспрепятствовать возвращению в главный цикл более чем на 500 мс.

Каждый раз, когда вы применяете команду сброса контроля времени, вы должны также использовать команду непосредственной записи для одного выходного байта (QB) на каждый цифровой модуль вывода, чтобы сбросить контроль времени модуля расширения.

Если вы используете команду сброса контроля времени, чтобы можно было выполнить программу с большим временем цикла, то перевод переключателя режимов работы в положение STOP вызывает переход S7-200 в состояние STOP в течение 1,4 секунды.

Пример: Команды останова, завершения и сброса контроля времени



Команды формирования программного цикла For-Next

С помощью команд FOR и NEXT вы можете управлять программными циклами, которые повторяются определенное количество раз. Каждая команда FOR требует наличия команды NEXT. Вы можете вкладывать циклы FOR/NEXT друг в друга (помещать цикл FOR-NEXT внутри другого цикла FOR-NEXT). Глубина вложения не может превышать восьми.

Команда FOR выполняет команды, расположенные между операторами FOR и NEXT. Вы должны задать значение индекса или счетчик цикла INDX, начальное значение INIT и конечное значение FINAL.

Команда NEXT отмечает конец цикла FOR.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Если вы активизируете цикл FOR-NEXT, то процесс циклического выполнения продолжается, пока не закончатся итерации, если только вы не измените конечное значение изнутри самого цикла. Вы можете изменять эти значения, пока цикл FOR-NEXT выполняет циклическую обработку. Когда цикл активизируется снова, он копирует начальное значение в индекс (счетчик цикла).

Команда FOR-NEXT сбрасывает себя каждый раз, когда она активизируется.

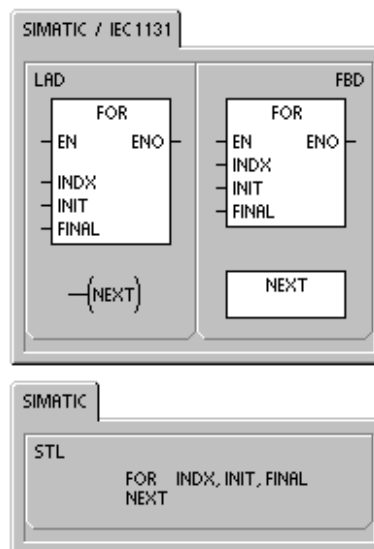
Например, если значение INIT равно 1, значение FINAL равно 10, то команды между FOR и NEXT исполняются 10 раз, причем значение INDX каждый раз увеличивается на единицу: 1, 2, 3, ...10.

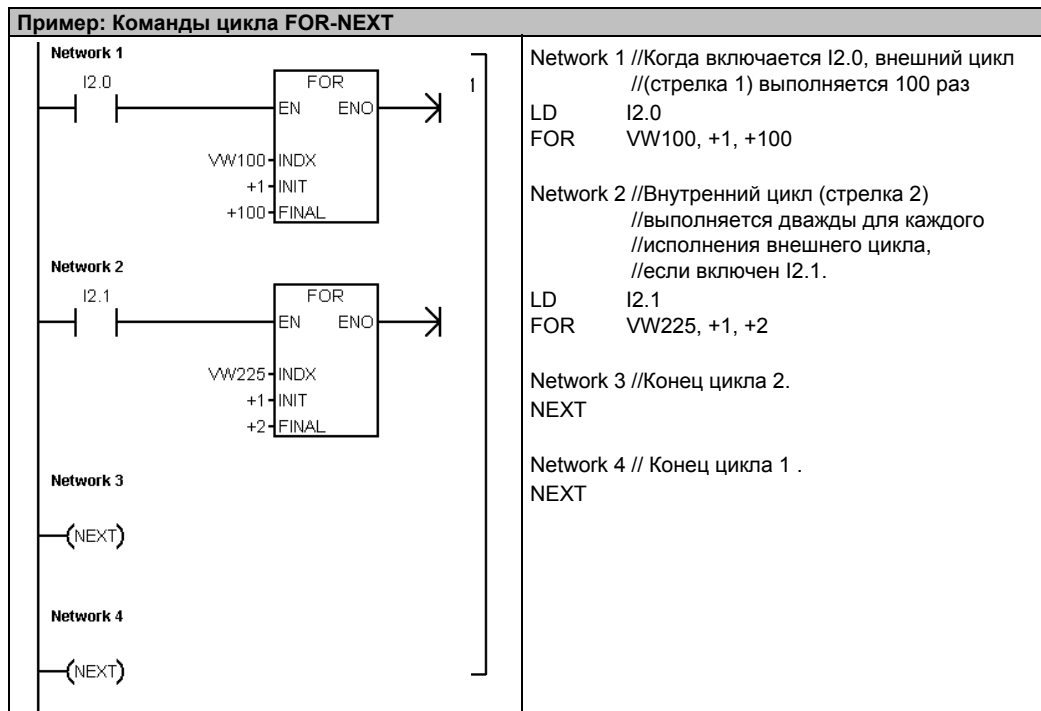
Если начальное значение больше конечного, то цикл не выполняется. После каждого исполнения команд между FOR и NEXT значение INDX увеличивается, а результат сравнивается с конечным значением. Если INDX больше конечного значения, то цикл завершается.

Если вершина стека равна 1, когда ваша программа входит в цикл FOR-NEXT, то вершина стека останется равной 1, когда ваша программа покинет цикл FOR-NEXT.

Таблица 6–57. Допустимые операнды для команд FOR и NEXT

Входы/выходы	Типы данных	Операнды
INDX	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
INIT, FINAL	INT	VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа





Команды перехода

Команда перехода на метку (JMP) осуществляет переход к указанной метке N внутри программы.

Команда "Метка" (LBL) отмечает положение цели перехода N.

Команду перехода на метку можно использовать в основной программе, в подпрограммах и в программах обработки прерываний. Команда перехода и соответствующая метка всегда должны находиться внутри одного и того же сегмента кода (в основной программе, подпрограмме или программе обработки прерываний).

Вы не можете перейти из главной программы на метку в подпрограмме или в программе обработки прерывания. Аналогично, вы не можете перейти из подпрограммы или программы обработки прерывания на метку вне этой подпрограммы или программы обработки прерывания.

Команду перехода на метку можно использовать внутри сегмента SCR, но соответствующая метка должна находиться внутри того же сегмента SCR.

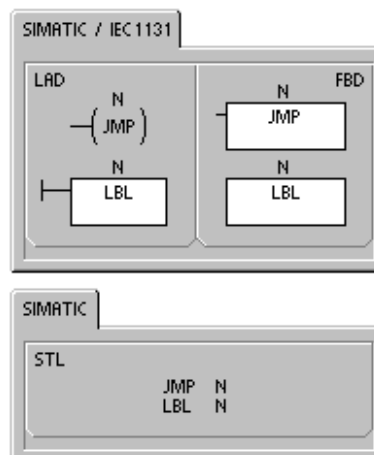


Таблица 6–58. Допустимые операнды для команд перехода

Входы/выходы	Типы данных	Операнды
N	WORD	константа (от 0 до 255)

Пример: Команда перехода на метку		
<p>Network 1</p> <p>Network 2</p>	<p>Network 1 //Если сохраняемые данные не потеряны, перейти на //метку LBL 4</p> <p>LDN SM0.2</p> <p>JMP 4</p> <p>Network 2</p> <p>LBL 4</p>	

Команды для реле управления очередностью (SCR)

Команды управления очередностью (SCR) предоставляют в ваше распоряжение простое и вместе с тем мощное средство для программирования пошагового управления, которое естественно встраивается в программу LAD, FBD или STL.

Если ваше приложение состоит из последовательности функций, которые должны неоднократно выполняться, то с помощью реле управления очередностью, вы можете так структурировать свою программу, что она будет непосредственно соответствовать вашему приложению. В результате вам будет проще и быстрее программировать и тестировать ваше приложение.

Команда загрузки реле управления очередностью (LSCR) загружает стек SCR и логический стек значением S-бита, на который указывает команда N.

Сегмент SCR активизируется или деактивизируется результирующим значением стека SCRE. Значение стека SCR копируется в вершину логического стека, так что блоки или выходные катушки могут быть непосредственно связаны с левой силовой шиной без промежуточных контактов.

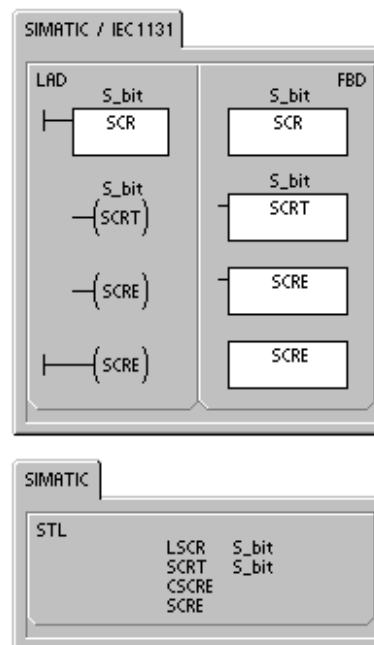
Ограничения

Используя SCR, примите во внимание следующие ограничения:

- Нельзя использовать один и тот же S-бит более чем в одной программе. Например, если вы используете S0.1 в главной программе, не используйте его в подпрограмме.
- Нельзя использовать команды JMP и LBL для перехода в сегмент SCR или из него; но вы можете использовать эти команды для перепрыгивания через сегменты SCR или для переходов внутри сегмента SCR.
- В сегменте SCR нельзя использовать команду END.

Таблица 6–59. Допустимые операнды для команд реле управления очередностью

Входы/выходы	Типы данных	Операнды
S_bit	BOOL	S



На рис. 6–32 показаны S-стек и логический стек и воздействие команды загрузки SCR. Для команд реле управления очередностью имеет силу следующее:

- Команда загрузки реле управления очередностью (LSCR) отмечает начало сегмента SCR, а команда завершения реле управления очередностью (SCRE) отмечает конец сегмента SCR. Исполнение всех операций между командами LSCR и SCRE зависит от значения S-стека. Исполнение операций между SCRE и следующей командой LSCR не зависит от значения S-стека.

- Команда перехода к следующему реле управления очередностью (SCRT) предоставляет средство для передачи управления от активного сегмента SCR другому сегменту SCR.

Команда перехода к следующему реле управления очередностью выполняется, когда ее достигает поток сигнала. Она сбрасывает S-бит активного в данный момент сегмента и устанавливает S-бит указанного сегмента. Сброс S-бита активного сегмента не оказывает влияния на S-стек на время выполнения команды перехода к следующему реле. Поэтому сегмент SCR остается активизированным, пока он не будет покинут.

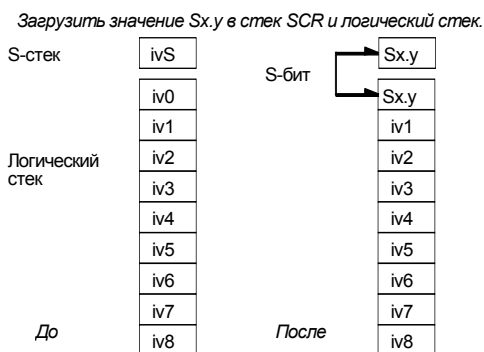
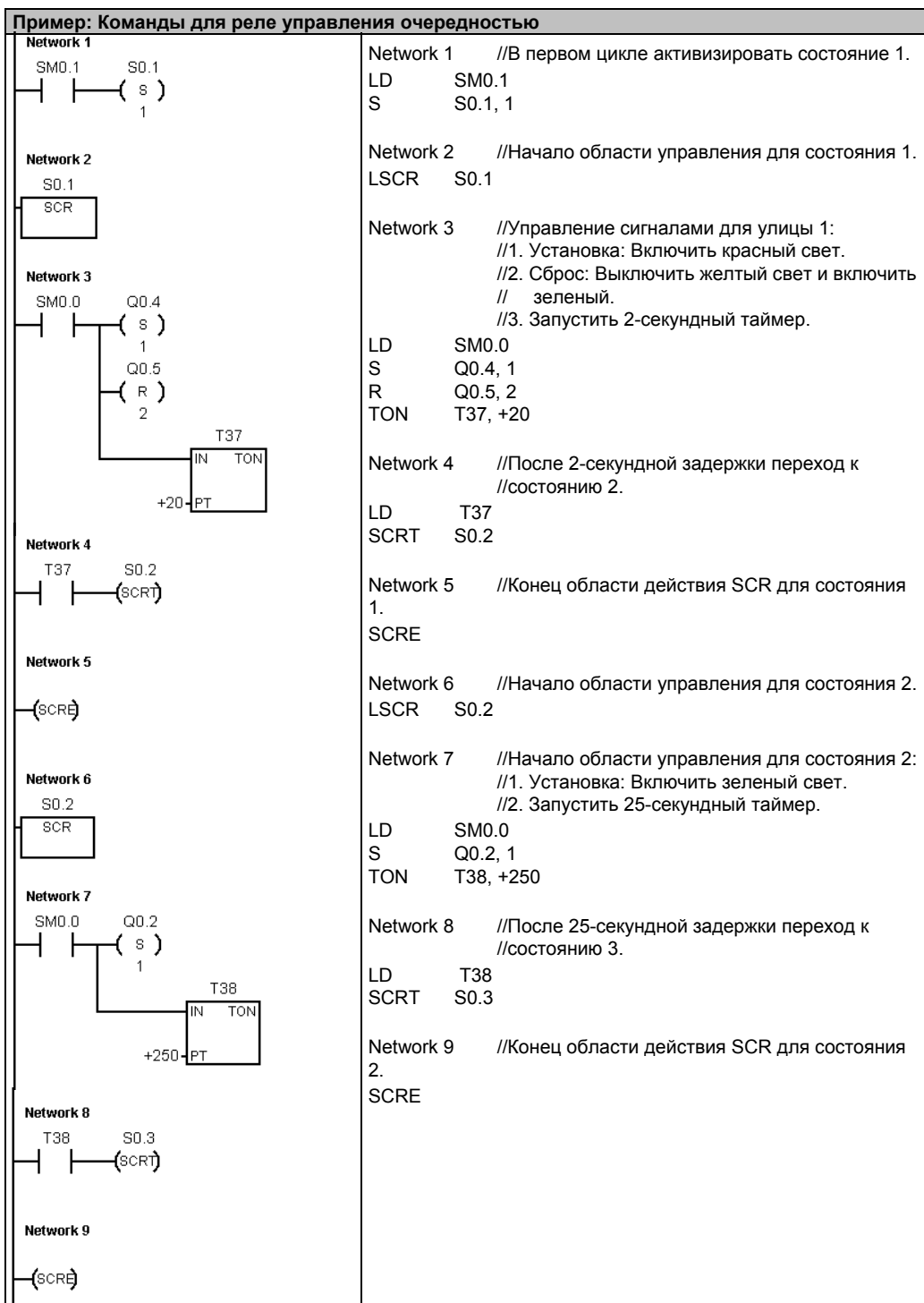


Рис. 6–32. Влияние LSCR на логический стек

- Команда условного завершения SCR (CSCRE) предоставляет средство для выхода из активного сегмента SCR без выполнения команд, расположенных между командой условного завершения SCR и командой завершения реле управления очередностью. Команда условного завершения SCR не оказывает влияния ни на S-бит, ни на S-стек.

В следующем примере бит первого цикла SM0.1 устанавливает S0.1, который в первом цикле будет в активном состоянии 1. После 2-секундной задержки T37 вызывает переход к состоянию 2. Этот переход деактивизирует сегмент SCR для состояния 1 (S0.1) и активизирует сегмент SCR для состояния 2 (S0.2).



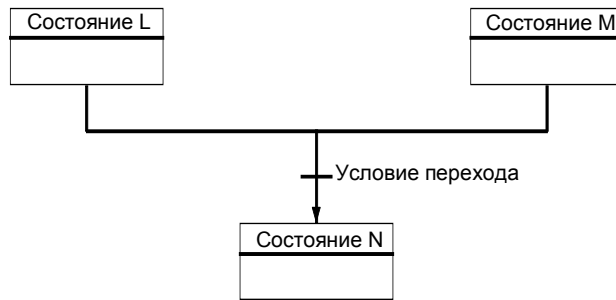
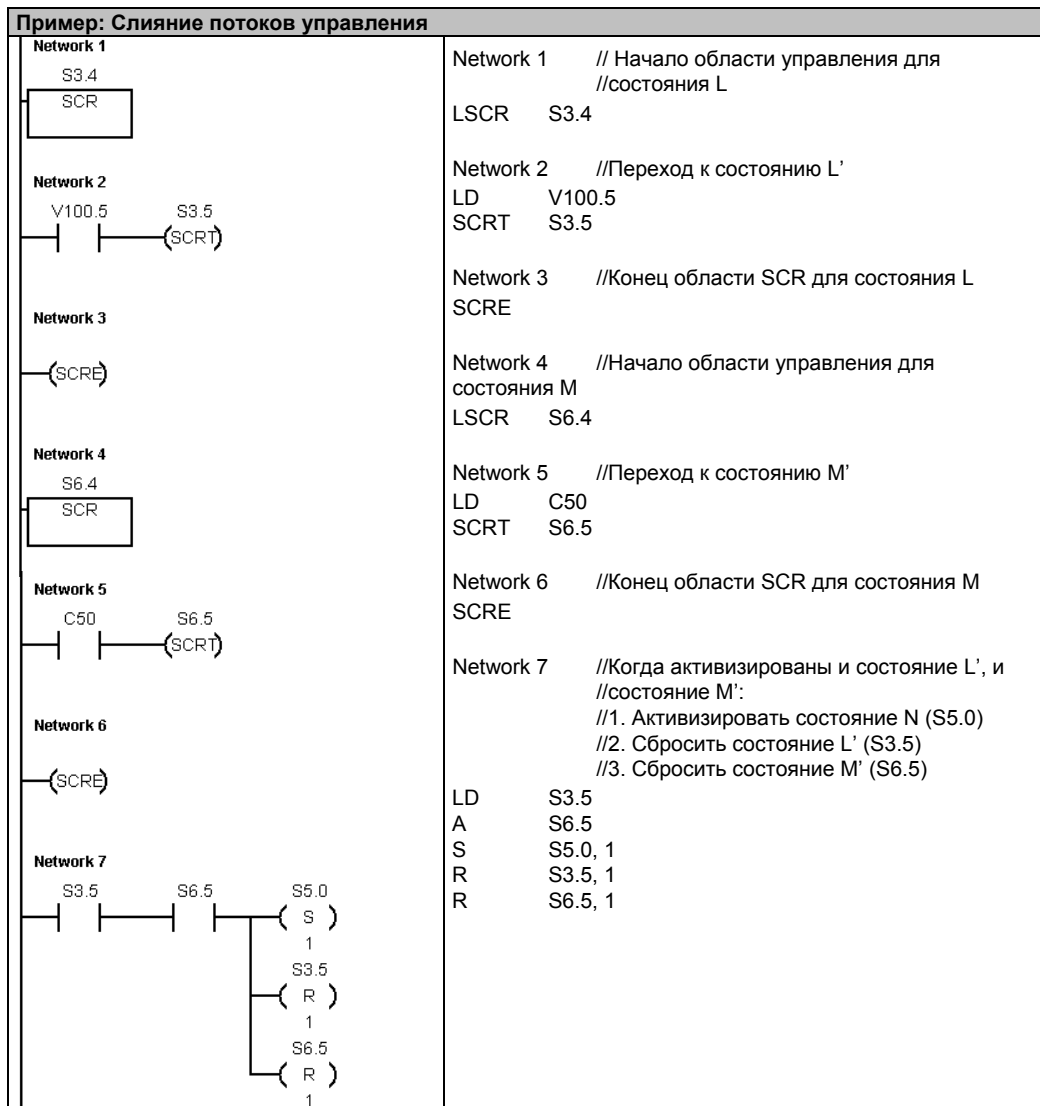


Рис. 6–34. Слияние потоков управления



В других ситуациях поток управления может быть направлен в один из нескольких возможных потоков управления в зависимости от того, какое условие перехода выполнилось первым. Такая ситуация изображена на рис. 6-35, где представлена эквивалентная программа SCR.

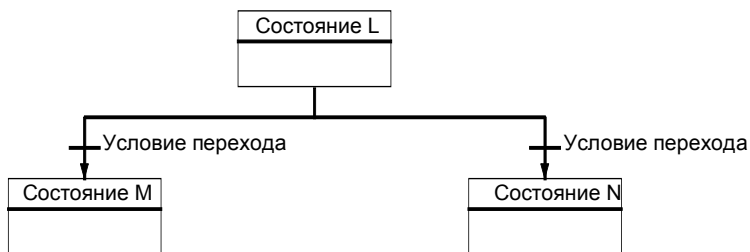


Рис. 6-35. Разделение потока управления в зависимости от условия перехода

Пример: Условные переходы	
Network 1 S3.4 SCR	Network 1 //Начало области управления для состояния L LSCR S3.4
Network 2 M2.3 S3.5 (SCRT)	Network 2 //Переход к состоянию M LD M2.3 SCRT S3.5
Network 3 I3.3 S6.5 (SCRT)	Network 3 //Переход к состоянию N LD I3.3 SCRT S6.5
Network 4 (SCRE)	Network 4 //Конец области SCR для состояния L SCRE

Команда управления диагностическим светодиодом

Если входной параметр IN имеет значение ноль, то диагностический светодиод выключается. Если входной параметр IN имеет значение больше нуля, то диагностический светодиод включается (желтый).

Светодиод CPU, обозначенный SF/ DIAG, может быть сконфигурирован таким образом, что он загорается желтым светом, когда выполняются условия, указанные в системном блоке данных или когда выполняется команда DIAG_LED с параметром IN, отличным от нуля.

Триггерные кнопки для системного блока данных (конфигурирование светодиода):

- светодиод SF/ DIAG включен (желтый), если некоторый элемент в CPU управляется принудительно
- светодиод SF/ DIAG LED включен (желтый), если в модуле имеет место ошибка ввода-вывода

Деактивизируйте обе опции для такого конфигурирования светодиода, чтобы желтым светом светодиода управляла только команда DIAG_LED. Системная ошибка CPU (SF) отображается красным светом.

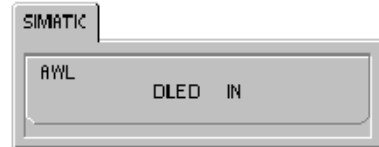
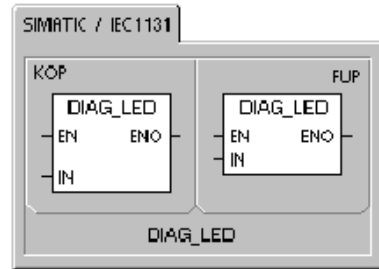
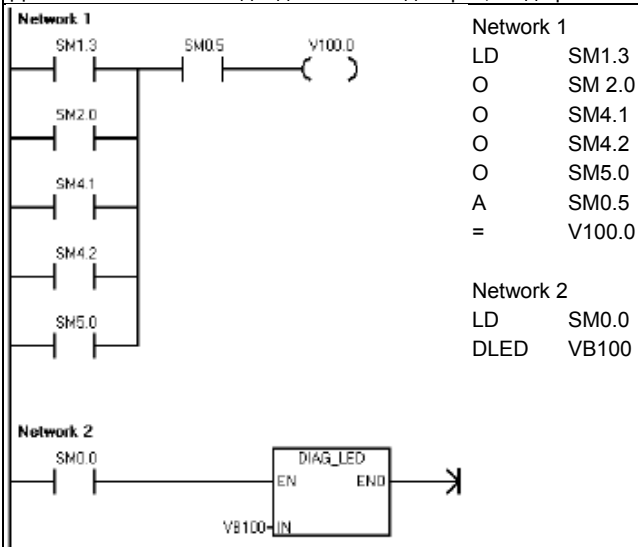


Таблица 6–60. Допустимые операнды для команды управления диагностическим светодиодом

Входы/выходы	Типы данных	Операнды
IN	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *LD, *AC

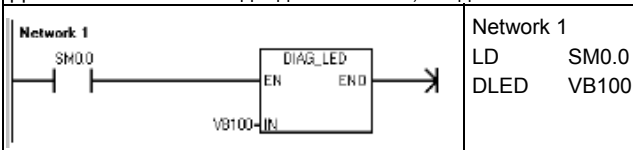
Пример 1 для команды управления диагностическим светодиодом

Диагностический светодиод мигает при обнаружении ошибки.
Диагностический светодиод мигает каждый раз, когда распознается одна из 5 сбойных ситуаций.



Пример 1 для команды управления диагностическим светодиодом

Диагностический светодиод включается, когда появляется ошибка.
Диагностический светодиод включается, когда в VB100 появляется код ошибки



Команды сдвига и циклического сдвига Marker 35

Команды сдвига вправо и сдвига влево

Команды сдвига сдвигают входную величину IN вправо или влево на число разрядов, указанное в N, и загружают результат в выход OUT.

Команды сдвига заполняют позиции выдвигаемых битов нулями. Если величина сдвига (N) больше или равна максимально допустимому значению (8 для операций с байтами, 16 для операций со словами и 32 для операций с двойными словами), то сдвиг производится на максимально возможную величину для данной операции. Если величина сдвига больше 0, то бит переполнения (SM1.1) принимает значение последнего выдвинутого бита. Бит нулевого значения (SM1.0) устанавливается, если результат операции сдвига равен нулю.

Байтовые операции являются беззнаковыми. Для операций со словами и двойными словами знаковый бит сдвигается, если вы используете типы данных со знаком.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.0 (ноль)
- SM1.1 (переполнение)

Команды циклического сдвига вправо и циклического сдвига влево

Команды циклического сдвига циклически сдвигают входную величину (IN) вправо или влево на число разрядов, указанное в (N), и загружают результат по адресу (OUT). Циклический сдвиг является кольцевым.

Если величина сдвига больше или равна максимально допустимому значению (8 для операций с байтами, 16 для операций со словами и 32 для операций с двойными словами), то S7-200 выполняет операцию по модулю с величиной сдвига (получение остатка от деления заданного сдвига на максимально допустимый), чтобы получить действительную величину циклического сдвига. Результатом является величина сдвига от 0 до 7 для операций с байтами, от 0 до 15 для операций со словами и от 0 до 31 для операций с двойными словами.

Если величина сдвига равна нулю, то циклический сдвиг не производится. Если циклический сдвиг выполняется, то значение последнего циклически сдвинутого бита копируется в бит переполнения (SM1.1).

Если величина сдвига не является целым кратным 8 (для операций с байтами), 16 (для операций со словами) или 32 (для операций с двойными словами), то последний циклически выдвинутый бит копируется в бит переполнения (SM1.1). Бит нулевого значения (SM1.0) устанавливается, если подлежащая циклическому сдвигу величина равна нулю.

Операции с байтами являются беззнаковыми. Для операций со словами и двойными словами знаковый бит сдвигается, если вы используете типы данных со знаком.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Биты специальной памяти, на которые действует команда:

- SM1.0 (ноль)
- SM1.1 (переполнение)

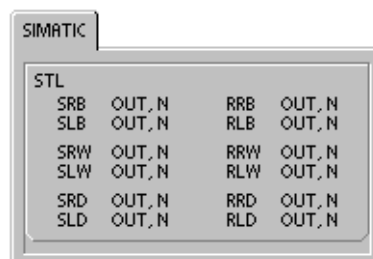
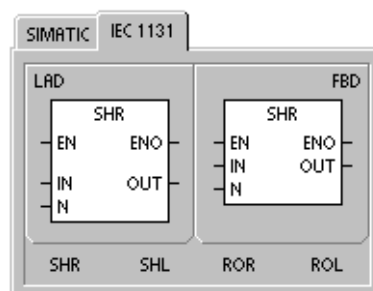
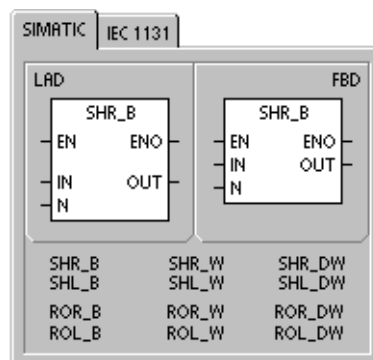


Таблица 6–61. Допустимые операнды для команд сдвига и циклического сдвига

Входы/выходы	Типы данных	Операнды
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа
OUT	BYTE	OB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа

Пример: Команды сдвига и циклического сдвига

<p>Network 1</p>		<p>Network 1</p> <pre>LD I4.0 RRW AC0, 2 SLW VW200, 3</pre>
<p>Циклический сдвиг</p> <p>Перед циклическим сдвигом</p> <p>AC0 0100 0000 0000 0001 x</p> <p>После 1-го сдвига</p> <p>AC0 1010 0000 0000 0000 1</p> <p>После 2-го сдвига</p> <p>AC0 0101 0000 0000 0000 0</p> <p>Бит нулевого результата (SM1.0) = 0 Бит переполнения (SM1.1) = 0</p>	<p>Сдвиг</p> <p>Перед сдвигом</p> <p>VW200 1110 0010 1010 1101 x</p> <p>После 1-го сдвига</p> <p>VW200 1100 0101 0101 1010 1</p> <p>После 2-го сдвига</p> <p>VW200 1000 1010 1011 0100 1</p> <p>После 3-го сдвига</p> <p>VW200 0001 0101 0110 1000 1</p> <p>Бит нулевого результата (SM1.0) = 0 Бит переполнения (SM1.1) = 1</p>	

Команда перемещения значения бита в регистр сдвига Marker 36

Команда перемещения значения бита в регистр сдвига перемещает значение в регистр сдвига. Эта команда предоставляет в распоряжение простой метод упорядочения и управления потоком изделий или данных. С помощью этой команды вы можете сдвигать весь регистр на один бит каждый цикл.

Команда перемещения значения бита в регистр сдвига вдвигает значение DATA в регистр сдвига. S_BIT указывает на младший бит регистра сдвига. N задает длину регистра сдвига и направление сдвига (положительный сдвиг = N, отрицательный сдвиг = -N).

Каждый бит, выдвинутый из регистра командой SHRB, помещается в бит переполнения (SM1.1).

Эта команда определяется младшим битом регистра сдвига (S_BIT) и количеством битов, задаваемых длиной (N).

Сбойные состояния, устанавливающие ENO = 0

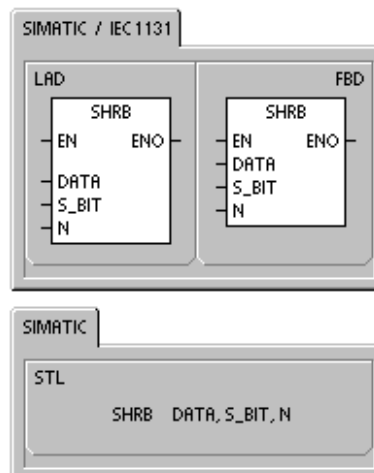
- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)
- 0092 (ошибка в поле счета)

Биты специальной памяти, на которые действует команда:

- SM1.1 (переполнение)

Таблица 6–62. Допустимые операнды для команды перемещения значения бита в регистр сдвига

Входы/выходы	Типы данных	Операнды
DATA, S_Bit	BOOL	I, Q, V, M, SM, S, T, C, L
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа



Адрес старшего бита регистра сдвига (MSB.b) можно рассчитать с помощью следующего уравнения:

$$MSB.b = [(\text{байт параметра } S_BIT) + ([N] - 1 + (\text{бит параметра } S_BIT)) / 8] \cdot [\text{остаток от деления на } 8]$$

Например: если S_BIT – это V33.4, и N = 14, то следующий расчет показывает, что MSB.b – это V35.1.

$$\begin{aligned} MSB.b &= V33 + ([14] - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2 \text{ с остатком } 1 \\ &= V35.1 \end{aligned}$$

При отрицательном сдвиге, который задается отрицательным значением длины (N), входные данные (DATA) вдвигаются на место старшего бита регистра сдвига и выдвигают из регистра младший бит (S_BIT). Данные, выдвинутые из регистра, помещаются в бит переполнения (SM1.1).

При положительном сдвиге, который задается положительным значением длины (N), входные данные (DATA) вдвигаются на место младшего бита регистра сдвига, указываемого параметром S_BIT, и выдвигают из регистра сдвига старший бит. Данные, выдвинутые из регистра, помещаются в бит переполнения (SM1.1).

Максимальная длина регистра сдвига (положительная или отрицательная) равна 64 битам. На рис. 6–36 показан сдвиг битов при отрицательном и положительном значении N.

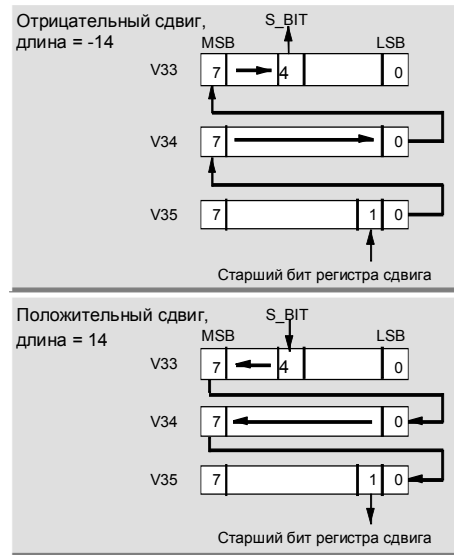
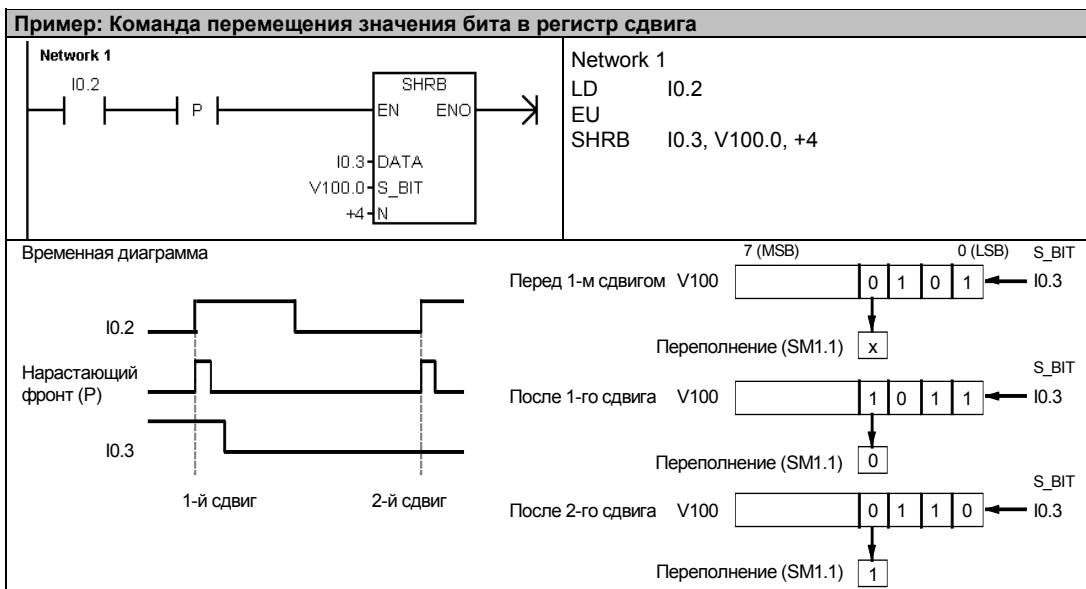


Рис. 6–36. Ввод и удаление данных из регистра сдвига



Команда перестановки байтов в слове

Команда перестановки байтов меняет местами старший и младший байты слова IN.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

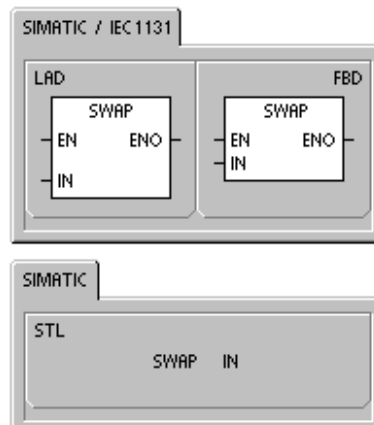
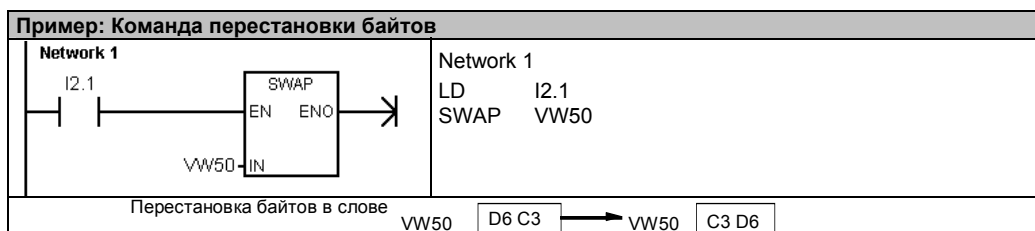


Таблица 6–63. Допустимые операнды для команды перестановки байтов в слове

Входы/выходы	Типы данных	Операнды
IN	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC



Строковые команды Marker 38

Длина строки

Команда определения длины строки (SLEN) возвращает длину строки, указанной в IN.

Копирование строки

Команда копирования строки (SCPY) копирует строку, указанную в IN, в строку, указанную в OUT.

Соединение строк

Команда соединения строк (SCAT) присоединяет строку, указанную в IN, к концу строки, указанной в OUT.

Биты специальной памяти и ENO

Для команд определения длины строки, копирования строки и соединения строк следующие условия влияют на ENO.

Сбойные состояния, устанавливающие ENO = 0

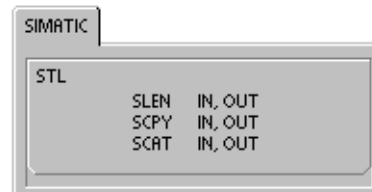
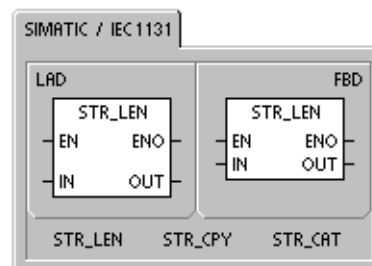
- 0006 (косвенный адрес)
- 0091 (ошибка диапазона)

Таблица 6–64. Допустимые операнды для команды определения длины строки

Входы/выходы	Типы данных	Операнды
IN	STRING	VB, LB, *VD, *LD, *AC, строковая константа
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Таблица 6–65. Допустимые операнды для команд копирования строки и соединения строк

Входы/выходы	Типы данных	Операнды
IN	STRING	VB, LB, *VD, *LD, *AC, строковая константа
OUT	STRING	VB, LB, *VD, *AC, *LD



Пример: Команды соединения строк, копирования строки и определения длины строки

<p>Network 1</p>	<p>Network 1</p> <pre>//1. Присоединить строку «WORLD» // к строке в VB0 //2. Скопировать строку, находящуюся // в VB0, в новую строку, // находящуюся в VB100 //3. Получить длину строки, // начинающейся по адресу VB100 LD I0.0 SCAT «WORLD», VB0 STRCPY VB0, VB100 STRLEN VB100, AC0</pre>
-------------------------	---

Перед выполнением программы

VB0	6	VB6
	'H' 'E' 'L' 'L' 'O' ''	

После выполнения программы

VB0	11	VB11
	'H' 'E' 'L' 'L' 'O' '' 'W' 'O' 'R' 'L' 'D'	
VB100	11	VB111
	'H' 'E' 'L' 'L' 'O' '' 'W' 'O' 'R' 'L' 'D'	
AC0	11	

Копирование подстроки из строки

Команда копирования подстроки из строки (SSCPY) копирует указанное количество символов N из строки, указанной в IN, начиная с индекса INDX, в новую строку, указанную OUT.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (ошибка диапазона)
- 009B (индекс=0)

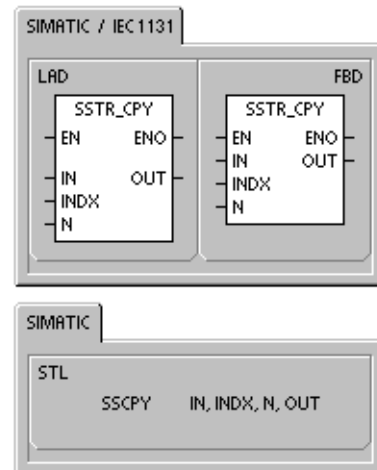
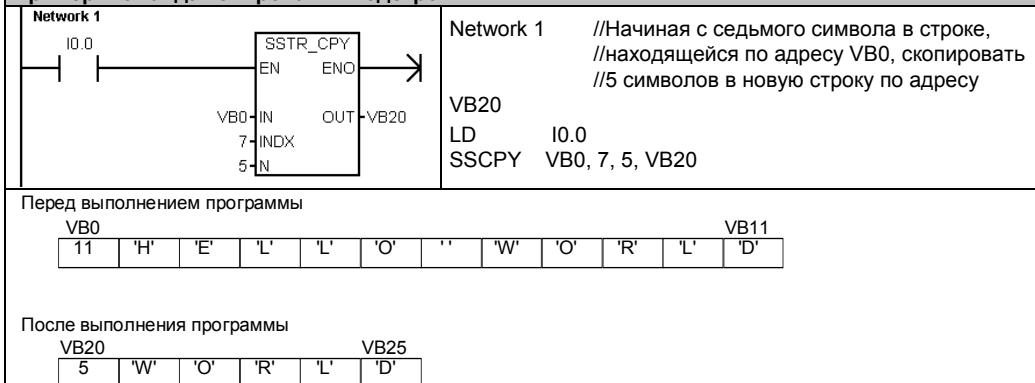


Таблица 6–66. Допустимые операнды для команды копирования подстроки из строки

Входы/выходы	Типы данных	Операнды
IN	STRING	VB, LB, *VD, *LD, *AC, строковая константа
OUT	STRING	VB, LB, *VD, *LD, *AC
INDX, N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа

Пример: Команда копирования подстроки



Поиск строки в строке

Команда поиска строки в строке (SFND) ищет первое появление строки IN2 в строке IN1. Поиск начинается с позиции, указанной в OUT. Если найдена последовательность символов, точно совпадающая со строкой IN2, то позиция первого символа в этой последовательности записывается в OUT. Если строка IN2 не найдена в строке IN1, то OUT устанавливается в 0.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (ошибка диапазона)
- 009B (индекс=0)

Поиск первого появления символа в строке

Команда поиска первого появления символа в строке (CFND) ищет в строке IN1 первое появление любого символа из набора символов, описываемого в строке IN2. Поиск начинается с позиции, указанной в OUT. Если совпадающий символ найден, то его положение записывается в OUT. Если совпадающего символа не найдено, то OUT устанавливается в 0.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (ошибка диапазона)
- 009B (индекс=0)

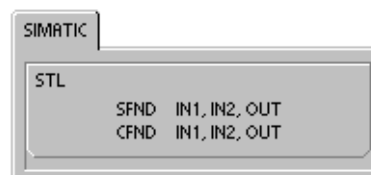
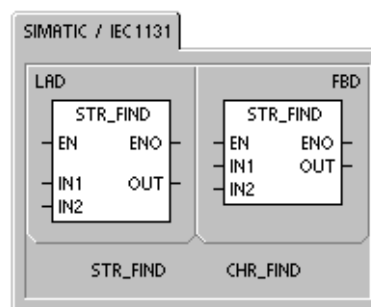


Таблица 6–67. Допустимые операнды для команд поиска строки в строке и первого символа в строке

Входы/выходы	Типы данных	Операнды
IN1, IN2	STRING	VB, LB, *VD, *LD, *AC, строковая константа
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Пример: Команда поиска строки в строке

В следующем примере строка, хранящаяся по адресу VB0, используется как команда для включения или выключения насоса. Строка 'On [Включить]' хранится по адресу VB20, а строка 'Off [Выключить]' хранится по адресу VB30. Результат команды поиска строки в строке сохраняется в AC0 (параметр OUT). Если результат не равен 0, то строка 'On' была найдена в командной строке (VB12).

Network 1

Network 1 //1. Установить AC0 в 1.
// (AC0 используется как параметр OUT.)
//2. Искать в строке, находящейся по адресу VB0, строку, находящуюся по адресу VB20 ('On [Включить]'), начиная с первой позиции (AC0=1).

```
LD I0.0
MOVB 1, AC0
SFND VB0, VB20, AC0
```

VB0	12	't'	'u'	'r'	'n'	''	'p'	'u'	'm'	'p'	''	'o'	'n'	VB12
VB20	2	'o'	'n'											
VB30	3	'o'	'f'	'f'										

Если строка из VB20 найдена: **AC0**

11

Если строка из VB20 не найдена: **AC0**

0

Пример: Команда поиска первого появления символа в строке

В следующем примере строка, хранящаяся по адресу VB0, содержит значение температуры. Строка, хранящаяся по адресу VB20, хранит все цифровые символы (в том числе + и -), которые могут указывать на температуру в строке. Данная программа-пример находит начальное положение числа в этой строке, а затем преобразует цифровые символы в вещественное число. VD200 хранит значение температуры в виде вещественного числа.

Network 1

Network 1 //1. Установить AC0 в 1.
// (AC0 используется в качестве параметра OUT и указывает на 1-ю позицию строки.)
//2. Найти цифровой символ в строке, находящейся по адресу VB0.
//3. Преобразовать строку в вещественное число.

```
LD I0.0
MOVB 1, AC0
CFND VB0, VB20, AC0
STR VB0, AC0, VD200
```

VB0	11	't'	'e'	'm'	'p'	''	''	'9'	'8'	'.'	'6'	'f'	VB11	
VB20	12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'.'	VB32

Начальная позиция температуры, хранящейся в VB0: **AC0**

7

Значение температуры в виде вещественного числа: **VD200**

98.6

Табличные команды

Ввод значения в таблицу

Команда ввода значений в таблицу вносит в таблицу (TBL) значения, имеющие размер слова (DATA). Первым значением таблицы является ее максимальная длина (TL). Второе значение – это количество записей в таблице (EC). Новые данные добавляются к таблице после последней записи. Каждый раз, когда к таблице добавляются новые данные, количество записей увеличивается на единицу.

Таблица может иметь до 100 записей.

Сбойные состояния, устанавливающие ENO = 0

- SM1.4 (переполнение таблицы)
- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

Биты специальной памяти, на которые действует команда:

- SM1.4 устанавливается в 1, если вы пытаетесь переполнить таблицу

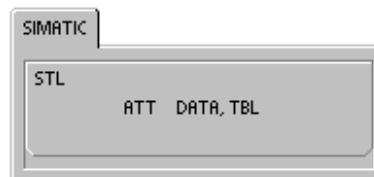
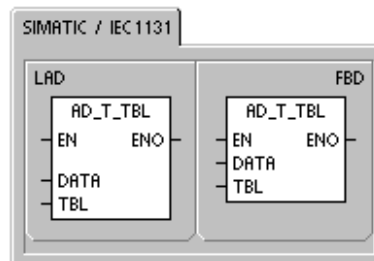


Таблица 6–68. Допустимые операнды для табличных команд

Входы/выходы	Типы данных	Операнды
DATA	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
TBL	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD, *AC

Пример: Команда ввода значения в таблицу

Network 1

Network 2

Network 1 //Загрузить максимальную длину таблицы

```
LD SM0.1
MOVW +6, VW200
```

Network 2

```
LD I0.0
ATT VW100, VW200
```

Перед выполнением ATT

VW100	1234
VW200	0006
VW202	0002
VW204	5431
VW206	8942
VW208	xxxx
VW210	xxxx
VW212	xxxx
VW214	xxxx

После выполнения ATT

VW200	0006	TL (макс. число записей.)
VW202	0003	EC (число записей)
VW204	5431	d0 (данные 0)
VW206	8942	d1 (данные 1)
VW208	1234	d2 (данные 2)
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

Удаление первой записи из таблицы и удаление последней записи из таблицы

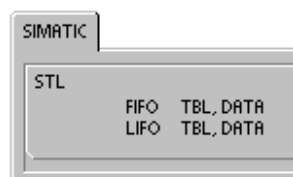
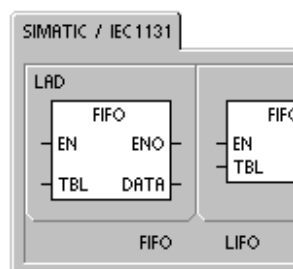
Таблица может иметь до 100 записей.

Удаление первой записи

Команда удаления первой записи (FIFO) удаляет первую (самую старую) запись из таблицы (TBL) и пересылает значение по адресу, указанному в DATA. Все остальные записи таблицы смещаются на одну позицию вверх. Количество записей в таблице после выполнения каждой команды уменьшается на 1.

Удаление последней записи

Команда удаления последней записи (LIFO) удаляет последнюю (самую новую) запись из таблицы (TBL) и пересылает значение по адресу, указанному в DATA. Количество записей в таблице после выполнения каждой команды уменьшается на 1.



Сбойные состояния, устанавливающие ENO = 0

- SM1.5 (пустая таблица)
- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

Биты специальной памяти, на которые действует команда:

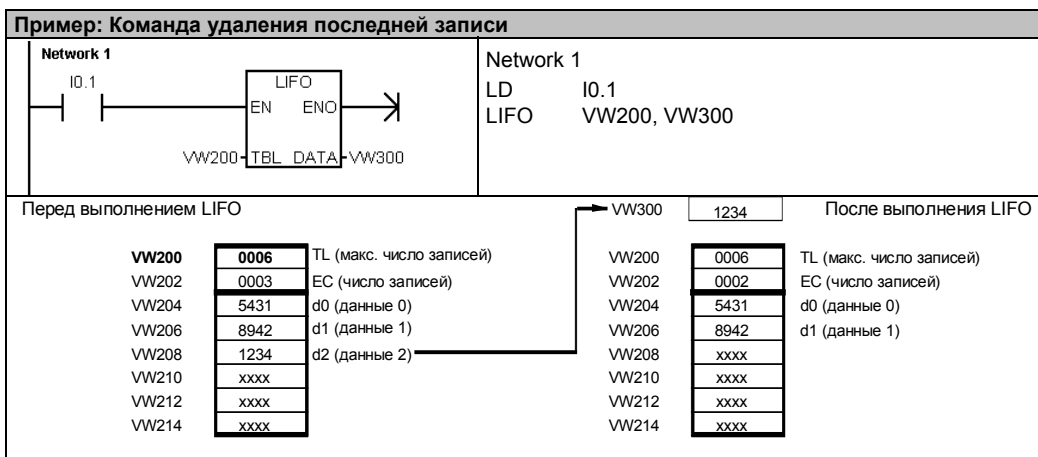
- SM1.5 устанавливается в 1, если вы пытаетесь удалить запись из пустой таблицы

Таблица 6–69. Допустимые операнды для команд удаления первой и последней записи

Входы/выходы	Типы данных	Операнды
TBL	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD, *AC
DATA	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC

Пример: Команда удаления первой записи

<p>Network 1</p>		<p>Network 1</p> <pre>LD I4.1 FIFO VW200, VW400</pre>																																	
<p>Перед выполнением FIFO</p> <table border="1"> <tr><td>VW200</td><td>0006</td></tr> <tr><td>VW202</td><td>0003</td></tr> <tr><td>VW204</td><td>5431</td></tr> <tr><td>VW206</td><td>8942</td></tr> <tr><td>VW208</td><td>1234</td></tr> <tr><td>VW210</td><td>xxxx</td></tr> <tr><td>VW212</td><td>xxxx</td></tr> <tr><td>VW214</td><td>xxxx</td></tr> </table> <p>TL (макс. число записей) EC (число записей) d0 (данные 0) d1 (данные 1) d2 (данные 2)</p>		VW200	0006	VW202	0003	VW204	5431	VW206	8942	VW208	1234	VW210	xxxx	VW212	xxxx	VW214	xxxx	<p>После выполнения FIFO</p> <table border="1"> <tr><td>VW200</td><td>0006</td></tr> <tr><td>VW202</td><td>0002</td></tr> <tr><td>VW204</td><td>8942</td></tr> <tr><td>VW206</td><td>1234</td></tr> <tr><td>VW208</td><td>xxxx</td></tr> <tr><td>VW210</td><td>xxxx</td></tr> <tr><td>VW212</td><td>xxxx</td></tr> <tr><td>VW214</td><td>xxxx</td></tr> </table> <p>TL (макс. число записей) EC (число записей) d0 (данные 0) d1 (данные 1)</p>		VW200	0006	VW202	0002	VW204	8942	VW206	1234	VW208	xxxx	VW210	xxxx	VW212	xxxx	VW214	xxxx
VW200	0006																																		
VW202	0003																																		
VW204	5431																																		
VW206	8942																																		
VW208	1234																																		
VW210	xxxx																																		
VW212	xxxx																																		
VW214	xxxx																																		
VW200	0006																																		
VW202	0002																																		
VW204	8942																																		
VW206	1234																																		
VW208	xxxx																																		
VW210	xxxx																																		
VW212	xxxx																																		
VW214	xxxx																																		



Заполнение памяти комбинацией двоичных разрядов

Команда заполнения памяти (FILL) записывает N последовательных слов, начиная с адреса OUT, значением слова, содержащегося по адресу IN.

N имеет диапазон от 1 до 255.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

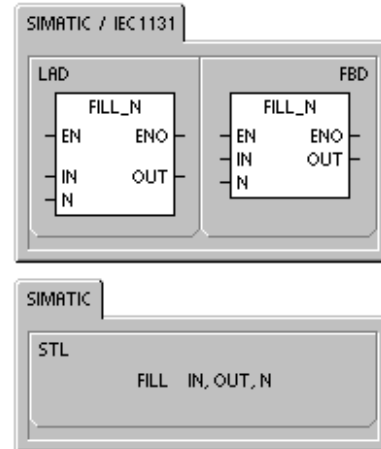


Таблица 6–70. Допустимые операнды для команды заполнения памяти

Входы/выходы	Типы данных	Операнды
IN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AQW, *VD, *LD, *AC



Поиск значения в таблице

Команда поиска значения в таблице (FND) ищет в таблице данные, удовлетворяющие определенному критерию. Команда поиска значения в таблице ищет в таблице TBL, начиная с записи INDX, значение или образец данных PTN, удовлетворяющее критерию поиска, определенному CMD. Командному параметру CMD дается числовое значение от 1 до 4, что соответственно означает =, <>, < и >.

Если найдена запись, удовлетворяющая критерию поиска, то INDX указывает на эту запись в таблице. Для нахождения следующей записи, удовлетворяющей критерию поиска, то INDX должен быть увеличен, прежде чем команда Поиск в таблице будет вызвана снова. Если запись, удовлетворяющая критерию поиска, не найдена, то INDX имеет значение, равное количеству записей.

Таблица может иметь до 100 записей. Записи данных (область поиска) нумеруются от 0 до максимального значения, равного 99.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (операнд вышел за пределы допустимого диапазона)

Таблица 6–71. Допустимые операнды для команды поиска значения в таблице

Входы/выходы	Типы данных	Операнды
TBL	WORD	IW, QW, VW, MW, SMW, T, C, LW, *VD, *LD, *AC
PTN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа
INDX	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
CMD	BYTE	(константа) 1: равно (=), 2: не равно (<>), 3: меньше (<), 4: больше (>)



Совет

Когда вы используете команду поиска в таблицах, созданных с помощью команд ввода значения в таблицу (ATT), удаления первой записи (LIFO) и удаления последней записи (FIFO), то количество записей и сами записи данных согласуются непосредственно. Слово, определяющее максимальное количество записей, требуемое для команд ATT, LIFO и FIFO, не требуется для команды поиска. См. рис. 6–37.

Поэтому вы должны установить операнд TBL команды поиска в таблице на один адрес слова (два байта) выше, чем операнд TBL соответствующей команды ATT, LIFO или FIFO.

Формат таблицы для ATT, LIFO и FIFO

VW200	0006	TL (макс. кол-во записей)
VW202	0006	ЕС (число записей)
VW204	xxxx	d0 (данные 0)
VW206	xxxx	d1 (данные 1)
VW208	xxxx	d2 (данные 2)
VW210	xxxx	d3 (данные 3)
VW212	xxxx	d4 (данные 4)
VW214	xxxx	d5 (данные 5)

Формат таблицы для TBL_FIND

VW202	0006	ЕС (число записей)
VW204	xxxx	d0 (данные 0)
VW206	xxxx	d1 (данные 1)
VW208	xxxx	d2 (данные 2)
VW210	xxxx	d3 (данные 3)
VW212	xxxx	d4 (данные 4)
VW214	xxxx	d5 (данные 5)

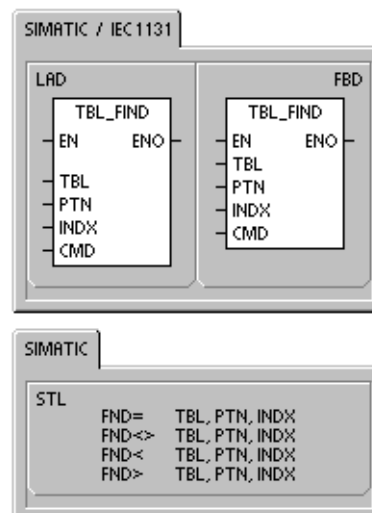


Рис. 6–37. Различия в форматах таблиц у команд "Поиск в таблице" и команд ATT, LIFO и FIFO

Пример: Команда поиска значения в таблице

Network 1

Network 1

LD I2.1
FND= VW202, 16#3130, AC1

Если I2.1 включен, то в таблице ищется число, равное 3130 HEX.

VW202	0006	EC (число записей)
VW204	3133	d0 (данные 0)
VW206	4142	d1 (данные 1)
VW208	3130	d2 (данные 2)
VW210	3030	d3 (данные 3)
VW212	3130	d4 (данные 4)
VW214	4541	d5 (данные 5)

Если таблица была создана командой ATT, LIFO или FIFO, то VW200 содержит максимально разрешенное количество записей и не требуется для команд поиска.

AC1

Для поиска с самой верхней записи в таблице AC1 должен быть установлен в 0

Поиск в таблице

AC1

AC1 содержит номер записи данных, которая соответствует критерию поиска.

AC1

Перед поиском остальных записей в таблице увеличить INDX на 1

Поиск в таблице

AC1

AC1 содержит номер записи данных, которая соответствует второму совпадению, найденному в таблице (d4)

AC1

Перед поиском остальных записей в таблице увеличить INDX на 1.

Поиск в таблице

AC1

AC1 содержит значение, равное количеству записей. Вся таблица была просмотрена в безуспешных поисках еще одного совпадения.

AC1

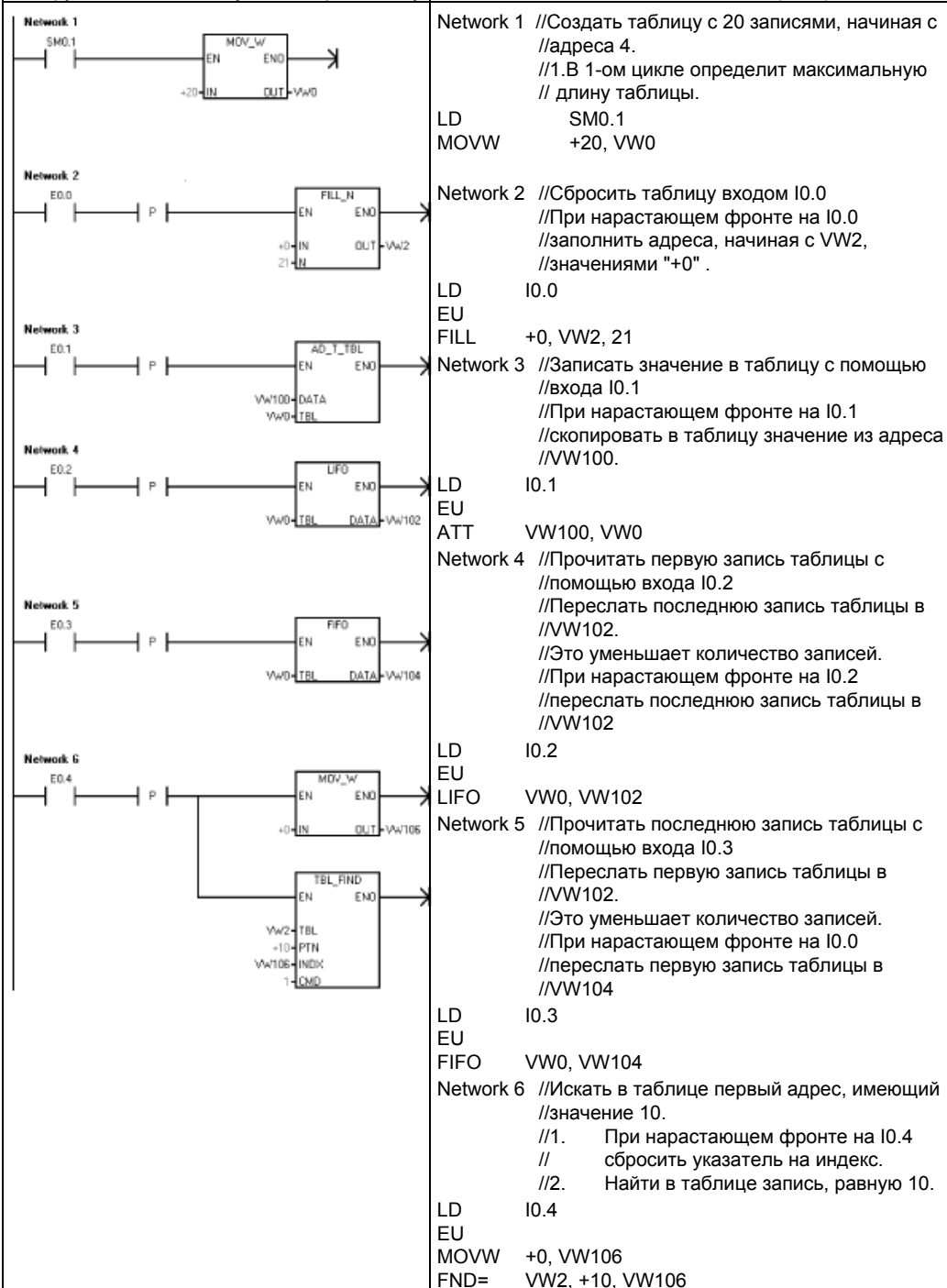
Перед новым поиском в таблице INDX должен быть установлен в 0.

Пример: Создание таблицы

Следующая программа создает таблицу с 20 записями. Первый адрес таблицы содержит длину таблицы (в данном случае 20 записей). Второй адрес показывает текущее количество записей в таблице. Остальные адреса содержат записи. Таблица может иметь до 100 записей. Сюда не входят параметры, определяющие максимальную длину таблицы и текущее количество записей (здесь VW0 и VW2). Текущее количество записей в таблице (здесь VW2) CPU автоматически увеличивает или уменьшает на 1 при выполнении каждой команды.

Перед началом работы с таблицей назначьте максимальное количество записей таблицы. В противном случае вы не сможете вводить записи в таблицу. Кроме того, обеспечьте, чтобы все команды чтения и записи активизировались с помощью фронтов.

Для поиска в таблице индекс (VW106) перед выполнением поиска должен быть установлен в 0. Если найдено совпадение, индекс будет указывать номер соответствующей записи, а если совпадение не обнаружено, то индекс будет содержать текущее количество записей для таблицы (VW2).



Таймерные команды

Таймерные команды SIMATIC

Таймер с задержкой включения Таймер с задержкой включения с запоминанием

Команды "Таймер с задержкой включения" (TON) и "Таймер с задержкой включения с запоминанием" (TONR) отсчитывают время, когда включен разрешающий вход. Номер таймера (Тхх) определяет его разрешающую способность, и эта разрешающая способность теперь отображается в блоке команды.

Таймер с задержкой выключения

Таймер с задержкой выключения (TOF) используется для задержки выключения выхода на фиксированный интервал времени после выключения входа. Номер таймера (Тхх) определяет его разрешающую способность, и эта разрешающая способность теперь отображается в блоке команды.

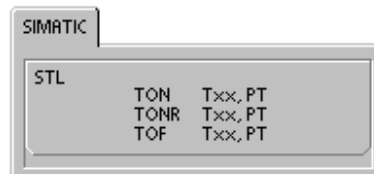
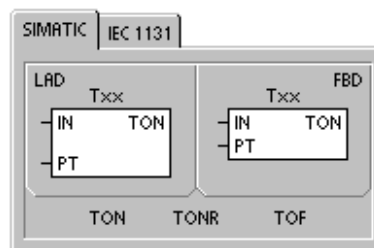


Таблица 6–72. Допустимые операнды для таймерных команд SIMATIC

Входы/выходы	Типы данных	Операнды
Txx	WORD	константа (от T0 до T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
PT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа



Совет

Таймерам с задержкой выключения (TOF) и с задержкой включения (TON) нельзя задавать один и тот же номер (Тхх). Например, у вас не могут одновременно использоваться TON T32 и TOF T32.

Как показано в таблице 6–73, эти три вида таймеров выполняют различные задачи измерения времени:

- Таймер с задержкой включения TON может использоваться для отсчета отдельного интервала.
- Таймер с задержкой включения с запоминанием TONR может использоваться для накопления нескольких отсчитанных интервалов времени.
- Таймер с задержкой выключения TOF может использоваться для увеличения интервала времени после выключения (или сбоя), например, для охлаждения двигателя после его отключения.

Таблица 6–73. Действия таймеров

Тип	Текущее время >= предустановленному	Состояние разрешающего входа (IN)	Выключение-включение питания / первый цикл
TON	Бит таймера установлен Отсчет текущего значения продолжается до 32 767	ON: Текущее значение отсчитывает время OFF: Бит таймера сброшен, текущее значение = 0	Бит таймера сброшен Текущее значение = 0
TONR	Бит таймера установлен Отсчет текущего значения продолжается до 32 767	ON: Текущее значение отсчитывает время OFF: Бит таймера и текущее значение сохраняют последнее состояние	Бит таймера сброшен Текущее значение может быть сохранено ¹
TOF	Бит таймера сброшен Текущее время = предустановленному, отсчет времени прекращен	ON: Бит таймера установлен, текущее значение = 0 OFF: Таймер выполняет отсчет времени после перехода из включенного состояния в выключенное	Бит таймера сброшен Текущее значение = 0

¹ Текущее значение таймера с запоминанием может быть выбрано в качестве сохраняемого значения на время выключения питания. Информацию о сохранении памяти для CPU S7–200 вы найдете в главе 4.



Советы по
программированию

Советы по программированию на компакт-диске с документацией содержат пример программы, использующей таймер с задержкой включения (TON). См. Совет 31

Команды TON и TONR отсчитывают время, когда включен разрешающий вход. Когда текущее значение становится больше или равно предустановленному времени, бит таймера устанавливается.

- Текущее значение таймера TON сбрасывается, когда выключается разрешающий вход, тогда как текущее значение таймера TONR сохраняется, когда этот вход выключается.
- Вы можете использовать таймер TONR для накопления времени, когда этот вход включается и выключается. Для стирания текущего значения TONR используется команда Сброс (R).
- Таймеры TON и TONR продолжают счет после достижения предустановленного значения, они останавливают счет при достижении максимального значения, равного 32767.

Команда TOF используется для задержки выключения выхода на фиксированный интервал времени после выключения входа. Когда включается разрешающий вход, немедленно включается бит таймера, а текущее значение устанавливается в 0. Когда вход выключается, таймер ведет отсчет времени, пока истекшее время не достигнет предустановленного значения.

- Когда предустановленное время достигнуто, бит таймера сбрасывается, а отсчет текущего значения прекращается; однако, если вход включается снова, прежде чем TOF достигнет предустановленного значения, то бит таймера остается установленным.
- Чтобы таймер TOF начал отсчет времени, к его разрешающему входу должен быть приложен падающий фронт.
- Если таймер TOF находится внутри области SCR, и область SCR не активна, то текущее значение устанавливается в 0, бит таймера выключается, и отсчет текущего значения не ведется.



Совет

Таймер TONR может быть сброшен только командой сброса (R). Команду сброса можно использовать также для сброса любого таймера TON или TOF. Команда сброса выполняет следующие операции:

- Бит таймера сбрасывается
- Текущее значение таймера = 0

После сброса таймеры TOF требуют для своего перезапуска появления отрицательного фронта на разрешающем входе.

Определение разрешающей способности таймера

Таймеры отсчитывают интервалы времени. Разрешающая способность (или база времени) таймера определяет промежуток времени на один интервал. Например, TON с разрешающей способностью 10 мс отсчитывает количество 10-миллисекундных интервалов, прошедших после активизации TON: отсчет 50 на 10-миллисекундном таймере представляет 500 мс. Имеются таймеры SIMATIC с тремя разрешающими способностями: 1 мс, 10 мс и 100 мс. Как показано в таблице 6-74, номер таймера определяет его разрешающую способность.


	<p>Совет</p> <p>Чтобы гарантировать минимальный интервал времени, увеличьте предустановленное значение (PV) на 1. Например: чтобы гарантировать минимальный интервал времени не менее 2100 мс для 100–миллисекундного таймера, установите PV на 22.</p>
---	--

Таблица 6–74. Номера таймеров и разрешения

Тип таймера	Разрешение	Максимальное значение	Номер таймера
TONR (с запоминанием)	1 мс	32,767 с (0,546 мин.)	T0, T64
	10 мс	327,67 с (5,46 мин.)	T1 – T4, T65 – T68
	100 мс	3276,7 с (54,6 мин.)	T5 – T31, T69 – T95
TON, TOF (без запоминания)	1 мс	32,767 с (0,546 мин.)	T32, T96
	10 мс	327,67 с (5,46 мин.)	T33 – T36, T97 – T100
	100 мс	3276,7 с (54,6 мин.)	T37 – T63, T101 – T255

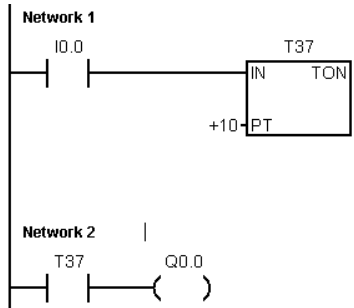
Влияние разрешающей способности на работу таймера

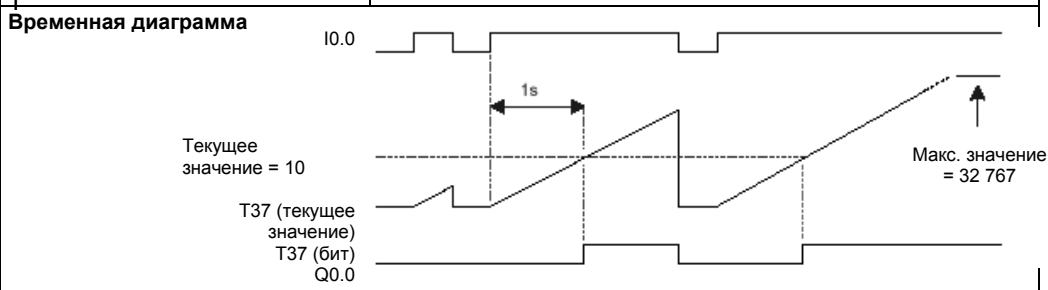
У таймера с разрешающей способностью 1 мс бит таймера и текущее значение обновляются асинхронно с циклом обработки программы. Для циклов, превышающих 1 мс, бит таймера и текущее значение обновляются несколько раз в течение цикла.

У таймера с разрешающей способностью 10 мс бит таймера и текущее значение обновляются в начале каждого цикла обработки программы. Текущее значение и бит таймера остаются неизменными на протяжении цикла, и интервалы времени, накопленные в течение цикла, добавляются к текущему значению в начале каждого цикла обработки программы.

У таймера с разрешающей способностью 100 мс бит таймера и текущее значение обновляются, когда выполняется команда; поэтому, чтобы таймер сохранял правильное значение времени, обратите внимание на то, чтобы ваша программа выполняла команду для 100–миллисекундного таймера только один раз за цикл.

Пример: Таймер SIMATIC с задержкой включения

	<pre> Network 1 //100-миллисекундный таймер T37 заканчивает //отсчет времени по истечении (10 x 100 мс = 1 с) //I0.0 установлен = активизировать T37, //I0.0 сброшен = деактивизировать и сбросить T37 LD I0.0 TON T37, +10 Network 2 //Бит T37 управляется таймером T37 LD T37 = Q0.0 </pre>
---	--

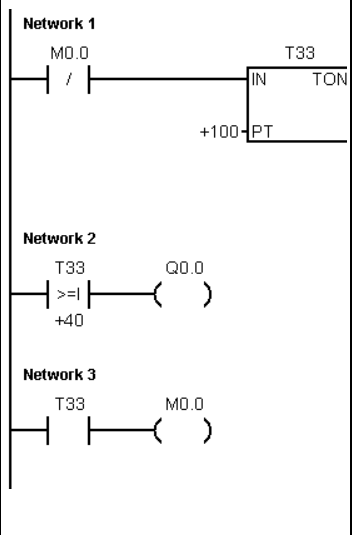




Совет

Чтобы гарантировать, что выход самосбрасывающегося таймера включался на время одного цикла каждый раз, когда таймер достигает предустановленного значения, вы должны использовать на разрешающем входе размыкающий контакт вместо бита таймера.

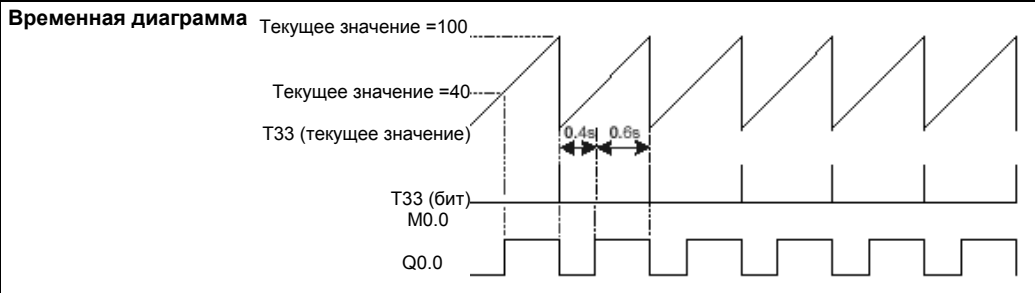
Пример: Самосбрасывающийся таймер SIMATIC с задержкой включения



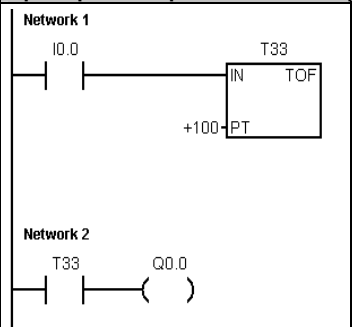
Network 1 //10-миллисекундный таймер T33 заканчивает отсчет //времени по истечении (100 x 10 мс = 1 с) //Импульс M0.0 слишком быстр для наблюдения в //представлении статуса программы
LDN M0.0
TON T33, +100

Network 2 //Сравнение осуществляется с темпом, позволяющим //наблюдать его результат в представлении статуса //программы. Включать Q0.0 через (40 x 10 мс) для //получения сигнала 40% выкл./60% вкл.
LDW>= T33, +40
= Q0.0

Network 3 //Импульс T33 (бит) слишком быстр для наблюдения в //представлении статуса программы. //Сбросить таймер с помощью M0.0 по истечении //периода (100 x 10 мс)
LD T33
= M0.0

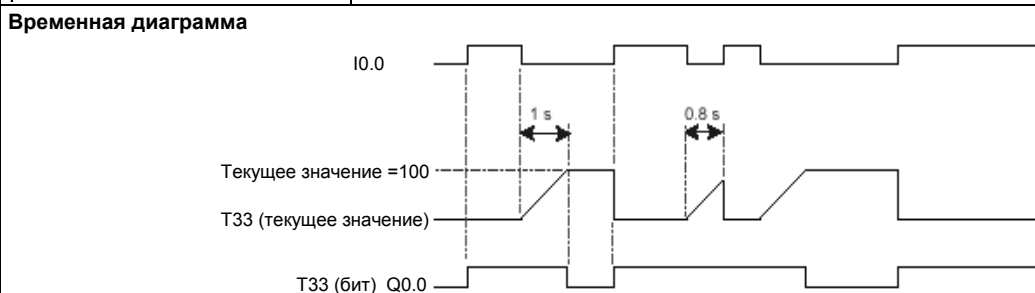


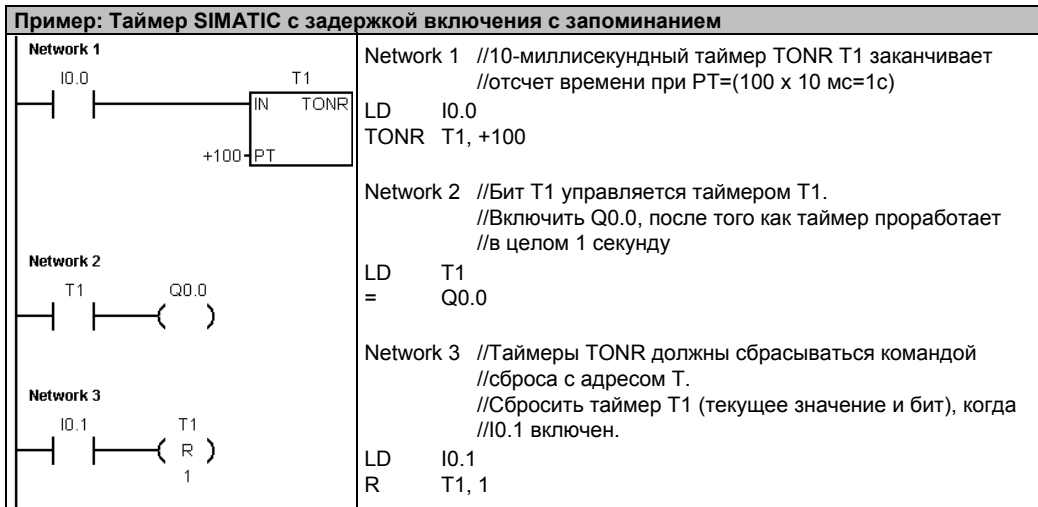
Пример: Таймер SIMATIC с задержкой выключения



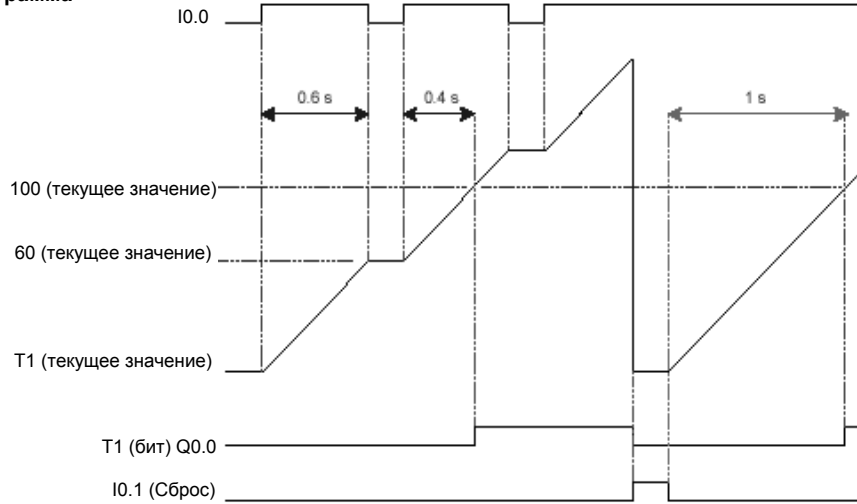
Network 1 //10-миллисекундный таймер T33 заканчивает отсчет //времени по истечении (100 x 10 мс = 1 с) //При выключении I0.0 T33 активизируется // При включении I0.0 T33 деактивируется и //сбрасывается
LD I0.0
TOF T33, +100

Network 2 //Таймер T33 управляет Q0.0 через контакт таймера //T33
LD T33
= Q0.0





Временная диаграмма



Таймерные команды IEC

Таймер с задержкой включения

Команда "Таймер с задержкой включения" (TON) ведет отсчет времени, когда включен разрешающий вход.

Таймер с задержкой выключения

Команда "Таймер с задержкой выключения" (TOF) задерживает выключение выхода на фиксированный интервал времени после выключения входа.

Импульсный таймер

Импульсный таймер (TP) генерирует импульсы заданной длительности.

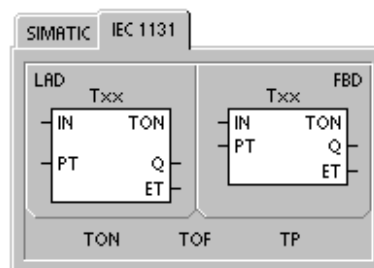


Таблица 6–75. Допустимые операнды для таймерных команд IEC

Входы/выходы	Типы данных	Операнды
Txx	TON, TOF, TP	константа (T32 – T63, T96 – T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
PT	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, константа
Q	BOOL	I, Q, V, M, SM, S, L
ET	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC



Совет

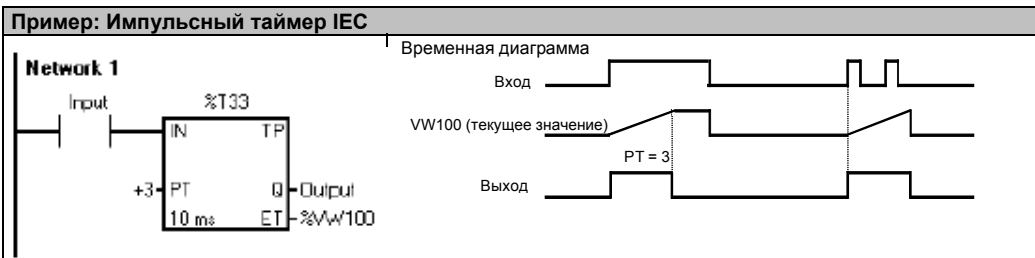
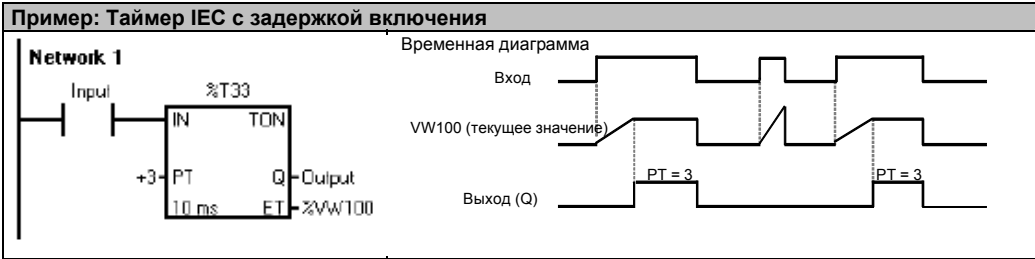
Нельзя совместно использовать одинаковые номера таймеров для TOF, TP и TON. Например, вы не можете иметь TON T32 и TOF T32.

- Команда TON ведет отсчет времени до предварительно заданного значения с момента, когда разрешающий вход (IN) становится истинным. Когда истекшее время (ET) равно заданному времени (PT), выходной бит таймера (Q) включается. Выходной бит сбрасывается, когда разрешающий вход выключается. Когда заданное время достигнуто, отсчет времени прекращается и таймер блокируется.
- Команда TOF задерживает выключение выхода на фиксированный интервал времени после выключения входа. Она ведет отсчет времени до предварительно заданного значения с момента, когда разрешающий вход (IN) выключается. Когда истекшее время (ET) равно предварительно установленному времени (PT), выходной бит таймера (Q) выключается. Как только предварительная установка достигнута, выходной бит таймера выключается, а истекшее время сохраняется до тех пор, пока разрешающий вход (IN) не включится снова. Если разрешающий вход переходит в выключенное состояние на интервал времени, более короткий, чем предустановленное время, то выходной бит остается включенным.
- Команда TP генерирует импульсы заданной длительности. Когда разрешающий вход (IN) включается, то включается и выходной бит (Q). Выходной бит остается включенным для указанного импульса в течение заранее заданного времени (PT). Как только истекшее время (ET) достигнет предварительной установки (PT), выходной бит выключается. Истекшее время сохраняется, пока не выключится разрешающий вход. Когда выходной бит включается, он остается включенным до истечения времени импульса.

Каждый отсчет текущего значения кратен базе времени. Например, значение 50 в 10–миллисекундном таймере представляет величину 500 мс. Таймеры IEC (TON, TOF и TP) имеют с тремя разрешающими способностями. Разрешающая способность определяется номером таймера, как показано в таблице 6–76.

Таблица 6–76. Разрешающая способность таймеров IEC

Разрешающая способность	Максимальное значение	Номер таймера
1 мс	32,767 с (0,546 минуты)	T32, T96
10 мс	327,67 с (5,46 минуты)	T33 – T36, T97 – T100
100 мс	3276,7 с (54,6 минуты)	T37 – T63, T101 – T255



Интервальные таймеры

Начало отсчета времени интервала

Команда начала отсчета времени интервала (BITIM) считывает текущее значение встроенного 1-миллисекундного счетчика и сохраняет это значение в OUT. Максимальный интервал времени для миллисекундного значения типа DWORD составляет 2 в 32-ой степени или 49,7 дня.

Расчет времени интервала

Команда расчета времени интервала (CITIM) рассчитывает разность между текущим временем и временем, поступающие в IN. Эта разность сохраняется в OUT. Максимальный интервал времени для миллисекундного значения типа DWORD составляет 2 в 32-ой степени или 49,7 дня. CITIM автоматически обрабатывает переполнение 1-миллисекундного таймера, которое возникает внутри максимального интервала, в зависимости от того, когда была исполнена команда BITIM.

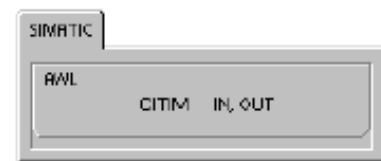
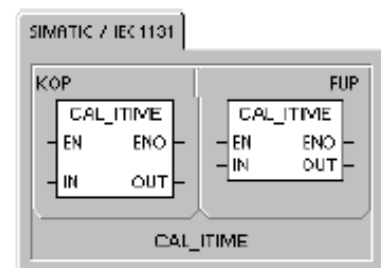
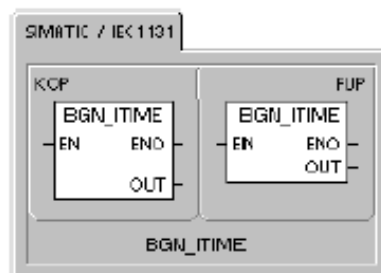
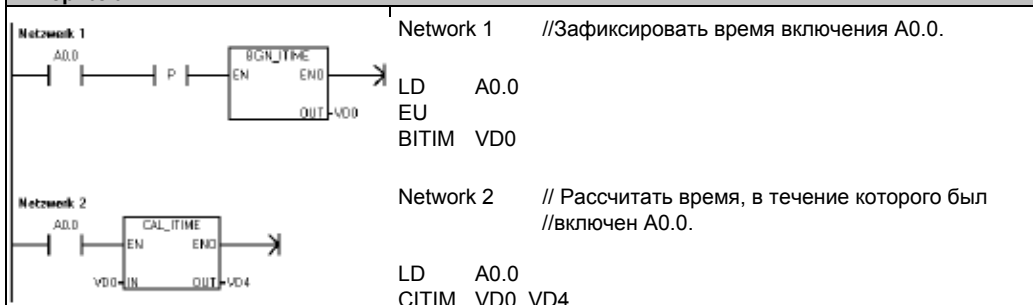


Таблица 6–77. Допустимые операнды для интервальных таймеров

Входы/выходы	Типы данных	Операнды
IN	DWORD	VD, ID, QD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC
OUT	DWORD	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Пример: Команды SIMATIC для начала отсчета времени интервала и расчета времени интервала



Операции с подпрограммами

Команда вызова подпрограммы (CALL) передает управление подпрограмме SBR_N. Команду вызова подпрограммы можно использовать с параметрами или без них. Как только исполнение подпрограммы завершается, управление возвращается команде, следующей за вызовом подпрограммы.

Команда условного возврата из подпрограммы (CRET) завершает подпрограмму в зависимости от результата предшествующей логической операции.

Для добавления подпрограммы выберите команду меню **Edit → Insert → Subroutine [Редактировать → Вставить → Подпрограмма]**.

Сбойные состояния, устанавливающие ENO = 0

- 0008 (превышена максимальная вложенность для подпрограмм)
- 0006 (косвенный адрес)

В главной программе вы можете вкладывать подпрограммы друг в друга (помещать вызов подпрограммы внутри другой подпрограммы) на глубину до восьми уровней. В программе обработки прерывания вложение подпрограмм друг в друга невозможно.

Вызов подпрограммы не может быть помещен ни в какую другую подпрограмму, вызываемую из программы обработки прерывания. Рекурсия (вызов подпрограммы, вызывающей саму себя) не запрещена, но с подпрограммами ее следует использовать с осторожностью.

Таблица 6–78. Допустимые операнды для команды вызова подпрограммы

Входы/выходы	Типы данных	Операнды
SBR_N	WORD	константа для CPU 221, CPU 222, CPU 224: от 0 до 63 для CPU 224XP и CPU 226 от 0 до 127
IN	BOOL BYTE WORD, INT DWORD, DINT STRING	V, I, Q, M, SM, S, T, C, L, поток сигнала VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC ¹ , константа VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC ¹ , константа VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC ¹ , &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, константа *VD, *LD, *AC, константа
IN/OUT	BOOL BYTE WORD, INT DWORD, DINT	V, I, Q, M, SM ² , S, T, C, L VB, IB, QB, MB, SMB ² , SB, LB, AC, *VD, *LD, *AC ¹ VW, T, C, IW, QW, MW, SMW ² , SW, LW, AC, *VD, *LD, *AC ¹ VD, ID, QD, MD, SMD ² , SD, LD, AC, *VD, *LD, *AC ¹
OUT	BOOL BYTE WORD, INT DWORD, DINT	V, I, Q, M, SM ² , S, T, C, L VB, IB, QB, MB, SMB ² , SB, LB, AC, *VD, *LD, *AC ¹ VW, T, C, IW, QW, MW, SMW ² , SW, LW, AC, AQW, *VD, *LD, *AC ¹ VD, ID, QD, MD, SMD ² , SD, LD, AC, *VD, *LD, *AC ¹

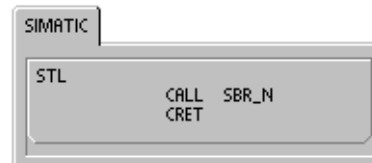
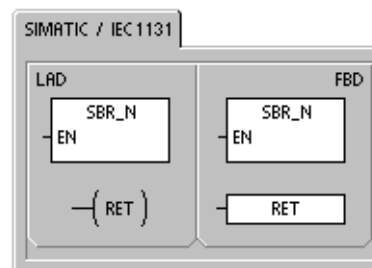
¹ Смещение должно быть не меньше 1
² Смещение должно быть не меньше 30



Совет

STEP 7-Micro/WIN автоматически добавляет команду безусловного возврата из каждой подпрограммы.

Когда вызывается подпрограмма, весь логический стек сохраняется, вершина стека устанавливается в единицу, все остальные ячейки стека устанавливаются в ноль и управление передается вызываемой подпрограмме. Когда эта подпрограмма завершается, стек восстанавливается со значениями, сохраненными в точке вызова, а управление возвращается в вызывающую программу.



Аккумуляторы являются общими для подпрограмм и вызывающей программы. При использовании подпрограммы операции сохранения и восстановления к аккумуляторам не применяются.

Если подпрограмма вызывается в одном и том же цикле несколько раз, то нельзя применять команды «Нарастающий фронт», « Падающий фронт», а также таймеры и счетчики.

Вызов подпрограммы с параметрами

Подпрограмма может содержать передаваемые параметры. Параметры определяются в таблице локальных переменных подпрограммы. Параметру должно быть назначено символическое имя (не более 23 символов), тип переменной и тип данных. В подпрограмму и из нее может быть передано шестнадцать параметров.

Поле типа переменной в таблице локальных переменных определяет, передается ли переменная в подпрограмму (IN), в подпрограмму и из нее (IN_OUT), или она передается из подпрограммы (OUT). Типы параметров для подпрограммы описаны в таблице 6–79. Для добавления параметра поместите курсор на поле того типа параметров (IN, IN_OUT или OUT), который вы хотите добавить. Щелкните правой кнопкой мыши, чтобы вызвать меню для выбора. Выберите пункт Insert [Вставить], а затем пункт Row Below [Строка снизу]. Под текущей записью появится место для записи еще одного параметра выбранного типа.

Таблица 6–79. Типы параметров для подпрограммы

Параметр	Описание
IN	Параметры передаются в подпрограмму. Если параметр является прямым адресом (например, VB10), то в подпрограмму передается значение, находящееся по указанному адресу. Если параметр является косвенным адресом (например, *AC1), то в подпрограмму передается значение, находящееся по адресу, на который сделана ссылка. Если параметр является константой (16#1234) или адресом (&VB100), то в подпрограмму передается значение константы или адреса.
IN_OUT	Значение, находящееся по указанному адресу параметра, передается в подпрограмму, а результирующее значение возвращается по тому же самому адресу. Константы (например, 16#1234) и адреса (например, &VB100) не могут быть параметрами типа IN_OUT.
OUT	Результирующее значение из подпрограммы возвращается по указанному адресу параметра. Константы (например, 16#1234) и адреса (например, &VB100) не могут быть параметрами типа OUT. Так как выходные параметры не сохраняют значения, присвоенного последним исполнением подпрограммы, то вы должны присваивать значения выходам при каждом вызове подпрограммы. Обратите внимание, что команды SET и RESET влияют на значения булевых операндов только в том случае, если поток сигнала включен ON.
TEMP	Локальная память, не используемая для передаваемых параметров, может использоваться для временного хранения данных внутри подпрограммы.

Как показано на рис. 6–38, поле типа данных в таблице локальных переменных определяет размер и формат параметра. Ниже перечислены типы параметров:

- BOOL: Этот тип данных используется для отдельных битовых входов и выходов. IN3 в следующем примере является булевым входом.
- BYTE, WORD, DWORD: Эти типы данных определяют входной или выходной параметр без знака размером 1, 2 или 4 байта соответственно.

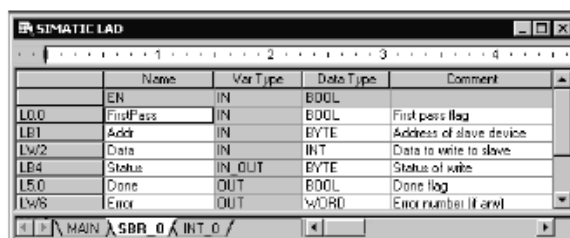


Рис. 6–38. Таблица локальных переменных

- INT, DINT: Эти типы данных определяют входной или выходной параметр без знака размером 1, 2 или 4 байта соответственно.
- REAL: Этот тип данных определяет число с плавающей точкой IEEE однократной точности (4 байта).
- STRING: Этот тип данных используется в качестве четырехбайтового указателя на строку.
- Power Flow [Поток сигнала]: Булев поток сигнала разрешен только для битовых (булевых) входов. Это описание сообщает STEP 7–Micro/WIN 32, что этот входной параметр является результатом достижения подпрограммы потоком сигнала, основанным на комбинации битовых логических операций. Входы с булевым потоком сигнала должны находиться в таблице локальных переменных перед любым другим типом входов. Таким способом можно использовать только входные параметры. Разрешающий вход (EN) и вход IN1 в следующем примере используют булеву логику.

Пример: Вызов подпрограммы	
<p>Ниже приведены два примера на STL. Первый набор команд STL может быть отображен только в редакторе STL, так как булевы параметры, используемые в качестве входов типа "Поток сигнала", не сохраняются в локальной памяти.</p> <p>Второй набор команд STL может быть отображен также и в редакторах LAD и FBD, так как для сохранения состояний булевых входных параметров, которые в LAD и FBD показаны как входы, принимающие поток сигнала, используется локальная память.</p>	
	<p>Только STL:</p> <pre>Network 1 LD IO.0 CALL SBR_0, IO.1, VB10, I1.0, &VB100, *AC1, VD200</pre> <p>Для правильного отображения в LAD и FBD:</p> <pre>Network 1 LD IO.0 = L60.0 LD IO.1 = L63.7 LD L60.0 CALL SBR_0, L63.7, VB10, I1.0, &VB100, *AC1, VD200</pre>

Адресные параметры, например, IN4 (&VB100) передаются в подпрограмму как DWORD (двойное слово без знака). Тип постоянного параметра должен быть указан для параметра в вызывающей программе с помощью описателя константы перед значением константы. Например, чтобы передать в качестве параметра константу, имеющую размер двойного слова без знака, со значением 12 345, постоянный параметр должен быть задан как DW#12345. Если описатель константы для параметра опущен, то константа может быть воспринята как имеющая другой тип.

Автоматическое преобразование типов для входных и выходных параметров не производится. Например, если таблица локальных переменных указывает, что параметр имеет тип данных REAL, а вызывающая программа задает для этого параметра двойное слово (DWORD), то это значение в подпрограмме будет рассматриваться как двойное слово.

Когда значения передаются в подпрограмму, они помещаются в локальную память подпрограммы. Самый левый столбец таблицы локальных переменных показывает адрес в локальной памяти для каждого передаваемого параметра. Значения входных параметров копируются в локальную память подпрограммы, когда подпрограмма вызывается. Значения выходных параметров копируются из локальной памяти подпрограммы в указанные адреса выходных параметров, когда исполнение подпрограммы завершается.

Размер и тип элемента данных представляются в коде параметра. Значения параметров ставятся в соответствие локальной памяти в подпрограмме следующим образом:

- Значения параметров ставятся в соответствие локальной памяти в порядке, задаваемом командой вызова подпрограммы с параметрами, начиная с L.0.
- От одного до восьми последовательных битовых значений параметров ставятся в соответствие отдельному байту, начиная с Lx.0 и вплоть до Lx.7.
- Значения, имеющие тип байт, слово или двойное слово ставятся в соответствие локальной памяти на границах байтов (LBx, LWx или LDx).

В команде вызова подпрограммы с параметрами параметры должны быть расположены в следующем порядке: сначала входные параметры, за ними параметры типа IN_OUT, а затем выходные параметры.

выходные параметры.

Если вы программируете на STL, то формат команды CALL имеет вид:

CALL номер подпрограммы, параметр 1, параметр 2, ... , параметр n

Пример: Команды вызова подпрограммы и возврата из подпрограммы		
M A I N	<p>Network 1</p>	<p>Network 1 //В первом цикле вызвать подпрограмму 0 для //инициализации.</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p> <p>Network 2</p>	<p>Network 1 //Вы можете использовать условный возврат // для выхода из подпрограммы перед //последним сегментом.</p> <pre>LD M14.3 CRET</pre> <p>Network 2 //Этот сегмент будет пропущен, если M14.3 //установлен.</p> <pre>LD SM0.0 MOVB 10, VB0</pre>

Пример: Вызов подпрограммы со строками		
<p>В этом примере различный строковый литерал копируется по определенному адресу в зависимости от входа. Уникальный адрес этой строки сохраняется. Затем адрес строки передается в подпрограмму с помощью косвенного адреса. Типом данных входного параметра подпрограммы является строка. Затем подпрограмма смещает строку на другой адрес.</p> <p>Строковый литерал также может быть передан в подпрограмму. Ссылка на строку в подпрограмме всегда одна и та же.</p>		
M A I N	<p>Network 1</p> <p>Network 2</p> <p>Network 3</p> <p>Network 4</p>	<p>Network 1 //</p> <pre>LD I0.0 SSCPY «string1», VB100 AENO MOVD &VB100, VD0</pre> <p>Network2 //</p> <pre>LD I0.1 SSCPY «string2», VB200 AENO MOVD &VB200, VD0</pre> <p>Network3 //</p> <pre>LD I0.2 CALL SBR_0, *VD0</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 //</p> <pre>LD SM0.0 SSCPY *LD0, VB300</pre>

Обмен данными в сети

7

S7–200 спроектирован так, чтобы удовлетворять ваши потребности в обмене данными и объединении в сети путем поддержки как простых, так и более сложных сетей. S7–200 предоставляет также инструментальные средства, позволяющие обмениваться данными с другими устройствами, например, с принтерами и весами, использующими свои собственные протоколы связи.

STEP 7-Micro/WIN делает создание и настройку вашей сети легкой, как детская игра.

В этой главе

Основы обмена данными в сети для S7–200	212
Выбор протокола связи для вашей сети	216
Инсталляция и деинсталляция интерфейсов для обмена данными	222
Построение сети	223
Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными	227
Использование в сети модемов и STEP 7-Micro/WIN	230
Для опытных пользователей	235
Настройка кабеля RS–232/CPI Multi–Master на дистанционный режим	241

Основы обмена данными в сети для S7–200

Выбор коммуникационного интерфейса для вашей сети

S7–200 поддерживает несколько видов коммуникационных интерфейсов. Выбор сети осуществляется в диалоговом окне Set PG/PC Interface property [Настройка интерфейса PG/PC]. Выбранная сеть обозначается как интерфейс. Имеются следующие виды интерфейсов для доступа к этим сетям для обмена данными:

- Кабели PPI Multi–Master
- Коммуникационные платы CP
- Коммуникационные платы Ethernet

Чтобы выбрать коммуникационный интерфейс для STEP 7-Micro/WIN, действуйте следующим образом. См. рис. 7–1.

1. Дважды щелкните на этом символе в диалоговом окне Communications [Обмен данными].
2. Выберите параметр интерфейса для STEP 7-Micro/WIN.

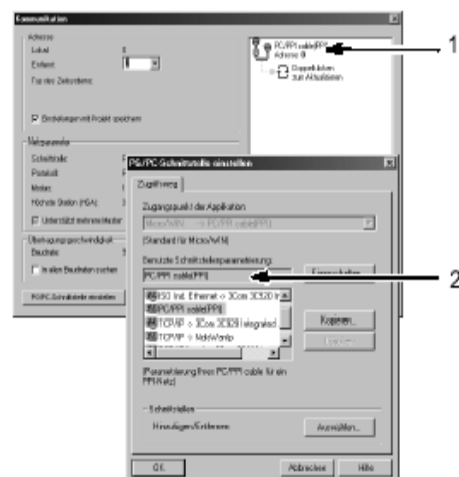


Рис. 7–1. Коммуникационный интерфейс STEP 7-Micro/WIN

Кабели PPI Multi–Master

S7–200 поддерживает обмен данными через два различных вида кабелей PPI Multi–Master. Эти кабели делают возможным обмен данными через интерфейс RS–232 или USB.

Как видно из рис. 7–2, выбор типа кабеля PPI Multi–Master очень прост. Действуйте следующим образом:

1. Щелкните на кнопке Properties [Свойства] в диалоговом окне Set PG/PC Interface property [Настройка свойств интерфейса PG/PC].
2. В диалоговом окне Properties [Свойства] откройте закладку Local Connection [Местное соединение].
3. Выберите порт USB или желаемый COM-порт.

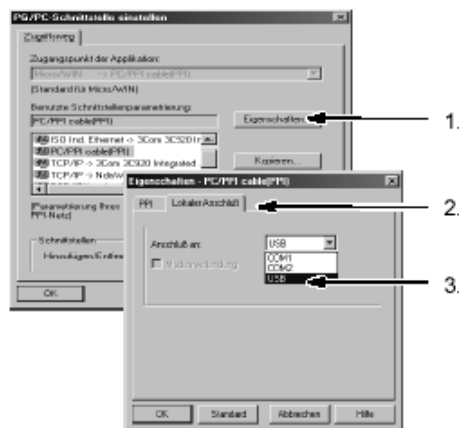


Рис. 7–2. Выбор кабеля PPI Multi–Master



Совет

Обратите, пожалуйста, внимание, что в настоящее время может использоваться только один кабель USB.



Совет

В примерах в этом руководстве используется только кабель RS–232/PPI Multi–Master. Кабель RS–232/PPI Multi–Master заменяет ранее использовавшийся кабель PC/PP1. В продаже имеется также кабель USB/PPI Multi–Master. Номера для заказа вы найдете в Приложении E.

Использование Master- и Slave-устройств в сети PROFIBUS

S7-200 поддерживает сеть с master- и slave-устройствами и может работать в сети PROFIBUS как в качестве ведущего (master) устройства, так и в качестве ведомого (slave) устройства, тогда как STEP 7-Micro/WIN всегда является master-устройством.

Master-устройства

Устройство, являющееся в сети ведущим (master) устройством, может посылать запросы другому устройству в этой сети. Master может также отвечать на запросы других master-устройств в сети. Типичными ведущими устройствами являются STEP 7-Micro/WIN, устройства человеко-машинного (ЧМ) интерфейса, например, TD 200 и ПЛК S7-300 или S7-400. S7-200 функционирует в качестве ведущего устройства, когда он запрашивает данные от других S7-200 (связь между равноправными узлами сети).



Совет

TP070 не работает в сети с другими master-устройствами.

Slave-устройства

Устройство, установленное в сети в качестве ведомого (slave) устройства, может только отвечать на запросы ведущего (master) устройства; slave не может посылать запросы. В большинстве сетей S7-200 функционирует как slave. В качестве slave-устройства S7-200 отвечает на запросы сетевого master-устройства, например, панели оператора или STEP 7-Micro/WIN.

Установка скорости передачи и сетевого адреса

Скорость, с которой данные передаются в сети, обычно измеряется в килободах (кБод) или мегабодах (МБод). Скорость передачи показывает, сколько данных может быть передано в течение определенного промежутка времени. Например, скорость передачи 19,2 кБод означает 19 200 бит в секунду.

Все устройства, которые обмениваются данными через определенную сеть, должны быть настроены на передачу с одинаковой скоростью. Поэтому наивысшая скорость передачи в сети определяется самым медленным устройством, подключенным к сети.

В таблице 7-1 приведены скорости передачи, поддерживаемые S7-200.

Сетевой адрес – это уникальный номер, назначаемый каждому устройству в сети. Уникальный сетевой адрес гарантирует, что данные передаются или извлекаются из нужного устройства. S7-200 поддерживает сетевые адреса от 0 до 126. Для S7-200 с двумя портами каждый порт может иметь сетевой адрес. В таблице 7-2 приведены настройки по умолчанию (заводские) для устройств S7-200.

Таблица 7-1. Скорости передачи, поддерживаемые S7-200

Сеть	Скорость передачи
Стандартная сеть	от 9,6 кБод до 187,5 кБод
При использовании EM 277	от 9,6 кБод до 12 МБод
Режим свободно программируемого обмена данными (Freeport)	от 1200 Бод до 115,2 кБод

Таблица 7-2. Адреса устройств S7-200, установленные по умолчанию

Устройство S7-200	Адрес по умолчанию
STEP 7-Micro/WIN	0
ЧМ-интерфейс (TD 200, TP или OP)	1
CPU S7-200	2

Установка скорости передачи и сетевого адреса для STEP 7-Micro/WIN

Вы должны установить скорость передачи и сетевой адрес для STEP 7-Micro/WIN. Скорость передачи должна быть такой же, как у других устройств в сети, а сетевой адрес должен быть уникальным.

Обычно сетевой адрес (0) для STEP 7-Micro/WIN не изменяется. Если ваша сеть содержит другой пакет программирования, то вам, возможно, потребуется изменить сетевой адрес для STEP 7-Micro/WIN.

Как видно из рис. 7–3, установка скорости передачи и сетевого адреса для STEP 7-Micro/WIN проста. После щелчка на пиктограмме Communications [Обмен данными], находящейся на навигационной панели, действуйте следующим образом:

1. Дважды щелкните на этом символе в диалоговом окне Communications [Обмен данными].
2. Щелкните на кнопке Properties [Свойства] в диалоговом окне Set PG/PC Interface [Установка интерфейса PG/PC].
3. Выберите сетевой адрес для STEP 7-Micro/WIN.
4. Выберите скорость передачи для STEP 7-Micro/WIN.

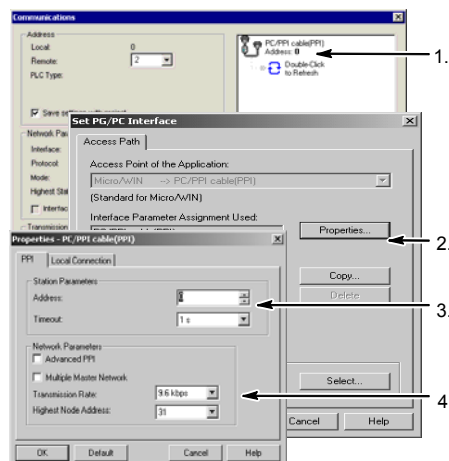


Рис. 7–3. Настройка STEP 7-Micro/WIN

Установка скорости передачи и сетевого адреса для S7–200

Вы должны также установить скорость передачи и сетевой адрес для S7–200. Скорость передачи и сетевой адрес S7–200 хранятся в системном блоке данных. Выбрав параметры для S7–200, вы должны загрузить системный блок данных в S7–200.

По умолчанию скорость передачи для каждого порта S7–200 составляет 9,6 КБод, а сетевой адрес равен 2.

Как видно из рис. 7–4, для установки скорости передачи и сетевого адреса для S7–200 используется STEP 7-Micro/WIN. После выбора пиктограммы System Block [Системный блок], находящейся на навигационной панели, или выбора команды меню **View > Component > System Block [Вид > Компонент > Системный блок]** действуйте следующим образом:

1. Выберите сетевой адрес для S7–200.
2. Выберите скорость передачи для S7–200.
3. Загрузите системный блок в S7–200.

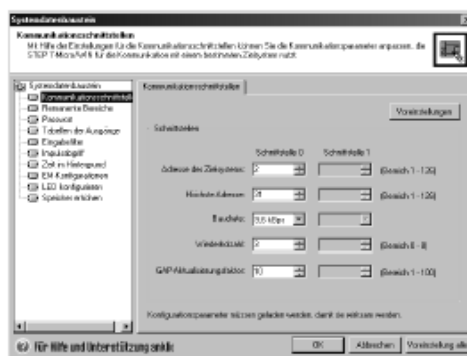


Рис. 7–4. Настройка CPU S7–200



Совет

Допускается выбор любого варианта скорости передачи. STEP 7-Micro/WIN проверяет этот выбор во время загрузки системного блока. Варианты скорости передачи, при которых STEP 7-Micro/WIN не может обмениваться данными с S7–200, не загружаются.

Установка удаленного адреса

Перед тем как вы сможете загрузить измененные настройки в S7-200, вы должны установить коммуникационный порт (COM) STEP 7-Micro/WIN (локальный) и удаленный адрес S7-200 таким образом, чтобы они соответствовали текущим настройкам удаленного S7-200. См. рис. 7-5.

После того как вы загрузили измененные настройки, вам, возможно, придется снова сконфигурировать скорость передачи для интерфейса PG/PC (если имеются отличия от настроек, использованных для удаленного S7-200). Как настраивается скорость передачи, видно на рис. 7-3.

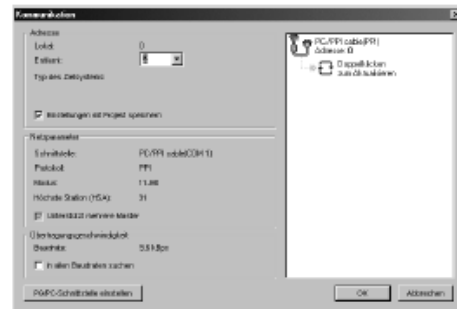


Рис. 7-5. Настройка STEP 7-Micro/WIN

Поиск CPU S7-200 в сети

Вы можете искать и идентифицировать CPU S7-200, присоединенные к вашей сети. Вы можете также вести поиск в сети при определенной скорости передачи или на всех скоростях, когда вы хотите идентифицировать CPU S7-200.

Поиск при всех скоростях передачи возможен только при использовании кабелей PPI Multi-Master. Эта функция недоступна, если обмен данными производится через плату CP. Поиск начинается при скорости, установленной в данный момент.

1. Откройте диалоговое окно Communications [Обмен данными] и дважды щелкните на пиктограмме.
2. Для поиска на всех скоростях активизируйте триггерную кнопку Search All baud rates [Искать на всех скоростях передачи].

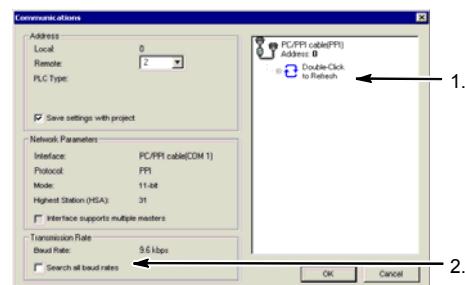


Рис. 7-6. Поиск CPU в сети

Выбор протокола связи для вашей сети

Ниже приведены протоколы, поддерживаемые CPU S7–200.

- двухточечный интерфейс (PPI)
- многоточечный интерфейс (MPI)
- PROFIBUS

Эти протоколы основаны на семиуровневой модели взаимодействия открытых систем (Open System Interconnection, OSI) архитектуры связи. Они реализованы в кольцевой сети с маркерным доступом, удовлетворяющей стандарту PROFIBUS, определенному в Европейском стандарте EN 50170. Эти протоколы являются асинхронными, основанными на символах протоколами, с одним стартовым битом, восемью битами данных, контролем четности и одним стоповым битом. Передаваемые блоки данных включают в себя специальные стартовый и стоповый символы, адреса передающей и принимающей станций, длину кадра и контрольную сумму для проверки целостности данных. Эти протоколы могут быть активными в сети одновременно, не оказывая влияния друг на друга, пока скорость передачи одинакова для всех протоколов.

Для CPU S7–200 с модулями расширения CP243-1 и CP243-1 IT доступен также Ethernet.

Протокол PPI

PPI является протоколом типа master–slave [ведущий-подчиненный]: master-устройства посылают запросы slave-устройствам, а slave-устройства реагируют. См. рис. 7–7. Slave-устройства не инициируют сообщений, а ждут, пока ведущее устройство (master) не пошлет им запрос или не будет опрашивать их реакцию.

Ведущие устройства обмениваются данными с подчиненными устройствами через совместно используемое соединение, управляемое протоколом PPI. PPI не ограничивает количества master-устройств, которые могут обмениваться данными со slave-устройствами; однако в сети можно установить не более 32 master-устройств.

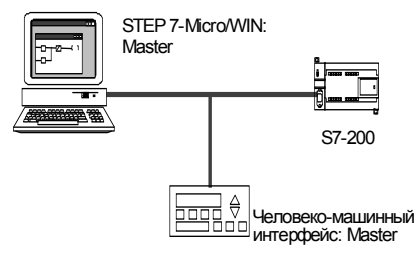


Рис. 7–7. Сеть PPI

CPU S7–200 могут действовать как master-устройства, пока они находятся в режиме RUN, если вы активизировали в своей пользовательской программе привилегированный режим PPI. (См. описание SMB30 в Приложении D.) После активизации привилегированного режима PPI вы можете использовать команды чтения из сети и записи через сеть для чтения сообщений от других S7–200 или записи их в другие S7–200. Когда S7–200 используется как master-устройство PPI, он все же реагирует как slave на запросы других master-устройств.

Если вы выберете расширенный протокол PPI (PPI Advanced), то сетевые устройства могут формировать между собой логические соединения. При использовании PPI Advanced каждое устройство предоставляет в распоряжение ограниченное количество соединений. В таблице 7–3 приведено количество соединений, поддерживаемых S7–200.

Все CPU S7–200 поддерживают как протокол PPI, так и расширенный протокол PPI Advanced, тогда как модуль EM 277 поддерживает только протокол PPI Advanced.

Таблица 7–3. Количество соединений для модулей CPU S7–200 и EM 277

Модуль	Скорость передачи	Соединения
CPU S7–200	Порт 0	9,6 кБод, 19,2 кБод или 187,5 кБод
	Порт 1	9,6 кБод, 19,2 кБод или 187,5 кБод
Модуль EM 277	от 9,6 кБод до 12 МБод	6 на модуль

Протокол MPI

MPI допускает обмен данными типа master–master и master–slave. См. рис. 7–8. Для обмена данными с CPU S7–200 STEP 7-Micro/WIN устанавливает соединение типа master–slave. Протокол MPI не обменивается данными с CPU S7–200, работающим в качестве master-устройства.

Сетевые устройства обмениваются данными через отдельные соединения (управляемые протоколом MPI) между любыми двумя устройствами. Обмен данными между устройствами ограничен количеством соединений, поддерживаемых модулями CPU S7–200 или EM 277. Количество соединений, поддерживаемых S7–200, приведено в таблице 7–3.

В случае протокола MPI ПЛК S7–300 и S7–400 используют для чтения и записи данных в CPU S7–200 команды XGET и XPUT. Информацию об этих командах вы найдете в руководстве по программированию для S7–300 или S7–400.

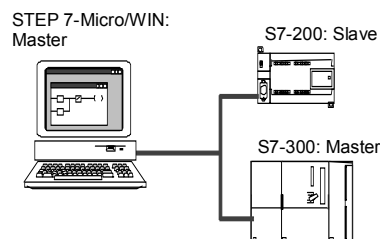


Рис. 7–8. Сеть MPI

Протокол PROFIBUS

Протокол PROFIBUS спроектирован для скоростного обмена данными с устройствами децентрализованной периферии (удаленного ввода/вывода). Имеется много устройств PROFIBUS различных изготовителей. Спектр этих устройств простирается от простых модулей ввода или вывода до устройств управления двигателями и программируемых логических контроллеров.

Сети PROFIBUS обычно имеют одно master-устройство и несколько slave-устройств ввода/вывода. См. рис. 7–9. Master-устройство конфигурируется таким образом, что ему известно, какие виды slave-устройств ввода/вывода и под какими адресами присоединены к сети. Master-устройство инициализирует сеть и проверяет соответствие slave-устройств, находящихся в сети, ее конфигурации. Master непрерывно записывает выходные данные в slave-устройства и считывает из них входные данные.

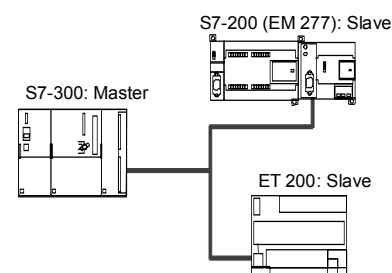


Рис. 7–9. Сеть PROFIBUS

Если master-устройство DP успешно сконфигурировало slave, то оно становится владельцем этого slave-устройства. Если в сети имеется второе master-устройство, то оно имеет ограниченный доступ к slave-устройствам, принадлежащим первому master-устройству.

Протокол TCP/IP

S7–200 может поддерживать обмен данными через протокол TCP/IP Ethernet путем использования одного из модулей расширения Ethernet (CP 243-1) или Internet (CP 243-1 IT). В таблице 7–4 приведена скорость передачи и количество соединений, поддерживаемых этими модулями.

Таблица 7–4. Количество соединений для модулей Ethernet (CP 243-1) и Internet (CP 243-1 IT)

Модуль	Скорость передачи	Соединения
Модуль Ethernet (CP 243-1)	от 10 до 100 МБод	8 соединений общего назначения
Модуль Internet (CP 243-1 IT)		1 соединение STEP 7-Micro/WIN

Дополнительную информацию вы найдете в руководствах *SIMATIC NET CP 243-1 Communications Processor for Industrial Ethernet* [Коммуникационный процессор SIMATIC NET CP 243-1 для Industrial Ethernet] и *SIMATIC NET CP 243-1 IT Communications Processor for Industrial Ethernet and Information Technology* [Коммуникационный процессор SIMATIC NET CP 243-1 IT для Industrial Ethernet и информационных технологий].

Примеры сетевых конфигураций, использующих только устройства S7-200

Сети PPI с одним master-устройством

Для построения простой сети с одним master-устройством устройство программирования и CPU S7-200 соединяются через кабель PPI Multi-Master или через плату коммуникационного процессора (CP), установленную в устройстве программирования.

В примере сети, представленном в верхней части рис. 7-10, master-устройством сети является устройство программирования (STEP 7-Micro/WIN).

В примере сети, представленном в нижней части рис. 7-10, master-устройством сети является устройство ЧМ-интерфейса (например, TD 200, TP или OP).

В обоих примерах CPU S7-200 является slave-устройством, которое отвечает на запросы master-устройства.

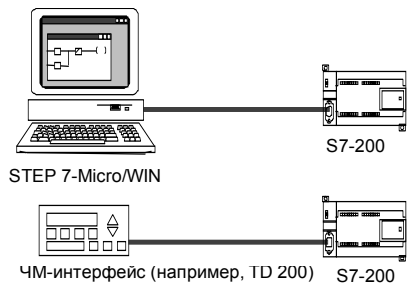


Рис. 7-10. Сеть PPI с одним master-устройством

В сети PPI с одним master-устройством вы настраиваете STEP 7-Micro/WIN на использование протокола PPI. Деактивируйте триггерные кнопки Multiple Master Network [Сеть с несколькими master-устройствами] и PPI Advanced [Расширенный протокол PPI].

Сеть PPI с несколькими master-устройствами

На рис. 7-11 показан пример сети с несколькими master-устройствами и одним slave-устройством. Устройство программирования (STEP 7-Micro/WIN) использует плату CP или кабель PPI Multi-Master. STEP 7-Micro/WIN и устройство ЧМ-интерфейса используют сеть совместно.

STEP 7-Micro/WIN и устройство ЧМ-интерфейса являются master-устройствами и должны иметь отдельные сетевые адреса. Если используется кабель PPI Multi-Master, то этот кабель является master-устройством и использует сетевой адрес, предоставленный STEP 7-Micro/WIN. CPU S7-200 является slave-устройством.

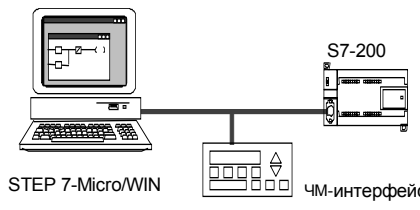


Рис. 7-11. Несколько master-устройств с одним slave-устройством

На рис. 7-12 показана сеть PPI с несколькими master-устройствами, обменивающимися данными с несколькими slave-устройствами. В этом примере STEP 7-Micro/WIN и ЧМ-интерфейс могут запрашивать данные от любого slave-устройства CPU S7-200. STEP 7-Micro/WIN и устройство ЧМ-интерфейса используют сеть совместно.

Все устройства (ведущие и ведомые) имеют разные сетевые адреса. Если используется кабель PPI Multi-Master, то этот кабель является master-устройством и использует сетевой адрес, предоставленный STEP 7-Micro/WIN. CPU S7-200 являются slave-устройствами.

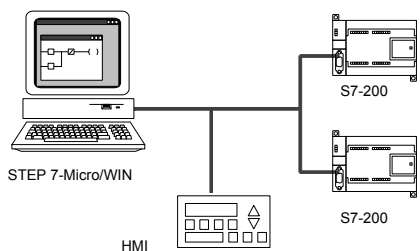


Рис. 7-12. Несколько master- и slave-устройств

Для сети PPI с несколькими master-устройствами и одним или несколькими slave-устройствами настройте STEP 7-Micro/WIN на использование протокола PPI и активизируйте триггерные кнопки Multiple Master Network [Сеть с несколькими master-устройствами] и PPI Advanced [Расширенный протокол PPI], если они доступны. Если вы используете кабель PPI Multi-Master, то триггерные кнопки Multiple Master Network и PPI Advanced игнорируются.

Сложные сети PPI

На рис. 7–13 показан пример сети, использующей несколько master-устройств с обменом данными между равноправными STEP 7-Micro/WIN и устройством ЧМ-интерфейса имеют через сеть доступ на чтение и запись к модулям CPU S7–200, а CPU S7–200 используют команды чтения и записи через сеть для чтения и записи между собой (обмен данными между равноправными узлами).

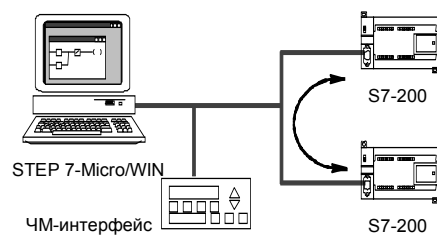


Рис. 7–13. Обмен данными между равноправными узлами

На рис. 7–14 показан еще один пример сложной сети PPI, использующей несколько master-устройств с обменом данными между равноправными узлами. В этом примере каждый ЧМ-интерфейс контролирует один CPU S7–200.

CPU S7–200 используют команды NETR и NETW для чтения и записи между собой (обмен данными между равноправными узлами).

Для сложных сетей PPI настройте STEP 7-Micro/WIN на использование протокола PPI и активизируйте триггерные кнопки Multiple Master Network [Сеть с несколькими master-устройствами] и PPI Advanced [Расширенный протокол PPI], если они доступны. Если вы используете кабель PPI Multi-Master, то триггерные кнопки Multiple Master Network и PPI Advanced игнорируются.

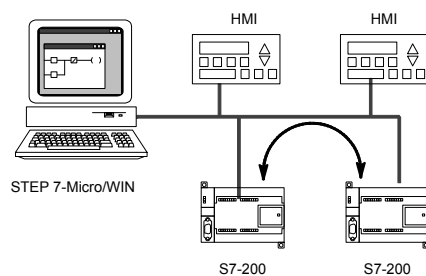


Рис. 7–14. Устройства ЧМ-интерфейса и обмен данными между равноправными узлами

Примеры сетевых конфигураций с устройствами S7–200, S7–300 и S7–400

Сети со скоростями передачи до 187,5 кБод

В примере сети, показанном на рис. 7–15, S7–300 использует для обмена данными с CPU S7–200 команды XPUT и XGET. S7–300 не может обмениваться данными с CPU S7–200 в привилегированном режиме.

Для обмена данными с CPU S7 настройте STEP 7-Micro/WIN на использование протокола PPI и активизируйте триггерные кнопки Multiple Master Network [Сеть с несколькими master-устройствами] и PPI Advanced [Расширенный протокол PPI], если они доступны. Если вы используете кабель PPI Multi-Master, то триггерные кнопки Multiple Master Network и PPI Advanced игнорируются.

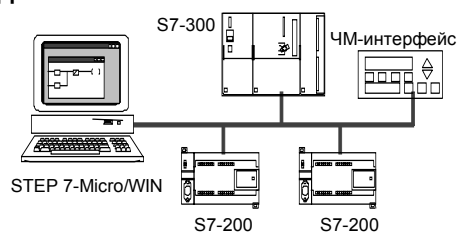


Рис. 7–15. Скорости передачи до 187,5 кБод

Сети со скоростями передачи свыше 187,5 кБод

При скоростях передачи свыше 187,5 кБод CPU S7-200 должен использовать для присоединения к сети модуль EM 277. См. рис. 7-16. STEP 7-Micro/WIN должен быть присоединен через плату коммуникационного процессора (CP).

В этой конфигурации S7-300 может обмениваться данными с модулями S7-200 с помощью команд XPUT и XGET, а ЧМ-интерфейс может контролировать модули S7-200 или S7-300.

EM 277 всегда является slave-устройством.

STEP 7-Micro/WIN может программировать или контролировать любой CPU S7-200 через присоединенный EM 277. Для обмена данными с EM 277 со скоростью более 187,5 кБод настройте STEP 7-Micro/WIN на использование протокола MPI с платой CP. Максимальная скорость передачи для кабелей PPI Multi-Master составляет 187,5 кБод.

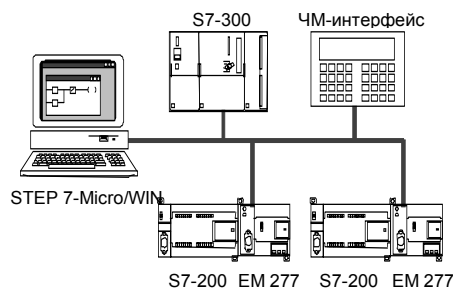


Рис. 7-16. Скорости передачи свыше 187,5 кБод

Пример конфигурации для сети PROFIBUS-DP

Сети с S7-315-2 DP в качестве master-устройства PROFIBUS и EM 277 в качестве slave-устройства PROFIBUS

На рис. 7-17 показан пример сети PROFIBUS, использующей S7-315-2 DP в качестве master-устройства PROFIBUS. Модуль EM 277 является slave-устройством PROFIBUS.

S7-315-2 DP может читать и записывать в EM 277 данные объемом от 1 до 128 байт. S7-315-2 DP считывает или записывает в S7-200 ячейки памяти переменных.

Эта сеть поддерживает скорости передачи от 9600 Бод до 12 МБод.

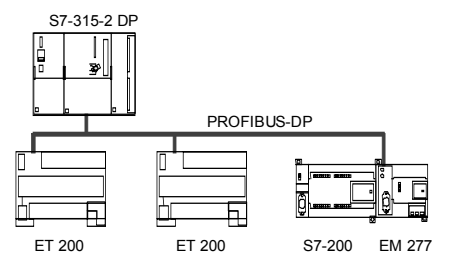


Рис. 7-17. Сеть с S7-315-2 DP

Сети со STEP 7-Micro/WIN и ЧМ-интерфейсом

На рис. 7-18 показан пример сети с S7-315-2 DP в качестве master-устройства PROFIBUS и EM 277 в качестве slave-устройства PROFIBUS. В этой конфигурации ЧМ-интерфейс контролирует S7-200 через EM 277. STEP 7-Micro/WIN программирует S7-200 через EM 277.

Эта сеть поддерживает скорости передачи от 9600 Бод до 12 МБод. STEP 7-Micro/WIN требует платы CP для скоростей передачи свыше 187,5 кБод.

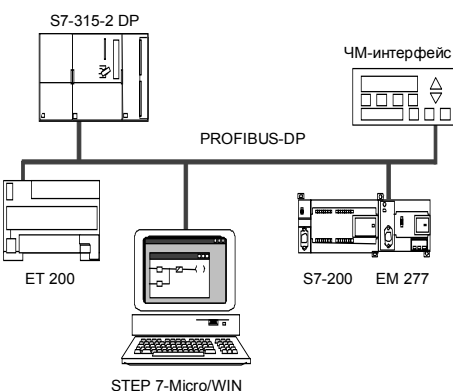


Рис. 7-18. Сеть PROFIBUS

Настройте STEP 7-Micro/WIN для использования протокола PROFIBUS и платы CP. Если в сети имеются только устройства DP, то выберите профиль DP или стандартный профиль. Если в сети имеются не только устройства DP, но и другие устройства, например, TD 200, то выберите универсальный профиль (DP/FMS) для всех master-устройств. Чтобы сеть была работоспособной, все master-устройства в сети должны быть настроены на использование одного и того же профиля PROFIBUS (DP, стандартный или универсальный).

Кабели PPI Multi-master функционируют в сетях с максимальной скоростью 187,5 кБод только в том случае, если все master-устройства используют универсальный профиль (DP/FMS).

Примеры сетевых конфигураций, использующих устройства Ethernet и/или Internet

В конфигурации, показанной на рис. 7–19, используется соединение с Ethernet, чтобы дать возможность STEP 7-Micro/WIN обмениваться данными с любым из CPU S7–200, которые используют модуль Ethernet (CP 243-1) или модуль Internet (CP 243-1 IT). CPU S7–200 могут обмениваться данными через соединение с Ethernet. Для доступа к базовой странице модуля Internet (CP 243-1 IT) может быть использован стандартный браузер, работающий на ПК со STEP 7-Micro/WIN.

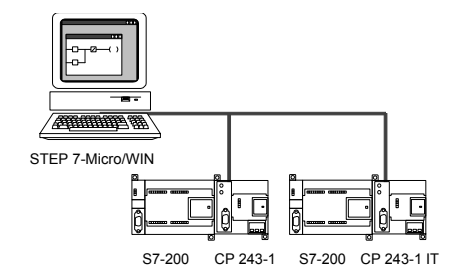


Рис. 7–19. Сеть Ethernet 10/100 МБод

Для сетей Ethernet настройте STEP 7-Micro/WIN на использование протокола TCP/IP.



Совет

В диалоговом окне Set PG/PC Interface [Настройка интерфейса PG/PC] имеются две возможности для выбора TCP/IP. S7–200 не поддерживает вариант TCP/IP -> NdisWanlp.

- В диалоговом окне Set PG/PC Interface [Настройка интерфейса PG/PC] вариант(ы) выбора зависят от типа интерфейса Ethernet, предоставляемого вашим ПК. Выберите тот из них, который соединяет ваш компьютер с сетью Ethernet, к которой присоединен модуль CP 243-1 или CP 243-1 IT.
- В диалоговом окне Communications [Обмен данными] вы должны ввести удаленный IP-адрес(а) каждого из модулей Ethernet/Internet, с которыми вы хотите обмениваться данными.

Инсталляция и деинсталляция интерфейсов для обмена данными

Для инсталляции или деинсталляции коммуникационных интерфейсов для своего компьютера вызовите из диалогового окна Set PG/PC Interface [Установка интерфейса PG/PC] диалоговое окно Installing/Uninstalling Interfaces [Инсталляция/деинсталляция интерфейсов].

1. В диалоговом окне Set PG/PC Interface [Установка интерфейса PG/PC] щелкните на кнопке Select [Выбрать], чтобы получить доступ к диалоговому окну Installing/Uninstalling Interfaces [Инсталляция/деинсталляция интерфейсов].
Доступные интерфейсы перечислены в поле для выбора, а поле Installed [Установлено] отображает интерфейсы, которые уже были установлены на вашем компьютере.
2. *Чтобы добавить интерфейс для обмена данными:* Выберите коммуникационную аппаратуру, установленную на вашем компьютере, и щелкните на Install [Установить]. При закрытии диалогового окна Installing/Uninstalling Interfaces [Инсталляция/деинсталляция интерфейсов] диалоговое окно Set PG/PC Interface [Установка интерфейса PG/PC] отображает этот интерфейс в поле Interface Parameter Assignment Used [Используемая параметризация интерфейса].
3. *Чтобы деинсталлировать интерфейс для обмена данными:* Выберите интерфейс, который вы хотите деинсталлировать, и щелкните на Uninstall [Деинсталлировать]. При закрытии диалогового окна Installing/Uninstalling Interfaces [Инсталляция/деинсталляция интерфейсов] диалоговое окно Set PG/PC Interface [Установка интерфейса PG/PC] удаляет этот интерфейс из поля Interface Parameter Assignment Used [Используемая параметризация интерфейса].

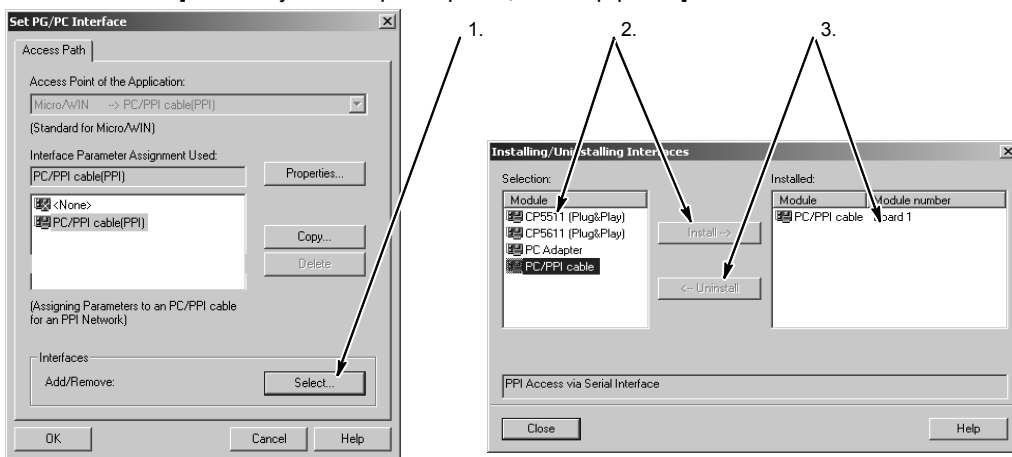


Рис. 7–20. Диалоговые окна Set PG/PC Interface и Installing/Uninstalling Interfaces

Изменение настроек порта вашего компьютера для работы в режиме PPI с несколькими ведущими узлами (Multi-Master)

Если вы используете кабель USB/PPI Multi-Master или кабель RS-232/PPI Multi-Master в режиме PPI, то нет необходимости изменять настройки порта вашего компьютера, и работа в сетях с несколькими ведущими узлами (multi-master) возможна с использованием операционной системы Windows NT.

Если вы используете кабель RS-232/PPI Multi-Master в режиме PPI/свободно программируемый обмен данными (Freeport) для обмена данными между CPU S7-200 и STEP 7-Micro/WIN в операционной системе, которая поддерживает конфигурацию PPI Multi-Master (Windows NT не поддерживает PPI Multi-Master), то вам, возможно, придется изменить настройки порта на вашем компьютере:

1. Щелкните правой кнопкой мыши на пиктограмме My Computer [Мой компьютер] на рабочем столе и выберите команду меню Properties [Свойства].
2. Выберите закладку Device Manager [Администратор устройств]. У Windows 2000 сначала выберите закладку Hardware [Аппаратура], а затем кнопку Device Manager [Администратор устройств].
3. Дважды щелкните на Ports [Порты] (COM & LPT).
4. Выберите коммуникационный порт, который вы в настоящее время используете (например, COM1).
5. В закладке Port Settings [Настройки порта] щелкните на кнопке Advanced [Расширенные].
6. Установите для приемного буфера (Receive Buffer) и для передающего буфера (Transmit Buffer) наименьшее значение (1).
7. Щелкните на ОК для приема изменения, закройте все окна и перезагрузите компьютер, чтобы сделать новые настройки активными.

Построение сети

Общие указания

Всегда снабжайте линии, которые могут быть подвергнуты ударам молнии, надлежащей защитой от перенапряжений.

Избегайте прокладывать линии для низковольтных сигналов и кабели связи в тех же кабельных трассах, что и линии переменного тока и быстро переключаемые линии постоянного тока. Провода должны прокладываться парами: нейтральный или нулевой провод в паре с фазовым или сигнальным проводом.

Коммуникационный порт CPU S7–200 не имеет потенциальной развязки. В случае необходимости для потенциальной развязки своей сети используйте повторитель RS–485 или модуль EM 277.

Осторожно

Соединение друг с другом устройств, имеющих разные опорные потенциалы, может вызвать протекание нежелательных токов по соединительному кабелю.

Эти нежелательные токи могут вызвать ошибки при обмене данными или повредить оборудование.

Обеспечьте, чтобы все устройства, которые вы собираетесь соединить друг с другом кабелем связи, имели в цепи тока одинаковый опорный потенциал или были гальванически развязаны, чтобы предотвратить протекание нежелательных токов. Дополнительные указания о заземлении и опорных точках в гальванически развязанных цепях тока вы найдете в главе 3.

Определение расстояний, скоростей передачи и кабелей для вашей сети

Как показано в таблице 7–5, максимальная длина сегмента сети определяется двумя факторами: потенциальной развязкой (с помощью повторителя RS–485) и скоростью передачи.

Потенциальная развязка требуется, если вы соединяете устройства с различными потенциалами земли. Различные потенциалы земли могут возникнуть, если точки заземления физически разделены большим расстоянием. Но и при коротких расстояниях токи нагрузки тяжелых машин могут вызвать различие в потенциалах земли.

Таблица 7–5. Максимальная длина сетевого кабеля

Скорость передачи	Порт CPU без потенциальной развязки ¹	Порт CPU с повторителем или EM 277
от 9,6 до 187,5 кБод	50 м	1 000 м
500 кБод	Не поддерживается	400 м
от 1 до 1,5 МБод	Не поддерживается	200 м
от 3 до 12 МБод	Не поддерживается	100 м

¹ Максимально допустимое расстояние без использования потенциальной развязки или повторителя составляет 50 м. Это расстояние измеряется от первого до последнего узла в сегменте.

Использование в сети повторителей

Повторитель RS-485 обеспечивает напряжение смещения и оконечную нагрузку сегмента сети. Повторитель можно использовать для следующих целей:

- *Для увеличения длины сети:* Добавление повторителя к вашей сети дает возможность удлинить сеть еще на 50 м. Если соединить два повторителя при отсутствии между ними других узлов (как показано на рис. 7-21), то сеть можно расширить до максимальной длины кабеля для данной скорости передачи. В сети можно использовать последовательно до 9 повторителей, но общая длина сети не должна превышать 9600 м.
- *Для добавления устройств к сети:* Каждый сегмент может содержать не более 32 устройств, соединенных друг с другом на расстоянии до 50 м при скорости передачи 9600 Бод. Использование повторителя позволяет добавить к сети еще один сегмент (32 устройства).
- *Для гальванической развязки различных сегментов сети:* гальваническая развязка в сети улучшает качество передачи, разделяя сегменты сети, обладающие разными потенциалами земли.

Повторитель в сети считается одним из узлов сегмента, хотя ему и не присвоен сетевой адрес.



Рис. 7-21. Пример сети с повторителями

Выбор сетевого кабеля

В сетях S7-200 используются кабели в виде витой пары, удовлетворяющие стандарту RS-485. В таблице 7-6 приведены технические данные сетевого кабеля. К сегменту сети можно присоединить до 32 устройств.

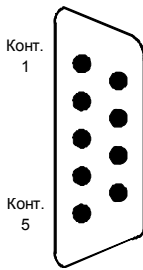
Таблица 7-6. Общие технические данные сетевого кабеля

Технические данные	Описание
Тип кабеля	Экранированная витая пара
Сопротивление контура	$\leq 115 \text{ Ом/км}$
Эффективная емкость	30 пФ/м
Номинальный импеданс	Примерно от 135 до 160 Ом (частота = от 3 до 20 МГц)
Ослабление	0,9 дБ/100 м (частота = 200 кГц)
Поперечное сечение жил	от 0,3 до 0,5 мм ²
Диаметр кабеля	8 мм $\pm 0,5$ мм

Назначение контактов

Коммуникационные порты CPU S7-200 работают с уровнями сигнала RS-485 и снабжены 9-контактным миниатюрным D-образным разъемом в соответствии со стандартом PROFIBUS, определенным в Европейском стандарте EN 50170. В таблице 7-7 показан разъем, обеспечивающий физическое соединение с коммуникационным портом, и описано назначение его контактов.

Таблица 7-7. Назначение контактов коммуникационного порта S7-200

Разъем	Номер контакта	Сигнал PROFIBUS	Порт 0/Порт 1
	1	Экран	Заземляющий контакт
	2	Обратный провод 24 В	Общий провод логики
	3	RS-485, сигнал В	RS-485, сигнал В
	4	Запрос на передачу	RTS (TTL)
	5	Обратный провод 5 В	Общий провод логики
	6	+5 В	+5 В, последовательно включенный резистор 100 Ом
	7	+24 В	+24 В
	8	RS-485, сигнал А	RS-485, сигнал А
	9	Не используется	10-битовый протокол (вход)
Корпус разъема	Экран	Заземляющий контакт	

Присоединение сетевого кабеля

Фирма Siemens предлагает два вида сетевых разъемов, с помощью которых можно быстро и легко подключить к сети несколько устройств: стандартный штекер для подключения к шине (назначение контактов см. в таблице 7-7) и штекер с дополнительным портом для устройства программирования, который позволяет подключать к сети устройство программирования или ЧМ-интерфейса, не нарушая существующих сетевых соединений. Штекер с портом для устройства программирования передает в этот порт все сигналы (включая питающее напряжение) от S7-200, что особенно полезно для подключения устройств, получающих питание от S7-200 (например, TD 200).

Оба штекера имеют два набора соединительных винтов, дающие возможность закрепить входящий и выходящий сетевой кабель. На обоих штекерах также имеются переключатели, с помощью которых вы можете подключить оконечное сопротивление (терминатор). На рис. 7-22 показано типовое присоединение для кабельных разъемов.

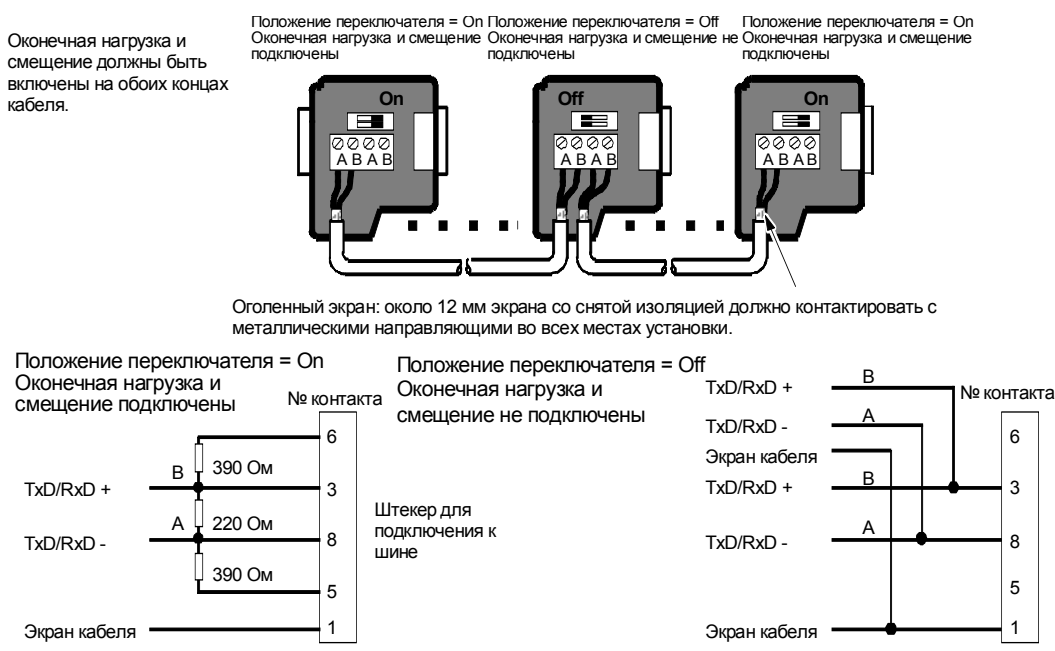


Рис. 7-22. Присоединение сетевого кабеля

Выбор кабеля PPI Multi-Master или платы CP для вашей сети

Как показано в таблице 7–8, STEP 7-Micro/WIN поддерживает кабель RS–232/PPI Multi-Master и кабель USB/PPI Multi-Master, а также несколько плат CP, которые дают возможность устройству программирования (вашему компьютеру или устройству программирования SIMATIC) работать в сети в качестве master-устройства.

Для скоростей передачи до 187,5 кБод кабели PPI Multi-Master обеспечивают самое простое и наиболее экономичное соединение между STEP 7-Micro/WIN и одним CPU S7–200 или сетью S7–200. Имеются два типа кабелей PPI Multi-Master, и оба могут быть использованы для локального соединения между STEP 7-Micro/WIN и сетью S7–200.

Кабель USB/PPI Multi-Master является устройством типа plug and play [включай и работай], которое может быть использовано с ПК, поддерживающими USB версии 1.1. Он обеспечивает гальваническую развязку между вашим ПК и сетью S7–200 и поддерживает обмен данными через PPI при скоростях передачи до 187,5 кБод. При этом не требуется устанавливать никакие переключатели; просто подключите кабель, выберите в качестве интерфейса кабель PC/PPI, выберите протокол PPI и выберите для интерфейса опцию USB в закладке PC Connection [Соединение с ПК]. Для использования STEP 7-Micro/WIN к ПК можно одновременно подсоединить только один кабель USB/PPI Multi-Master

Кабель RS–232/PPI Multi-Master имеет восемь DIP-переключателей: два из этих переключателей используются для настройки кабеля для работы со STEP 7-Micro/WIN.

- Если вы подключаете кабель к ПК, выберите режим PPI (переключатель 5 = 1) и локальный режим (переключатель 6 = 0).
- Если вы подключаете кабель к модему, выберите режим PPI (переключатель 5 = 1) и дистанционный режим (переключатель 6 = 1).

Этот кабель обеспечивает гальваническую развязку между вашим ПК и сетью S7–200. Выберите в качестве интерфейса кабель PC/PPI и выберите порт RS–232, который вы хотите использовать в закладке PC Connection [Соединение с ПК]. В закладке PPI выберите адрес станции и скорость передачи в сети. Вам нет необходимости выполнять еще какие-либо настройки, так как выбор протокола осуществляется кабелем RS–232/PPI Multi-Master автоматически.

Кабели USB/PPI и RS–232/PPI Multi-Master снабжены светодиодами, которые сигнализируют об активности обмена данными с ПК, а также об активности обмена данными в сети.

- Светодиод Tx показывает, что кабель передает информацию в ПК.
- Светодиод Rx показывает, что кабель принимает информацию из ПК.
- Светодиод PPI показывает, что кабель передает данные в сети. Так как кабели Multi-Master являются держателями маркера, то светодиод PPI включен постоянно, после того как STEP 7-Micro/WIN инициализировал обмен данными. Светодиод PPI выключается, когда закрывается связь со STEP 7-Micro/WIN. Светодиод PPI будет также мигать с частотой 1 Гц, когда ожидается включение в сеть.

Платы CP содержат специализированные аппаратные средства, которые помогают устройству программирования управлять сетью с несколькими master-устройствами и могут поддерживать различные протоколы при нескольких скоростях передачи.

Все платы CP снабжены одним портом RS–485 для подключения к сети. Плата CP 5511 PCMCIA имеет адаптер с 9-контактным D-образным портом. Один конец кабеля подключается к порту RS–485 платы CP, а другой конец – к штекеру с портом для устройства программирования в вашей сети.

Если вы используете плату CP с обменом данными через PPI, то примите во внимание, что STEP 7-Micro/WIN не поддерживает одновременную работу двух разных приложений на одной и той же плате CP. Перед тем как соединить STEP 7-Micro/WIN с сетью через плату CP, вы должны закрыть другое приложение. Если вы используете обмен данными через MPI или PROFIBUS, то допускается одновременный обмен данными через сеть нескольких приложений STEP 7-Micro/WIN.

Осторожно

Использование преобразователя RS–485/RS–232 без гальванической развязки может повредить порт RS–232 вашего компьютера.

Кабели RS–232/PPI и USB/PPI Multi-Master фирмы Siemens (номер для заказа соответственно 6ES7 901-3CB30-0XA0 или 6ES7 901-3DB30-0XA0) обеспечивает гальваническую развязку между портом RS–485 на CPU S7–200 и портом RS–232 или USB, используемым для подключения на вашем компьютере. Если вы не пользуетесь кабелем Multi-Master фирмы Siemens, то вы должны позаботиться о гальванической развязке для порта RS–232 вашего компьютера.

Таблица 7–8. Платы CP и протоколы, поддерживаемые STEP 7-Micro/WIN

Конфигурация	Скорость передачи	Протокол
Кабель RS–232/PPI Multi–Master или USB/PPI Multi–Master ¹ Подключен к порту на устройстве программирования	от 9,6 кБод до 187,5 кБод	PPI
CP 5511 Тип II, плата PCMCIA (для ноутбука)	от 9,6 кБод до 12 МБод	PPI, MPI и PROFIBUS
CP 5512 Тип II, плата PCMCIA (для ноутбука)	от 9,6 кБод до 12 МБод	PPI, MPI и PROFIBUS
CP 5611 (версия 3 или выше) Плата PCI	от 9,6 кБод до 12 МБод	PPI, MPI и PROFIBUS
CP 1613, S7–1613 Плата PCI	10 МБод или 100 МБод	TCP/IP
CP 1612, SoftNet–S7 Плата PCI	10 МБод или 100 МБод	TCP/IP
CP 1512, SoftNet–S7 Плата PCMCIA (для ноутбука)	10 МБод или 100 МБод	TCP/IP

¹ Кабели Multi–Master обеспечивают гальваническую развязку между портом RS–485 (на CPU S7–200) и портом, через который производится подключение к вашему компьютеру. Использование преобразователя RS–485/RS–232 без гальванической развязки может повредить порт RS–232 вашего компьютера.

Использование устройств ЧМ-интерфейса в вашей сети

CPU S7–200 поддерживает многие виды устройств ЧМ-интерфейса фирмы Siemens, а также других изготовителей. В то время как некоторые из этих устройств ЧМ-интерфейса (например, TD 200 или TP070) не позволяют выбирать протокол связи, используемый этим устройством, другие устройства (например, OP7 и TP170) предоставляют такую возможность.

Если ваше устройство ЧМ-интерфейса позволяет выбирать протокол для обмена данными, обратите внимание на следующие указания:

- Если устройство ЧМ-интерфейса подключено к коммуникационному порту CPU S7–200, и в сети нет других устройств, то выберите для этого устройства ЧМ-интерфейса протокол PPI или MPI.
- Если устройство ЧМ-интерфейса подключено к модулю EM 277 PROFIBUS, выберите протокол MPI или PROFIBUS.
 - Если сеть с устройством ЧМ-интерфейса содержит ПЛК S7–300 или S7–400, то выберите для устройства ЧМ-интерфейса протокол MPI.
 - Если сеть с устройством ЧМ-интерфейса является сетью PROFIBUS, то установите для этого устройства ЧМ-интерфейса протокол PROFIBUS и выберите профиль, соответствующий другим master-устройствам в сети PROFIBUS.
- Если устройство ЧМ-интерфейса подключено к коммуникационному порту CPU S7–200, сконфигурированного в качестве master-устройства, выберите для этого устройства ЧМ-интерфейса протокол PPI. Оптимальным решением является протокол PPI Advanced. Протоколы MPI и PROFIBUS не поддерживают CPU S7–200 в качестве master-устройства.

Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными

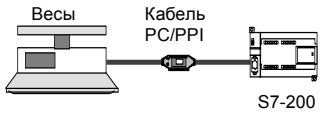
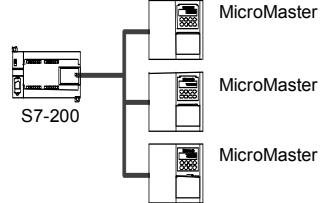

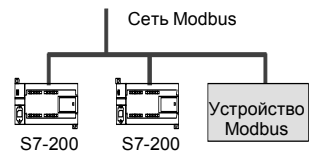

При свободно программируемом обмене данными (Freerport) ваша программа может управлять коммуникационным портом CPU S7–200. Режим свободно программируемого обмена данными можно использовать для реализации определяемых пользователем протоколов для обмена данными со многими типами интеллектуальных устройств. Режим свободно программируемого обмена данными поддерживает протоколы ASCII и двоичные протоколы.

Режим свободно программируемого обмена данными активизируется с помощью байтов специальной памяти SMB30 (для порта 0) и SMB130 (для порта 1). Ваша программа управляет работой коммуникационного порта, используя следующие средства:

- Команда передачи (XMT) и прерывание по передаче: Команда передачи дает возможность модулю S7-200 передать до 255 символов из COM-порта. Прерывание по передаче извещает вашу программу в S7-200, когда передача завершена.
- Прерывание по приему символа: Прерывание по приему символа извещает программу пользователя о том, что в COM-порту был принят символ. После этого ваша программа может реагировать на этот символ в зависимости от реализованного протокола.
- Команда приема (RCV): Команда приема принимает все сообщение из COM-порта, а затем генерирует прерывание для вашей программы, когда это сообщение полностью принято. Чтобы настроить команду приема на определенные вами условия начала и прекращения приема сообщений, используется специальная память S7-200. С помощью команды приема ваша программа может начинать или останавливать прием сообщения после появления определенных символов или через определенные интервалы времени. С помощью команды приема можно реализовать большинство протоколов.

Режим свободно программируемого обмена данными активен только тогда, когда S7-200 находится в режиме RUN. Перевод S7-200 в состояние STOP останавливает свободно программируемый обмен данными, и коммуникационный порт возвращается к протоколу PPI с настройками, установленными в системном блоке S7-200.

Таблица 7-9. Использование режима свободно программируемого обмена данными

Конфигурация сети		Описание
Использование режима свободно программируемого обмена данными через соединение RS-232		<p>Пример: Использование S7-200 с электронными весами, имеющими порт RS-232.</p> <ul style="list-style-type: none"> • Кабель RS-232/PPI Multi-Master соединяет порт RS-232 на весах с портом RS-485 на CPU S7-200. (Настройте кабель на режим PPI/Freepoint (свободно программируемый обмен данными), переключатель 5=0.) • CPU S7-200 использует режим свободно программируемого обмена данными для связи с весами. • Скорость передачи может принимать значения от 1200 Бод до 115,2 кБод. • Протокол определяется программой пользователя.
Использование протокола USS		<p>Пример: Использование S7-200 с приводами SIMODRIVE MicroMaster.</p> <ul style="list-style-type: none"> • STEP 7-Micro/WIN предоставляет в распоряжение библиотеку USS. • CPU S7-200 является master-устройством, а приводы – slaves-устройствами. <p> Пример программы, использующей USS, вы найдете в советах по программированию на компакт-диске с документацией. См. совет 28</p> <p><small>Советы по программированию</small></p>
Создание программы пользователя, эмулирующей slave-устройство в другой сети		<p>Пример: Соединение CPU S7-200 с сетью Modbus.</p> <ul style="list-style-type: none"> • Программа пользователя в S7-200 эмулирует slave-устройство Modbus. • STEP 7-Micro/WIN предоставляет в распоряжение библиотеку Modbus. <p> Пример программы, использующей Modbus, вы найдете в советах по программированию на компакт-диске с документацией. См. совет 28</p> <p><small>Советы по программированию</small></p>

Использование кабеля RS-232/PPI Multi-Master и режима свободно программируемого обмена данными с устройствами

Кабель RS-232/PPI Multi-Master и функции свободно программируемого обмена данными можно использовать для связи CPU S7-200 со многими устройствами, совместимыми со стандартом RS-232. Кабель должен быть настроен на режим PPI/Freepoint (свободно программируемый обмен данными) (переключатель 5 = 0). Переключатель 6 выбирает локальный режим (DCE) (переключатель 6 = 0) и ли дистанционный режим (DTE) (переключатель 6 = 1).

Кабель RS-232/PPI Multi-Master находится в режиме передачи, когда данные передаются из порта RS-232 в порт RS-485. Кабель находится в режиме приема, когда он бездействует или передает данные из порта RS-485 в порт RS-232. Кабель переходит из режима приема в режим передачи немедленно после обнаружения символов в линии передачи RS-232.

Кабель RS-232/PPI Multi-Master поддерживает скорости передачи между 1200 Бод и 115,2 кБод. Для настройки кабеля на нужную скорость передачи используются DIP-переключатели на корпусе кабеля RS-232/PPI Multi-Master. В таблице 7-10 показаны скорости передачи и положения переключателей.

Кабель возвращается к режиму приема, когда линия передачи RS-232 бездействует в течение определенного интервала времени, определяемого как время переключения кабеля. Время переключения определяется скоростью передачи, как показано в таблице 7-10.

Если вы работаете с кабелем RS-232/PPI Multi-Master в системе, где используется свободно программируемый обмен данными, то программа в S7-200 должна учитывать время переключения в следующих ситуациях:

- S7-200 реагирует на сообщения, передаваемые устройством RS-232.
После получения модулем S7-200 запроса от устройства RS-232 модуль S7-200 должен задержать передачу ответного сообщения на интервал времени, не меньший чем время переключения кабеля.
- Устройство RS-232 реагирует на сообщения, передаваемые из S7-200.
После получения модулем S7-200 реакции устройства RS-232 модуль S7-200 должен задержать передачу следующего запроса на интервал времени, не меньший чем время переключения кабеля.

В обеих ситуациях задержка предоставляет кабелю RS-232/PPI Multi-Master достаточное время для переключения из режима передачи в режим приема, чтобы можно было передавать данные из порта RS-485 в порт RS-232.

Таблица 7-10, Время переключения и настройки

Скорость передачи	Время переключения	Настройки (1 = наверху)
115200	0,15 мс	110
57600	0,3 мс	111
38400	0,5 мс	000
19200	1,0 мс	001
9600	2,0 мс	010
4800	4,0 мс	011
2400	7,0 мс	100
1200	14,0 мс	101

Использование в сети модемов и STEP 7-Micro/WIN

STEP 7-Micro/WIN версии 3.2 и выше использует для выбора и конфигурирования телефонных модемов обычно принятые в Windows настройки телефона и модема. Настройки телефона и модема находятся в панели управления (Control Panel) Windows. Использование настроек Windows для модемов дает возможность:

- Применять большинство внутренних и внешних модемов, поддерживаемых Windows.
- Применять стандартные конфигурации для большинства модемов, поддерживаемых Windows.
- Применять обычные для Windows правила набора для выбора мест расположения, кода страны и области, импульсного или тонального способа набора и при поддержке «визитных карточек» (calling card).
- Применять более высокие скорости передачи при обмене данными с модемом EM 241.

Диалоговое окно Modems Properties [Свойства модемов] вызывается через панель управления Windows. В этом диалоговом окне вы можете настроить локальный модем. Вы выбираете свой модем из списка модемов, поддерживаемых Windows. Если тип вашего модема не отображается в диалоговом окне Modem Properties, выберите тип, наиболее близкий к вашему, или обратитесь к изготовителю вашего модема, чтобы получить файлы с конфигурацией модема для Windows.

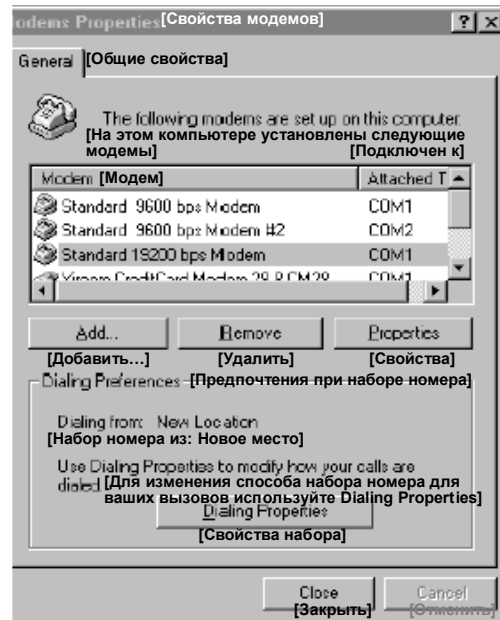


Рис. 7–23. Настройка локального модема

STEP 7-Micro/WIN допускает также использование радиомодемов и модемов сотовой связи. Эти типы модемов не появляются в диалоговом окне Modems Properties [Свойства модемов] Windows, но они имеются в распоряжении при настройке соединения для STEP 7-Micro/WIN.

Настройка соединения через модем

Соединение ставит в соответствие идентификационное имя физическим свойствам соединения. Для телефонного модема эти свойства включают в себя тип модема, 10- или 11-битовый протокол и значения для времени простоя. Для модемов сотовой связи соединение допускает установку PIN-кода и других параметров. Свойства радиомодемов включают в себя скорость передачи, контроль четности, контроль потока и другие параметры.



Мастер соединения

Добавление соединения

Для добавления нового соединения, удаления или редактирования соединения используйте мастер соединения (Connection wizard), как показано на рис. 7–24.

1. Дважды щелкните на символе в диалоговом окне Communication [Связь].
2. Двойным щелчком на Interface PG/PC вызовите интерфейс PG/PC. Выберите кабель PPI и щелкните на кнопке Properties [Свойства]. В закладке Local Connection [Локальное соединение] отметьте триггерную кнопку Modem Connection [Соединение через модем].
3. В диалоговом окне Communication [Связь] дважды щелкните на пиктограмме для соединения через модем.
4. Щелкните на кнопке Settings [Настройки], чтобы вызвать диалоговое окно Modem Connection Settings [Настройки соединения через модем].
5. Щелкните на кнопке Add [Добавить], чтобы вызвать мастер соединения через модем (Add Modem Connection wizard).
6. Настройте соединение с помощью подсказок мастера.

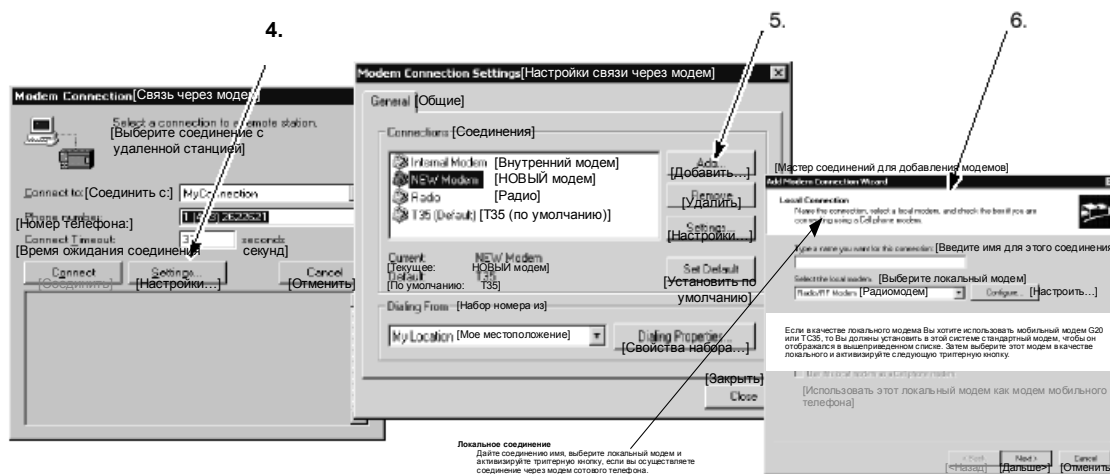


Рис. 7–24. Добавление соединения через модем

Соединение с S7–200 с помощью модема

Добавив соединение через модем, вы можете связаться с CPU S7–200.

1. Откройте диалоговое окно Communications [Связь] и дважды щелкните на пиктограмме Connect [Соединить], чтобы вызвать диалоговое окно Modem Connection [Соединение через модем].
2. В диалоговом окне Modem Connection щелкните на Connect [Соединить], чтобы набрать код модема.

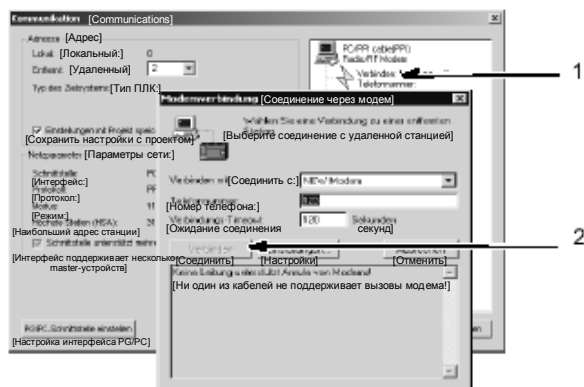


Рис. 7–25. Соединение с S7–200



Мастер настройки модемного модуля

Настройка удаленного модема

Удаленный модем – это модем, присоединенный к S7–200. Если этот модем является модемным модулем EM 241, то настройка не требуется. Если же вы устанавливаете соединение с автономным модемом или с модемом сотовой связи, то вы должны настроить соединение.

Настройка удаленного модема, подключенного к CPU S7–200, производится с помощью мастера настройки модемного модуля (Modem Expansion wizard). Особые конфигурации модема необходимы, чтобы надлежащим образом обмениваться данными с полудуплексным портом RS–485 CPU S7–200. Просто выберите тип модема и введите информацию по подсказке мастера. Подробную информацию по этому вопросу вы найдете в оперативной помощи в режиме online.

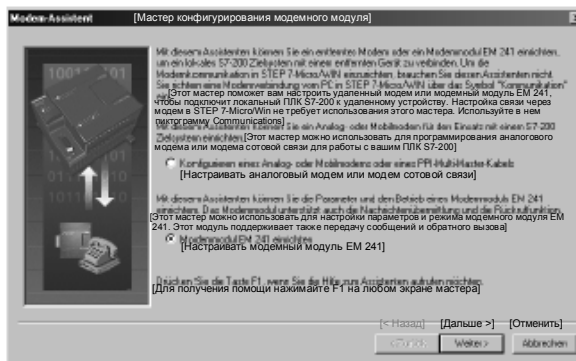


Рис. 7–26. Мастер настройки модемного модуля

Настройка кабеля PPI Multi–Master для работы с удаленным модемом

Кабель RS–232 PPI Multi–Master может посылать AT-команды для модемов при включении кабеля. Обратите, пожалуйста, внимание, что эта настройка требуется только в том случае, если настройки модема, установленные по умолчанию, должны быть изменены. См. рис. 7–27.

Команды модема могут быть указаны в общих командах. Единственной настройкой по умолчанию является команда для автоматических ответов.

Команды авторизации сотовых телефонов и PIN-коды могут быть указаны в поле Cell Phone Authorization [Авторизация сотового телефона], например, +CPIN=1234.

Каждая командная строка посылается модему отдельно. Перед каждой командной строкой посылается AT-команда для привлечения внимания модема.

Эти команды инициализируются в кабеле через кнопку «Program/Test [Программировать/тестировать]».

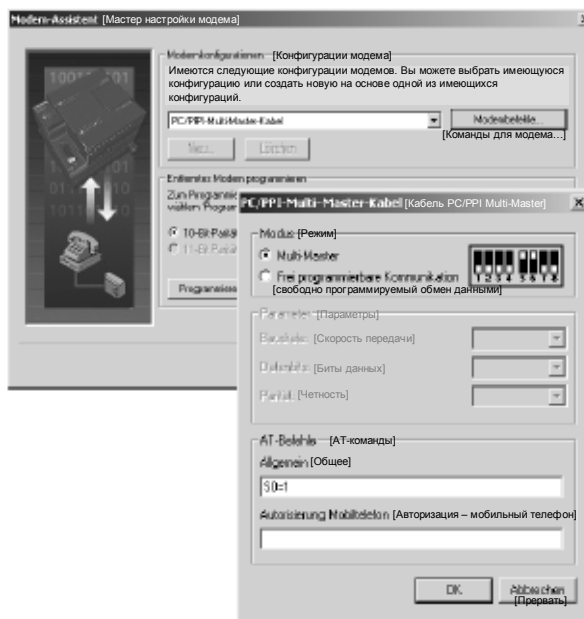


Рис. 7–27. Мастер настройки модема - передача команд для модема

Обратите внимание, что битовый образ отображает рекомендуемые настройки переключателей в зависимости от выбранных параметров.

При настройке кабеля RS–232/PPI Multi–Master с помощью STEP 7-Micro/WIN вы должны подключить разъем RS–485 к CPU S7–200. Это источник питания 24 В, необходимый для работы кабеля. Обеспечьте подачу напряжения питания на CPU S7–200.

После завершения настройки кабеля RS–232/PPI Multi–Master в STEP 7-Micro/WIN отсоедините кабель от ПК и присоедините его к модему. Выключите и снова включите питание модема и кабеля. Теперь вы готовы к использованию кабеля в дистанционном режиме в сети PPI с несколькими master-устройствами.

	<p>Совет Для использования модема с кабелем PPI Multi–Master модем имеет заводские настройки, установленные по умолчанию.</p>
--	--

Настройка кабеля PPI Multi–Master для работы в режиме свободно программируемого обмена данными

Кабель RS–232 PPI Multi–Master также может посылать AT-команды для модемов, когда кабель настроен для работы в режиме свободно программируемого обмена данными (Freerport). Обратите, пожалуйста, внимание, что эта настройка требуется только в том случае, если настройки модема, установленные по умолчанию, должны быть изменены. Однако, кабель должен быть также настроен на совпадение с установленными для порта S7–200 скоростью передачи, проверкой четности и количеством битов данных. Это необходимо, так как прикладная программа S7–200 управляет конфигурацией этих параметров.

Скорости передачи могут выбираться между 1,2 кБод и 115,2 кБод.

Можно выбрать семь или восемь битов данных.

Можно выбрать контроль на четность, нечетность или отсутствие контроля.

Обратите внимание, что битовый образ отображает рекомендуемые настройки переключателей в зависимости от выбранных параметров.

При настройке кабеля RS–232/PPI Multi–Master с помощью STEP 7-Micro/WIN вы должны подключить разъем RS–485 к CPU S7–200. Это источник питания 24 В, необходимый для работы кабеля. Обеспечьте подачу напряжения питания на CPU S7–200.

После завершения настройки кабеля RS–232/PPI Multi–Master в STEP 7-Micro/WIN отсоедините кабель от ПК и присоедините его к модему. Выключите и снова включите питание модема и кабеля. Теперь вы готовы к использованию кабеля в дистанционном режиме в сети PPI с несколькими master-устройствами.

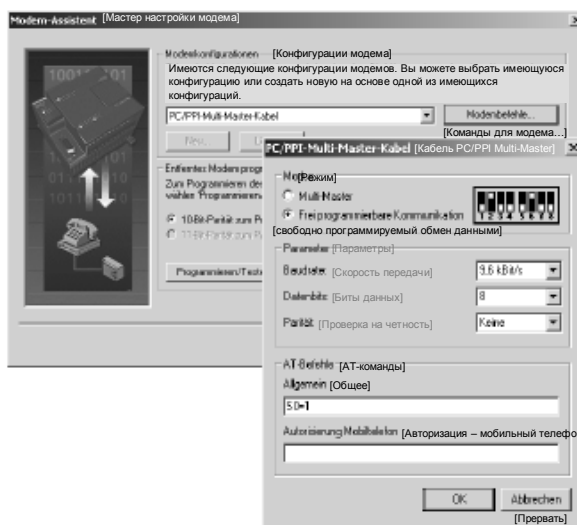


Рис. 7–28 Мастер настройки модема - передача команд для модема в режиме свободно программируемого обмена данными



Совет

Для использования модема с кабелем PPI Multi–Master модем имеет заводские настройки, установленные по умолчанию.

Использование телефонного модема с кабелем RS–232/PPI Multi–Master

Кабель RS–232/PPI Multi–Master можно использовать для соединения коммуникационного порта RS–232 модема с CPU S7–200. См. рис. 7–29.

- Переключатели 1, 2 и 3 устанавливают скорость передачи.
- Переключатель 5 выбирает режим PPI или PPI/Freerport.
- Переключатель 6 выбирает режим локальный (эквивалентно Data Communications Equipment [Оборудование для обмена данными] - DCE) или дистанционный (эквивалентно Data Terminal Equipment [Терминальное оборудование] - DTE).
- Переключатель 7 выбирает 10– или 11–битовый протокол PPI.

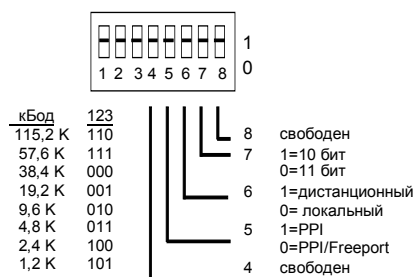


Рис. 7–29. Настройки для кабеля RS–232/PPI Multi–Master

Переключатель 5 выбирает работу в режиме PPI или в режиме PPI/Freerport (свободно программируемый обмен данными). Если, используя STEP 7-Micro/WIN, вы общаетесь данными с S7–200 через модемы, то выберите режим PPI (переключатель 5 = 1). В противном случае выберите режим PPI/Freerport (переключатель 5 = 0).

Переключатель 7 кабеля RS-232/PPI Multi-Master выбирает 10-битовый или 11-битовый режим для режима PPI/Freerport (свободно программируемый обмен данными). Используйте переключатель 7 только тогда, когда S7-200 соединен со STEP 7-Micro/WIN через модем в режиме PPI/Freerport. В противном случае установите с помощью переключателя 7 11-битовый режим, чтобы обеспечить его надлежащую работу с другими устройствами.

Переключатель 6 кабеля RS-232/PPI Multi-Master позволяет настроить порт RS-232 кабеля на локальный (DCE) или дистанционный (DTE) режим.

- Если вы используете кабель RS-232/PPI Multi-Master со STEP 7-Micro/WIN или если кабель RS-232/PPI Multi-Master подключен к компьютеру, настройте кабель RS-232/PPI Multi-Master на локальный (DCE) режим.
- Если вы используете кабель RS-232/PPI Multi-Master с модемом (который является DCE-устройством), настройте кабель RS-232/PPI Multi-Master на дистанционный (DTE) режим.

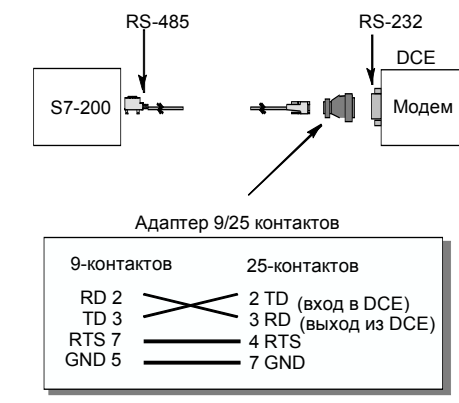


Рис. 7-30. Назначение контактов адаптера

Это устраняет необходимость устанавливать безмодемный адаптер между кабелем RS-232/PPI Multi-Master и модемом. В зависимости от разъема модема вам, возможно, потребуется адаптер 9/25 контактов.

На рис. 7-30 показано назначение контактов для обычного адаптера модема.

Дополнительную информацию о кабеле RS-232/PPI Multi-Master вы найдете в Приложении А. Номера контактов и функции для портов RS-485 и RS-232 кабеля RS-232/PPI Multi-Master в локальном (DCE) режиме показаны в таблице А-66. В таблице А-67 приведены номера контактов и функции для портов RS-485 и RS-232 кабеля RS-232/PPI Multi-Master в дистанционном (DTE) режиме. Кабель RS-232/PPI Multi-Master поставляет сигнал RTS только тогда, когда он находится в дистанционном (DTE) режиме.

Использование радиомодема с кабелем RS-232/PPI Multi-Master

Вы можете использовать кабель RS-232/PPI Multi-Master для подключения коммуникационного порта RS-232 радиомодема к CPU S7-200. Однако работа с радиомодемами отличается от работы с телефонными модемами.

Режим PPI

Если кабель RS-232/PPI Multi-Master настроен на режим PPI (переключатель 5 = 1), то обычно для работы с модемом выбирается дистанционный режим (переключатель 6 = 1). Однако выбор дистанционного режима заставляет кабель при каждом включении посылать строку символов 'AT' и ждать от модема ответа 'OK'. В то время как телефонные модемы используют эту последовательность для установления скорости передачи, радиомодемы обычно не принимают AT-команды.

Поэтому для работы с радиомодемами вы должны выбрать локальный режим (переключатель 6 = 0) и использовать безмодемный адаптер между разъемом RS-232 кабеля и портом RS-232 на вашем радиомодеме. Безмодемные адаптеры имеются в конфигурациях 9/9 контактов и 9/25 контактов.

Настройте радиомодем на работу со скоростью 9,6, 19,2, 38,4, 57,6 или 115,2 кБод. Кабель RS-232/PPI Multi-Master автоматически настраивается на одну из этих скоростей при получении первого символа, переданного радиомодемом.

Режим PPI/свободно программируемый обмен данными (Freerport)

Если кабель RS-232/PPI Multi-Master настроен на режим PPI/Freerport (переключатель 5 = 0), выберите для работы с радиомодемом дистанционный режим (переключатель 6 = 1). Настройте кабель так, чтобы он не посылал AT-команд для настройки модема.

Переключатели 1, 2 и 3 на кабеле RS-232/PPI Multi-Master устанавливают скорость передачи. См. рис. 7-29. Выберите настройку скорости передачи, соответствующую скорости передачи ПЛК и радиомодема.

Для опытных пользователей

Оптимизация производительности сети

На производительность сети влияют следующие факторы (причем наибольшее воздействие оказывают скорость передачи и количество master-устройств):

- Скорость передачи: Если в сети все устройства поддерживают наивысшую скорость передачи, то это оказывает наибольшее воздействие на производительность сети.
- Количество master-устройств в сети: Минимизация количества master-устройств в сети также увеличивает производительность сети. Каждое master-устройство в сети увеличивает потребности в передаче служебных данных в сети; уменьшение количества master-устройств снижает непроизводительные издержки.
- Выбор адресов master- и slave-устройств: Адреса master-устройств должны быть установлены так, чтобы все master-устройства имели последовательные адреса без пропусков между ними. Если между адресами master-устройств есть пропуск, то master-устройства постоянно проверяют пропущенные адреса, чтобы проверить, нет ли еще одного master-устройства, желающего перейти в режим online. Эта проверка требует времени и увеличивает непроизводительные издержки сети. Если между адресами master-устройств нет пропусков, то проверка не производится, и непроизводительные издержки минимизируются. Адресам slave-устройств можно присваивать любое значение, не оказывая влияния на производительность сети, до тех пор, пока эти адреса не окажутся между адресами master-устройств. Наличие адресов slave-устройств между адресами master-устройств увеличивает непроизводительные издержки сети точно так же, как и наличие пропусков между адресами master-устройств.
- Коэффициент актуализации пропусков (Gap Update Factor, GUF): Используется только при работе CPU S7-200 в качестве master-устройства PPI. GUF указывает S7-200, как часто нужно проверять пропуски между адресами других master-устройств. GUF устанавливается в конфигурации CPU для каждого его порта с помощью STEP 7-Micro/WIN. Благодаря этому S7-200 проверяет пропуски между адресами лишь периодически. При GUF=1 S7-200 проверяет пропуски между адресами при каждом получении маркера; при GUF=2 S7-200 проверяет пропуски между адресами при каждом втором получении маркера. При наличии пропусков между адресами master-устройств более высокий GUF сокращает непроизводительные издержки сети. Если пропуски между адресами master-устройств отсутствуют, то GUF не оказывает влияния на производительность. Установка большого значения для GUF вызывает большие задержки при переводе новых master-устройств в режим online, так как адреса проверяются менее часто. По умолчанию GUF устанавливается равным 10.
- Наибольший адрес станции (Highest Station Address, HSA): Используется только при работе CPU S7-200 в качестве master-устройства PPI. HSA определяет наибольший адрес, по которому master-устройство должно искать новое master-устройство. HSA устанавливается в конфигурации CPU для каждого его порта с помощью STEP 7-Micro/WIN. Установка HSA ограничивает диапазон адресов, которые должны быть проверены последним master-устройством (обладающим наибольшим адресом) в сети. Ограничение диапазона адресов минимизирует время, необходимое для поиска и перевода в режим online нового master-устройства. Наибольший адрес станции не влияет на адреса slave-устройств: master-устройства могут обмениваться данными со slave-устройствами, имеющими адреса, большие, чем HSA. Как общее правило, следует устанавливать наибольший адрес станции на одно и то же значение на всех master-устройствах. Этот адрес должен быть не меньше наибольшего адреса master-устройства. Значение по умолчанию для HSA равно 31.

Расчет времени оборота маркера

В сети с передачей маркера обмен данными может инициировать только станция, получившая маркер. Время оборота маркера (время, необходимое маркеру для обхода всех master-устройств в логическом кольце) измеряет производительность сети.

На рис. 7-31 показана сеть с несколькими master-устройствами в качестве примера для расчета времени оборота маркера. В этом примере TD 200 (станция 3) обменивается данными с CPU 222 (станция 2), TD 200 (станция 5) обменивается данными с CPU 222 (станция 4) и т.д. Оба модуля CPU 224 используют команды чтения из сети и записи через сеть для сбора данных от других S7-200: CPU 224 (станция 6) посылает сообщения станциям 2, 4 и 8, а CPU 224 (станция 8) посылает сообщения станциям 2, 4 и 6. В этой сети имеется шесть master-устройств (четыре TD 200 и два CPU 224) и два slave-устройства (два CPU 222).

Описание сетей с передачей маркера вы найдете в советах по программированию на компакт-диске с документацией. См. Совет 42.

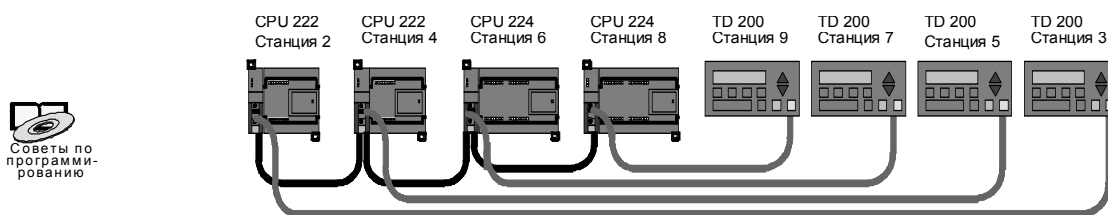


Рис. 7–31. Пример сети с передачей маркера

Чтобы master-устройство могло послать сообщение, оно должно обладать маркером. Например: Если станция 3 обладает маркером, она инициирует запрос станции 2, а затем передает маркер станции 5. После этого станция 5 инициирует запрос станции 4, после чего передает маркер станции 6. Затем станция 6 инициирует сообщение станции 2, 4 или 8 и передает маркер станции 7. Этот процесс инициирования сообщения и дальнейшей передачи маркера продолжается по всему логическому кольцу от станции 3 к станции 5, к станции 6, к станции 7, к станции 8, к станции 9 и, наконец, обратно к станции 3. Маркер должен совершить полный оборот по логическому кольцу, чтобы master был способен послать запрос на получение информации. Для логического кольца из шести станций, посылающих за время пребывания маркера одно сообщение с запросом на чтение или запись одного двойного слова (четыре байта данных), время оборота маркера составляет приблизительно 900 мс при скорости передачи 9600 Бод. Увеличение количества байтов данных, к которым нужно получить доступ за одно сообщение, или увеличение количества станций увеличивает время оборота маркера.

Время оборота маркера определяется тем, как долго станция удерживает маркер. Время оборота маркера для вашей сети с несколькими master-устройствами определяется суммой времен обладания маркером каждым из master-устройств. Если активизирован привилегированный режим PPI (в протоколе PPI вашей сети), то вы можете посылать из S7–200 сообщения другим S7–200 с помощью команд чтения из сети и записи через сеть. При посылке сообщений с помощью этих команд вы можете использовать формулу для приблизительного расчета времени оборота маркера, представленную на следующем рисунке. Формула основана на следующих допущениях: за время обладания маркером каждая станция посылает один запрос, этот запрос является запросом на чтение или запись для последовательно расположенных адресов, отсутствуют конфликты за использование одного коммуникационного буфера в S7–200, и нет ни одного S7–200 со временем цикла, большим 10 мс.

Время обладания маркером (T_{hold}) = (непроизводительное время 128 + n символов данных) x 11 битов/символ x 1/скорость передачи	
Время оборота маркера (T_{rot}) = T_{hold} master-устройства 1 + T_{hold} master-устройства 2 + . . . + T_{hold} master-устройства m	
где	n – количество символов данных (байтов) m – количество master-устройств

Для примера, показанного на рис. 7–31, расчет времен оборота производится с помощью следующих уравнений (в них «время передачи бита» равно длительности передачи одного сигнала):

T (время обладания маркером) = (128 + 4 символа) x 11 битов/символ x 1/9600 времен передачи бита/с = 151,25 мс на master-устройство
 T (время оборота маркера) = 151,25 мс на master-устройство x 6 master-устройств = 907,5 мс

	Совет
	Программное обеспечение SIMATIC NET COM PROFIBUS предоставляет в распоряжение анализатор для определения производительности сети.

Сравнение времен оборота маркера

Таблица 7–11 дает сравнение времен оборота маркера при различном количестве станций, различном объеме данных и различных скоростях передачи. Эти времена относятся к случаю, когда в CPU S7–200 или других master-устройствах используются команды чтения из сети и записи через сеть.

Таблица 7–11. Время оборота маркера (в секундах)

Скорость передачи	Передано байтов	Количество master-устройств								
		2	3	4	5	6	7	8	9	10
9,6 кБод	1	0,30	0,44	0,59	0,74	0,89	1,03	1,18	1,33	1,48
	16	0,33	0,50	0,66	0,83	0,99	1,16	1,32	1,49	1,65
19,2 кБод	1	0,15	0,22	0,30	0,37	0,44	0,52	0,59	0,67	0,74
	16	0,17	0,25	0,33	0,41	0,50	0,58	0,66	0,74	0,83
187,5 кБод	1	0,009	0,013	0,017	0,022	0,026	0,030	0,035	0,039	0,043
	16	0,011	0,016	0,021	0,026	0,031	0,037	0,042	0,047	0,052

Обзор соединений между сетевыми устройствами

Сетевые устройства обмениваются данными через индивидуальные соединения, которые являются "частными" линиями связи между master- и slave-устройством. Как показано на рис. 7–32, протоколы обмена данными отличаются в зависимости от того, как эти соединения обрабатываются:

- Протокол PPI использует одно общее соединение для всех сетевых устройств.
- Протоколы PPI Advanced, MPI и PROFIBUS используют отдельные соединения для обмена данными между двумя устройствами.

При использовании протокола PPI Advanced, MPI или PROFIBUS второе master-устройство не может влиять на соединение, установленное между master- и slave-устройством. Модули CPU S7–200 и EM 277 всегда резервируют одно соединение для STEP 7-Micro/WIN и одно соединение для устройств ЧМ-интерфейса. Другие master-устройства не могут использовать эти зарезервированные соединения. Это гарантирует, что вы всегда можете подключить по крайней мере одно устройство программирования и по крайней мере одно устройство ЧМ-интерфейса к CPU S7–200 или EM 277, если master-устройство использует протокол, поддерживающий такие соединения, как, например, PPI Advanced.

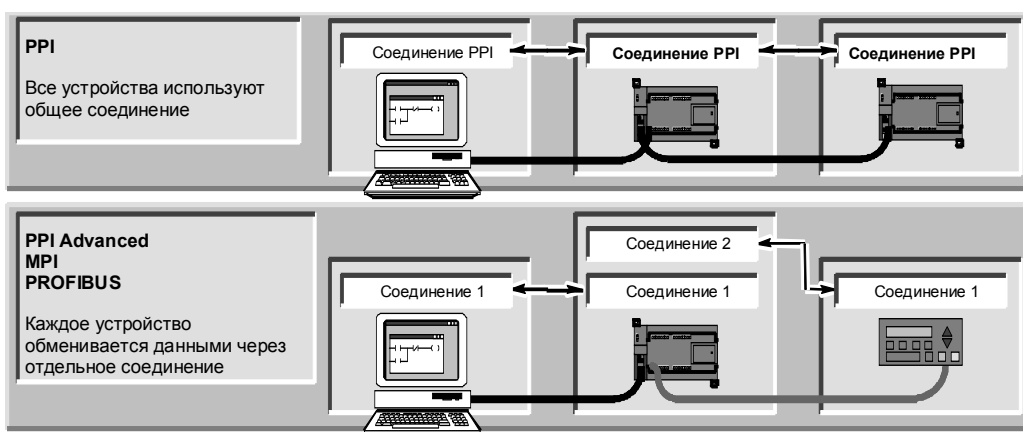


Рис. 7–32. Управление коммуникационными соединениями

Как показано в таблице 7–12, CPU S7–200 и EM 277 предоставляют в распоряжение определенное количество соединений. Каждый порт (порт 0 и порт 1) CPU S7–200 поддерживает до четырех отдельных соединений. (Таким образом, может иметь до восьми соединений CPU S7–200.) Кроме того, имеется совместно используемое соединение PPI. EM 277 поддерживает шесть соединений.

Таблица 7–12. Возможности модулей CPU S7–200 и EM 277

Точка присоединения	Скорость передачи	Соединения	Настройки в STEP 7-Micro/WIN на протокол	
CPU S7–200	Порт 0	9,6 кБод, 19,2 кБод или 187,5 кБод	4	PPI, PPI Advanced, MPI и PROFIBUS ¹
	Порт 1	9,6 кБод, 19,2 кБод, или 187,5 кБод	4	PPI, PPI Advanced, MPI и PROFIBUS ¹
Модуль EM 277	от 9,6 кБод до 12 МБод	6 на модуль	PPI Advanced, MPI и PROFIBUS	

¹ Если для подключения STEP 7-Micro/WIN к CPU S7–200 через порт 0 или порт 1 используется плата CP, то вы можете выбрать профиль MPI или DP PROFIBUS только в том случае, если устройство S7–200 сконфигурировано как slave.

Работа со сложными сетями

Для S7–200 сложные сети обычно имеют несколько master-устройств S7–200, которые используют команды чтения из сети (NETR) и записи через сеть (NETW) для обмена данными с другими устройствами в сети PPI. В сложных сетях обычно имеют место определенные проблемы, которые могут препятствовать master-устройству обмениваться данными со slave-устройством.

Если сеть работает с низкой скоростью передачи (например, 9,6 кБод или 19,2 кБод), то каждое master-устройство завершает транзакцию (чтение или запись) перед передачей маркера. Однако при скорости 187,5 кБод master-устройство выдает запрос slave-устройству, а затем передает маркер дальше, так что у slave-устройства остается невыполненный запрос.

На рис. 7–33 показана сеть с возможными коммуникационными конфликтами. В этой сети станция 1, станция 2 и станция 3 являются master-устройствами, использующими команды чтения из сети и записи через сеть для обмена данными со станцией 4. Команды чтения из сети и записи через сеть применяют протокол PPI так, что все устройства S7–200 совместно используют одно соединение PPI со станцией 4.

В этом примере станция 1 выдает запрос станции 4. При скоростях передачи свыше 19,2 кБод станция 1 после этого передает маркер станции 2. Если станция 2 пытается выдать запрос станции 4, то запрос от станции 2 отвергается, так как еще имеет место запрос от станции 1. Все запросы к станции 4 будут отвергаться, пока станция 4 не завершит ответ станции 1. Только после завершения этого ответа другое master-устройство сможет выдать запрос станции 4.

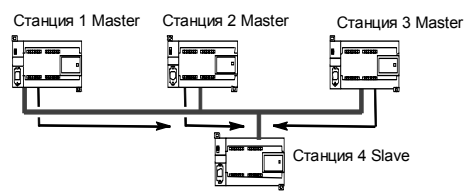


Рис. 7–33. Коммуникационный конфликт

Чтобы устранить этот конфликт для коммуникационного порта на станции 4, вы можете определить станцию 4 в качестве единственного master-устройства в сети, как показано на рис. 7–34. Тогда станция 4 выдает запросы на чтение и запись остальным S7–200.

Эта конфигурация не только обеспечивает устранение конфликта при обмене данными, но она также сокращает непроизводительные расходы, вызванные наличием нескольких master-устройств в сети, и дает возможность сети работать более эффективно.

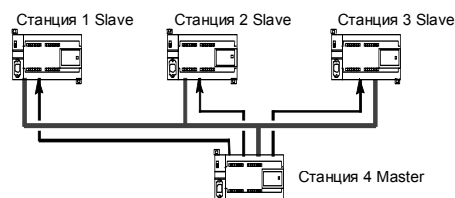


Рис. 7–34. Устранение конфликта

Однако для некоторых приложений сокращение количества master-устройств в сети невозможно. Если имеется несколько master-устройств, вы должны управлять временем оборота маркера, обращая внимание на то, чтобы сеть не превысила целевое время оборота маркера. (Время оборота маркера – это количество времени, проходящего с момента передачи master-устройством маркера до получения его снова.)

Таблица 7–13. Наибольший адрес станции (HSA) и целевое время оборота маркера

HSA	9,6 кБод	19,2 кБод	187,5 кБод
HSA=15	0,613 с	0,307 с	31 мс
HSA=31	1,040 с	0,520 с	53 мс
HSA=63	1,890 с	0,950 с	97 мс
HSA=126	3,570 с	1,790 с	183 мс

Если время, необходимое маркеру для возвращения к master-устройству, больше, чем целевое время оборота маркера, то master-устройство не может послать запрос. Master-устройство может выдать запрос только тогда, когда фактическое время оборота маркера меньше, чем целевое время оборота маркера.

Целевое время оборота маркера определяется настройками наибольшего адреса станции (HSA) и скорости передачи для S7–200. Целевые времена оборота маркера приведены в таблице 7–13.

При низких скоростях передачи, например, 9,6 кБод и 19,2 кБод, master-устройство ожидает ответа на свой запрос до того, как оно передаст маркер. Так как обработка цикла запрос/ответ может занимать относительно большое время по сравнению с временем цикла обработки программы, то велика вероятность того, что каждое master-устройство в сети может подготовить запрос для передачи каждый раз, как оно обладает маркером. Фактическое время оборота маркера при этом увеличилось бы, и некоторые master-устройства, возможно, не смогли бы вообще обрабатывать никакие запросы. В некоторых случаях master-устройство может обрабатывать запросы лишь изредка.

Например: Представьте себе сеть с наибольшим адресом станции HSA = 15, содержащую 10 master-устройств, которые передают 1 байт со скоростью 9,6 кБод. В этом примере каждое из master-устройств всегда имеет сообщение, готовое к передаче. Как показано в таблице 7–13, целевое время оборота маркера для этой сети равно 0,613 секунды. Однако, в соответствии с данными о производительности, приведенными в таблице 7–11, фактическое время оборота маркера, необходимое для этой сети, равно 1,48 секунды. Так как фактическое время оборота маркера больше, чем целевое время оборота маркера, то некоторым master-устройствам будет разрешено передать сообщение только при последующих оборотах маркера.

У вас есть две возможности для улучшения ситуации, когда фактическое время оборота маркера больше, чем целевое время оборота маркера:

- Вы можете сократить фактическое время оборота маркера, уменьшив количество master-устройств в своей сети. Для некоторых приложений это решение может оказаться непригодным.
- Вы можете увеличить целевое время оборота маркера, увеличив HSA для всех master-устройств в сети.

Увеличение HSA может вызвать другую проблему в вашей сети, так как это влияет на время, необходимое S7–200 для перехода в привилегированный режим и входа в сеть. Если вы используете таймер, чтобы гарантировать завершение команд чтения из сети и записи через сеть в течение заданного времени, то может оказаться, что задержка при инициализации привилегированного режима и включения S7–200 в сеть в качестве master-устройства вызовет превышение лимита времени при выполнении команды. Вы можете минимизировать задержку при добавлении master-устройств в сеть, уменьшая коэффициент актуализации пропусков между адресами (GUF) для всех master-устройств в сети.

Из-за способа, которым запросы передаются slave-устройству и там оставляются при скорости передачи 187,5 кБод, вам следует запланировать дополнительное время при установке целевого времени оборота маркера. При скорости 187,5 кБод фактическое время оборота маркера должно составлять примерно половину целевого времени оборота маркера.

При определении времени оборота маркера пользуйтесь данными о производительности, приведенными в таблице 7–11, чтобы определить время, необходимое для завершения операций чтения из сети и записи через сеть. Для расчета времени, необходимого для устройств ЧМ-интерфейса (например, TD 200), пользуйтесь данными о производительности для передачи 16 байтов. Рассчитайте время оборота маркера, сложив времена для всех устройств в сети. Сложение всех времен описывает наихудший сценарий, когда все устройства намерены обрабатывать запрос за время одного и того же оборота маркера. Это определяет максимальное время оборота маркера, необходимое для сети.

Например: Рассмотрим сеть, работающую со скоростью 9,6 кБод, с четырьмя TD 200 и четырьмя S7–200, где каждый S7–200 каждую секунду записывает 10 байтов данных другому S7–200. Воспользуемся таблицей 7–11 для расчета конкретных времен передачи для сети:

Передача 16 байтов данных 4 устройствами TD 200	=	0,66 с
Передача 10 байтов данных 4 модулями S7–200	=	<u>0,63 с</u>
Общее время оборота маркера	=	1,29 с

Чтобы обеспечить этой сети достаточное время для обработки всех запросов в течение одного оборота маркера, установите HSA равным 63. (См. таблицу 7–13.) Выбор целевого времени оборота маркера (1,89 с), большего, чем максимальное время оборота маркера (1,29 с), гарантирует, что каждое устройство сможет передавать данные при каждом обороте маркера.

Для улучшения надежности сети с несколькими master-устройствами вам следует также предусмотреть следующие мероприятия:

- Измените темп актуализации устройств ЧМ-интерфейса, чтобы увеличить время между актуализациями. Например, измените темп актуализации для TD 200 с «As fast as possible [Максимально быстро]» на «Once per second [Один раз в секунду]».
- Сократите количество запросов (и непроизводительные затраты сети для обработки запросов) путем объединения соответствующих операций чтения из сети и записи через сеть. Например, вместо двух операций чтения из сети, которые считывают по 4 байта каждая, используйте одну операцию чтения из сети, которая считывает 8 байтов. Время обработки двух запросов по 4 байта значительно больше, чем время обработки одного запроса на 8 байтов.
- Измените темп актуализации master-устройств S7–200 так, чтобы они не пытались выполнять актуализацию быстрее времени оборота маркера.

Настройка кабеля RS-232/PPI Multi-Master на дистанционный режим

Гипертерминал как инструментальное средство настройки

Если в вашем распоряжении для настройки кабеля RS-232/PPI Multi-Master на дистанционный режим нет STEP 7-Micro/WIN, то вы можете использовать для этой цели гипертерминал (HyperTerminal) или любой другой неинтеллектуальный терминальный пакет. Кабель RS-232/PPI Multi-Master предоставляет в ваше распоряжение встроенное меню, которое будет руководить вами при настройке кабеля на дистанционный режим.

При настройке кабеля RS-232/PPI Multi-Master с помощью гипертерминала вы должны присоединить разъем RS-485 к CPU S7-200. Это источник напряжения 24 В, необходимый для работы кабеля. Обеспечьте подачу напряжения на CPU S7-200.

Чтобы вызвать HyperTerminal на своем ПК, выберите **Start > Programs > Accessories > Communications > HyperTerminal [Пуск > Программы > Принадлежности > Обмен данными > Гипертерминал]**.

HyperTerminal запускается и просит вас ввести описание соединения. Вы должны ввести имя соединения (например, Multi-Master). Щелкните на ОК. Вы можете выбрать пиктограмму или принять стандартный символ для нового соединения. См. рис. 7-35.



Рис. 7-35. HyperTerminal: описание соединения

Появляется диалоговое окно Connect To [Соединить с]. Выберите коммуникационный порт, который вы будете использовать, и щелкните на ОК. После этого отображается диалоговое окно COMx Properties [Свойства COMx]. Примите настройки по умолчанию и щелкните на ОК. См. рис. 7-36.

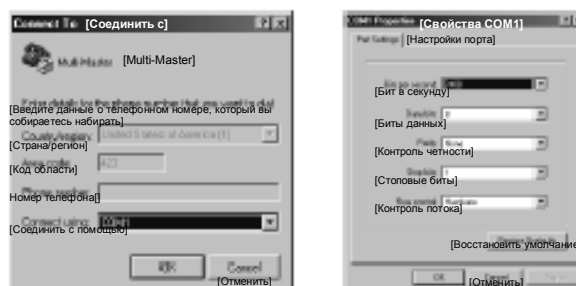


Рис. 7-36. HyperTerminal: Диалоговые окна Connect To и COMx Properties

После щелчка на ОК курсор помещается в окне редактирования гипертерминала, как показано на рис. 7-37. Обратите внимание, что строка состояния в нижней части окна гипертерминала показывает, что соединение установлено, а таймер показывает длительность соединения.

Выберите команду меню **Call > Disconnect [Вызвать > Разъединить]**. Строка состояния теперь показывает, что соединение разорвано.

Выберите **View > Font [Вид > Шрифт]**. Выберите Courier New и щелкните на ОК.

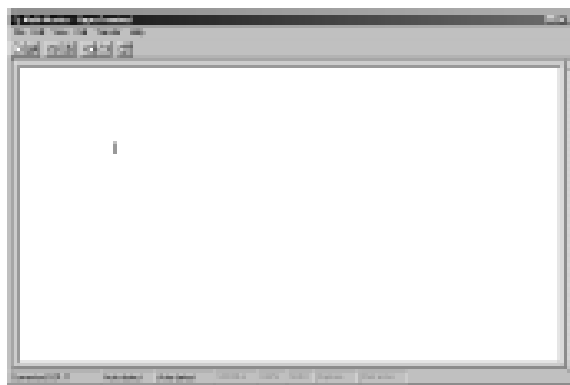


Рис. 7-37. HyperTerminal: Окно редактирования

Выберите **File > Properties [Файл > Свойства]**. В закладке **Connect To [Соединить с]** щелкните на кнопке **Configure ... [Настроить...]**, чтобы отобразить свойства коммуникационного порта. См. рис. 7–38.

В диалоговом окне COMx Properties [Свойства COMx] выберите в раскрывающемся поле для выбора скорость передачи в битах в секунду. Вы должны выбрать скорость передачи от 9600 до 115200 битов в секунду (обычно 9600). выберите 8 битов данных, отсутствие контроля четности, один стоповый бит и отсутствие управления потоком в соответствующих раскрывающихся полях для выбора.

Щелкните на **OK**, чтобы вернуться в закладку **Connect To [Соединить с]**.

Выберите закладку **Settings [Настройка]**. В раскрывающемся поле для выбора **Emulation [Эмуляция]** выберите **ANSI** и щелкните на **OK**. Это вернет вас в окно редактирования гипертерминала. В строке состояния в нижней части экрана отображается: «**Disconnected [Соединение разорвано] ANSI 9600 8-N-1**», как показано на рис. 7–39.

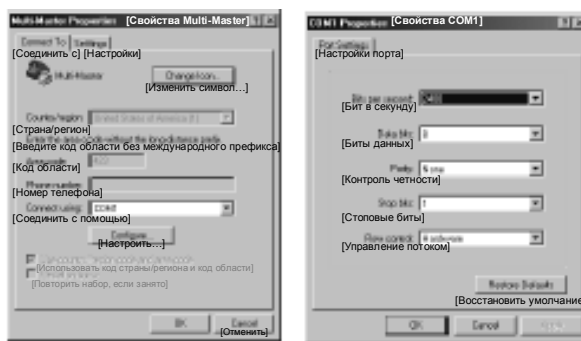


Рис. 7–38. Свойства Multi–Master и COMx

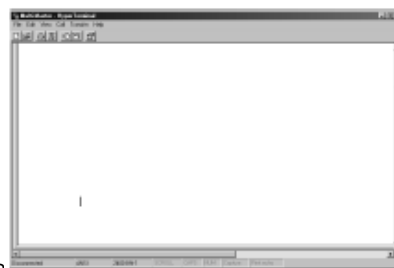


Рис. 7–39. Редактирование в окне гипертерминал – ANSI отсоединено

Для иницирования обмена данными с помощью кабеля RS–232/PPI Multi–Master введите «hhh». Светодиод Rx на кабеле при вводе «hhh» должен загореться примерно на 1 секунду. Светодиод TX кратковременно включается, когда кабель предоставляет в распоряжение выбор языка.

Введите номер, соответствующий желаемому языку (для стирания установки по умолчанию используйте клавишу забора) и нажмите клавишу ввода. На рис. 7–40 показано отображение для выбора языка, а также отображение для установки дистанционного режима для кабеля RS232/PPI.

Это отображение содержит также версию ПЗУ кабеля.

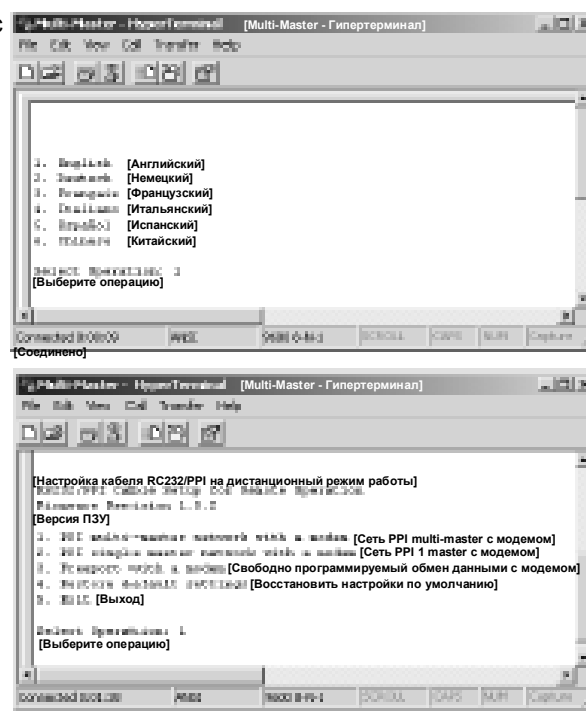


Рис. 7–40. HyperTerminal: выбор языка и настройка кабеля RS–232/PPI

В процессе настройки кабеля RS232/PPI на дистанционный режим работы вы получаете подсказки, какие шаги необходимо выполнить, чтобы настроить кабель на желаемый вид дистанционного режима.

- Если у вас имеется более ранняя версия STEP 7-Micro/WIN, выберите вариант 2 «PPI single master network with a modem [Сеть PPI с одним master-устройством и модемом]».
- Если вы используете свободно программируемый обмен данными с модемом, выберите вариант 3.

Выберите, например, вариант 1 для сети PPI с несколькими master-устройствами с модемом с помощью STEP 7-Micro/WIN 3.2 Service Pack 4 или выше.

Отображение гипертерминала, представленное на рис. 7–41, показывает положения переключателей, которые вам необходимы для настройки кабеля. Благодаря этим положениям переключателей STEP 7-Micro/WIN может через модемы принять участие в удаленной сети с одним или несколькими master-устройствами и одним или несколькими ПЛК S7–200. Такая сеть показана на рис. 7–41.

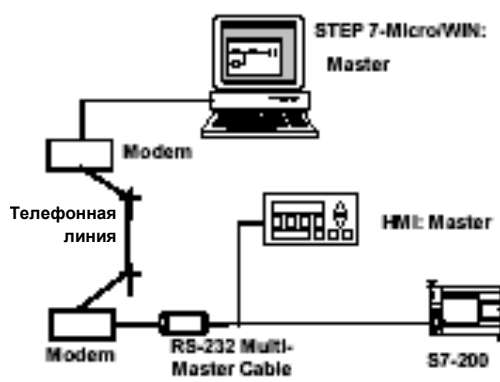
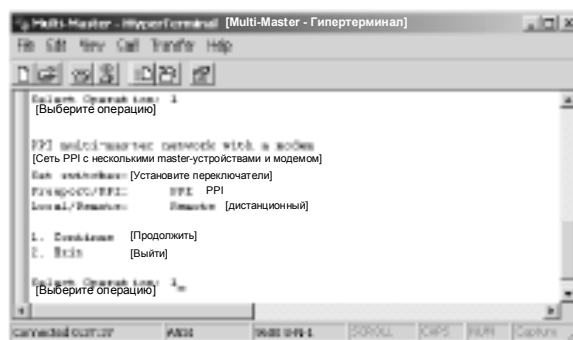


Рис. 7–41. HyperTerminal – настройка кабеля RS–232/PPI

После установки переключателей в соответствии с указаниями выберите Continue [Продолжить]. Появившееся после этого изображение в окне гипертерминала показано на рис. 7–42.

Удаленный модем (подключенный к кабелю RS–232/PPI Multi–Master) должен быть настроен в соответствии с заводскими установками по умолчанию. Когда для удаленного модема сделаны заводские установки, введите командные AT-строки, необходимые для программирования модема для работы с кабелем RS–232/PPI Multi–Master. Обычно должна быть послана только одна команда ATSO=1, которая настраивает модем так, чтобы он автоматически реагировал при первом звонке на поступающие вызовы.

Если вы используете сотовый модем, который требует PIN-кода, отправьте его во второй AT-команде (AT-команды, поддерживаемые вашим модемом, вы найдете в руководстве к модему). Если вам нужно изменить AT-команды, сделайте соответствующий выбор и введите требуемые команды при поступлении запроса на них. Эти запросы содержат пример командных AT-строк, чтобы помочь вам при форматировании команд.

Кабель RS–232/PPI Multi–Master посылает эти командные AT-строки модему при каждом включении кабеля. Обратите внимание, что модем должен быть включен раньше или одновременно с кабелем. Если вы выключаете, а затем включаете модем, то же самое нужно сделать и с кабелем. Это дает возможность кабелю надлежащим образом настраивать модем и работать с наибольшей доступной скоростью передачи.

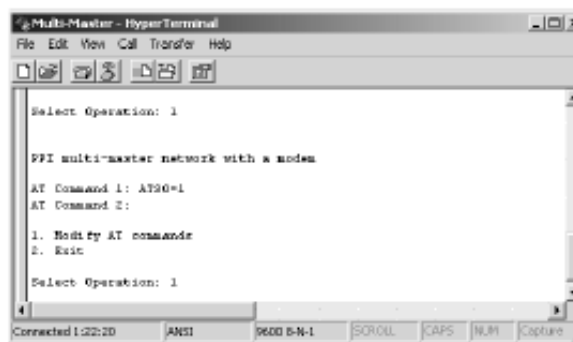


Рис. 7–42. HyperTerminal – Удаленный модем

Представления гипертерминала на рис. 7–43 показывают, как вводить AT-команды. Если вы при запросе ввода не должны вводить вторую AT-команду, нажмите клавишу ввода. Благодаря этому вы снова вернетесь к тому пункту, где вы можете выбрать, изменять ли AT-команды или покинуть редактирование. Если вы закончили ввод AT-команд, выберите Exit [Выйти].

После того как вы закончили настройку кабеля RS–232/PPI Multi–Master в окне гипертерминала, отсоедините кабель от ПК и присоедините его к модему. Выключите и снова включите модем и кабель. Теперь вы готовы использовать кабель в дистанционном режиме в сети PPI с несколькими master-устройствами.

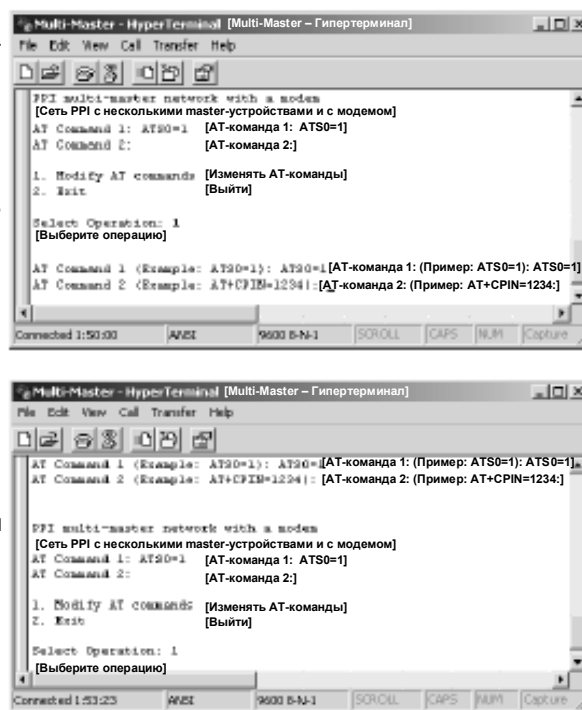


Рис. 7–43. HyperTerminal – AT-команды

Работа с гипертерминалом в режиме свободно программируемого обмена данными

Настройка кабеля RS–232/PPI Multi–Master для режима свободно программируемого обмена данными (Freeport) с помощью гипертерминала очень похожа на пример настройки, описанный выше. Следуйте подсказкам для настройки кабеля в соответствии с вашими потребностями.

Устранение неисправностей аппаратуры и инструментальные средства для тестирования программ

8

STEP 7-Micro/WIN предоставляет инструментальные программные средства, с помощью которых вы можете отлаживать и тестировать свою программу. Эти функции включают в себя просмотр состояния программы во время ее исполнения в S7-200, выполнение S7-200 определенного количества циклов и принудительное присваивание значений переменным.

С помощью таблицы 8-1 вы можете выяснить причину и возможный способ устранения неисправностей в аппаратном обеспечении S7-200.

В этой главе

Функции для отладки вашей программы	246
Отображение состояния программы	248
Использование таблицы состояний для контроля и изменения данных в S7-200	249
Принудительное присваивание значений	250
Исполнение программы в течение определенного количества циклов	250
Указания по устранению неисправностей аппаратуры	251

Функции для отладки вашей программы

STEP 7-Micro/WIN предлагает несколько функций для отладки вашей программы: закладки, таблицы перекрестных ссылок и возможность редактирования программы во время ее исполнения.

Использование закладок для быстрого доступа к программе

Вы можете устанавливать в своей программе закладки, облегчающие повторное нахождение определенных строк в своей программе. С помощью закладок вы можете перемещаться в своей программе к следующей или предыдущей помеченной строке.



Перекрестные ссылки

Использование таблицы перекрестных ссылок для проверки ссылок в вашей программе

Таблица перекрестных ссылок дает возможность отображать перекрестные ссылки и информацию об использовании элементов вашей программы.

Таблица перекрестных ссылок дает список всех операндов, используемых в программе, и все появления операндов в программных блоках, сегментах и строках, а также командах.

Вы имеете возможность переключаться между абсолютным и символическим представлением, чтобы изменять отображение всех операндов.

Рис. 8–1. Таблица перекрестных ссылок



Совет

Двойной щелчок на элементе в таблице перекрестных ссылок перемещает курсор в соответствующее место в программе или блоке.

Редактирование программы в режиме RUN

CPU S7–200, версии 2.0 (и выше) поддерживают редактирование в режиме RUN. Возможность редактирования в режиме RUN позволяет делать небольшие изменения в программе пользователя, не создавая помех процессу, управляемому программой. Однако эта функция позволяет выполнять и большие изменения в программе, которые могут оказать неблагоприятное воздействие на процесс, и даже могут быть опасными.

Предупреждение

Если вы загружаете изменения в S7–200 в режиме RUN, то эти изменения немедленно оказывают воздействие на работу процесса. Изменение в программе в режиме RUN может привести к неожиданному поведению системы, что может вызвать гибель или тяжкие телесные повреждения персонала и/или нанести вред имуществу.

Редактирование в режиме RUN должен выполнять только квалифицированный персонал, понимающий воздействие редактирования в режиме RUN на систему.

Для выполнения редактирования в режиме RUN CPU S7–200, находящийся в режиме online, должен поддерживать редактирование в режиме RUN и должен находиться в режиме RUN.

1. Выберите команду меню **Debug > Program Edit in RUN [Отладка > Редактирование программы в режиме RUN]**.
2. Если проект отличается от программы, находящейся в S7–200, то вы получаете запрос на ее сохранение. Редактировать в режиме RUN можно только программу, находящуюся в S7–200.
3. STEP 7-Micro/WIN предостерегает вас от редактирования в режиме RUN и спрашивает, желаете ли вы продолжить или прекратить операцию. Если вы щелкнете на Continue [Продолжить], STEP 7-Micro/WIN загружает программу из S7–200. Теперь вы можете ее редактировать в режиме RUN. При редактировании нет никаких ограничений.

**Совет**

В LAD команды обнаружения положительного фронта (EU) и обнаружения отрицательного фронта (ED) представляются с помощью операнда. Для просмотра информации об этих командах выберите в отображении программы кнопку Cross Reference [Перекрестные ссылки]. В закладке Edge Usage [Использование фронтов] перечислены номера этих команд в вашей программе. При редактировании программы обращайте внимание на то, чтобы не использовать в своей программе дважды одни и те же номера фронтов.

Загрузка программы в режиме RUN

Функция редактирования в режиме RUN позволяет загружать программный блок только в том случае, если S7-200 находится в режиме RUN. Перед загрузкой программного блока в режиме RUN подумайте о влиянии, которое оказывают изменения в режиме RUN на работу S7-200 в следующих ситуациях:

- При удалении логики управления для некоторого выхода S7-200 сохраняет последнее состояние этого выхода до выключения S7-200 или его перевода в STOP.
- При удалении скоростного счетчика или вывода последовательности импульсов, когда они работают, скоростной счетчик или вывод последовательности импульсов продолжает действовать до следующего включения или следующего перевода S7-200 в STOP.
- При удалении скоростного счетчика или вывода последовательности импульсов, когда они работают, скоростной счетчик или вывод последовательности импульсов продолжает действовать до следующего включения или следующего перевода S7-200 в STOP.
- При добавлении команды назначения прерывания, которая включается битом первого цикла, это событие не активизируется до следующего цикла выключения/включения питания или перевода S7-200 из STOP в RUN.
- При удалении команды разблокирования всех прерываний прерывания продолжают работать до следующего цикла выключения/включения питания или перевода S7-200 из RUN в STOP.
- Если вы изменили адрес принимающего блока в таблице, и принимающий блок был активен при переключении S7-200 со старой программы на измененную, то S7-200 продолжает записывать принимаемые данные по старому адресу. Команды чтения из сети и записи через сеть действуют таким же образом.
- Любая логика, активизируемая битом первого цикла, выполняется только после следующего цикла выключения/включения питания или перевода S7-200 из STOP в RUN. Бит первого цикла устанавливается только при переходе в режим RUN, и функция редактирования в режиме RUN на него влияния не оказывает.

**Совет**

Чтобы вы могли загрузить свою программу в режиме RUN, S7-200 должен поддерживать редактирование в режиме RUN, программа должна быть скомпилирована без ошибок, и обмен данными между STEP 7-Micro/WIN и S7-200 должен происходить без сбоев. Вы можете загрузить в S7-200 только программный блок.

Для загрузки вашей программы в режиме RUN щелкните на кнопке Download [Загрузить] на панели инструментов или выберите команду меню **File > Download [Файл > Загрузить]**. Если программа компилируется успешно, то STEP 7-Micro/WIN загружает программный блок в S7-200.

Выход из редактирования в режиме RUN

Для выхода из редактирования в режиме RUN выберите команду меню **Debug > Program Edit in RUN [Отладка > Редактирование программы в режиме RUN]** и удалите метку рядом с этим пунктом меню. Если у вас есть несохраненные изменения, STEP 7-Micro/WIN предлагает вам продолжить редактирование, загрузить изменения и выйти из редактирования в режиме RUN, или выйти из редактирования без загрузки.

Отображение состояния программы

STEP 7-Micro/WIN дает возможность наблюдать состояние программы во время ее исполнения. При наблюдении состояния программ редактор программ отображает состояние операндов команд.

Для отображения состояния щелкните на кнопке Program Status [Состояние программы] на панели инструментов или выберите команду меню **Debug > Program Status** [Отладка > Состояние программы].

Отображение состояния программы в LAD и FBD

STEP 7-Micro/WIN предоставляет две возможности для отображения состояния программ в LAD и FBD:

- Состояние в конце цикла: STEP 7-Micro/WIN запрашивает значения для отображения состояния в нескольких циклах, а затем обновляет отображение состояния на экране. Отображение состояния не отражает фактическое состояние каждого элемента во время исполнения. Состояние в конце цикла не показывает состояния памяти локальных данных и аккумуляторов.

Для состояния в конце цикла значения состояния обновляются во всех режимах работы CPU.

- Состояние во время исполнения: STEP 7-Micro/WIN отображает значения операндов во время исполнения команд в сегментах программы S7-200. Для отображения состояния во время исполнения выберите команду меню **Debug > Use Execution Status** [Отладка > Состояние во время исполнения].

Для состояния во время исполнения значения состояния обновляются только при нахождении CPU в режиме RUN.



Совет

STEP 7-Micro/WIN предоставляет простой метод для изменения состояния переменной. Просто выберите переменную и щелкните правой кнопкой мыши для вызова контекстного меню.

Настройка отображения состояния в программах LAD и FBD

STEP 7-Micro/WIN предоставляет несколько возможностей для отображения состояния программы.

Для настройки вариантов отображения выберите команду меню **Tools > Options [Инструментальные средства > Возможности]**, а затем выберите закладку Program Editor [Редактор программ], как показано на рис. 8-2.

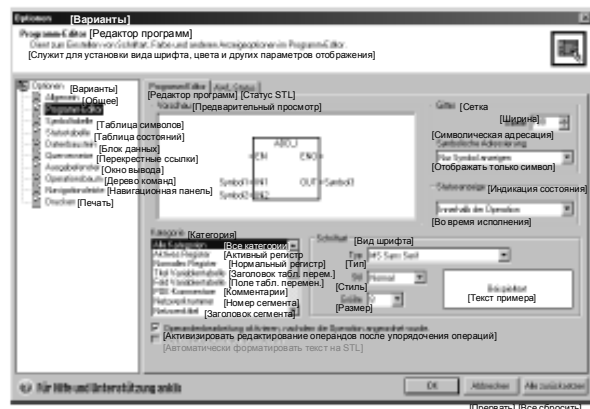


Рис. 8-2. Возможности отображения состояния

Отображение состояния программы в STL

Вы можете наблюдать исполнение своей программы на STL команда за командой. Для программы на STL STEP 7-Micro/WIN отображает состояние команд, которые видны на экране.

STEP 7-Micro/WIN собирает информацию о состоянии из S7-200, начиная с первого оператора STL на верхнем краю окна редактора. По мере листания окна редактора вниз из S7-200 собирается новая информация.

STEP 7-Micro/WIN постоянно обновляет значения на экране. Для прекращения обновления выделите кнопку Triggered Pause [Пауза активизирована]. Текущие данные останутся на экране, пока вы не отмените выделение кнопки Triggered Pause.

Настройка параметров, отображаемых в программе на STL

STEP 7-Micro/WIN позволяет отображать состояние ряда параметров для команд STL. Выберите команду меню **Tools > Options [Инструментальные средства > Варианты]**, выберите редактор программ (Program Editor) и щелкните на закладке STL Status [Состояние STL]. См. рис. 8-3.

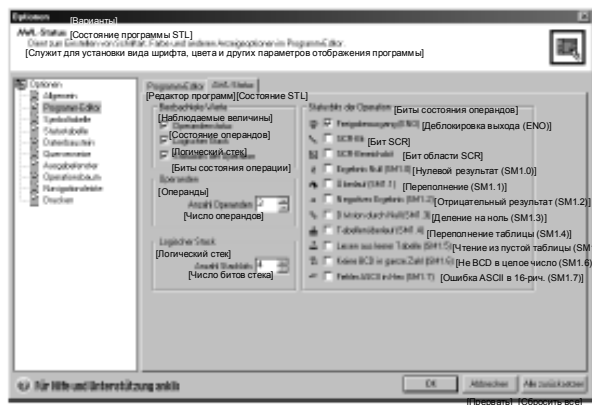


Рис. 8-3. Возможности отображения состояния STL

Использование таблицы состояний для контроля и изменения данных в S7-200

Таблица состояний дает возможность считывать, записывать, принудительно присваивать значения и наблюдать переменные, когда S7-200 исполняет вашу программу. Выберите команду меню **View > Component > Status Chart [Вид > Компонент > Таблица состояний]** для создания таблицы состояний. На рис. 8-4 показан пример таблицы состояний.

Вы можете создать несколько таблиц состояний.



Рис. 8-4. Таблица состояний

STEP 7-Micro/WIN имеет на панели инструментальных средств пиктограммы для работы с таблицей состояний: Sort Ascending [Расположить по возрастанию], Sort Descending [Расположить по убыванию], Single Read [Однократное чтение], Write All [Записать все], Force [Принудительно присвоить значение], Unforce [Отменить принудительное присваивание], Unforce All [Отменить все принудительные присваивания] и Read All Forced [Прочитать все принудительно присвоенные значения].

Чтобы выбрать формат для ячейки, выберите ячейку и щелкните правой кнопкой мыши для отображения контекстного меню.

Принудительное присваивание значений

S7-200 позволяет принудительно присваивать значения отдельным или всем входам и выходам (битам I и Q). Кроме того, вы можете принудительно присвоить значения 16 битам памяти (V или M) или аналоговым входам или выходам (AI или AQ). Значения в памяти переменных и значения в битовой памяти могут принудительно присваиваться байтами, словами или двойными словами. Аналоговые значения принудительно присваиваются только двойными словами по границам байтов с четными адресами, например, AIW6 или AQW14. Все принудительно присвоенные значения хранятся в устойчивом к потере питания памяти модуля S7-200.

Так как принудительно присвоенные значения данных могут изменяться в течение цикла (программой, актуализацией входов и выходов или в результате обработки коммуникаций), S7-200 снова записывает принудительно присвоенные значения в различные моменты времени внутри цикла.

- *Чтение входов:* S7-200 применяет принудительно присвоенные значения к входам при их считывании.
- *Исполнение логики управления в программе:* S7-200 применяет принудительно присвоенные значения ко всем входам и выходам с непосредственным доступом. Принудительно присвоенные значения применяются не более чем к 16 значениям в памяти после исполнения программы.
- *Обработка запросов на обмен данными:* S7-200 применяет принудительно присвоенные значения при всех обращениях на чтение или запись при обмене данными.
- *Запись в выходы:* S7-200 применяет принудительно присвоенные значения к выходам при их записи.

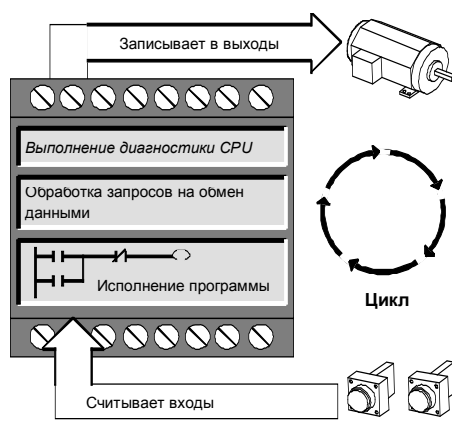


Рис. 8-5. Цикл S7-200

Для принудительного присваивания значений можно использовать таблицу состояний. Для принудительного присваивания нового значения введите это значение в столбец New Value [Новое значение] таблицы состояний, затем нажмите кнопку Force [Принудительно присвоить] на панели инструментов. Для принудительного присваивания существующего значения выделите это значение в столбце Current Value [Текущее значение], затем нажмите кнопку Force.



Совет

Функция принудительного присваивания значений перекрывает команду непосредственного чтения или записи. Функция принудительного присваивания значений перекрывает также таблицу выходов, которая была сконфигурирована для перехода в STOP. Если S7-200 переходит в STOP, то на выходе появляется принудительно присвоенное значение, а не значение, сконфигурированное в таблице выходов.

Исполнение программы в течение определенного количества циклов

Чтобы помочь вам в отладке программы, STEP 7-Micro/WIN дает возможность прогонять программу в течение определенного количества циклов.

Вы можете заставить S7-200 выполнить только первый цикл. Это позволяет наблюдать данные в S7-200 после первого цикла. Для прогона первого цикла выберите команду меню **Debug > First Scan [Отладка > Первый цикл]**.

Вы можете заставить S7-200 выполнять вашу программу в течение определенного количества циклов (от 1 до 65 535 циклов). Это позволяет наблюдать программу, когда она изменяет переменные. Чтобы указать количество циклов, подлежащих выполнению, выберите команду меню **Debug > Multiple Scans [Отладка > Несколько циклов]**.

Указания по устранению неисправностей аппаратуры

Таблица 8–1. Указания по устранению неисправностей аппаратуры для S7–200

Симптом	Возможные причины	Возможное решение
Прекращение работы выходов	<ul style="list-style-type: none"> ■ Управляемое устройство вызвало бросок напряжения, который повредил выход ■ Ошибка в программе пользователя ■ Плохо или неправильно подключена проводка ■ Чрезмерная нагрузка ■ Принудительный режим работы выхода 	<ul style="list-style-type: none"> ■ При подключении к индуктивной нагрузке (например, к двигателю или реле) необходимо использовать соответствующую гасящую цепочку. См. главу 3. ■ Исправьте программу пользователя ■ Проверьте и исправьте проводку ■ Проверьте нагрузку относительно номинала для выходов ■ Проверьте S7–200 на наличие принудительного режима для входов/выходов
На S7–200 загорелся сигнал SF (системная ошибка)	<p>В следующем списке приведены наиболее часто встречающиеся коды ошибок и их причины:</p> <ul style="list-style-type: none"> ■ Ошибка программирования пользователя <ul style="list-style-type: none"> - 0003 превышение контроля времени - 0011 косвенная адресация - 0012 недопустимое значение для числа с плавающей точкой - 0014 Ошибка области ■ Электрические помехи (от 0001 до 0009) ■ Повреждение компонентов (от 0001 до 0010) 	<p>Прочитайте код фатальной ошибки и обратитесь к Приложению С за информацией о типе ошибки:</p> <ul style="list-style-type: none"> ■ В случае ошибок программирования проверьте использование команд FOR, NEXT, JMP, LBL и сравнения. ■ В случае электрических помех: <ul style="list-style-type: none"> - Обратитесь к указаниям по монтажу электропроводки в главе 3. Очень важно, чтобы панель управления была хорошо заземлена и чтобы высоковольтная проводка не шла параллельно с низковольтной проводкой. - Подключите клемму M в источнике питания датчиков 24 В постоянного тока к заземлению.
Не включается ни один из светодиодов	<ul style="list-style-type: none"> ■ Сгорел предохранитель ■ Перепутана полярность проводов 24 В ■ Неправильное напряжение 	<p>Подключите к системе измерительный прибор, чтобы проверить величину и длительность пиков перенапряжения. С помощью этой информации добавьте к своей системе разрядник подходящего типа. Подробную информацию о монтаже полевой проводки вы найдете в главе 3.</p>
Неустойчивая работа, связанная с силовыми устройствами	<ul style="list-style-type: none"> ■ Плохое заземление ■ Прокладка проводки внутри шкафа управления ■ Слишком малое время задержки для входных фильтров 	<p>Обратитесь к указаниям по монтажу электропроводки в главе 3. Очень важно, чтобы панель управления была хорошо заземлена и чтобы высоковольтная проводка не шла параллельно с низковольтной проводкой. Подключите клемму M в источнике питания датчиков 24 В постоянного тока к заземлению. Увеличьте задержку входного фильтра в блоке системных данных.</p>

Таблица 8–1. Указания по устранению неисправностей аппаратуры для S7–200, продолжение

<p>Сеть связи повреждена при подключении к внешнему устройству. Поврежден или порт на компьютере, или порт на S7–200, или кабель PC/PPI.</p>	<p>Кабель связи может давать путь для протекания нежелательных токов, если все устройства, не имеющие гальванической развязки, напр., ПЛК, компьютеры или другие устройства, которые подключены к сети, не используют один и тот же опорный потенциал. Нежелательные токи могут вызвать ошибки связи и повреждения в цепях.</p>	<ul style="list-style-type: none"> ■ Обратитесь к указаниям по монтажу электропроводки в главе 3 и к руководству по сетям в главе 7. ■ Приобретите кабель PC/PPI, имеющий гальваническую развязку. ■ Приобретите повторитель RS–485/RS–485 с гальванической развязкой, если вы соединяете установки, не имеющие общей электрической опорной точки. <p>За номерами для заказа оборудования S7–200 обратитесь к Приложению E.</p>
<p>Другие коммуникационные проблемы (STEP 7-Micro/WIN)</p>	<p>Обратитесь к главе 7 за информацией об обмене данными в сети.</p>	
<p>Обработка ошибок</p>	<p>Обратитесь к Приложению C за информацией о кодах ошибок.</p>	

Управление перемещением без обратной связи с помощью S7-200

9

S7–200 предоставляет три метода управления перемещением без обратной связи:

- Широтно-импульсная модуляция (ШИМ, PWM) – встроена в S7–200 и служит для управления скоростью вращения, положением и относительной длительностью импульсов
- Вывод последовательности импульсов (PTO) – встроена в S7–200 и служит для управления скоростью вращения и позиционирования
- Модуль позиционирования EM 253 – дополнительный модуль для управления скоростью вращения и позиционирования

Для упрощения управления позиционированием в ваших приложениях STEP 7-Micro/WIN предоставляет в ваше распоряжение Мастер управления позиционированием (Position Control wizard), с помощью которого можно за несколько минут полностью сконфигурировать PWM, PTO или модуль позиционирования. Мастер генерирует команды позиционирования, с помощью которых вы можете динамически управлять скоростью вращения или положением в своем приложении. Для модуля позиционирования STEP 7-Micro/WIN снабжен также панелью управления, которая позволяет вам управлять, контролировать и тестировать процессы управления позиционированием.

В этой главе

Обзор	254
Использование выхода с широтно-импульсной модуляцией (ШИМ)	255
Основы управления перемещением без обратной связи с помощью шаговых двигателей или сервомоторов	257
Команды, создаваемые Мастером управления позиционированием	262
Коды ошибок для команд PTO	266
Функции модуля позиционирования	267
Настройка модуля позиционирования	269
Команды, создаваемые Мастером управления позиционированием для модуля позиционирования	275
Примеры программ для модуля позиционирования	287
Наблюдение за модулем позиционирования с помощью панели управления EM 253	292
Коды ошибок для модуля и команд позиционирования	294
Для опытных пользователей	296
Режимы поиска опорной точки, поддерживаемые модулем позиционирования	305

Обзор

S7–200 предоставляет три метода управления перемещением без обратной связи:

- широтно-импульсная модуляция (ШИМ, PWM) – встроена в S7–200 и служит для управления скоростью вращения, положением или относительной длительностью импульсов
- Вывод последовательности импульсов (PTO) – встроена в S7–200 и служит для управления скоростью вращения и позиционирования
- Модуль позиционирования EM 253 – дополнительный модуль для управления скоростью вращения и позиционирования



S7–200 имеет два цифровых выхода (Q0.0 и Q0.1), которые могут быть настроены с помощью Мастера управления позиционированием на использование в качестве PWM- или PTO-выходов. Мастер управления позиционированием может также использоваться для настройки модуля позиционирования EM 253.

Если выход настроен на режим PWM (ШИМ), то время цикла выхода (период следования выводимых импульсов) фиксировано, а ширина импульсов или их относительная длительность изменяются вашей программой. Изменения ширины импульсов может быть использовано для управления скоростью вращения или позиционированием в вашем приложении.

Если выход настроен на режим PTO, то для управления без обратной связи скоростью вращения и позиционированием для шаговых двигателей или серводвигателей генерируется последовательность импульсов с относительной длительностью 50%. Встроенная функция PTO поставляет только последовательность импульсов. Управление направлением перемещения и граничные значения должны обеспечиваться вашей прикладной программой с помощью входов и выходов ПЛК или модулями расширения.

Модуль позиционирования EM 253 поставляет отдельную последовательность импульсов со встроенным управлением направлением перемещения и выходами блокировки и сброса. Он также содержит специальные входы, которые позволяют настраивать модуль на различные режимы работы, включая автоматический поиск опорной точки. Этот модуль предлагает унифицированное решение для управления без обратной связи скоростью вращения и позиционированием для шаговых двигателей и серводвигателей.

Для упрощения управления позиционированием в вашем приложении STEP 7-Micro/WIN предоставляет в ваше распоряжение Мастер управления позиционированием, который дает вам возможность за несколько минут полностью настроить PWM, PTO или модуль позиционирования. Этот Мастер генерирует команды позиционирования, которые вы можете использовать для обеспечения динамического управления скоростью вращения и позиционированием в вашем приложении. Для модуля позиционирования STEP 7-Micro/WIN предоставляет также панель управления, которая дает вам возможность управлять, контролировать и тестировать ваши процессы позиционирования.

Использование выхода с широтно-импульсной модуляцией (ШИМ, PWM)

Функция «Широтно-импульсная модуляция (ШИМ, PWM)» предоставляет в ваше распоряжение выход с фиксированным периодом следования импульсов и переменной относительной длительностью включения. Выход PWM после запуска работает непрерывно с заданной частотой (периодом следования импульсов). Ширина импульсов меняется по потребности, чтобы добиться желаемого управления. Длительность импульсов может быть выражена в процентах от периода следования импульсов или как значение времени, соответствующее ширине импульса. Ширина импульсов может меняться от 0% (нет импульсов, постоянно выключено) до 100% (нет импульсов, постоянно включено). См. рис. 9–1.

Так как выход ШИМ может находиться в пределах от 0% до 100%, он представляет собой цифровой выход, который во многих отношениях похож на аналоговый выход. Например, выход ШИМ может быть использован для управления скоростью вращения двигателя от состояния покоя до максимальной скорости или для управления положением клапана от закрытого до полностью открытого.

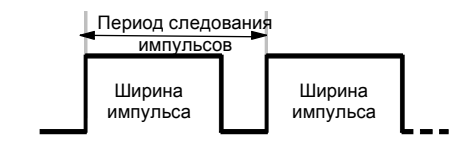


Рис. 9–1. Широтно-импульсная модуляция (ШИМ, PWM)

Настройка выхода ШИМ

Для настройки одного из встроенных выходов для управления с помощью ШИМ воспользуйтесь в STEP 7-Micro/WIN Мастером позиционирования. Для запуска Мастера позиционирования щелкните на символе Tools [Инструментальные средства] в навигационной панели, а затем дважды щелкните на символе Мастера позиционирования, или выберите команду меню **Tools > Position Control Wizard [Инструментальные средства > Мастер позиционирования]**. См. рис. 9–2.

1. Выберите опцию для настройки встроенной функции PTO/PWM ПЛК S7–200.
2. Выберите выход Q0.0 или Q0.1, который вы хотите настроить как выход PWM (ШИМ).
3. Затем выберите в раскрывающемся диалоговом окне Pulse Width Modulation (PWM) [Широтно-импульсная модуляция (ШИМ)], выберите в качестве базы времени (time base) микросекунды или миллисекунды и укажите период следования импульсов (cycle time).
4. Для завершения работы Мастера выберите Finish [Закончить].

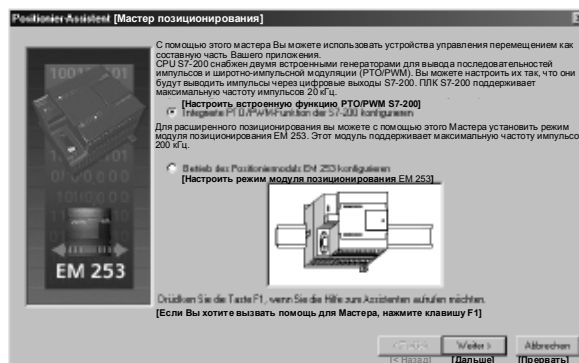


Рис. 9–2. Настройка выхода ШИМ

Мастер сгенерирует команду, с помощью которой вы можете управлять относительной длительностью импульсов на выходе ШИМ.

Команда PWMx_RUN

Команда PWMx_RUN дает вам возможность управлять относительной длительностью импульсов на выходе путем изменения ширины импульса от 0 до ширины, равной периоду следования импульсов.

Вход Cycle [Цикл] – это величина, имеющая размер слова, которая определяет период следования импульсов для выхода ШИМ. Допустимый диапазон составляет от 2 до 65535 единиц базы времени (микросекунд или миллисекунд), которая была определена во время работы Мастера.

Вход Pulse [Импульс] – это величина, имеющая размер слова, которая определяет ширину импульса для выхода ШИМ. Допустимый диапазон значений составляет от 0.0 до 65535 единиц базы времени (микросекунд или миллисекунд), которая была определена во время работы Мастера.

Error [Ошибка] – это величина, имеющая размер байта, которая возвращается командой PWMx_RUN и указывает на результат выполнения операции. Описание возможных кодов ошибок вы найдете в таблице.

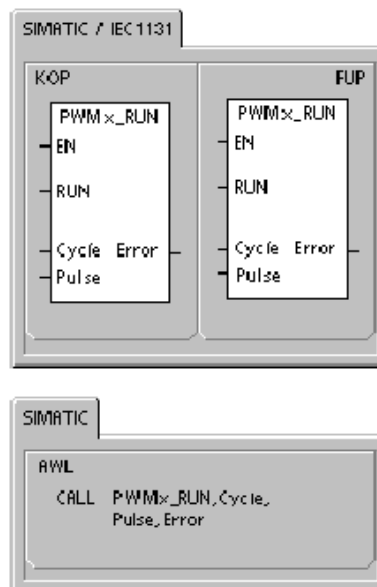


Таблица 9–1. Параметры команды PWMx_RUN

Входы/выходы	Типы данных	Операнды
Cycle, Pulse	Word	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, константа
Error	Byte	IB, QB, VB, MBV, SMB, LB, AC, *VD, *AC, *LD, константа

Таблица 9–2. Коды ошибок команды PWMx_RUN

Код ошибки	Описание
0	Нет ошибок, нормальное завершение
1	Выдана команда немедленного останова (STOP) во время движения. Команда STOP выполнена успешно

Основы управления перемещением без обратной связи с помощью шаговых двигателей или сервомоторов

И функция PTO, встроенная в ПЛК S7-200 PLC, и модуль позиционирования EM 253 используют последовательности импульсов для управления скоростью вращения и положением шагового двигателя или серводвигателя.

Использование функции PTO или модуля для управления перемещением без обратной связи требует опыта в области управления перемещением. Эта глава служит введением для неопытных пользователей. Она дает основную информацию, необходимую для использования Мастера управления позиционированием STEP 7-Micro/WIN для настройки PTO или модуля в вашем приложении.

Максимальная и стартстопная скорость

Мастер предложит вам ввести максимальную скорость вращения (MAX_SPEED) и стартстопную скорость вращения (SS_SPEED) для вашего приложения. См. рис. 9-3.

- MAX_SPEED: Введите значение для оптимальной рабочей скорости вращения вашего приложения в области возможных значений вращающего момента вашего двигателя. Вращающий момент, необходимый для приведения в движение нагрузки, определяется трением, инерцией и временами ускорения и замедления.
- Мастер управления позиционированием рассчитывает и отображает минимальную скорость вращения, которой может управлять модуль позиционирования, на основе указанной вами MAX_SPEED.
- Для выхода PTO вы должны указать желаемую стартстопную скорость вращения. Так как каждый раз, когда выполняется перемещение, генерируется по крайней мере один цикл стартстопной скорости вращения, то вы должны использовать стартстопную скорость вращения, период которой меньше времени ускорения и замедления.
- SS_SPEED: Введите значение в области, где двигатель в состоянии приводить в движение нагрузку с минимальными скоростями вращения. Если значение SS_SPEED слишком мало, то это может привести к колебаниям двигателя и нагрузки или к коротким скачкам в начале и в конце пути. Если значение SS_SPEED слишком велико, то двигатель может терять импульсы при запуске, а нагрузка может увлекать за собой двигатель при попытке останова.

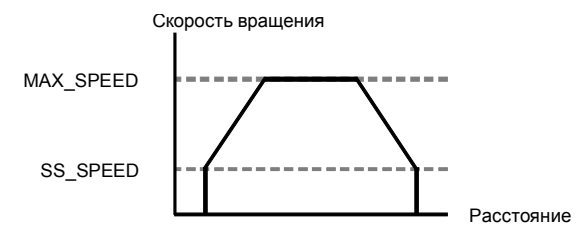


Рис. 9-3. Максимальная и стартстопная скорость

В спецификациях двигателей используются различные способы указания стартстопной скорости вращения двигателя при заданной нагрузке. Обычно значение SS_SPEED составляет от 5% до 15% от значения MAX_SPEED. Чтобы выбрать правильные скорости вращения для вашего приложения, обратитесь к спецификации вашего двигателя. На рис. 9-4 показана типичная кривая вращающий момент/скорость вращения двигателя.

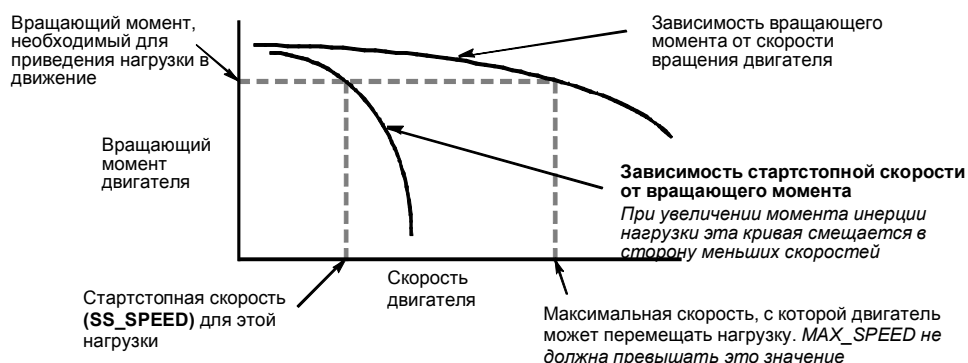


Рис. 9-4. Типовая характеристика вращающий момент/скорость вращения для двигателя

Ввод времен ускорения и замедления

Составной частью настройки является установка времен ускорения и замедления. По умолчанию эти времена устанавливаются равными 1 секунде. Обычно двигатели могут работать с временами, меньшими 1 секунды. См. рис. 9-5. Вы указываете следующие времена в миллисекундах:

- ACCEL_TIME: Время, необходимое для разгона двигателя от SS_SPEED до MAX_SPEED.
По умолчанию = 1000 мс
- DECEL_TIME: Время, необходимое для торможения двигателя от MAX_SPEED до SS_SPEED.
По умолчанию = 1000 мс

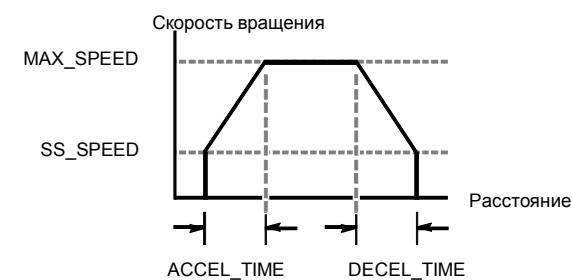


Рис. 9-5. Времена ускорения и замедления



Совет

Времена ускорения и замедления определяются методом проб и ошибок. Следует начинать с ввода большого значения. Оптимизируйте эти настройки для своего приложения, постепенно снижая эти времена, пока двигатель не начнет самопроизвольно останавливаться.

Настройка профилей движения

Профиль – это заранее определенное описание движения, состоящего из одной или более скоростей перемещения и приводящего к изменению положения от начальной до конечной точки. Вам нет необходимости определять профиль для использования функции РТО или модуля. Мастер управления позиционированием предоставляет в ваше распоряжение команды для управления перемещением, не требующие исполнения профиля.

Профиль программируется шагами, состоящими из ускорения и замедления, вплоть до целевой скорости вращения, вслед за которыми следует фиксированное количество импульсов при целевой скорости вращения. При перемещениях отдельными шагами или на последнем шаге движения имеет место также замедление от целевой скорости (последней целевой скорости) до остановки.

Функция РТО поддерживает до 100 профилей, тогда как модуль поддерживает не более 25 профилей.

Определение профиля движения

Мастер управления позиционированием ведет вас через определение профиля движения, где вы определяете все профили движения для своего приложения. Для каждого профиля вы выбираете рабочий режим и определяете отдельные шаги профиля. Мастер управления позиционированием позволяет также определить символическое имя для каждого профиля, просто вводя это символическое имя при определении профиля.

Выбор режима работы для профиля

Профиль настраивается в соответствии с желаемым режимом работы. РТО поддерживает относительное позиционирование и непрерывную работу с постоянной скоростью вращения. Модуль позиционирования поддерживает абсолютное позиционирование, относительное позиционирование, непрерывную работу с постоянной скоростью вращения и непрерывную работу с двумя скоростями вращения. На рис. 9–6 показаны различные режимы работы.

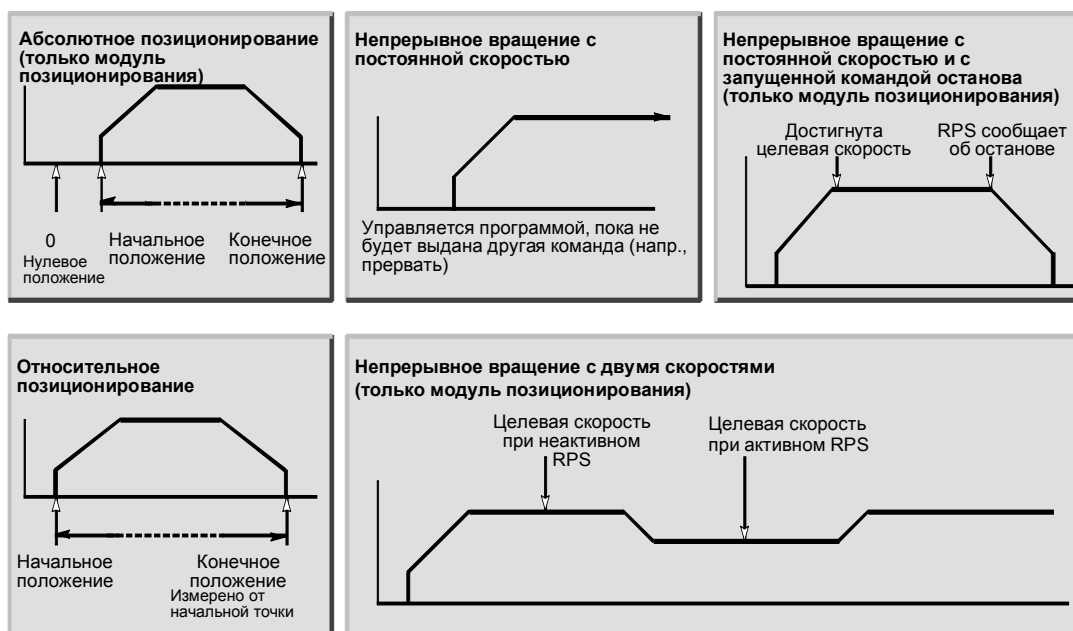


Рис. 9–6. Выбор режима для модуля позиционирования

Формирование шагов для профиля

Шаг – это фиксированное расстояние, на которое перемещается инструмент, включая расстояние, покрываемое за время ускорения и замедления. РТО поддерживает максимум 29 шагов на каждый профиль. Модуль поддерживает до 4 шагов в каждом профиле.

Вы определяете целевую скорость и конечное положение или количество импульсов для каждого шага. Дополнительные шаги вводятся одновременно. На рис. 9–7 показаны профили, состоящие из одного, двух, трех и четырех шагов.

Обратите внимание, что одношаговый профиль содержит один сегмент с постоянной скоростью вращения, профиль из двух шагов содержит два сегмента с постоянной скоростью вращения и т.д. Количество шагов в профиле совпадает с количеством сегментов с постоянной скоростью вращения.

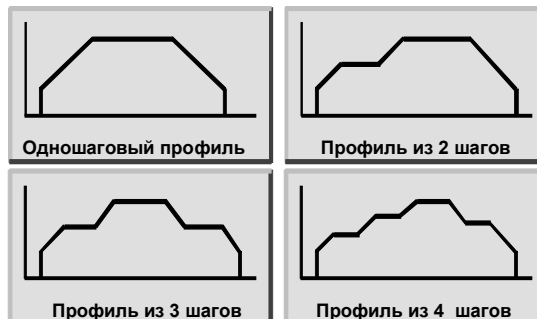


Рис. 9–7. Примеры профилей перемещения

Использование выхода РТО

РТО предоставляет в распоряжение выходной сигнал, состоящий из определенного количества прямоугольных импульсов (с относительной длительностью импульсов 50%). Частота или время цикла для каждого импульса линейно меняется при ускорении и замедлении и остается постоянной на участках с постоянной скоростью перемещения. Как только сгенерировано заданное количество импульсов, выход РТО выключается, и новые импульсы не генерируются, пока не будет загружена новая спецификация. См. рис. 9–8.

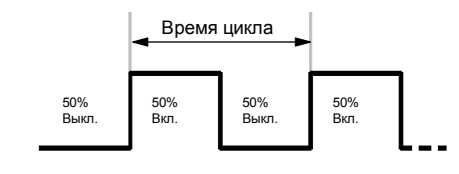


Рис. 9–8. Вывод последовательности импульсов (РТО)

Настройка выхода PTO

Для настройки одного из встроенных выходов для управления PTO используйте Мастер управления позиционированием STEP 7-Micro/WIN. Для запуска мастера управления позиционированием щелкните на символе Tools [Инструментальные средства] на навигационной панели, а затем дважды щелкните на символе Position Control Wizard [Мастер управления позиционированием], или выберите команду меню **Tools > Position Control Wizard [Инструментальные средства > Мастер управления позиционированием]**.

1. Выберите опцию для настройки встроенной функции PTO/PWM ПЛК S7-200.
2. Выберите выход Q0.0 или Q0.1, который вы хотите настроить как выход PTO.
3. В раскрывающемся диалоговом окне выберите Linear Pulse Train Output (PTO) [Линейная последовательность импульсов (PTO)].
4. Если вы хотите контролировать количество импульсов, сгенерированных PTO, активизируйте триггерную кнопку High Speed Counter [Скоростной счетчик].
5. Введите MAX_SPEED и SS_SPEED в предназначенных для этого полях.
6. Введите в предназначенных для этого полях времена ускорения и замедления.
7. В диалоговом окне для определения профилей перемещения щелкните на кнопке New profile [Новый профиль] для активизации определения профилей. Выберите желаемый режим работы.

Для профиля относительного позиционирования:

Введите целевую скорость и количество импульсов. Затем вы можете щелкнуть на кнопке Plot step [Изобразить шаг], чтобы увидеть графическое представление перемещения.

Если необходимо выполнить более одного шага, выберите кнопку New step [Новый шаг] и введите необходимую информацию о шаге.

Для непрерывного перемещения с постоянной скоростью вращения:

Введите в поле редактирования значение скорости вращения.

Если вы хотите завершить перемещение с постоянной скоростью вращения, активизируйте триггерную кнопку Program a Subroutine [Запрограммировать подпрограмму] и введите количество импульсов для перемещения после события, вызывающего останов.

8. Определите столько профилей и шагов, сколько необходимо для реализации желаемого перемещения
9. Выберите Finish [Завершить] для завершения работы Мастера.

Команды, создаваемые Мастером управления позиционированием

Мастер управления позиционированием облегчает управление встроенным PTO путем создания пяти однозначных подпрограмм-команд. Все команды позиционирования имеют префикс «PTOx_», где x – это положение модуля.

Подпрограмма PTOx_CTRL

Подпрограмма PTOx_CTRL (управление) активизирует и инициализирует выход PTO для использования с шаговым двигателем или с серводвигателем. Используйте эту подпрограмму в своей программе только один раз и обеспечьте, чтобы она выполнялась в каждом цикле. В качестве входа EN всегда используйте SMO.0.

Вход I_STOP (немедленный останов) представляет собой булев вход. При низком уровне сигнала на этом входе функция PTO работает нормально. При появлении высокого уровня сигнала на этом входе PTO немедленно прекращает вывод импульсов.

Вход D_STOP (останов с замедлением) представляет собой булев вход. При низком уровне сигнала на этом входе функция PTO работает нормально. При появлении высокого уровня сигнала на этом входе PTO генерирует последовательность импульсов, которая замедляет вращение двигателя до остановки.

Выход Done [Готово] представляет собой булев выход. Если бит Done установлен, это указывает, что CPU выполнил подпрограмму.

Если бит Done установлен, то байт ошибок (Error) сообщает о нормальном завершении без ошибок или с кодом ошибки. Коды ошибок вы найдете в таблице 9–7.

Параметр C_Pos содержит текущее положение модуля в виде количества импульсов, если в Мастере был активизирован скоростной счетчик HSC. В противном случае положение модуля всегда равно 0.

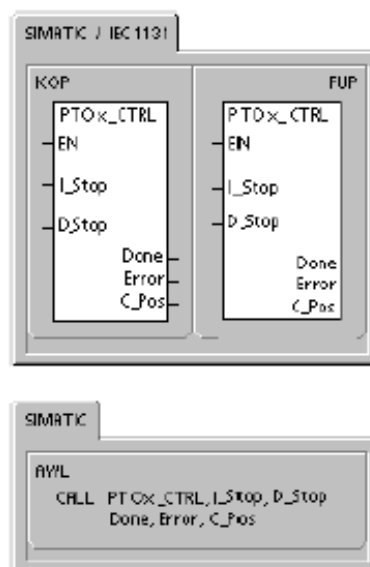


Таблица 9–3. Параметры команды PTOx_CTRL

Входы/выходы	Типы данных	Операнды
I_STOP	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
D_STOP	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Подпрограмма PTOx_RUN

Подпрограмма PTOx_RUN (исполнять профиль) предписывает ПЛК исполнить команду перемещения в определенном профиле, хранящемся в таблице конфигураций/профилей.

Установка бита EN активизирует подпрограмму. Обеспечьте, чтобы бит EN оставался установленным до тех пор, пока бит Done не сообщит, что исполнение подпрограммы завершено.

Установка параметра START инициализирует исполнение профиля. В каждом цикле, в котором параметр START включен, а функция PTO еще не активна, эта команда активизирует PTO. Чтобы гарантировать посылку только одной команды на исполнение профиля, параметр START следует включать путем распознавания фронта.

Параметр Profile [Профиль] содержит номер или символическое имя профиля перемещения.

Включение параметра Abort [Завершить] предписывает модулю позиционирования завершить текущий профиль и выполнить замедление, пока двигатель не остановится.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error содержит результат выполнения команды. Определение кодов ошибок вы найдете в таблице 9–7.

Параметр C_Profile содержит профиль, выполняемый в данный момент модулем позиционирования.

Параметр C_Step содержит выполняемый в данный момент шаг профиля.

Параметр C_Pos содержит текущее положение модуля в виде количества импульсов, если в Мастере был активизирован скоростной счетчик HSC. В противном случае текущее положение всегда равно 0.

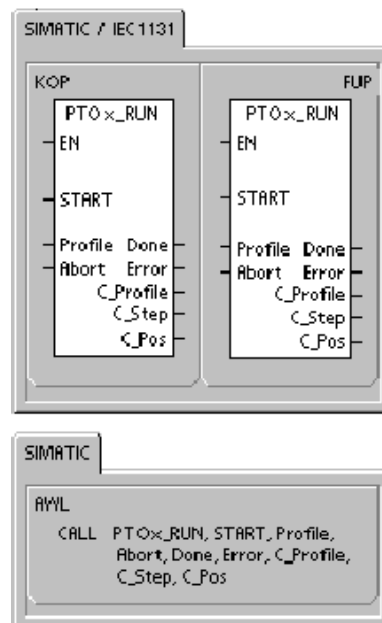


Таблица 9–4. Параметры команды PTOx_RUN

Входы/выходы	Типы данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Profile	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, константа
Abort, Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error, C_Profile, C_Step	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Подпрограмма PTOx_MAN

Подпрограмма PTOx_MAN (ручной режим) переводит выход PTO в ручной режим. Это позволяет запускать, останавливать двигатель и работать с ним на различных скоростях. Когда подпрограмма PTOx_MAN активизирована, никакая другая подпрограмма PTO не может исполняться.

Активизация параметра RUN (работа/останов) предписывает PTO ускоряться до заданной скорости (параметр Speed [Скорость]). Значение параметра Speed можно изменять во время работы двигателя. Деактивизация параметра RUN предписывает PTO замедляться до остановки двигателя.

Параметр Speed определяет скорость вращения, когда параметр RUN активизирован. Скорость вращения имеет тип данных DINT для импульсов в секунду. Этот параметр можно изменять, когда двигатель работает.

Параметр Error [Ошибка] содержит результат выполнения команды. Определение кодов ошибок вы найдете в таблице 9–7.

Параметр C_Pos содержит текущее положение модуля в виде количества импульсов, если в Мастере был активизирован скоростной счетчик HSC. В противном случае текущее положение всегда равно 0.

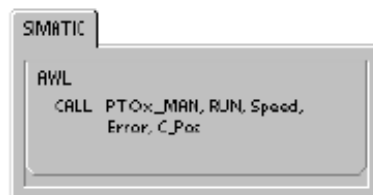
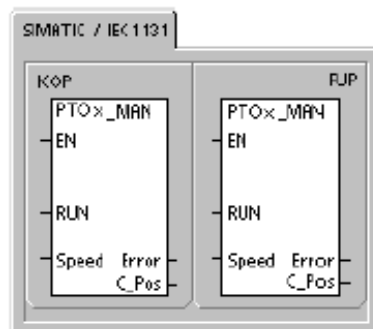


Таблица 9–5. Параметры команды PTOx_MAN

Входы/выходы	Типы данных	Операнды
RUN	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
SPEED	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, константа
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Совет

PTO может не среагировать на малые изменения параметра Speed, особенно, если запроецированное время ускорения или замедления невелико, а разница между установленной максимальной и стартовой скоростью велика.

Команда PTOx_LDPOS

Команда PTOx_LDPOS (загрузить позицию) заменяет текущее значение состояния счетчика импульсов PTO новым значением. Эту команду можно использовать также для установки нулевого положения для команды перемещения.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался установленным, пока бит Done [Готово] не сообщит, что исполнение команды завершено.

Включение параметра START загружает новую позицию в счетчик импульсов PTO. В каждом цикле, в котором параметр START включен, а функция PTO в это время не занята, эта команда загружает новую позицию в счетчик импульсов PTO. Чтобы гарантировать подачу только одной команды, для включения параметра START необходимо использовать распознавание фронта.

Параметр New_Pos поставляет новое значение для замены текущего значения положения, которое было сообщено. Значение положения выражается количеством импульсов.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определение кодов ошибок вы найдете в таблице 9–7.

Параметр C_Pos содержит текущее положение модуля, выраженное количество импульсов, если в Мастере был активизирован скоростной счетчик HSC. В противном случае текущее положение всегда равно 0.

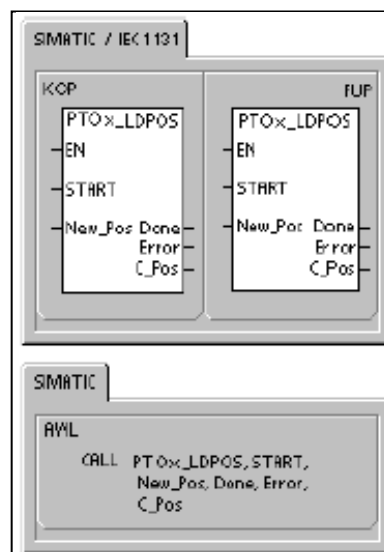
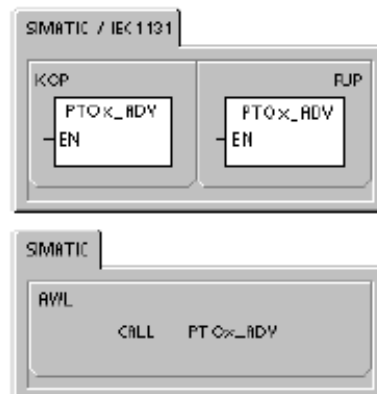


Таблица 9–6. Параметры команды PTOx_LDPOS

Входы/выходы	Типы данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
New_Pos, C_Pos	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Подпрограмма PTOx_ADV

Подпрограмма PTOx_ADV останавливает выполнение текущего профиля непрерывного перемещения и увеличивает количество импульсов, указанное при определении профиля с помощью Мастера. Эта подпрограмма создается, если вы задали по крайней мере одно непрерывное перемещение с постоянной скоростью вращения при активизированной опции PTOx_ADV в Мастере управления позиционированием.



Коды ошибок для команд PTO

Таблица 9–7. Коды ошибок для команд PTO

Код ошибки	Описание
0	Нет ошибок, нормальное завершение
1	Выдана команда немедленного останова (STOP) во время движения. Команда STOP выполнена успешно
2	Команда останова с замедлением, выданная во время перемещения. Команда STOP выполнена успешно
3	Ошибка исполнения, обнаруженная в генераторе импульсов или в формате таблицы PTO
127	Команда HSC, PLS или PTO во время исполнения привела к ошибке ENO
128	Этот запрос не может быть обработан. Или CPU занят другим запросом, или при этом запросе не было импульса START.
129	Одновременно активизированы команды немедленного останова и останова с замедлением
130	Команде PTO в настоящий момент дается указание остановиться
132	Для указанного профиля не сформирован профильный блок

Функции модуля позиционирования

Модуль позиционирования предоставляет функциональные возможности и мощности, необходимые для одноосного позиционирования без обратной связи:

- управление высокими скоростями в диапазоне от 20 до 200 000 импульсов в секунду
- поддержка бестолчковой (по S-образной кривой) и линейного ускорения и замедления
- настраиваемая измерительная система, позволяющая вводить данные как в инженерных единицах (например, дюймах или сантиметрах), так и в виде количества импульсов
- настраиваемая компенсация зазоров
- поддержка абсолютного, относительного и ручного методов управления позиционированием
- непрерывный режим работы
- до 25 профилей перемещения с возможностью до 4 изменений скорости на профиль
- четыре различных способа поиска опорной точки с выбором начального направления поиска и направления завершающего подхода для каждой последовательности
- съемные клеммные блоки для полевой проводки для упрощения монтажа и демонтажа

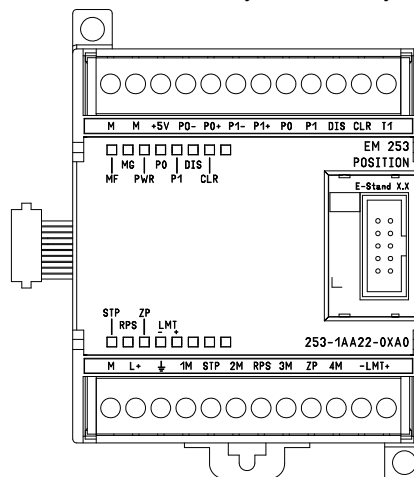


Рис. 9–9. Модуль позиционирования EM 253

Вся информация о настройке и профилях для модуля позиционирования создается в STEP 7-Micro/WIN. Эта информация загружается в S7-200 вместе с программными блоками. Так как вся информация, необходимая для управления позиционированием, хранится в S7-200, то модуль позиционирования можно заменять без необходимости нового программирования и настройки модуля.

S7-200 резервирует 8 битов регистра выходов образа процесса (область памяти Q) для интерфейса с модулем позиционирования. Ваша прикладная программа в S7-200 использует эти биты для управления режимом работы модуля позиционирования. Эти 8 выходных битов не связаны ни с одним из физических полевых выходов модуля позиционирования.

Модуль позиционирования обладает пятью цифровыми входами и четырьмя цифровыми выходами, которые обеспечивают интерфейс с вашим приложением. См. таблицу 9–8. Эти входы и выходы находятся непосредственно на модуле позиционирования. В приложении А вы найдете подробные технические данные модуля позиционирования, включая схемы его подключения к некоторым приводам и усилительным устройствам.


Таблица 9–8. Входы и выходы модуля позиционирования


Сигнал	Описание
STP	Вход STP заставляет модуль остановить текущее перемещение. Вы можете выбрать желаемый режим STP в Мастере управления позиционированием.
RPS	Вход RPS (Reference Point Switch – переключатель опорной точки) устанавливает опорную точку или исходное положение для операций абсолютного перемещения.
ZP	Вход ZP (Zero Pulse – нулевой импульс) помогает установить опорную точку или исходное положение. Обычно устройство управления двигателем или усилитель выдает один нулевой импульс на оборот двигателя.
LMT+ LMT-	Входы LMT+ и LMT- устанавливают границы перемещения. Мастер управления позиционированием дает возможность настроить режим входов LMT+ и LMT-.
P0 P1 P0+, P0- P1+, P1-	P0 и P1 – это импульсные выходы на транзисторах с открытым стоком (open drain), которые управляют движением и направлением вращения двигателя. P0+, P0- и P1+, P1- – это дифференциальные импульсные выходы, которые предоставляют такие же функции, как P0 и P1, соответственно, обеспечивая при этом более высокое качество сигнала. Выходы с активным стоком и дифференциальные выходы активны все одновременно. Выбор импульсных выходов производится в зависимости от требований интерфейса устройства управления двигателем или усилителя.
DIS	DIS – это выход на транзисторе с открытым стоком, используемый для деактивизации и активизации устройства управления двигателем или усилителя.
CLR	CLR – это выход на транзисторе с открытым стоком, используемый для стирания регистра счета служебных импульсов.

Программирование модуля позиционирования

STEP 7-Micro/WIN предоставляет удобные инструментальные средства для настройки и программирования модуля позиционирования. Действуйте следующим образом:

1. Настройте модуль позиционирования. STEP 7-Micro/WIN предоставляет в ваше распоряжение Мастер управления позиционированием для создания таблицы настройки/профилей и команд позиционирования. Информацию о настройке модуля позиционирования вы найдете на стр. 269 под соответствующим заголовком.
2. Протестируйте работу модуля позиционирования. STEP 7-Micro/WIN предоставляет в ваше распоряжение панель управления EM 253 для тестирования проводки, подключенной к входам и выходам, настройки модуля позиционирования и работы профилей перемещения. Информацию о панели управления EM 253 вы найдете на стр. 292.
3. Создайте программу для исполнения в S7-200. Мастер управления позиционированием автоматически создает команды позиционирования, которые вы вставляете в свою программу. Информацию о командах позиционирования вы найдете на стр. 275. Вставьте в свою программу следующие команды:
 - Для активизации модуля позиционирования вставьте команду POSx_CTRL. Чтобы обеспечить выполнение этой команды в каждом цикле, используйте SM0.0 (Постоянно включен).
 - Для приведения двигателя в заданное положение используйте команду POSx_GOTO или POSx_RUN. Команда POSx_GOTO перемещает в положение, указанное входами вашей программы. Команда POSx_RUN выполняет профили перемещения, спроектированные с помощью Мастера управления позиционированием.
 - Если вы хотите использовать для управления позиционированием абсолютные координаты, вы должны установить в своем приложении нулевое положение. Для установки нулевого положения используйте команду POSx_RSEEK или POSx_LDPOS.
 - Другие команды, создаваемые Мастером управления позиционированием, обеспечивают функциональные возможности для типовых приложений и не обязательны для вашего конкретного приложения.
4. Скомпилируйте свою программу и загрузите системный блок, блок данных и программный блок в S7-200.

	<p>Совет Дополнительную информацию о подключении модуля позиционирования к различным устройствам управления шаговыми двигателями вы найдете в Приложении А.</p>
---	--

	<p>Совет Для согласования с настройками, установленными в Мастере управления позиционированием по умолчанию, установите DIP-переключатели на устройстве управления шаговым двигателем на 10000 импульсов на оборот.</p>
---	--

Настройка модуля позиционирования



Управление
позиционированием

Чтобы модуль мог управлять позиционированием в вашем приложении, вы должны создать таблицу настройки и профилей. Мастер управления позиционированием выполняет процесс настройки быстро и легко, ведя вас шаг за шагом через этот процесс. Подробную информацию о таблице настройки и профилей вы найдете на стр. 296 по заголовком «Для опытных пользователей».

Мастер управления позиционированием позволяет создавать таблицу настройки и профилей также и в автономном режиме (offline). Вы можете выполнить настройку без соединения с CPU S7-200 с подключенным модулем позиционирования.

Для запуска Мастера управления позиционированием ваш проект должен быть скомпилирован и установлен в режим символической адресации.

Для вызова Мастера управления позиционированием щелкните на символе Tools [Инструментальные средства] в навигационной панели, а затем дважды щелкните на символе Position Control Wizard [Мастер управления позиционированием], или выберите команду меню **Tools > Position Control Wizard [Инструментальные средства > Мастер управления позиционированием]**.

Для настройки модуля позиционирования используйте Мастер управления позиционированием STEP 7-Micro/WIN. Выберите опцию для настройки модуля позиционирования EM 253.

Ввод положения модуля

Введите положение гнезда для модуля (от модуль 0 до модуль 6). Если STEP 7-Micro/WIN связан с ПЛК, то вам достаточно щелкнуть на кнопке Read Modules [Считать модули]. В CPU S7-200 с ПЗУ версии ниже, чем 1.2, модуль должен быть установлен сразу за CPU.

Выбор вида измерения

Выберите систему измерения. Вы можете выбрать инженерные единицы или импульсы. При выборе импульсов никакой другой информации не требуется. При выборе инженерных единиц необходимо ввести количество импульсов, необходимое для выполнения одного оборота двигателя (см. спецификацию своего двигателя или привода), базу единиц измерения (напр., дюймы, футы, сантиметры или миллиметры) и расстояние, проходимое за один оборот двигателя.

- STEP 7-Micro/WIN предоставляет в ваше распоряжение панель управления EM253, которая позволяет изменять количество единиц на оборот после настройки модуля позиционирования.
- Если вы изменяете систему измерения позже, то вы должны удалить всю настройку, включая все команды, сгенерированные Мастером управления позиционированием. Затем необходимо ввести данные, согласованные с новой системой измерения.

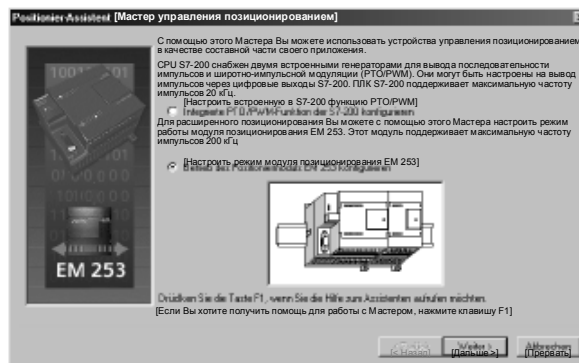


Рис. 9–10. Мастер управления позиционированием

Редактирование установленной по умолчанию настройки входов и выходов

Для изменения или отображения установленной по умолчанию настройки встроенных входов и выходов выберите кнопку **Advanced Options** [Дополнительные возможности].

- В закладке **Input Active Levels** [Уровни активности входов] выберите активный уровень (высокий или низкий). Если в качестве активного принят высокий уровень сигнала (High), то логическая 1 считывается, когда на входе имеет место поток сигнала. Если в качестве активного принят низкий уровень сигнала (Low), то логическая 1 считывается, когда поток сигнала на входе отсутствует. Уровень сигнала 1 всегда интерпретируется как условие активности. Светодиоды горят, когда на входе имеет место поток сигнала, независимо от уровня активности. (По умолчанию активным является высокий уровень)
- В закладке **Input Filter Times** [Времена входных фильтров] вы можете ввести постоянную времени фильтра (от 0,20 мс до 12,80 мс) для входов STP, RPS, LMT+ и LMT-. Увеличение постоянной времени фильтра лучше устраняет помехи, но это также увеличивает время реакции на изменение сигнала. (По умолчанию = 6,4 мс)
- В закладке **Pulse and Directional Outputs** [Импульсные выходы и выходы направления] выберите полярность выходов и установить метод управления направлением. Воздействия настроек полярности и выбора направления показаны на рисунках 9–11 и 9–12.

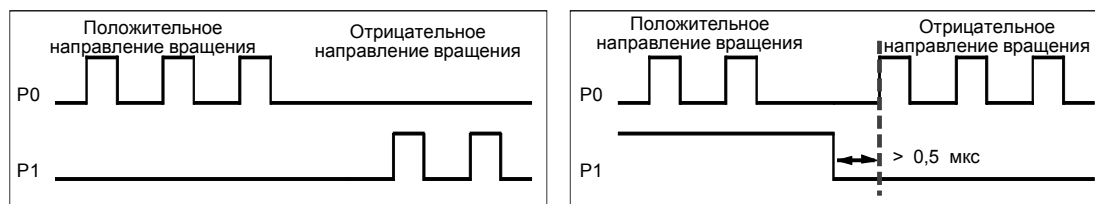


Рис. 9–11. Возможности вращения при положительной полярности

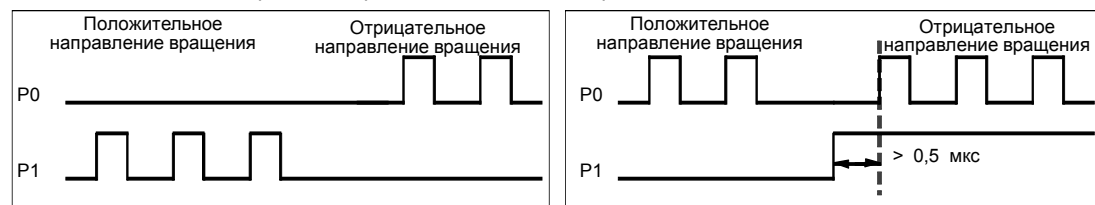


Рис. 9–12. Возможности вращения при отрицательной полярности

Предупреждение

Устройств управления могут выходить из строя в условиях, не обеспечивающих безопасность, что приводит к неконтролируемой работе управляемых устройств. Это может привести к гибели или серьезным травмам обслуживающего персонала, и/или выходу из строя оборудования.

Функции ограничения и останова в модуле позиционирования реализованы с помощью электронной логики, которая не обеспечивает такого же уровня защиты, как электромеханические управляющие устройства. Поэтому позаботьтесь о том, чтобы функция аварийного отключения была реализована с помощью электромеханических или резервных устройств защиты, независимых от модуля позиционирования и CPU S7-200.

Настройка реакции модуля на физические входы

Затем выберите реакцию модуля на входы LMT+, LMT- и STP. В разворачивающемся окне со списком выберите: **no action** [бездействие] (игнорировать условия на входе), **decelerate to a stop** [замедлить до остановки] (по умолчанию) или **immediate stop** [немедленный останов].

Ввод максимальной и стартстопной скорости вращения

Введите максимальную (MAX_SPEED) и стартстопную (SS_SPEED) скорость вращения для вашего приложения

Ввод параметров толчкового режима

Затем введите значения JOG_SPEED и JOG_INCREMENT.

- JOG_SPEED: JOG_SPEED (толчковая скорость вращения для двигателя) – это максимальная скорость вращения, которая может быть получена при активной команде JOG [толчок].
- JOG_INCREMENT: расстояние, на которое перемещается инструмент при получении кратковременной команды JOG.

На рис. 9–13 показан принцип действия команды Jog. Когда модуль позиционирования получает команду Jog, он запускает таймер. Если команда Jog завершается до истечения 0,5 секунды, то модуль позиционирования перемещает инструмент на величину, указанную в JOG_INCREMENT, со скоростью, определяемой параметром SS_SPEED. Если команда Jog по истечении 0,5 секунды еще активна, то модуль позиционирования ускоряется до JOG_SPEED. Движение продолжается до завершения команды Jog. Затем модуль позиционирования выполняет останов с замедлением. Команду Jog можно активизировать или из панели управления EM 253, или с помощью команды позиционирования.

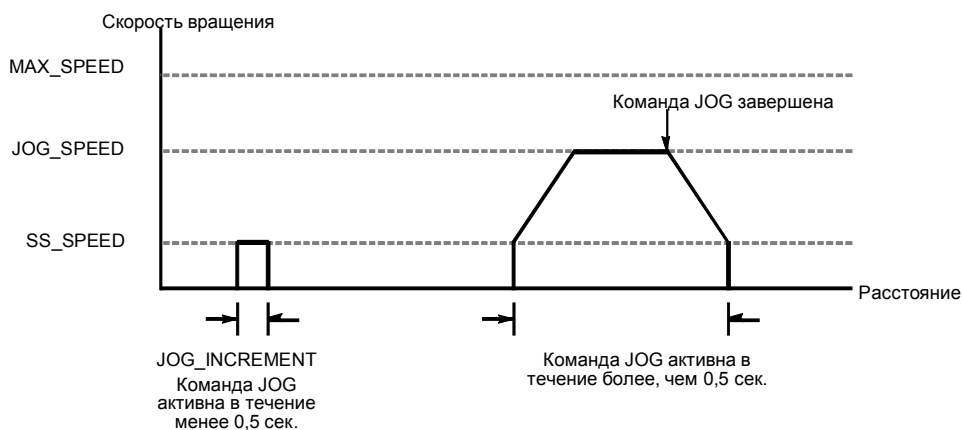


Рис. 9–13. Представление команды JOG

Ввод времени ускорения

Введите в окнах редактирования времена ускорения и замедления.

Ввод времени сглаживания толчка

Для одношаговых перемещений введите время сглаживания толчка. Сглаживание толчка приводит к бестолчковому управлению позиционированием путем уменьшения темпа изменения скорости на этапах ускорения и замедления профиля перемещения. См. рис. 9–14.

Время сглаживания толчка называют также «профилированием с помощью S-образной кривой». Сглаживание толчка производится равным образом в начале и в конце кривой ускорения и замедления. Сглаживание толчка не применяется к первому и к последнему шагу между нулевой скоростью и SS_SPEED.

Для сглаживания толчка вводится значение времени (JERK_TIME). Это время, необходимое, чтобы ускорение изменилось от нуля до максимального значения. Увеличение времени сглаживания толчка обеспечивает бестолчковый режим при меньшем увеличении общего времени цикла, чем этого бы удалось достичь увеличением ACCEL_TIME или DECEL_TIME. Значение ноль указывает, что сглаживание не производится.

(По умолчанию = 0 мс)

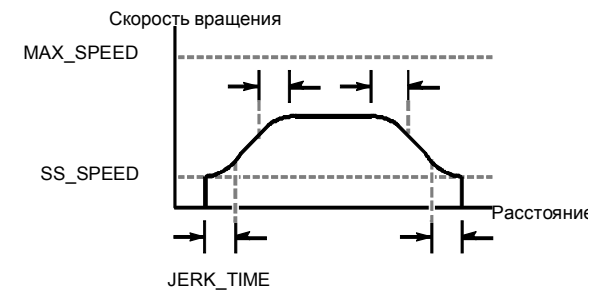


Рис. 9–14. Сглаживание толчка

	<p>Совет Хорошим начальным значением для JERK_TIME является 40% от ACCEL_TIME.</p>
--	---

Настройка опорной точки и параметров поиска

Выберите, хотите ли вы использовать в своем приложении опорную точку или нет.

- Если в вашем приложении перемещения должны отсчитываться от некоторой абсолютной позиции, то вы должны установить опорную точку (RP) или нулевое положение, которое фиксирует измерения положения относительно известной точки в физической системе.
- Если используется опорная точка, то должна существовать возможность ее автоматического обнаружения. Процесс автоматического обнаружения опорной точки называется поиском опорной точки. Определение поиска опорной точки требует выполнения двух шагов в Мастере.

Введите скорости поиска опорной точки (большую скорость поиска и малую скорость поиска). Определите начальное направление поиска и направление конечного приближения к опорной точке. С помощью кнопки Advanced RP Options [Дополнительные опции опорной точки] введите смещение опорной точки и значения для компенсации зазора.

RP_FAST – это начальная скорость, которую модуль использует при выполнении команды поиска опорной точки. Обычно значение RP_FAST равняется примерно 2/3 от значения MAX_SPEED.

RP_SLOW – это скорость конечного приближения к опорной точке. Для приближения к опорной точке используется малая скорость, чтобы не проскочить ее. Обычно значение RP_SLOW равно значению SS_SPEED.

RP_SEEK_DIR – это начальное направление для операции поиска опорной точки. Обычно это направление от рабочей зоны к окрестности опорной точки. Важную роль в определении области, где производится поиск опорной точки, играют конечные выключатели. Если при выполнении поиска опорной точки встречается конечный выключатель, то это может привести к изменению направления, чтобы поиск можно было продолжать. (По умолчанию = Negative [отрицательное])

RP_APPR_DIR – это направление конечного приближения к опорной точке. Для уменьшения зазора и получения большей точности приближение к опорной точке должно производиться в том же направлении, что и перемещение от опорной точки к рабочей зоне. (По умолчанию = Positive [положительное])

- Мастер управления позиционированием имеет дополнительные опции опорной точки, позволяющие указать смещение опорной точки (RP_OFFSET), которое представляет собой расстояние от опорной точки до нулевого положения. См. рис. 9–15.

RP_OFFSET: Расстояние от опорной точки до нулевой позиции физической измерительной системы. По умолчанию = 0

Компенсация зазора:

Расстояние, на которое двигатель должен осуществить перемещение, чтобы устранить зазор в системе после изменения направления.

Компенсация зазора – всегда положительная величина. По умолчанию = 0

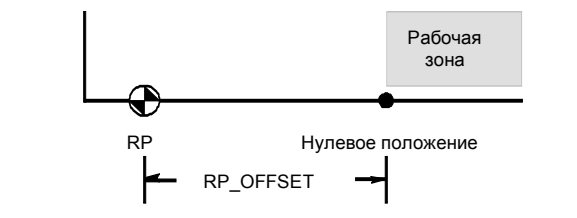


Рис. 9–15. Соотношение между опорной точкой и нулевым положением

Выберите последовательность поиска опорной точки.

- Модуль позиционирования имеет вход – переключатель опорной точки (RPS), который используется при поиске опорной точки. Опорная точка идентифицируется методом определения точного положения относительно RPS. Опорная точка может находиться в центре активной зоны RPS, на границе активной зоны RPS или на определенном расстоянии от активной зоны RPS, определяемым количеством фронтов нулевых импульсов (ZP) на входе.

Вы можете спроектировать последовательность, которую модуль позиционирования будет использовать для поиска опорной точки. На рис. 9–16 показана упрощенная диаграмма принятой по умолчанию последовательности поиска опорной точки. Вы можете выбрать следующие варианты последовательности поиска опорной точки:

Режим поиска опорной точки 0: Последовательность поиска опорной точки не выполняется

Режим поиска опорной точки 1: Опорная точка находится там, где вход RPS становится активным при приближении со стороны рабочей зоны. (По умолчанию)

Режим поиска опорной точки 2: Опорная точка находится в центре активной зоны входа RPS.

Режим поиска опорной точки 3: Опорная точка находится вне активной зоны входа RPS. RP_Z_CNT указывает, сколько нулевых импульсов ZP (Zero Pulse) должно быть отсчитано после деактивизации входа RPS.

Режим поиска опорной точки 4: Опорная точка находится внутри активной зоны входа RPS. RP_Z_CNT указывает, сколько нулевых импульсов ZP (Zero Pulse) должно быть отсчитано после активизации входа RPS.

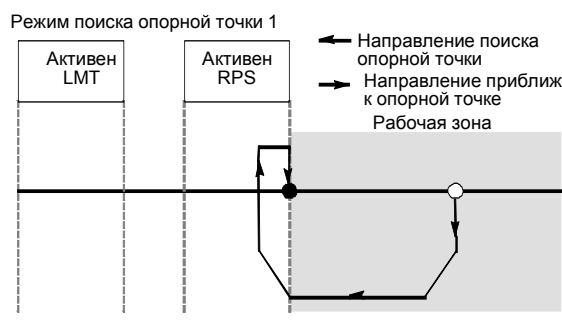


Рис. 9–16. Принятая по умолчанию последовательность поиска опорной точки (упрощено)

	<p>Совет</p> <p>Зона активности RPS (т.е. расстояние, на протяжении которого вход RPS остается активным) должна быть больше, чем расстояние, необходимое для замедления от скорости RP_FAST до скорости RP_SLOW. Если это расстояние слишком мало, то модуль позиционирования выдает ошибку.</p>
--	---

Командный байт

Затем введите байтовый адрес выхода для командного байта. Командный байт – это 8 цифровых выходов, зарезервированных в регистре образа процесса для интерфейса с модулем позиционирования. На рис. 4–10 в главе 4 вы найдете описание нумерации входов и выходов.

Определение профиля перемещения

В диалоговом окне для определения профиля перемещения щелкните на кнопке New profile [Новый профиль], чтобы активизировать определение профиля. Выберите желаемый режим работы.

- Для определения профиля абсолютного позиционирования действуйте следующим образом:
Введите целевую скорость вращения и конечное положение. Затем вы можете щелкнуть на кнопке Plot step [Отобразить шаг графически], чтобы увидеть графическое представление перемещения.
Если необходимо выполнить более одного шага, щелкните на кнопке New step [Новый шаг] и введите необходимую информацию о шаге.
- Для определения профиля относительного позиционирования действуйте следующим образом:
Введите целевую скорость вращения и конечное положение. Затем вы можете щелкнуть на кнопке Plot step [Отобразить шаг графически], чтобы увидеть графическое представление перемещения.
Если необходимо выполнить более одного шага, щелкните на кнопке New step [Новый шаг] и введите необходимую информацию о шаге.
- Для определения непрерывного перемещения с фиксированной скоростью вращения действуйте следующим образом:
Введите фиксированную скорость вращения в окне редактирования.
Выберите направление вращения
Если вы хотите завершить непрерывное перемещение с фиксированной скоростью вращения с помощью входа RPS, активизируйте соответствующую триггерную кнопку.
- Для определения непрерывного перемещения с двумя скоростями вращения действуйте следующим образом:
Введите в окне редактирования значение целевой скорости вращения при высоком уровне сигнал на входе RPS.
Введите в окне редактирования значение целевой скорости вращения при низком уровне сигнал на входе RPS.
Выберите направление вращения

Определите количество профилей и шагов, необходимое для выполнения желаемого перемещения.

Завершите настройку

Настроив режим работы модуля позиционирования, щелкните на кнопке Finish [Закончить], и Мастер управления позиционированием выполнит следующие действия:

- вставит настройки модуля и таблицу профилей в блок данных вашей программы для S7–200
- создаст таблицу глобальных символов для параметров перемещения
- добавит подпрограммы позиционирования в программный блок проекта, чтобы вы могли использовать его в своем приложении

Чтобы изменить данные настройки или профили, вы можете снова запустить Мастер управления позиционированием.



Совет

Так как Мастер управления позиционированием производит изменения в программном блоке, блоке данных и системном блоке, обеспечьте загрузку всех трех блоков в CPU S7–200. Иначе модуль позиционирования, возможно, не будет иметь всех программных компонентов, которые ему необходимы для надлежащего функционирования.

Команды, создаваемые Мастером управления позиционированием для модуля позиционирования

Мастер управления позиционированием облегчает управление модулем позиционирования путем создания подпрограмм на основе установленной вами позиции модуля и выбранных вами вариантов настройки. Каждая команда позиционирования имеет префикс «POSx_», где x – это положение модуля. Так как каждая команда позиционирования представляет собой подпрограмму, то 11 команд позиционирования используют 11 подпрограмм.



Совет

Команды позиционирования увеличивают объем памяти, необходимой для вашей программы на величину до 1700 байтов. Вы можете удалить неиспользуемые команды позиционирования для уменьшения требуемого объема памяти. Для восстановления удаленных команд позиционирования просто снова запустите Мастер управления позиционированием.

Указания по использованию команд позиционирования

Вы должны обеспечить, чтобы в каждый данный момент времени была активна только одна команда позиционирования.

Команды POSx_RUN и POSx_GOTO можно выполнять из программы обработки прерываний. Однако, очень важно, чтобы вы не пытались запустить команду в программе обработки прерываний, если модуль занят обработкой другой команды. Если вы запускаете команду в программе обработки прерываний, то вы можете использовать выходы команды POSx_CTRL для контроля завершения перемещения модулем позиционирования.

Мастер управления позиционированием автоматически настраивает значения параметров скорости (Speed и C_Speed) и параметров положения (Pos и C_Pos) в соответствии с выбранной вами системой измерения. Для импульсов эти параметры имеют значения типа DINT. Для инженерных единиц эти параметры имеют значения типа REAL в выбранных вами единицах измерения. Например: выбор сантиметров (см) сохраняет параметры положения как значения типа REAL в сантиметрах, а параметры скорости как значения типа REAL в сантиметрах в секунду (см/с).

Для конкретных задач управления позиционированием необходимы следующие команды:

- Вставьте в свою программу команду POSx_CTRL и используйте контакт SM0.0, чтобы исполнять ее в каждом цикле.
- Для задания перемещения в некоторое абсолютное положение вы должны сначала использовать команду POSx_RSEEK или POSx_LDPOS для установления нулевого положения.
- Для перемещения в определенное положение в соответствии с входами вашей программы используйте команду POSx_GOTO.
- Для запуска спроектированных вами профилей перемещения с помощью Мастера управления позиционированием используйте команду POSx_RUN.

Другие команды позиционирования являются необязательными.

Команда POSx_CTRL

Команда POSx_CTRL (управление) активизирует и инициализирует модуль позиционирования, автоматически заставляя модуль позиционирования загружать таблицу настроек и профилей каждый раз, когда S7-200 переходит в режим RUN.

Используйте эту команду в своем проекте только один раз и обеспечьте, чтобы ваша программа вызывала эту команду в каждом цикле. Используйте SM0.0 (всегда включен) в качестве входа для параметра EN.

Параметр MOD_EN должен быть включен, чтобы дать возможность другим командам позиционирования воздействовать на модуль позиционирования. Если параметр MOD_EN выключен, то модуль позиционирования прерывает все обрабатываемые команды.

Выходные параметры команды POSx_CTRL дают информацию о текущем состоянии модуля позиционирования.

Параметр Done включается при завершении модулем позиционирования любой команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

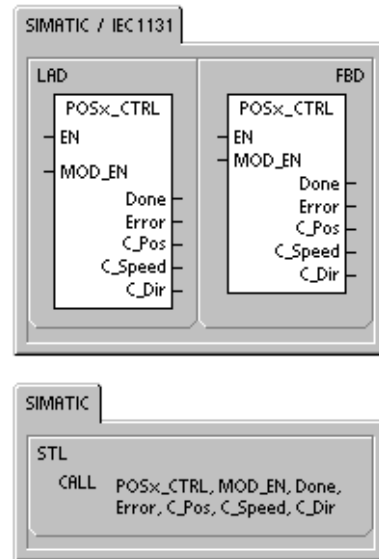
Параметр C_Pos определяет текущее положение модуля. В зависимости от единиц измерения это значение является количеством импульсов (DINT) или количеством инженерных единиц (REAL).

Параметр C_Speed дает информацию о текущей скорости модуля. Если вы настроили систему измерения для модуля позиционирования на импульсы, то C_Speed является значением типа DINT, содержащим количество импульсов в секунду. Если вы настроили систему измерения на инженерные единицы, то C_Speed является значением типа REAL, содержащим количество выбранных инженерных единиц в секунду (REAL).

Параметр C_Dir показывает текущее направление вращения двигателя.

Таблица 9–9. Параметры команды POSx_CTRL

Входы/выходы	Тип данных	Операнды
MOD_EN	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done, C_Dir	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Совет

Модуль позиционирования считывает таблицу настроек и профилей только при запуске или при получении предписания на загрузку настроек.

- Если вы используете для изменения настройки Мастер управления позиционированием, то команда POSx_CTRL автоматически предписывает модулю позиционирования загружать таблицу настроек и профилей каждый раз, когда CPU S7-200 переходит в режим RUN.
- Если вы используете для изменения настройки панель управления EM 253, то щелчок на кнопке Update Configuration [Изменить настройку] предписывает модулю позиционирования загрузить новую таблицу настроек и профилей.
- При использовании другого метода изменения настройки вы тоже должны выдать модулю позиционирования команду на перезагрузку таблицы настроек и профилей. Иначе модуль позиционирования будет использовать старую таблицу.

Команда POSx_MAN

Команда POSx_MAN (ручной режим) переводит модуль позиционирования в ручной режим. Это дает возможность двигателю работать с различными скоростями или перемещаться толчками в положительном или отрицательном направлении. Когда команда POSx_MAN активизирована, то допустимы только команды POSx_CTRL и POSx_DIS.

В каждый данный момент времени можно активизировать только один из входов RUN, JOG_P или JOG_N.

Активизация параметра RUN (пуск/останов) предписывает модулю позиционирования ускориться до заданной скорости (параметр Speed) в заданном направлении (параметр Dir). Вы можете изменить значение параметра Speed во время работы двигателя, но параметр Dir должен оставаться неизменным. Деактивизация параметра RUN предписывает модулю позиционирования замедляться до остановки двигателя.

Активизация параметра JOG_P (вращение толчками в положительном направлении) или JOG_N (вращение толчками в отрицательном направлении) предписывает модулю позиционирования выполнять перемещение толчками в положительном или отрицательном направлении. Если параметр JOG_P или JOG_N остается в активном состоянии менее, чем 0,5 секунды, то модуль позиционирования выдает импульсы для перемещения на расстояние, указанное в JOG_INCREMENT. Если параметр JOG_P или JOG_N остается в активном состоянии в течении 0,5 секунды или дольше, то модуль позиционирования ускоряется до скорости JOG_SPEED.

Параметр Speed определяет скорость, когда активизирован параметр RUN. Если вы настроили систему измерения модуля позиционирования на импульсы, то этот параметр имеет значение типа DINT для импульсов в секунду. Если вы настроили систему измерения модуля позиционирования на инженерные единицы, то этот параметр имеет значение типа REAL для выбранных единиц в секунду. Этот параметр можно изменять во время работы двигателя.



Совет

Модуль позиционирования может не среагировать на малые изменения параметра Speed, особенно если установленное при настройке время ускорения или замедления невелико, а разность между установленными при настройке максимальной и стартовой скоростью велика.

Параметр Dir определяет направление перемещения, когда активизирован параметр RUN. Когда параметр RUN активен, изменять это значение нельзя.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

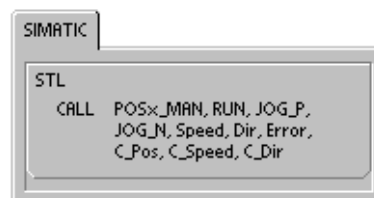
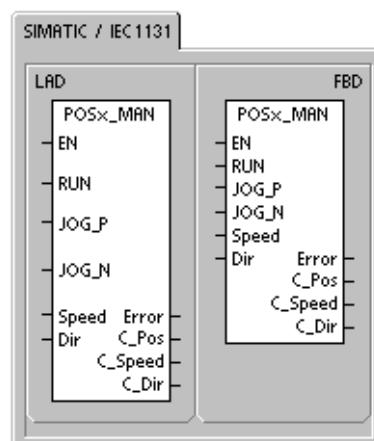
Параметр C_Pos содержит текущее положение модуля. В зависимости от выбранных единиц измерения это значение является или количеством импульсов (DINT), или количеством инженерных единиц (REAL).

Параметр C_Speed содержит текущую скорость модуля. В зависимости от выбранных единиц измерения это значение является или количеством импульсов в секунду (DINT), или количеством инженерных единиц в секунду (REAL).

Параметр C_Dir показывает текущее направление вращения двигателя.

Таблица 9–10. Параметры команды POSx_MAN

Входы/выходы	Тип данных	Операнды
RUN, JOG_P, JOG_N	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, константа
Dir, C_Dir	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Команда POSx_GOTO

Команда POSx_GOTO предписывает модулю позиционирования перейти в желаемое положение.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует о завершении выполнения команды.

Включение параметра START посылает модулю позиционирования предписание GOTO [перейти]. В каждом цикле, когда параметр START включен, а модуль позиционирования в данный момент не занят, команда посылает модулю позиционирования предписание GOTO. Чтобы обеспечить передачу только одного предписания GOTO, используйте для включения параметра START элемент обнаружения фронта.

Параметр Pos содержит значение, указывающее место, куда нужно переместиться (для абсолютного перемещения) или расстояние, на которое нужно переместиться (для относительного перемещения). В зависимости от выбранных единиц измерения это значение является или числом импульсов (DINT), или числом инженерных единиц (REAL).

Параметр Speed [Скорость] определяет максимальную скорость этого перемещения. В зависимости от выбранных единиц измерения это значение является или числом импульсов в секунду (DINT), или числом инженерных единиц в секунду (REAL).

Параметр Mode [Режим] определяет вид перемещения:

- 0 - Абсолютное позиционирование
- 1 - Относительное позиционирование
- 2 - Непрерывное вращение с постоянной скоростью в положительном направлении
- 3 - Непрерывное вращение с постоянной скоростью в отрицательном направлении

Параметр Done [Готово] включается, когда модуль позиционирования завершает выполнение этой команды.

Включение параметра Abort [Прервать] предписывает модулю остановить выполнение текущего профиля и замедляться до остановки двигателя.

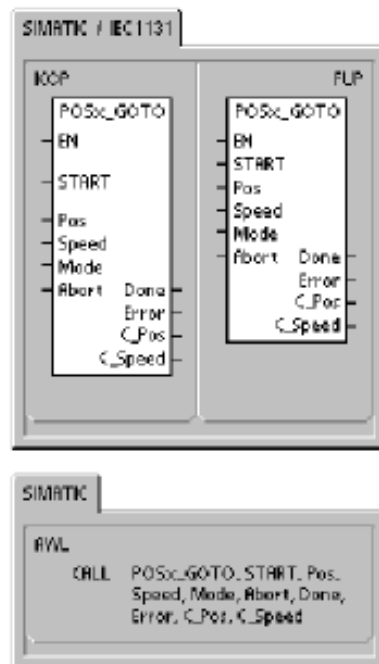
Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

Параметр C_Pos содержит текущее положение модуля. В зависимости от единиц измерения это значение является или числом импульсов (DINT), или числом инженерных единиц (REAL).

Параметр C_Speed содержит текущую скорость модуля. В зависимости от выбранных единиц измерения это значение является или числом импульсов в секунду (DINT), или числом инженерных единиц в секунду (REAL).

Таблица 9–11. Параметры команды POSx_GOTO

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Pos, Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, константа
Mode	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, константа
Abort, Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Команда POSx_RUN

Команда POSx_RUN (исполнить профиль) предписывает модулю позиционирования выполнить операцию по перемещению в соответствии с определенным профилем, хранящимся в таблице настроек и профилей.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

Включение параметра START посылает модулю позиционирования предписание RUN (начать исполнение). Эта команда посылает модулю позиционирования предписание RUN в каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр Profile [Профиль] содержит номер или символическое имя профиля перемещения. Вы можете также выбрать дополнительные команды позиционирования (со 118 по 127). Другие команды позиционирования вы найдете в таблице 9–26.

Включение параметра Abort предписывает модулю позиционирования прекратить исполнение текущего профиля и выполнить замедление до остановки двигателя.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

Параметр C_Profile содержит профиль, исполняемый в данный момент модулем позиционирования.

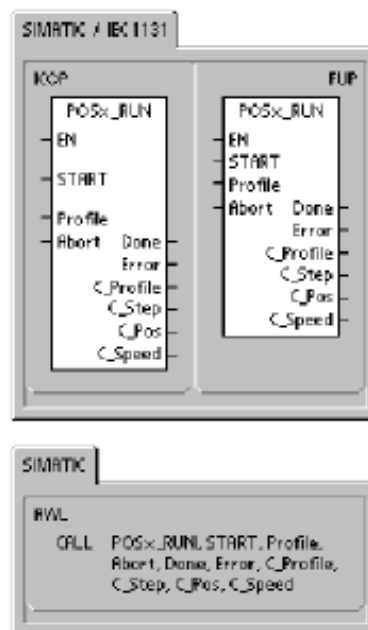
Параметр C_Step содержит исполняемый в данный момент шаг профиля.

Параметр C_Pos содержит текущее положение модуля. В зависимости от единиц измерения это значение является или числом импульсов (DINT), или числом инженерных единиц (REAL).

Параметр C_Speed содержит текущую скорость модуля. В зависимости от выбранных единиц измерения это значение является или числом импульсов в секунду (DINT), или числом инженерных единиц в секунду (REAL).

Таблица 9–12. Параметры команды POSx_RUN

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Profile	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, константа
Abort, Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error, C_Profile, C_Step	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Команда POSx_RSEEK

Команда POSx_RSEEK (поиск положения опорной точки) инициализирует операцию поиска опорной точки, используя метод поиска, указанный в таблице настроек и профилей. Когда модуль позиционирования обнаруживает опорную точку и завершает перемещение, он загружает значение параметра RP_OFFSET в текущем положении и генерирует 50–миллисекундный импульс на выходе CLR.

Значением по умолчанию для RP_OFFSET является 0. Для изменения значения RP_OFFSET можно использовать Мастер управления позиционированием, панель управления EM253 или команду POSx_LDOFF (загрузить смещение).

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

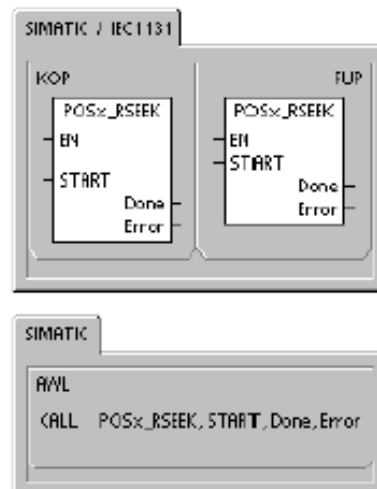
Включение параметра START посылает модулю позиционирования предписание RSEEK на поиск опорной точки. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание RSEEK. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

Таблица 9–13. Параметры команды POSx_RSEEK

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Команда POSx_LDOFF

Команда POSx_LDOFF (загрузить смещение опорной точки) устанавливает новое нулевое положение, отличающееся от положения опорной точки.

Перед выполнением этой команды вы сначала должны определить положение опорной точки. Вы должны также перевести станок в исходное положение. Когда команда посылает предписание LDOFF, модуль позиционирования вычисляет смещение между исходным (текущим) положением и положением опорной точки. Затем модуль позиционирования сохраняет вычисленное смещение в параметре RP_OFFSET и устанавливает текущее положение в 0. Благодаря этому нулевое положение устанавливается в качестве исходного.

В случае потери двигателем своего положения (например, из-за исчезновения питания или перестановки двигателя в новое положение вручную) команду POSx_RSEEK можно использовать для автоматического восстановления нулевого положения.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

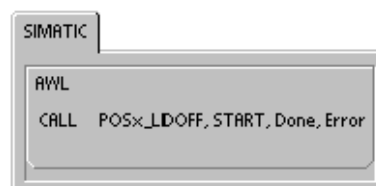
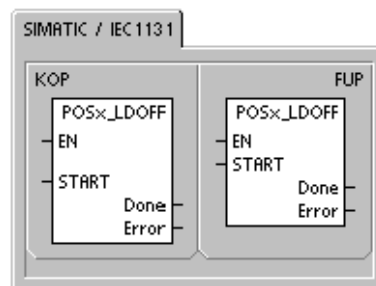
Включение параметра START посылает модулю позиционирования предписание LDOFF. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание LDOFF. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20s.

Таблица 9–14. Параметры команды POSx_LDOFF

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Команда POSx_LDPOS

Команда POSx_LDPOS (загрузить положение) заменяет текущее значение положения в модуле позиционирования новым значением. Эту команду можно использовать также для установки нового нулевого положения для команды абсолютного перемещения.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

Включение параметра START посылает модулю позиционирования предписание LDPOS. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание LDPOS. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр New_Pos дает новое значение, которым должно быть заменено текущее значение положения, сообщаемое модулем позиционирования и используемое для абсолютных перемещений. В зависимости от единиц измерения это значение является числом импульсов (DINT) или инженерных единиц (REAL).

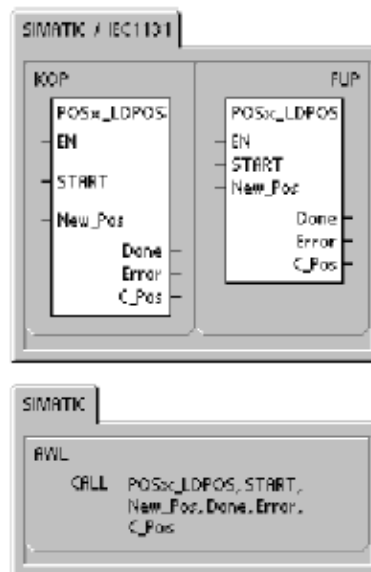
Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

Параметр C_Pos содержит текущее положение модуля. В зависимости от единиц измерения это значение является или числом импульсов (DINT), или числом инженерных единиц (REAL).

Таблица 9–15. Параметры команды POSx_LDPOS

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
New_Pos, C_Pos	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Команда POSx_SRATE

Команда POSx_SRATE (установить темп) предписывает модулю позиционирования изменить времена ускорения, замедления и сглаживания толчка.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

Включение параметра START копирует новые значения времени в таблиц настроек и профилей и посылает модулю позиционирования предписание SRATE на установку темпа. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание SRATE. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметры ACCEL_Time, DECEL_Time и JERK_Time определяют новые времена ускорения, замедления и сглаживания толчков в миллисекундах (мс).

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

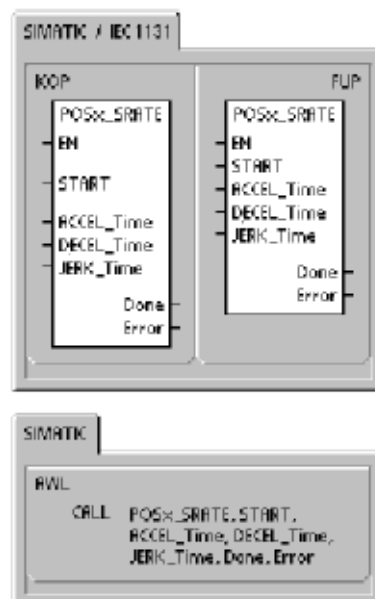


Таблица 9–16. Параметры команды POSx_SRATE

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L
ACCEL_Time, DECEL_Time, JERK_Time	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, константа
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Команда POSx_DIS

Команда POSx_DIS выключает или включает выход DIS модуля позиционирования. Это дает вам возможность использовать выход DIS для деактивизации и активизации устройства управления двигателем. Если вы используете выход DIS в модуле позиционирования, то эта команда может вызываться в каждом цикле или только тогда, когда вам нужно изменить значение выхода DIS.

Когда включается бит EN для активизации команды, параметр DIS_ON управляет выходом DIS модуля позиционирования. Дополнительную информацию о выходе DIS вы найдете в таблице 9–8 или в технических данных модуля позиционирования в Приложении A.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

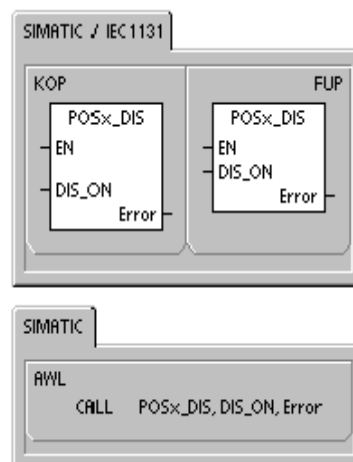


Таблица 9–17. Параметры команды POSx_DIS

Входы/выходы	Тип данных	Операнды
DIS_ON	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, константа
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Команда POSx_CLR

Команда POSx_CLR (сгенерировать импульс на выходе CLR) предписывает модулю позиционирования сгенерировать 50–миллисекундный импульс на выходе CLR.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

Включение параметра START посылает модулю позиционирования предписание CLR. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание CLR. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

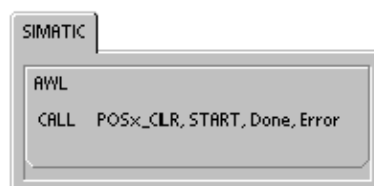
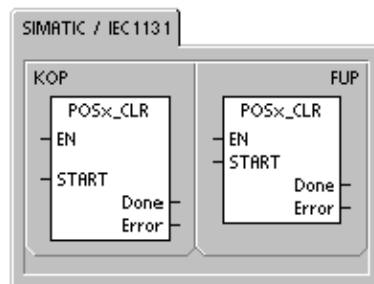


Таблица 9–18. Параметры команды POSx_CLR

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Команда POSx_CFG

Команда POSx_CFG (снова загрузить настройки) предписывает модулю позиционирования считать конфигурационный блок по адресу, содержащемуся в указателе на таблицу настроек и профилей. Затем модуль позиционирования сравнивает новую настройку с существующей и выполняет все необходимые изменения и новые расчеты.

Включение бита EN активизирует команду. Обеспечьте, чтобы бит EN оставался включенным, пока бит Done не сообщит о завершении команды.

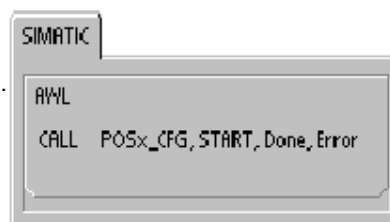
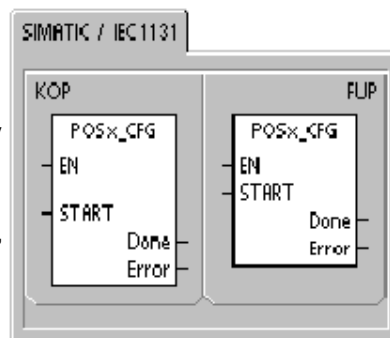
Включение параметра START посылает модулю позиционирования предписание CFG. В каждом цикле, когда параметр START включен, а модуль позиционирования в это время не занят, команда посылает модулю позиционирования предписание CFG. Чтобы обеспечить посылку только одной команды, используйте для включения параметра START элемент обнаружения фронта.

Параметр Done [Готово] включается, когда модуль завершает выполнение команды.

Параметр Error [Ошибка] содержит результат выполнения этой команды. Определения кодов ошибок вы найдете в таблице 9–20.

Таблица 9–19. Параметры команды POSx_CFG

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Примеры программ для модуля позиционирования

Первый образец программы показывает простое относительное перемещение, использующее команды POSx_CTRL и POSx_GOTO для выполнения операции раскрой по длине. Программа не требует использования режима поиска опорной точки и профиля перемещения, а длина может измеряться в импульсах или инженерных единицах. Введите длину (VD500) и целевую скорость (VD504). Когда включается I0.0 (Start), станок начинает работать. Когда включается I0.1 (Stop), станок завершает текущую операцию и останавливается. Когда включается I0.2 (E_Stop), станок прерывает перемещение и немедленно останавливается.

Второй образец программы дает пример использования команд POSx_CTRL, POSx_RUN, POSx_RSEEK и POSx_MAN. Вы должны настроить режим поиска опорной точки и профиль перемещения.

Пример программы 1: Простое относительное перемещение (раскрой по длине)

<p>Network 1</p>	<p>Network 1 //Команда управления (модуль в //гнезде 0).</p> <pre>LD SM0.0 = L60.0 LDN I0.2 = L63.7 LD L60.0 CALL POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906, V910.0</pre>
<p>Network 2</p>	<p>Network 2 //Start переводит станок в //автоматический режим</p> <pre>LD I0.0 AN I0.2 EU S Q0.2, 1 S M0.1, 1</pre>
<p>Network 3</p>	<p>Network 3 //E_Stop: немедленно останавливает и //выключает автоматический режим.</p> <pre>LD I0.2 R Q0.2, 1</pre>
<p>Network 4</p>	<p>Network 4 //Перемещение в определенную точку: //Ввод длины для отрезания. //Ввод целевой скорости в Speed. //Установка режима в 1 (режим //относительного перемещения).</p> <pre>LD Q0.2 = L60.0 LD M0.1 EU = L63.7 LD L60.0 CALL POS0_GOTO, L63.7, VD500, VD504, 1, I0.2, Q0.4, VB920, VD922, VD926</pre>
<p>Network 5</p>	<p>Network 5 //При занятии положения, включить //резак на 2 секунды для выполнения //отрезания.</p> <pre>LD Q0.2 A Q0.4 TON T33, +200 AN T33 = Q0.3</pre>

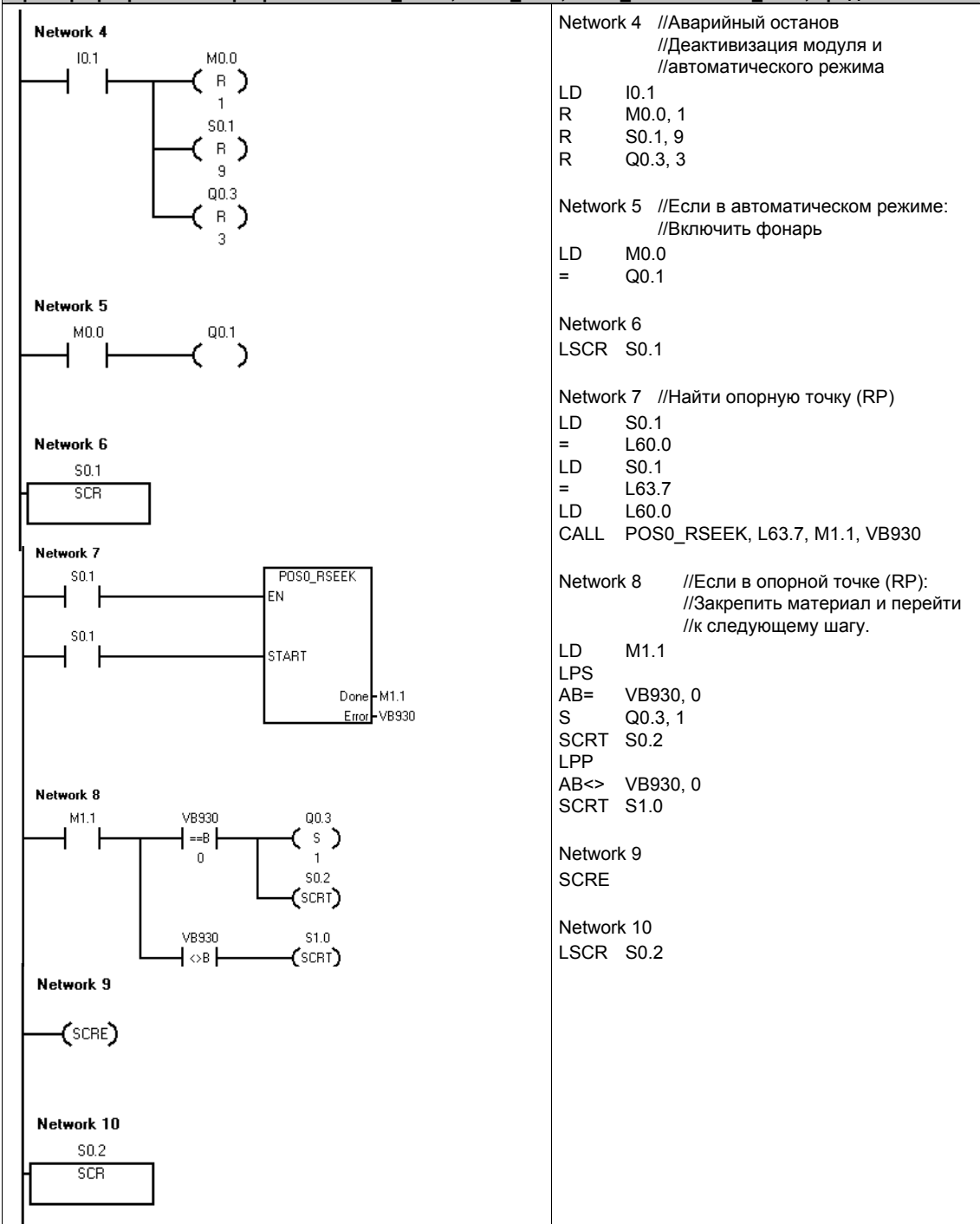
Пример программы 1: Простое относительное перемещение (раскрой по длине), продолжение

<p>Network 6</p>	<p>Network 6 //По окончании отрезания запустить //снова, если Stop не активен.</p> <pre> LD Q0.2 A T33 LPS AN I0.1 = M0.1 LPP A I0.1 R Q0.2, 1 </pre>
-------------------------	---

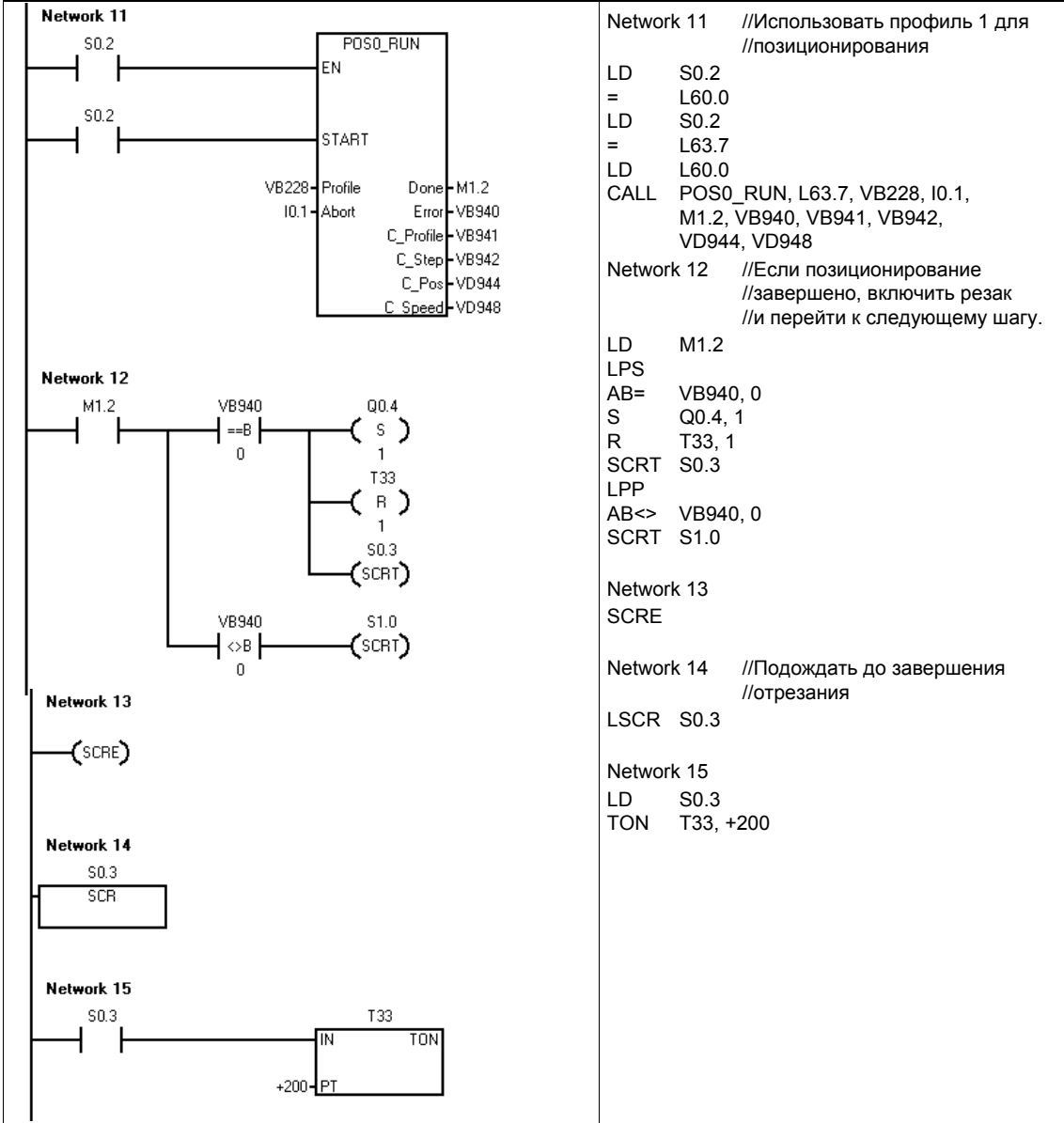
Пример программы 2: Программа с POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p>	<p>Network 1 //Активизировать модуль //позиционирования</p> <pre> LD SM0.0 = L60.0 LDN I0.1 = L63.7 LD L60.0 CALL POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906, V910.0 </pre> <p>Network 2 //Ручной режим, если не находится в //автоматическом режиме</p> <pre> LD I1.0 AN M0.0 = L60.0 LD I1.1 = L63.7 LD I1.2 = L63.6 LD I1.4 = L63.5 LD L60.0 CALL POS0_MAN, L63.7, L63.6, L63.5, +100000, 1.5, VB920, VD902, VD906, V910.0 </pre> <p>Network 3 //Активизация автоматического //режима</p> <pre> LD I0.0 EU S M0.0, 2 S S0.1, 1 R S0.2, 8 </pre>
---	---

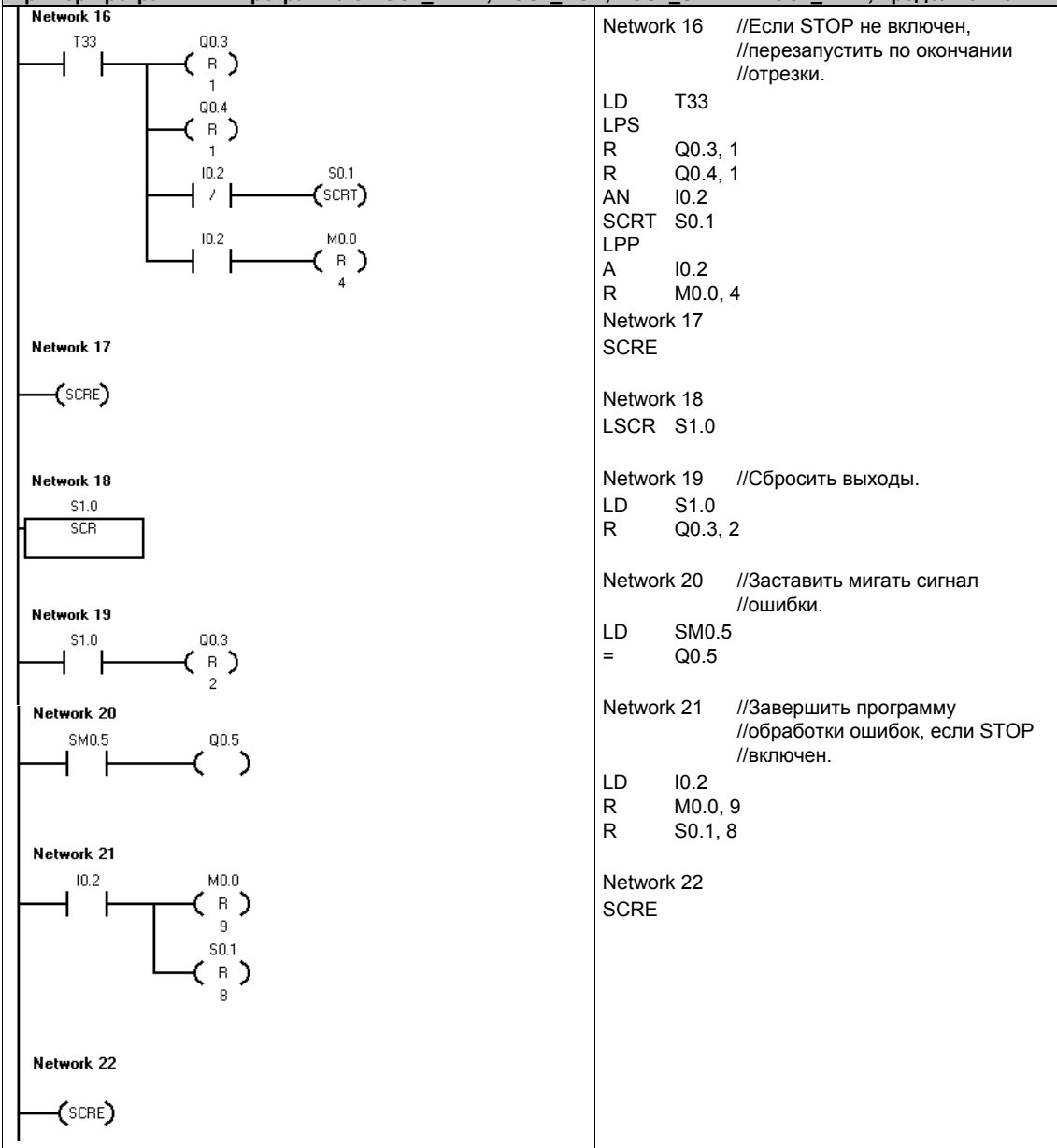
Пример программы 2: Программа с POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение



Пример программы 2: Программа с POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение



Пример программы 2: Программа с POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение



Наблюдение за модулем позиционирования с помощью панели управления EM 253

Для поддержки решения задач управления позиционированием в STEP 7-Micro/WIN имеется панель управления EM 253. Закладки Operation [Работа], Configuration [Настройка] и Diagnostics [Диагностика] облегчают наблюдение и управление работой модуля позиционирования на этапах ввода в эксплуатацию и тестирования вашего процесса разработки.

Панель управления EM 253 можно использовать для проверки правильности подключения модуля позиционирования, для согласования данных настройки и тестирования любого профиля перемещения.

Отображение и управление работой модуля позиционирования

Закладка Operation [Работа] в панели управления дает вам возможность влиять на функционирование модуля позиционирования. Панель управления отображает текущую скорость, текущее положение и текущее направление модуля позиционирования. Вы можете увидеть также состояние светодиодов на входах и выходах (кроме импульсных светодиодов).

Панель управления дает вам возможность влиять на функционирование модуля позиционирования, изменяя скорость и направление, останавливая и запуская перемещения и управляя инструментом в толчковом режиме (если движение остановлено). Вы можете также генерировать следующие команды позиционирования:

- Активизация ручного режима. Эта команда дает вам возможность позиционирования инструмента с помощью устройств ручного управления.
- Исполнение профиля перемещения. Эта команда дает вам возможность выбрать профиль для исполнения. Панель управления отображает состояние профиля, который выполняется модулем позиционирования.

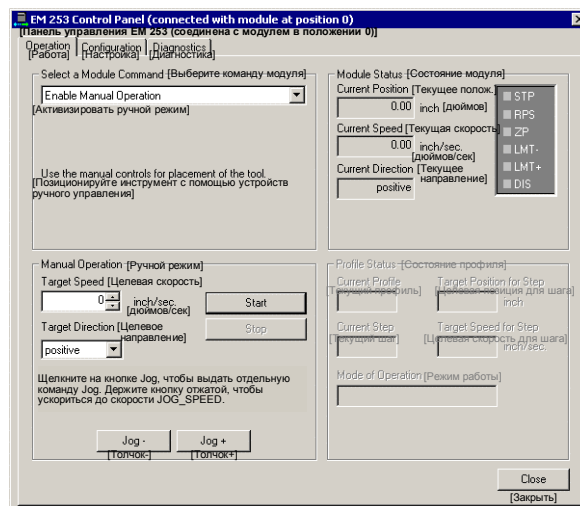


Рис. 9–17. Закладка Operation в панели управления EM 253

- Поиск опорной точки. Эта команда ищет опорную точку, используя установленный при настройке режим поиска. Загрузка смещения опорной точки. После того как вы с помощью устройств ручного управления переместили инструмент в новое нулевое положение, загрузите смещение опорной точки.
- Новая загрузка опорной точки. Эта команда обновляет значение текущего положения и устанавливает новое нулевое положение.
- Активизация и деактивизация выхода DIS. Эти команды включают и выключают выход DIS модуля позиционирования.
- Выдать импульс на выходе CLR. Эта команда генерирует 50-миллисекундный импульс на выходе CLR модуля позиционирования.
- Обучение профилю перемещения. Эта команда дает вам возможность сохранить при ручном позиционировании инструмента целевую позицию и скорость для профиля перемещения и шага. Панель управления отображает состояние профиля, исполняемого модулем позиционирования.
- Загрузка настройки модуля. Эта команда загружает новую настройку, предписывая модулю позиционирования считать конфигурационный блок из памяти переменных (V) S7-200.

- Перемещение к абсолютной позиции. Эта команда дает вам возможность перейти в заданное положение с целевой скоростью. Перед использованием этой команды вы должны уже установить нулевое положение.
- Выполнить относительное перемещение. Эта команда дает вам возможность переместиться из текущего положения на заданное расстояние с целевой скоростью. Вы можете ввести положительное или отрицательное расстояние.
- Сброс интерфейса команды. Эта команда очищает командный байт для модуля позиционирования и устанавливает бит Done [Готово]. Используйте эту команду, если вам кажется, что модуль позиционирования не реагирует на команды.

Отображение и изменение настройки модуля позиционирования

Закладка Configuration [Настройка] панели управления дает возможность просматривать и изменять настройки модуля позиционирования, которые хранятся в блоке данных S7-200.

Изменив настройки, просто щелкните на кнопке, чтобы обновить настройки в проекте STEP 7-Micro/Win и в блоке данных S7-200.

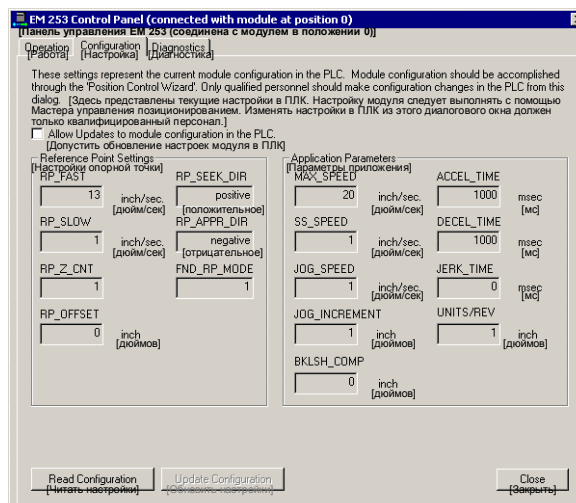


Рис. 9–18. Закладка Configuration панели управления EM 253

Отображение диагностической информации для модуля позиционирования

Закладка Diagnostics панели управления дает возможность просмотреть диагностическую информацию о модуле позиционирования.

Вы можете увидеть определенную информацию о модуле позиционирования, например, положение модуля в цепи входов и выходов, тип модуля и номер версии программы ПЗУ и выходной байт, используемый в качестве командного байта для модуля.

Панель управления отображает все сбойные состояния, вызванные предписанным процессом. Сбойные состояния для команд вы найдете в таблице 9–20.

Вы можете увидеть также сбойные состояния, сообщаемые модулем позиционирования. Сбойные состояния для модуля вы найдете в таблице 9–21.

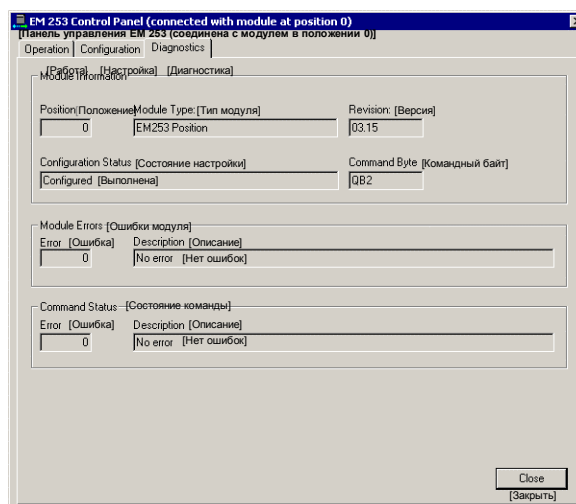


Рис. 9–19. Закладка Diagnostics панели управления EM 253

Коды ошибок для модуля и команд позиционирования

Таблица 9–20. Коды ошибок команд

Код ошибки	Описание
0	Нет ошибок
1	Прервано пользователем
2	Ошибка настройки Для просмотра кодов ошибок используйте закладку Diagnostics [Диагностика] панели управления EM 253
3	Недопустимая команда
4	Прервано из-за недопустимой настройки Для просмотра кодов ошибок используйте закладку Diagnostics [Диагностика] панели управления EM 253
5	Прервано из-за отсутствия пользовательского напряжения
6	Прервано из-за неопределенной опорной точки
7	Прервано из-за активного входа STP
8	Прервано из-за активного входа LMT-
9	Прервано из-за активного входа LMT+
10	Прервано из-за проблемы при выполнении перемещения
11	Для указанного профиля не сформирован блок
12	Недопустимый режим работы
13	Режим работы для этой команды не поддерживается
14	Недопустимое количество шагов в профильном блоке
15	Недопустимое изменение направления
16	Недопустимое расстояние
17	Срабатывание переключателя RPS до достижения целевой скорости
18	Недостаточная ширина области активности RPS
19	Скорость вне допустимого диапазона
20	Недостаточное расстояние для выполнения желаемого изменения скорости
21	Недопустимое положение
22	Неизвестно нулевое положение
с 23 по 127	Резерв
128	Модуль позиционирования не может обработать эту команду: или модуль позиционирования занят другой командой, или в этой команде не было стартового импульса
129	Ошибка модуля позиционирования: Неверен идентификатор модуля или отменена регистрация модуля. Другие сбойные состояния вы найдете в SMB8 – SMB21 (идентификатор модуля ввода-вывода и регистр ошибок).
130	Модуль позиционирования не активизирован
131	Модуль позиционирования недоступен из-за ошибки модуля или отсутствия его активизации (см. состояние POSx_CTRL)
132	Адрес в области памяти выходов (Q), установленный Мастером управления позиционированием, не соответствует фактическому адресу модуля в памяти на этом месте.

Таблица 9–21. Коды ошибок модуля

Код ошибки	Описание
0	Нет ошибок
1	Отсутствует пользовательское напряжение
2	Отсутствует блок настройки
3	Ошибка указателя блока настройки
4	Размер блока настройки превышает доступную память переменных (V)
5	Недопустимый формат блока настройки
6	Задано слишком много профилей
7	Недопустимое задание STP_RSP
8	Недопустимое задание LMT-_RPS
9	Недопустимое задание LMT+_RPS
10	Недопустимое задание FILTER_TIME
11	Недопустимое задание MEAS_SYS
12	Недопустимое задание RP_CFG
13	Недопустимое значение PLS/REV
14	Недопустимое значение UNITS/REV
15	Недопустимое значение RP_ZP_CNT
16	Недопустимое значение JOG_INCREMENT
17	Недопустимое значение MAX_SPEED
18	Недопустимое значение SS_SPD
19	Недопустимое значение RP_FAST
20	Недопустимое значение RP_SLOW
21	Недопустимое значение JOG_SPEED
22	Недопустимое значение ACCEL_TIME
23	Недопустимое значение DECEL_TIME
24	Недопустимое значение JERK_TIME
25	Недопустимое значение BKLSH_COMP

Для опытных пользователей

Описание таблицы настроек и профилей

Мастер управления позиционированием разработан для того, чтобы упростить приложения в области управления позиционированием путем автоматической генерации информации о настройках и профилях на основе ответов, которые вы даете о своей системе управления позиционированием. Информация о таблице настроек и профилей предназначена для опытных пользователей, которые хотят создавать свои собственные программы управления позиционированием.

Таблица настроек и профилей находится в области памяти переменных (V) S7-200. Как видно из таблицы 9–22, настройки хранятся в следующих видах данных:

- Конфигурационный блок содержит данные, которые используются для подготовки модуля к выполнению команд позиционирования.
- Интерактивный блок поддерживает непосредственную установку параметров перемещения через программу пользователя.
- Каждый профильный блок описывает заранее определенную операцию перемещения, подлежащую выполнению модулем позиционирования. Вы можете создать до 25 блоков профилей.


	<p>Совет Для создания большего, чем 25, количества профилей перемещения вы можете заменить таблицу настроек и профилей, изменить значение, хранящееся в указателе на таблицу настроек и профилей.</p>
---	--

Таблица 9–22. Таблица настроек и профилей

Смещение	Имя	Описание функций	Тип																				
Конфигурационный блок																							
0	MOD_ID	Поле для идентификатора модуля	--																				
5	CB_LEN	Длина конфигурационного блока в байтах (1 байт)	--																				
6	IB_LEN	Длина интерактивного блока в байтах (1 байт)	--																				
7	PF_LEN	Длина отдельного профиля в байтах (1 байт)	--																				
8	STP_LEN	Длина отдельного шага в байтах (1 байт)	--																				
9	STEPS	Допустимое количество шагов на профиль (1 байт)	--																				
10	PROFILES	Количество профилей от 0 до 25 (1 байт)	--																				
11	Резерв	Установлен на 0x0000	--																				
13	IN_OUT_CFG	<p>Определяет использование входов и выходов модуля (1 байт)</p> <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">LSB</td> <td style="text-align: center;">0</td> </tr> <tr> <td></td> <td style="text-align: center;">P/D</td> <td style="text-align: center;">POL</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">STP</td> <td style="text-align: center;">RPS</td> <td style="text-align: center;">LMT-</td> <td style="text-align: center;">LMT+</td> <td></td> </tr> </table> <p>P/D Этот бит определяет использование P0 и P1. Положительная полярность (POL = 0): 0 – Импульсы P0 для положительного направления вращения Импульсы P1 для отрицательного направления вращения 1 – Импульсы P0 для вращения P1 управляет направлением вращения (0 – положительное, 1 – отрицательное) Отрицательная полярность (POL = 1): 0 – Импульсы P0 для положительного направления вращения Импульсы P1 для отрицательного направления вращения 1 – Импульсы P0 для вращения P1 управляет направлением вращения (0 – положительное, 1 – отрицательное) POL Этот бит устанавливает соглашение о полярности P0 и P1 (0 – положительная полярность; 1 – отрицательная полярность) STP Этот бит управляет уровнем активности входа для остановки RPS Этот бит управляет уровнем активности для входа RPS LMT- Этот бит управляет уровнем активности для входа Отрицательная граница перемещения LMT+ Этот бит управляет уровнем активности для входа Положительная граница перемещения 0 – активен высокий уровень 1 – активен низкий уровень</p>	MSB	7	6	5	4	3	2	1	LSB	0		P/D	POL	0	0	STP	RPS	LMT-	LMT+		--
MSB	7	6	5	4	3	2	1	LSB	0														
	P/D	POL	0	0	STP	RPS	LMT-	LMT+															

Таблица 9–22. Таблица настроек и профилей, продолжение

Смещение	Имя	Описание функций	Тип																				
14	STP_RSP	<p>Определяет реакцию привода на вход STP (1 байт)</p> <p>0 Нет действий. Игнорировать условия на входе. 1 Замедлиться до остановки и отобразить, что вход STP активен. 2 Завершить импульсы и отобразить вход STP 3 – 255 Резерв (ошибка, если указан)</p>	--																				
15	LMT_-RSP	<p>Определяет реакцию привода на вход "Отрицательная граница перемещения" (1 байт)</p> <p>0 Нет действий. Игнорировать условия на входе. 1 Замедлиться до остановки и отобразить, что граница достигнута. 2 Завершить импульсы и отобразить, что граница достигнута. 3 – 255 Резерв (ошибка, если указан)</p>	--																				
16	LMT+_RSP	<p>Определяет реакцию привода на вход "Положительная граница перемещения" (1 байт)</p> <p>0 Нет действий. Игнорировать условия на входе. 1 Замедлиться до остановки и отобразить, что граница достигнута. 2 Завершить импульсы и отобразить, что граница достигнута. 3 – 255 Резерв (ошибка, если указан)</p>	--																				
17	FILTER_TIME	<p>Определяет время фильтра для входов LMT-, LMT+ и RPS (1 байт)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td></td> <td colspan="4" style="text-align: center;">STP, LMT-, LMT+</td> <td colspan="4" style="text-align: center;">RPS</td> <td></td> </tr> </table> <p>'0000' 200 мкс '0101' 3200 мкс '0001' 400 мкс '0110' 6400 мкс '0010' 800 мкс '0111' 12800 мкс '0011' 1600 мкс '1000' Нет фильтра '0100' 1600 мкс от '1001' до '1111' Резерв (ошибка, если указано)</p>	MSB	7	6	5	4	3	2	1	0	LSB		STP, LMT-, LMT+				RPS					--
MSB	7	6	5	4	3	2	1	0	LSB														
	STP, LMT-, LMT+				RPS																		
18	MEAS_SYS	<p>Определяет систему измерения (1 байт)</p> <p>0 Импульсы (скорость измеряется в импульсах в секунду, а значение положения – в импульсах). Значения хранятся как данные типа DINT. 1 Инженерные единицы (скорость измеряется в единицах в секунду, а значение положения – в единицах). Значения хранятся как данные одинарной точности типа REAL. 2 – 255 Резерв (ошибка, если указано)</p>	--																				
19	--	Резерв (установлено в 0)	--																				
20	PLS/REV	<p>Определяет количество импульсов на оборот двигателя (4 байта)</p> <p>Имеет силу только в том случае, когда MEAS_SYS установлен в 1.</p>	DINT																				
24	UNITS/REV	<p>Определяет количество инженерных единиц на оборот двигателя (4 байта)</p> <p>Имеет силу только в том случае, когда MEAS_SYS установлен в 1.</p>	REAL																				
28	UNITS	Зарезервирован для STEP 7-Micro/WIN для хранения единицы, определяемой пользователем (4 байта)	--																				
32	RP_CFG	<p>Определяет настройки для поиска опорной точки (1 байт)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td colspan="4" style="text-align: center;">MODE</td> <td></td> </tr> </table> <p style="margin-left: 40px;">↑ RP_ADDR_DIR ↑ RP_SEEK_DIR</p> <p>RP_SEEK_DIR Этот бит определяет начальное направление для поиска опорной точки. (0 – положительное направление; 1 – отрицательное направление)</p> <p>RP_APPR_DIR Этот бит определяет направление приближения при завершении поиска опорной точки. (0 – положительное направление; 1 – отрицательное направление)</p> <p>MODE Определяет метод поиска опорной точки '0000' Поиск опорной точки деактивизирован '0001' Опорная точка находится там, где вход RPS становится активным '0010' Опорная точка находится в центре диапазона активности входа RPS '0011' Опорная точка находится вне области активности входа RPS '0100' Опорная точка находится внутри области активности входа RPS от '0101' до '1111' Резерв (ошибка, если выбрано)</p>	MSB	7	6	5	4	3	2	1	0	LSB				0	0	MODE					--
MSB	7	6	5	4	3	2	1	0	LSB														
			0	0	MODE																		

Таблица 9–22. Таблица настроек и профилей, продолжение

Смещение	Имя	Описание функций	Тип
33	--	Резерв (установлен в 0)	--
34	RP_Z_CNT	Количество импульсов входа ZP для определения опорной точки (4 байта)	DINT REAL
38	RP_FAST	Высокая скорость для операции поиска опорной точки: MAX_SPD или меньше (4 байта)	DINT REAL
42	RP_SLOW	Низкая скорость для операции поиска опорной точки: максимальная скорость, с которой двигатель может немедленно остановиться или меньше (4 байта)	DINT REAL
46	SS_SPEED	Стартстопная скорость. (4 байта) стартовая скорость – это максимальная скорость, на которую двигатель может немедленно перейти из состояния покоя, и максимальная скорость, с которой двигатель может немедленно остановиться. Работа ниже этой скорости допустима, но времена ускорения и замедления в этом случае не имеют смысла.	DINT REAL
50	MAX_SPEED	Максимальная рабочая скорость двигателя (4 байта)	DINT REAL
54	JOG_SPEED	Толчковая скорость. MAX_SPEED или меньше (4 байта)	
58	JOG_INCREMENT	Расстояние (или количество импульсов), проходимое в ответ на единичный толчок. (4 байта)	DINT REAL
62	ACCEL_TIME	Время, необходимое для ускорения от минимальной до максимальной скорости в миллисекундах (4 байта)	DINT
66	DECEL_TIME	Время, необходимое для замедления от максимальной до минимальной скорости в миллисекундах (4 байта)	DINT
70	BKLSH_COMP	Компенсация зазора: расстояние, используемое для компенсации зазора в системе при изменении направления (4 байта)	DINT REAL
74	JERK_TIME	Время, в течение которого производится сглаживание толчка на начальном и конечном участках кривой ускорения и замедления (S-кривая). Задание значения 0 деактивирует сглаживание толчка. Время сглаживания толчка задается в миллисекундах. (4 байта)	DINT
Интерактивный блок			
78	MOVE_CMD	Устанавливает режим работы (1 байт) 0 Абсолютное позиционирование 1 Относительное позиционирование 2 Непрерывное перемещение в положительном направлении с постоянной скоростью 3 Непрерывное перемещение в отрицательном направлении с постоянной скоростью 4 Ручное управление скоростью, положительное направление 5 Ручное управление скоростью, отрицательное направление 6 Непрерывное перемещение в положительном направлении с постоянной скоростью и активизированной командой останова (вход RPS сообщает об останове) 7 Непрерывное перемещение в отрицательном направлении с постоянной скоростью и активизированной командой останова (вход RPS сообщает об останове) с 8 по 255 - Резерв (ошибка, если указано)	--
79	--	Резерв. Установлен в 0	--
80	TARGET_POS	Целевое положение для этого перемещения (4 байта)	DINT REAL
84	TARGET_SPEED	Целевая скорость для этого перемещения (4 байта)	DINT REAL
88	RP_OFFSET	Абсолютное позиционирование опорной точки (4 байта)	DINT REAL

Таблица 9–22. Таблица настроек и профилей, продолжение

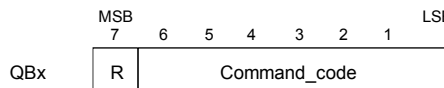
Смещение	Имя		Описание функций	Тип
Профильный блок 0				
92 (+0)	STEPS		Количество шагов в этой последовательности перемещений(1 байт)	--
93 (+1)	MODE		Устанавливает режим работы для этого профильного блока (1 байт) 0 Абсолютное позиционирование 1 Относительное позиционирование 2 Непрерывное перемещение с постоянной скоростью, положительное направление вращения 3 Непрерывное перемещение с постоянной скоростью, отрицательное направление вращения 4 Резерв (ошибка, если указано) 5 Резерв (ошибка, если указано) 6 Непрерывное перемещение с постоянной скоростью, положительное направление вращения с активной командой останова (скорость устанавливается RPS) 7 Непрерывное перемещение с постоянной скоростью, отрицательное направление вращения с активной командой останова (RPS сообщает об останове) 8 Непрерывное перемещение в положительном направлении с двумя скоростями (скорость устанавливается RPS) 9 Непрерывное перемещение в отрицательном направлении с двумя скоростями (скорость устанавливается RPS) с 10 по 255 - Резерв (ошибка, если указано)	--
94 (+2)	0	POS	Положение, в которое нужно перейти на шаге перемещения 0 (4 байта)	DINT REAL
98 (+6)		SPEED	Целевая скорость для шага перемещения 0 (4 байта)	DINT REAL
102 (+10)	1	POS	Положение, в которое нужно перейти на шаге перемещения 1 (4 байта)	DINT REAL
106 (+14)		SPEED	Целевая скорость для шага перемещения 1 (4 байта)	DINT REAL
110 (+18)	2	POS	Положение, в которое нужно перейти на шаге перемещения 2 (4 байта)	DINT REAL
114 (+22)		SPEED	Целевая скорость для шага перемещения 2 (4 байта)	DINT REAL
118 (+26)	3	POS	Положение, в которое нужно перейти на шаге перемещения 3 (4 байта)	DINT REAL
122 (+30)		SPEED	Целевая скорость для шага перемещения 3 (4 байта)	DINT REAL
Профильный блок 1				
126 (+34)	STEPS		Количество шагов в этой последовательности перемещений(1 байт)	--
127 (+35)	MODE		Устанавливает режим работы для этого профильного блока (1 байт)	--
128 (+36)	0	POS	Положение, в которое нужно перейти на шаге перемещения 0 (4 байта)	DINT REAL
132 (+40)		SPEED	Целевая скорость для шага перемещения 0 (4 байта)	DINT REAL
...

Описание командного байта для модуля позиционирования

Модуль позиционирования предоставляет один байт цифровых выходов, который используется как командный байт. На рис. 9–20 приведено определение командного байта. В таблице 9–20 показаны определения в Command_code.

Запись в командный байт, при которой бит R меняет значение с 0 на 1, интерпретируется как новая команда.

Если модуль обнаруживает переход к холостому ходу (бит R меняет состояние на 0), когда команда активна, то исполнение команды прерывается и, если производится перемещение, то выполняется замедление до остановки.



R 0 = холостой ход
1 = выполнение команды, указанной в Command_code (см. табл. 9-25)

Рис. 9–20. Определение командного байта

По окончании процесса модуль сначала должен перейти к холостому ходу, чтобы можно было принять новую команду. Если операция прервана, то модуль до приема новой команды должен завершить замедление. Любое изменение значения Command_code, когда команда активна, игнорируется.

Реакция модуля позиционирования на изменение режима работы S7–200 или на ошибочное состояние регулируется тем, что S7–200 управляет цифровыми выходами в соответствии с текущей функцией S7–200:

- Если S7–200 переходит из STOP в RUN: Программа в S7–200 управляет работой модуля позиционирования.
- Если S7–200 переходит из RUN в STOP: Вы можете установить состояние, которое цифровые выходы должны принять при переходе в STOP, или выбрать, что выходы должны сохранить свое последнее состояние.
 - Если бит R выключен при переходе в STOP: Модуль позиционирования замедляет любые текущие перемещения до остановки
 - Если бит R включен при переходе в STOP: Модуль позиционирования завершает все текущие команды. Если никакая команда не выполняется, то модуль позиционирования выполняет команду, указанную в битах Command_code.
 - Если бит R сохраняется в своем последнем состоянии: Модуль позиционирования завершает все текущие перемещения.
- Если S7–200 обнаруживает фатальную ошибку и выключает все цифровые выходы: Модуль позиционирования замедляет все текущие перемещения до остановки.

Command_code	Команда
000 0000 – 000 1111	0 – 24 Исполнять перемещение, указанное в профильных блоках с 0 по 24
100 0000 – 111 0101	25 – 117 Резерв (Ошибка, если указан)
111 0110	118 Активизировать выход DIS
111 0111	119 Деактивизировать выход DIS
111 1000	120 Импульс на выходе CLR
111 1001	121 Снова загрузить текущее положение
111 1010	122 Исполнять перемещение, указанное в интерактивном блоке
111 1011	123 Зарегистрировать смещение опорной точки
111 1100	124 Толчковый режим, положительное направление вращения
111 1101	125 Толчковый режим, отрицательное направление вращения
111 1110	126 Искать положение опорной точки
111 1111	127 Снова загрузить настройки

Модуль позиционирования осуществляет контроль времени, который выключает выходы при потере связи с S7–200. Если время контроля истекает, то модуль позиционирования замедляет все текущие перемещения до остановки.

Если обнаруживается фатальная ошибка в аппаратных средствах или в ПЗУ модуля, то модуль позиционирования устанавливает выходы P0, P1, DIS и CLR в неактивное состояние.

Таблица 9–26. Команды позиционирования

Команда	Описание
<p>Команды 0 – 24: <i>Исполняет перемещение, указанное в профильных блоках с 0 по 24</i></p>	<p>Когда эта команда исполняется, модуль позиционирования выполняет операцию перемещения, указанного в поле MODE [РЕЖИМ] профильного блока, которое, в свою очередь, указано в разделе Command_code команды.</p> <ul style="list-style-type: none"> • В режиме 0 (абсолютное позиционирование), профильный блок перемещения определяет от одного до четырех шагов, каждый из которых содержит положение (POS) и скорость (SPEED), которые описывают участок перемещения. Задание POS представляет собой абсолютное положение, отсчитываемое от положения, называемого опорной точкой. Направление перемещения определяется соотношением между текущим положением и положением первого шага в профиле. При многошаговом перемещении изменение направления движения запрещено и приводит к сообщению об ошибке. • В режиме 1 (относительное позиционирование), профильный блок перемещения определяет от одного до четырех шагов, каждый из которых содержит положение (POS) и скорость (SPEED), которые описывают участок перемещения. Знак величины, определяющей положение (POS), определяет направление перемещения. При многошаговом перемещении изменение направления движения запрещено и приводит к сообщению об ошибке. • В режимах 2 и 3 (режимы работы с непрерывным перемещением с постоянной скоростью) задание (POS) игнорируется, и модуль ускоряется до скорости, указанной в поле SPEED первого шага. Режим 2 используется для положительного направления вращения, а режим 3 для отрицательного. Движение прекращается, когда командный байт переходит в состояние, соответствующее холостому ходу. • В режимах 6 и 7 (режимы работы с непрерывным перемещением с постоянной скоростью и с активной командой останова) модуль ускоряется до скорости, указанной в поле SPEED первого шага. Если и когда вход RPS активизируется, движение останавливается после прохождения расстояния, указанного в поле POS первого шага. (Расстояние, указанное в поле POS, должно включать в себя путь, проходимый при замедлении.) Если поле POS равно нулю, когда активизируется вход RPS, то модуль позиционирования выполняет замедление до остановки. Режим 6 используется для положительного направления вращения, а режим 7 для отрицательного. • В режимах 8 и 9 двоичное значение входа RPS устанавливает одно из двух значений скорости, указанное в первых двух шагах профильного блока. <ul style="list-style-type: none"> - Если RPS не активен: скоростью привода управляет шаг 0. - Если RPS активен: скоростью привода управляет шаг 1. Режим 8 используется для положительного направления вращения, а режим 9 для отрицательного. Значение SPEED управляет скоростью перемещения. Значения POS в этом режиме игнорируются.
<p>Команда 118 <i>Активизирует выход DIS</i></p>	<p>Когда эта команда выполняется, модуль позиционирования активизирует выход DIS.</p>
<p>Команда 119 <i>Деактивизирует выход DIS</i></p>	<p>Когда эта команда выполняется, модуль позиционирования деактивизирует выход DIS.</p>
<p>Команда 120 <i>Импульс на выходе CLR</i></p>	<p>Когда эта команда выполняется, модуль позиционирования генерирует 50–миллисекундный импульс на выходе CLR.</p>
<p>Команда 121 <i>Снова загружает текущее положение</i></p>	<p>Когда эта команда выполняется, модуль позиционирования устанавливает текущее положение на значение, находящееся в поле TARGET_POS интерактивного блока.</p>

Таблица 9–26. Команды позиционирования, продолжение

<p>Команда 122 <i>Выполнить перемещение, указанное в интерактивном блоке</i></p>	<p>Когда эта команда выполняется, модуль позиционирования выполняет операцию перемещения, указанную в поле MOVE_CMD интерактивного блока.</p> <ul style="list-style-type: none"> • В режимах 0 и 1 (режимы абсолютного и относительного позиционирования) выполняется одношаговое перемещение на основе информации о целевой скорости и положении, содержащейся в полях TARGET_SPEED и TARGET_POS интерактивного блока. • В режимах 2 и 3 (режимы работы с непрерывным перемещением с постоянной скоростью) задание положения игнорируется, и модуль позиционирования выполняет ускорение до скорости, указанной в поле TARGET_SPEED интерактивного блока. Движение прекращается, когда командный байт переходит в состояние, соответствующее холостому ходу. • В режимах 4 и 5 (режимы ручного управления скоростью) задание положения игнорируется, и ваша программа загружает значение изменений скорости в поле TARGET_SPEED интерактивного блока. Модуль позиционирования постоянно контролирует это положение и соответствующим образом реагирует на изменения скорости.
<p>Команда 123 <i>Регистрация смещения опорной точки</i></p>	<p>Когда эта команда выполняется, модуль позиционирования устанавливает нулевое положение, отличающееся от положения опорной точки. Перед выдачей этой команды вы должны определить положение опорной точки, а также перевести станок в толчковом режиме в начальное рабочее положение. После получения этой команды модуль позиционирования рассчитывает смещение между начальным рабочим положением (текущим положением) и положением опорной точки и записывает вычисленное смещение в поле RP_OFFSET интерактивного блока. Затем текущее положение устанавливается в 0, чтобы установить начальное рабочее положение в качестве нулевого положения.</p> <p>В случае, если шаговый двигатель "забывает" свое положение (например, при потере питания или изменении положения в ручном режиме), может быть выдана команда поиска опорной точки, чтобы автоматически восстановить нулевое положение.</p>
<p>Команда 124 <i>Толчковый режим, положительное направление вращения</i></p>	<p>Эта команда дает возможность выдавать импульсы вручную для перемещения шагового двигателя в положительном направлении. Если эта команда остается активной в течение менее, чем 0,5 секунды, то модуль позиционирования выдает импульсы для перемещения на расстояние, указанное в JOG_INCREMENT.</p> <p>Если команда остается активной в течение 0,5 секунды или дольше, модуль позиционирования выполняет ускорение до скорости, заданной в JOG_SPEED.</p> <p>Если обнаруживается переход в состояние холостого хода, то модуль позиционирования выполняет замедление до остановки.</p>
<p>Команда 125 <i>Толчковый режим, отрицательное направление вращения</i></p>	<p>Эта команда дает возможность выдавать импульсы вручную для перемещения шагового двигателя в отрицательном направлении. Если эта команда остается активной в течение менее, чем 0,5 секунды, то модуль позиционирования выдает импульсы для перемещения на расстояние, указанное в JOG_INCREMENT.</p> <p>Если команда остается активной в течение 0,5 секунды или дольше, модуль позиционирования выполняет ускорение до скорости, заданной в JOG_SPEED.</p> <p>Если обнаруживается переход в состояние холостого хода, то модуль позиционирования выполняет замедление до остановки.</p>
<p>Команда 126 <i>Поиск положения опорной точки</i></p>	<p>Когда эта команда выполняется, модуль позиционирования инициирует операцию поиска опорной точки, используя указанный метод поиска. Когда опорная точка найдена и перемещение остановлено, модуль позиционирования загружает значение, считанное из поля RP_OFFSET интерактивного блока, в текущее положение и генерирует на выходе CLR импульс длительностью 50 миллисекунд.</p>
<p>Команда 127 <i>Вновь загрузить настройки</i></p>	<p>Когда эта команда выполняется, модуль позиционирования считывает указатель на таблицу настроек и профилей из соответствующего адреса в специальной (SM) памяти, а затем считывает конфигурационный блок из адреса, содержащегося в указателе на таблицу настроек и профилей. Модуль позиционирования сравнивает только что полученные данные о настройках с существующей настройкой модуля и выполняет все необходимые изменения настройки или новые расчеты. Все буферизованные профили отбрасываются.</p>

Описание буфера профилей модуля позиционирования

Модуль позиционирования сохраняет необходимые для исполнения данные не более, чем для 4 профилей в буферной памяти. Когда модуль позиционирования получает команду на выполнение профиля, он проверяет, хранятся ли запрошенный профиль в буферной памяти. Если данные, необходимые для исполнения профиля, находятся в буферной памяти, то модуль позиционирования немедленно исполняет этот профиль. Если данные, необходимые для исполнения профиля, не находятся в буферной памяти, то модуль позиционирования считывает информацию профильного блока из таблицы настроек и профилей в S7-200 и рассчитывает необходимые для исполнения данные перед исполнением профиля.

Команда 122 (Выполнить перемещение, указанное в интерактивном блоке) не использует буферную память для хранения данных, необходимых для исполнения, а всегда считывает интерактивный блок из таблицы настроек и профилей в S7-200 и рассчитывает необходимые для исполнения перемещения данные.

Новая настройка модуля позиционирования удаляет из буферной памяти все хранящиеся в ней данные, необходимые для исполнения профилей.

Создание ваших собственных команд управления позиционированием

Мастер управления позиционированием создает команды позиционирования для управления работой модуля позиционирования; однако, вы тоже можете создавать свои собственные команды. Следующий сегмент кода STL дает пример того, как можно создавать свои собственные команды для модуля позиционирования.

Этот пример использует CPU 224 S7-200 с модулем позиционирования, расположенным в слоте 0. Модуль позиционирования настраивается при запуске. CMD_STAT – это символ для SMB234, CMD – это символ для QB2 и NEW_CMD – это символ для профиля.

Пример программы: Управление модулем позиционирования

```

Network 1      //Состояние: новая команда позиционирования
LSCR      State_0

Network 2      //CMD_STAT – это символ для SMB234
              //CMD – это символ для QB2
              //NEW_CMD – это символ для профиля.
              //
              //1. Очистить бит Done [Готово] модуля позиционирования.
              //2. Очистить командный байт модуля позиционирования.
              //3. Отдать новую команду.
              //4. Ожидать исполнения команды.
LD          SM0.0
MOVB      0, CMD_STAT
BIW      0, CMD
BIW      NEW_CMD, CMD
SCRT      State_1

Network 3
SCRE

Network 4      // Ожидать завершения команды.
LSCR      State_1

Network 5      //Если команда завершается без ошибок, перейти в режим холостого хода.
LDB=      CMD_STAT, 16#80
SCRT      Idle_State

Network 6      // Если команда завершается с ошибкой, перейти в режим обработки ошибок.
LDB>     CMD_STAT, 16#80
SCRT      Error_State

Network 7
SCRE

```

Режимы поиска опорной точки, поддерживаемые модулем позиционирования

На следующих рисунках представлены диаграммы различных вариантов поиска опорной точки.

- На рис. 9–21 показано два варианта для 1-го режима поиска опорной точки. В этом режиме положение опорной точки ищется там, где вход RPS становится активным при приближении со стороны рабочей зоны.
- На рис. 9–22 показано два варианта для 2-го режима поиска опорной точки. В этом режиме положение опорной точки ищется в центре области активности входа RPS.
- На рис. 9–23 показано два варианта для 3-го режима поиска опорной точки. В этом режиме положение опорной точки ищется на указанном числе нулевых импульсов (ZP) расстоянии вне области активности входа RPS.
- На рис. 9–24 показано два варианта для 4-го режима поиска опорной точки. В этом режиме положение опорной точки ищется на указанном числе нулевых импульсов (ZP) расстоянии внутри области активности входа RPS.

Для каждого режима имеется четыре комбинации направления поиска опорной точки и направления приближения к опорной точке. (Показаны только две из этих комбинаций.) Эти комбинации дают образец поиска опорной точки. Для каждой комбинации имеются также четыре исходных точки.

Рабочие зоны для каждой диаграммы расположены таким образом, что перемещение из опорной точки к рабочей зоне требует перемещения в том же направлении, что и при приближении к опорной точке. При таком выборе положения рабочей зоны все зазоры механической системы передач устраняются из первого перемещения к рабочей зоне после поиска опорной точки.

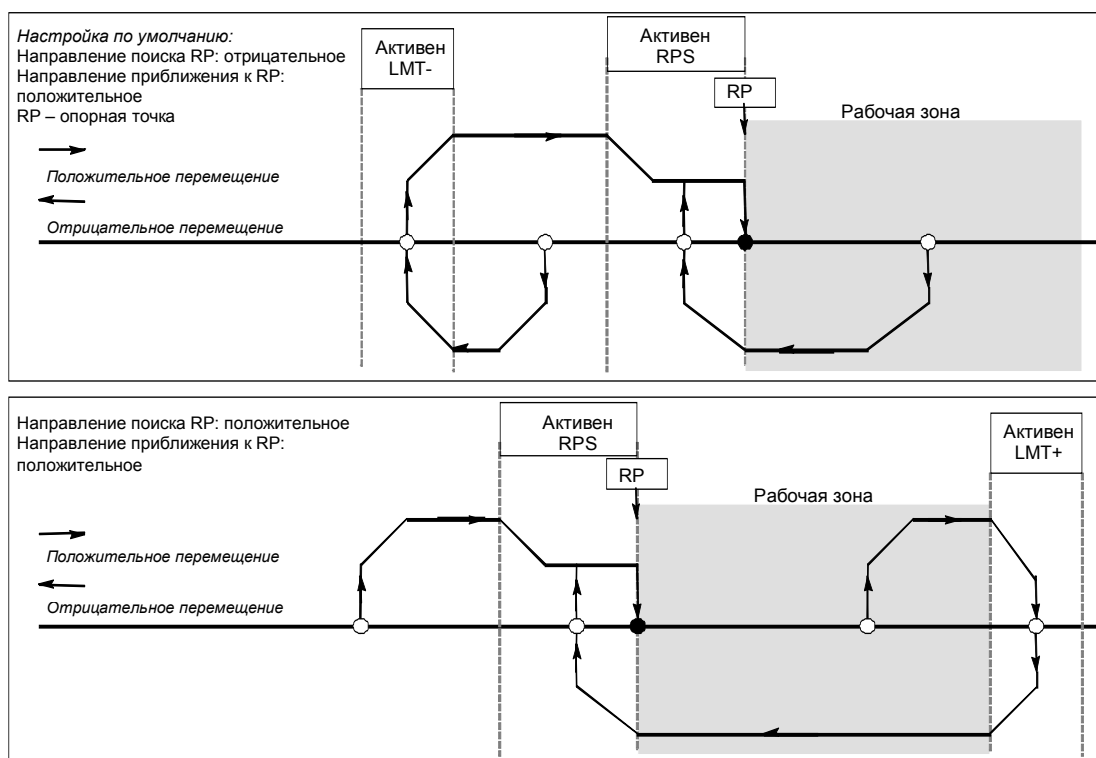


Рис. 9–21. 1-й режим поиска опорной точки

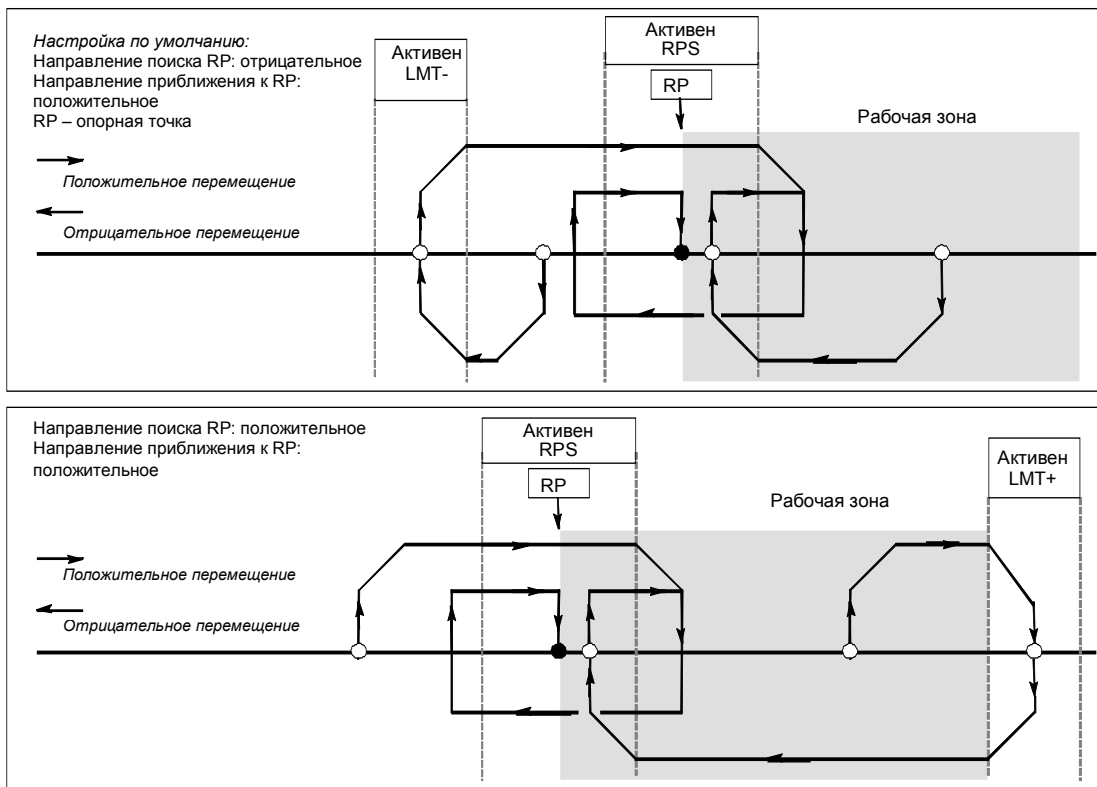


Рис. 9–22. Поиск опорной точки: Режим 2

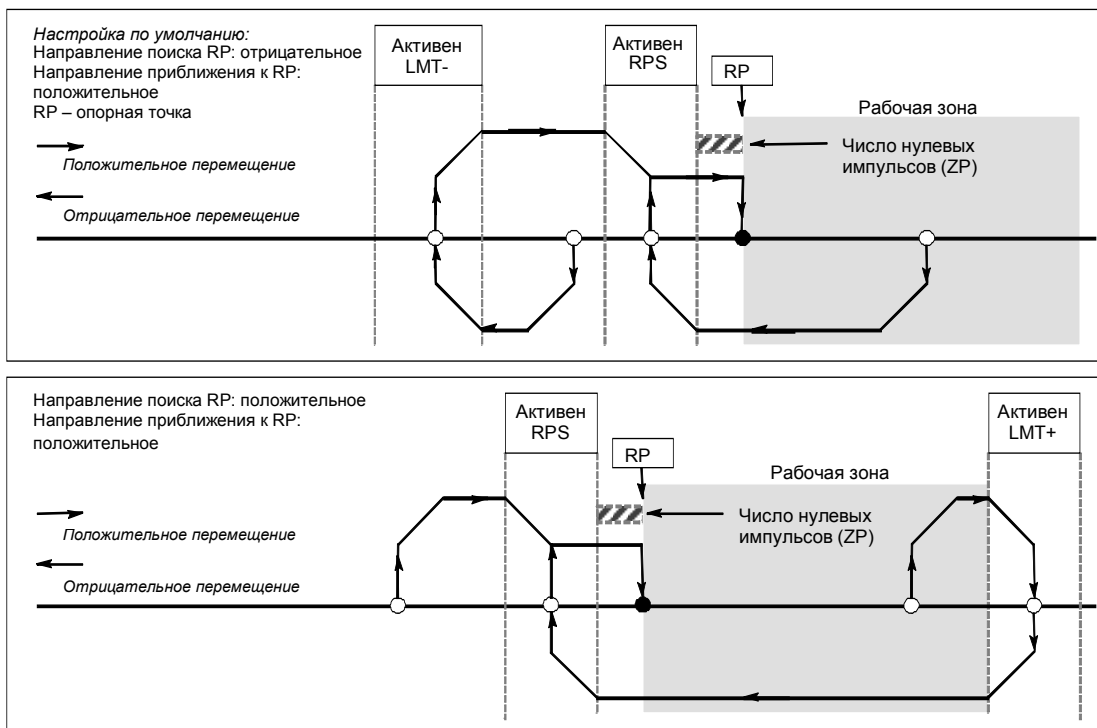


Рис. 9–23. Поиск опорной точки: Режим 3

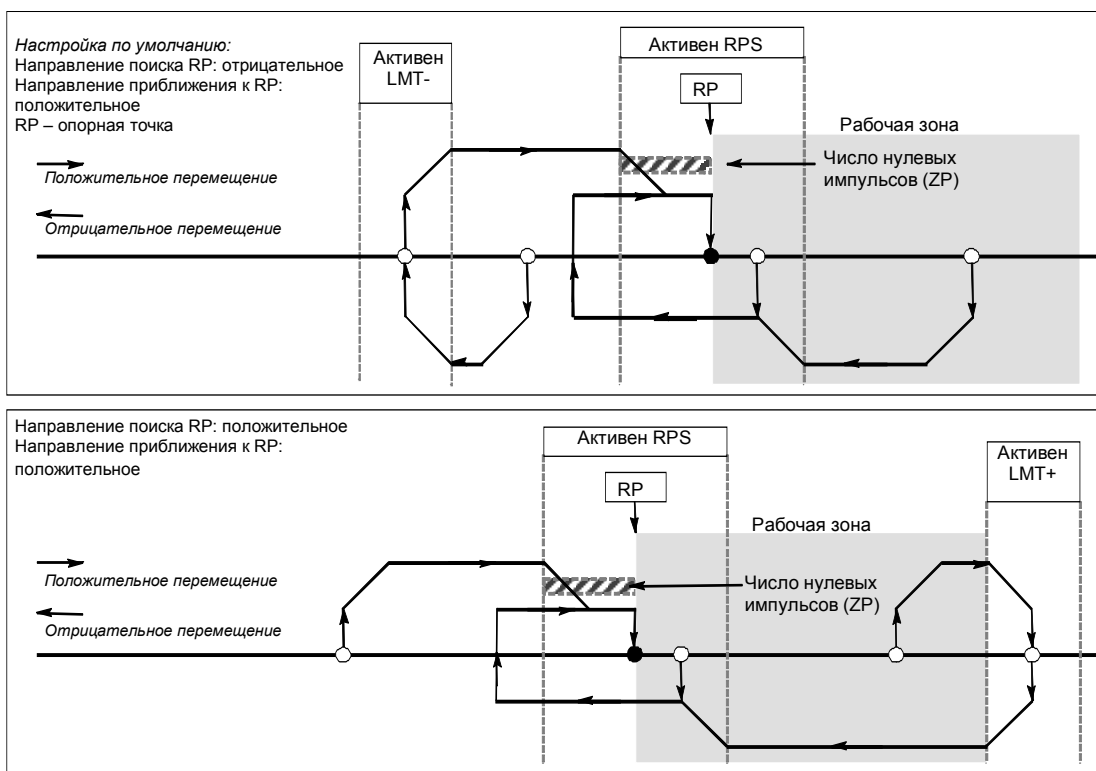


Рис. 9–24. Поиск опорной точки: Режим 4

Выбор положения рабочей зоны для устранения зазоров

На рис. 9–25 показано положение рабочей зоны относительно опорной точки (RP), зоны активности RPS и конечных выключателей (LMT+ и LMT-) для направления приближения, устраняющего зазоры. Во второй части рисунка рабочая зона расположена так, что зазор не устраняется. На рис. 9–25 показан поиск опорной точки в режиме 3. Подобное положение рабочей зоны возможно, хотя и не рекомендуется, для каждой из последовательностей поиска других режимов поиска опорной точки.

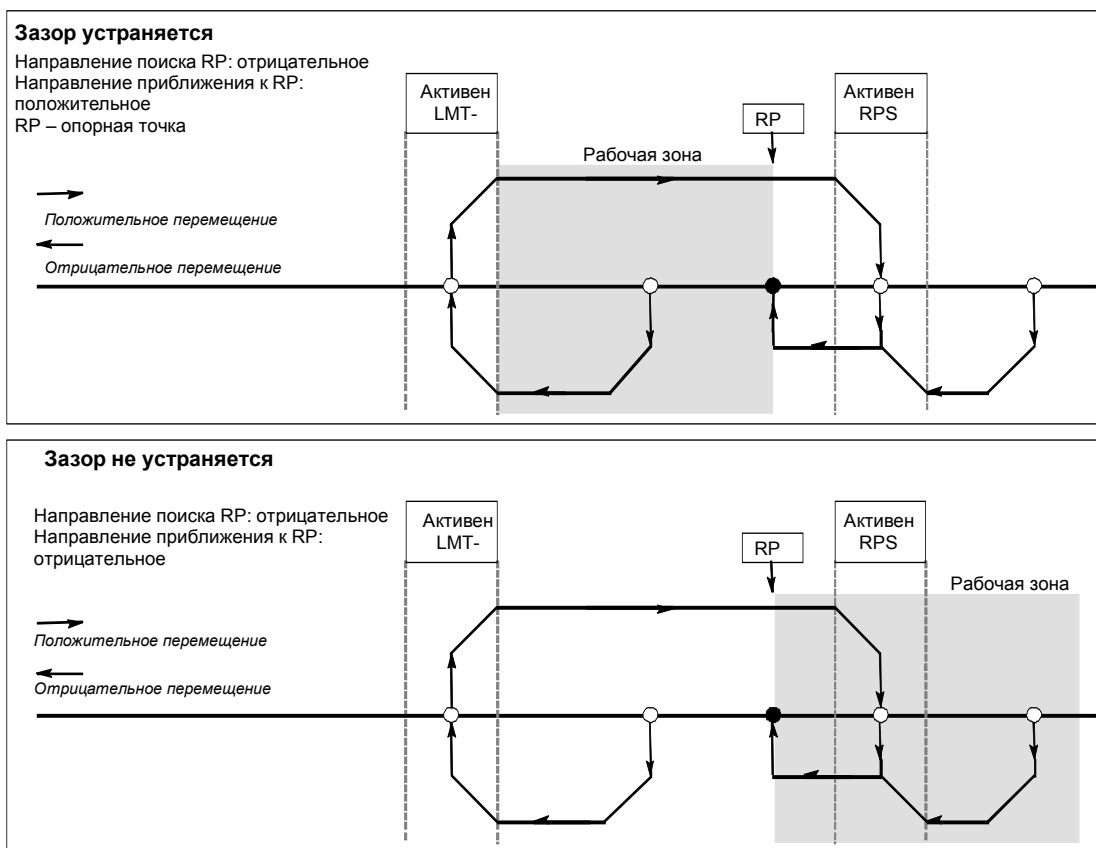


Рис. 9–25- Расположение рабочей зоны с устранением и без устранения зазора

Создание программы для модемного модуля

10

Модемный модуль EM 241 дает возможность подключать S7-200 непосредственно к аналоговой телефонной линии и поддерживает обмен данными между вашим S7-200 и STEP 7-Micro/WIN. Модемный модуль поддерживает также протокол Modbus slave RTU. Обмен данными между модемным модулем и S7-200 осуществляется через шину расширения ввода/ вывода.

С помощью Мастера расширения функций модема (Modem Expansion wizard), предоставляемого STEP 7-Micro/WIN, вы можете настроить удаленный модем или модемный модуль для соединения локального S7-200 с удаленным устройством.

В этой главе

Функции модемного модуля	310
Использование Мастера расширения функций модема для настройки модемного модуля	316
Обзор команд и ограничений модема	320
Команды для модемного модуля	321
Пример программы для модемного модуля	325
CPU S7-200, поддерживающие интеллектуальные модули	325
Биты специальной памяти для модемного модуля	326
Для опытных пользователей	328
Формат телефонных номеров для передачи сообщений	330
Формат текстовых сообщений	331
Формат сообщений для передачи данных CPU	332

Функции модемного модуля

Модемный модуль дает возможность подключать S7-200 непосредственно к аналоговой телефонной линии и имеет следующие характеристики:

- предоставляет интерфейс с международными телефонными линиями
- предоставляет модемный интерфейс со STEP 7-Micro/WIN для программирования и устранения неисправностей (телесервис)
- поддерживает протокол Modbus RTU
- поддерживает цифровую и текстовую пейджинговую связь
- поддерживает передачу коротких сообщений (SMS)
- допускает передачу данных от CPU к CPU и от CPU к Modbus
- предоставляет парольную защиту
- обеспечивает обратный вызов для предотвращения несанкционированного доступа к данным
- Конфигурация модемного модуля хранится в CPU

Для настройки модемного модуля можно использовать Мастер расширения функций модема (Modem Expansion wizard) STEP 7-Micro/WIN. Технические данные модемного модуля вы найдете в Приложении А.

Интерфейс с международными телефонными линиями

Модемный модуль представляет собой стандартный 10-битовый модем V.34 (33,6 кБод), совместимый с большинством внутренних и внешних модемов персональных компьютеров. Модемный модуль не обменивается данными с 11-битовыми модемами.

Контакт	Описание	
3	Звонок	Допустимо обратное присоединение
4	Оконеч. контакт	

Рис. 10-2. Вид штекера RJ11

Модемный модуль подключается к телефонной линии через шестиконтактный четырехпроводный разъем RJ11, находящийся на передней стороне модуля. См. рис. 10-2.

В зависимости от страны вам, возможно, понадобится переходное устройство, чтобы приспособить RJ11 к стандартной телефонной линии этой страны.

Подробную информацию о штекере переходного устройства вы найдете в документации к этому устройству.

Модем и интерфейс с телефонной линией получают питание от внешнего источника 24 В постоянного тока. Их можно подключить к источнику питания датчиков CPU или к внешнему источнику. Соедините заземляющую клемму на модемном модуле с землей системы.

При подаче на модемный модуль питания он автоматически настраивает интерфейс с телефонной линией для работы с конкретной страной. Код страны набирается с помощью двух поворотных переключателей на передней стороне модуля. Эти переключатели необходимо установить на желаемую страну до подачи питания на модемный модуль. Положения переключателей для поддерживаемых стран приведены в таблице 10-1.

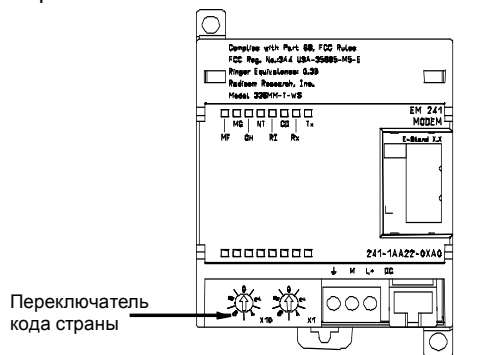


Рис. 10-1. Модемный модуль EM 241

Таблица 10-1. Страны, поддерживаемые EM 241

Положение переключателей	Страна
01	Австрия
02	Бельгия
05	Канада
08	Дания
09	Финляндия
10	Франция
11	Германия
12	Греция
16	Ирландия
18	Италия
22	Люксембург
25	Нидерланды
27	Норвегия
30	Португалия
34	Испания
35	Швеция
36	Швейцария
38	Великобритания
39	США

Интерфейс со STEP 7-Micro/WIN

Модемный модуль допускает обмен данными со STEP 7-Micro/WIN через телефонную линию (телесервис). При работе со STEP 7-Micro/WIN вам не нужно настраивать или программировать CPU S7-200, чтобы использовать модемный модуль в качестве удаленного модема.

Для использования модемного модуля со STEP 7-Micro/WIN действуйте следующим образом:

1. Снимите питание с CPU S7-200 и присоедините модемный модуль к шине расширения ввода/вывода. Не подключайте модули ввода/вывода, пока CPU S7-200 находится под напряжением.
2. Подключите к модемному модулю телефонную линию. В случае необходимости используйте переходное устройство.
3. Подключите питание 24 В постоянного тока к клеммному блоку модемного модуля.
4. Подключите заземляющую клемму клеммного блока модемного модуля к земле системы.
5. Установите в нужное положение переключатели кода страны.
6. Подайте напряжение на CPU S7-200 и модемный модуль.
7. Настройте STEP 7-Micro/WIN на обмен данными с 10-битовым модемом.

Протокол Modbus RTU

Модемный модуль можно настроить так, чтобы он функционировал как slave-устройство Modbus RTU. Модемный модуль принимает запросы Modbus через модемный интерфейс, интерпретирует эти запросы и передает данные в CPU или из него. После этого модемный модуль генерирует ответ Modbus и передает его через модемный интерфейс.



Совет

Если модемный модуль настроен как slave-устройство Modbus RTU, то STEP 7-Micro/WIN не может обмениваться данными с модемным модулем через телефонную линию.

Модемный модуль поддерживает функции Modbus, приведенные в таблице 10-2.

Функции 4 и 16 Modbus позволяют считывать или записывать до 125 регистров временного хранения информации (250 байтов в памяти переменных) в одном запросе. Функции 5 и 15 ведут запись в регистр выходов образа процесса CPU. Эти значения могут быть перезаписаны программой пользователя.

Адреса Modbus обычно записываются в виде значений из 5 или 6 символов, содержащих тип данных и смещение. Один или два первых символа определяют тип данных, а последние четыре символа выбирают надлежащее значение внутри этого типа данных. Master-устройство Modbus отображает эти адреса на соответствующие функции Modbus.

Таблица 10-2. Функции Modbus, поддерживаемые модемным модулем

Функция	Описание
Функция 01	Прочитать состояние выхода (катушки)
Функция 02	Прочитать состояние входа
Функция 03	Прочитать регистр временного хранения информации
Функция 04	Прочитать регистр входов (аналоговых)
Функция 05	Записать в один выход (катушку)
Функция 06	Заранее установить один регистр
Функция 15	Записать в несколько выходов (катушек)
Функция 16	Заранее установить несколько регистров

В таблице 10–3 показаны адреса Modbus, поддерживаемые модемным модулем, и отображение адресов Modbus в адреса CPU S7–200.

Создайте с помощью Мастера расширения функций модема конфигурационный блок, чтобы модемный модуль поддерживал протокол Modbus RTU. Прежде чем вы сможете использовать протокол Modbus, конфигурационный блок модемного модуля должен быть загружен в блок данных CPU.

Таблица 10–3. Отображение адресов Modbus в адреса CPU S7–200

Адрес Modbus	Адрес CPU S7–200
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	VW0
040002	VW2
040003	VW4
...	...
04xxxx	VW 2*(xxxx-1)

Пейджинговая связь и передача сообщений SMS

Модемный модуль поддерживает передачу цифровых и текстовых пейджинговых сообщений и сообщений SMS (Short Message Service [Служба коротких сообщений]) на сотовые телефоны (если они поддерживаются провайдером сотовой связи). Сообщения и телефонные номера хранятся в конфигурационном блоке модемного модуля, который должен быть загружен в блок данных CPU S7–200. Для создания сообщений и телефонных номеров конфигурационного блока модемного модуля можно воспользоваться Мастером расширения функций модема. Мастер расширения функций модема создает также программный код, позволяющий вашей программе инициировать передачу сообщений.

Цифровая пейджинговая связь

Цифровая пейджинговая связь использует для передачи цифровых значений на пейджер тональные сигналы кнопочного телефона. Модемный модуль набирает запрашиваемую пейджинговую службу, ожидает конца речевого сообщения, а затем передает тональные сигналы, соответствующие цифрам пейджингового сообщения. В пейджинговом сообщении разрешены цифры от 0 до 9, астериск (*), A, B, C и D. Символы, фактически отображаемые пейджером вместо астериска и A, B, C и D, не стандартизованы и определяются пейджером и провайдером пейджинговой связи.

Текстовая пейджинговая связь

Текстовая пейджинговая связь допускает передачу провайдеру пейджинговой связи и от него на пейджер алфавитно-цифровых сообщений. Провайдеры текстовой пейджинговой связи обычно имеют линию с модемом, которая принимает текстовые сообщения. Модемный модуль использует для передачи текстовых сообщений провайдеру пейджинговой связи алфавитно-цифровой протокол Telelocator (Telelocator Alphanumeric Protocol, TAP). Многие провайдеры текстовой пейджинговой связи используют этот протокол для приема сообщений.

Служба коротких сообщений (SMS)

Передача сообщений через Службу коротких сообщений (Short Message Service, SMS) поддерживается некоторыми службами сотовой телефонной связи, обычно теми, которые совместимы со стандартом GSM. SMS позволяет модемному модулю передавать сообщение по аналоговой телефонной линии провайдеру SMS. Затем провайдер SMS передает это сообщение на сотовый телефон, и сообщение появляется на текстовом дисплее телефона. Модемный модуль использует для передачи сообщений провайдеру SMS алфавитно-цифровой протокол Telelocator (Telelocator Alphanumeric Protocol, TAP) и универсальный компьютерный протокол (Universal Computer Protocol, UCP). SMS-сообщения можно передавать провайдеру SMS только в том случае, если он поддерживает эти протоколы на линии с модемом.

Переменные, вставленные в текстовые сообщения и в короткие сообщения SMS

Модемный модуль может вставлять значения данных из CPU в текстовые сообщения и форматировать значения этих данных на основе спецификации, имеющейся в сообщении. Вы можете задать количество цифр слева и справа от десятичной точки, и указать, должен ли этот разделитель быть точкой или запятой. Когда программа пользователя отдает модемному модулю команду на передачу текстового сообщения, модемный модуль извлекает это сообщение из CPU, определяет, какие значения из CPU должны находиться внутри сообщения, извлекает эти значения из CPU, а затем форматирует и помещает их внутри текстового сообщения перед его передачей соответствующему провайдеру.

Телефонный номер провайдера передачи сообщений, сообщение и переменные, вставленные в сообщение, считываются из CPU за несколько циклов CPU. Во время передачи сообщения ваша программа не должна изменять телефонные номера и тексты сообщений. Переменные, вставленные в сообщение, во время передачи сообщения могут продолжать обновляться. Если сообщение содержит несколько переменных, то эти переменные считываются за несколько циклов CPU. Если вы хотите, чтобы значения всех переменных, вставленных в сообщение, были непротиворечивы, вы не должны изменять ни одну из вставленных переменных после того, как вы отправили сообщение.

Передача данных

Модемный модуль дает возможность вашей программе передавать данные другому CPU или устройству Modbus через телефонную линию. Передаваемые данные и телефонные номера конфигурируются с помощью Мастера расширения функций модема и сохраняются в конфигурационном блоке модемного модуля. Затем конфигурационный блок загружается в блок данных в CPU S7-200. Мастер расширения функций модема создает также программный код, позволяющий вашей программе инициировать передачу данных.

Передача данных может быть или запросом на чтение данных из удаленного устройства, или запросом на запись данных в удаленное устройство. За одну передачу можно прочитать или записать от 1 до 100 слов данных, при этом данные передаются из памяти переменных или в память переменных присоединенного CPU.

С помощью Мастера расширения функций модема можно сконфигурировать передачу данных, во время которой производится одно считывание из удаленного устройства, одна запись в удаленное устройство или считывание из удаленного устройства и запись в него.

При передаче данных используется сконфигурированный протокол модемного модуля. Если модемный модуль настроен на поддержку протокола PPI (тогда он может реагировать на STEP 7-Micro/WIN), то и для передачи данных используется протокол PPI. Если модемный модуль настроен на поддержку протокола Modbus RTU, то данные передаются с помощью протокола Modbus.

Телефонный номер удаленного устройства, запрос на передачу данных и передаваемые данные считываются из CPU в течение нескольких циклов CPU. Во время передачи сообщения ваша программа не должна изменять телефонные номера и тексты сообщений. Вы тоже не должны изменять передаваемые данные, когда идет передача сообщения.

Если удаленным устройством является другой модемный модуль, то при передаче данных может быть применена функция защиты с помощью пароля путем ввода пароля удаленного модемного модуля в конфигурацию телефонного номера. Функция обратного вызова при передаче данных использоваться не может.

Защита с помощью пароля

Защита модемного модуля паролем не обязательна и активизируется с помощью мастера расширения функций модема. Пароль, используемый модемным модулем, отличается от пароля CPU. Модемный модуль использует собственный пароль из 8 символов, который абонент (вызывающая программа) должен предоставить модемному модулю, прежде чем он получит доступ к присоединенному CPU. Пароль хранится в памяти переменных CPU как часть конфигурационного блока модемного модуля. Конфигурационный блок модемного модуля должен быть загружен в блок данных присоединенного CPU.

Если в блоке системных данных активизирована парольная защита CPU, то абонент (вызывающая программа) должен предоставить этот пароль CPU, чтобы получить доступ к функциям, защищенным паролем.

Обратный вызов для защиты от несанкционированного доступа

Функция обратного вызова модемного модуля не обязательна и конфигурируется с помощью Мастера расширения функций модема. Функция обратного вызова обеспечивает дополнительную безопасность для присоединенного CPU, разрешая допуск к CPU только для заранее определенных телефонных номеров. Когда функция обратного вызова разблокирована, модемный модуль отвечает на все поступающие вызовы, проверяет абонента, а затем отсоединяет линию. Если абонент авторизован, то после этого модемный модуль набирает заранее определенный телефонный номер этого абонента и разрешает допуск к CPU.

Модемный модуль поддерживает три режима обратного вызова:

- Обратный вызов для одного заранее определенного телефонного номера
- Обратный вызов для нескольких заранее определенных телефонных номеров
- Обратный вызов для любых телефонных номеров

Режим обратного вызова выбирается проверкой соответствующей опции в Мастере расширения функций модема, а затем определением телефонных номеров для обратного вызова. Телефонные номера для обратного вызова хранятся в конфигурационном блоке модемного модуля, находящемся в блоке данных присоединенного CPU.

Простейшей формой обратного вызова является обратный вызов для одного заранее определенного телефонного номера. Если в конфигурационном блоке модемного модуля для обратного вызова хранится только один номер, всякий раз, когда модемный модуль отвечает на поступающий вызов, он уведомляет абонента, что обратный вызов активизирован, отсоединяет абонента, а затем набирает номер для обратного вызова, указанный в конфигурационном блоке.

Модемный модуль поддерживает обратный вызов также для нескольких заранее определенных телефонных номеров. В этом режиме у абонента запрашивается телефонный номер. Если указанный номер совпадает с одним из заранее определенных телефонных номеров в конфигурационном блоке модемного модуля, то модемный модуль отсоединяет абонента, а затем звонит обратно, используя совпадающий телефонный номер из конфигурационного блока. Пользователь может сформировать до 250 телефонных номеров для обратного вызова.

Если для обратного вызова заранее определено несколько телефонных номеров, то номер, сообщенный абонентом при соединении с модемным модулем, должен точно совпадать с номером в конфигурационном блоке модемного модуля кроме первых двух цифр. Например, если сформированный для обратного вызова номер равен 91(123)4569999 из-за необходимости набора внешней телефонной линии (9) и большого расстояния (1), то сообщенный абонентом номер для обратного вызова может быть одним из следующих:

- 91(123)4569999
- 1(123)4569999
- (123)4569999

Все вышеприведенные телефонные номера рассматриваются как совпадающие для обратного вызова. Модемный модуль при выполнении обратного вызова использует телефонный номер из своего конфигурационного блока, в данном примере 91(123)4569999. При формировании нескольких телефонных номеров для обратного вызова обратите внимание на то, чтобы все телефонные номера были уникальными кроме первых двух цифр. При сравнении телефонных номеров для обратного вызова в телефонном номере используются только цифровые символы. При сравнении телефонных номеров для обратного вызова символы, подобные запятым и скобкам, игнорируются.

Обратный вызов для любых телефонных номеров устанавливается в Мастере расширения функций модема во время конфигурирования обратных вызовов выбором опции «Enable callbacks to any phone number [Активизировать обратные вызовы для любых телефонных номеров]». Если эта опция выбрана, то модемный модуль отвечает на поступивший вызов и запрашивает телефонный номер для обратного вызова. После сообщения абонентом своего телефонного номера модемный модуль отсоединяется и набирает этот номер. Этот режим обратного вызова всего лишь дает средство возложить плату за телефонное соединение на абонента, которому принадлежит модемный модуль, и не обеспечивает безопасности для CPU S7-200. В этом режиме обратного вызова для обеспечения безопасности следует использовать пароль модемного модуля.

Пароль модемного модуля и функции обратного вызова могут быть активизированы одновременно. При активизированной парольной защите модемный модуль перед выполнением обратного вызова требует от абонента сообщить правильный пароль.

Конфигурационная таблица для модемного модуля

Все текстовые сообщения, телефонные номера, информация для передачи данных, номера для обратного вызова и другие опции хранятся в конфигурационной таблице модемного модуля, которая должна быть загружена в память переменных CPU S7-200. Мастер расширения функций модема руководит вами в процессе создания конфигурационной таблицы модемного модуля. Затем STEP 7-Micro/WIN помещает конфигурационную таблицу модемного модуля в блок данных, который загружается в CPU S7-200.

Модемный модуль считывает эту конфигурационную таблицу из CPU при запуске и в течение пяти секунд при каждом переходе CPU из STOP в RUN. Модемный модуль не считывает новую конфигурационную таблицу из CPU, пока модемный модуль находится в режиме online со STEP 7-Micro/WIN. Если новая конфигурационная таблица загружается, когда модемный модуль находится в режиме online, то модемный модуль считывает новую конфигурационную таблицу по окончании сеанса online.

Если модемный модуль обнаруживает ошибку в конфигурационной таблице, то на передней панели модуля мигает светодиод MG (Module Good – модуль в порядке). Проверьте экран с информацией о ПЛК в STEP 7-Micro/WIN или прочитайте значение в SMW220 (для слота 0 модуля), чтобы получить информацию об ошибке в конфигурации. Ошибки конфигурации модемного модуля приведены в таблице 10–4. Если для создания конфигурационной таблицы модемного модуля вы пользуетесь Мастером расширения функций модема, то STEP 7-Micro/WIN проверяет данные перед созданием конфигурационной таблицы.

Таблица 10–4. Ошибки конфигурации EM 241 (в шестнадцатеричном коде)

Ошибка	Описание
0000	Нет ошибки
0001	Отсутствует внешний источник питания 24 В пост. тока
0002	Вышел из строя модем
0003	Отсутствует идентификатор конфигурационного блока - Идентификация EM 241 в начале конфигурационной таблицы для этого модуля недействительна.
0004	Конфигурационный блок вне области - Указатель конфигурационной таблицы указывает не область памяти переменных, или часть таблицы находится вне области памяти переменных для присоединенного CPU.
0005	Ошибка конфигурации - Активизирован обратный вызов, а количество телефонных номеров для обратного вызова равно 0 или больше 250. Количество сообщений больше 250. Количество телефонных номеров для передачи сообщений больше 250, или длина телефонных номеров для передачи сообщений больше 120 байтов.
0006	Ошибочный код страны – Код страны, установленный с помощью двух поворотных переключателей, не поддерживается.
0007	Телефонный номер слишком велик – Активизирован обратный вызов, а длина номера обратного вызова больше максимума.
от 0008 до 00FF	Резерв
01xx	Ошибка в номере обратного вызова xx – В телефонном номере для обратного вызова xx имеются недопустимые символы. Значение xx равно 1 для первого номера обратного вызова, 2 для второго и т.д.
02xx	Ошибка в телефонном номере xx – Одно из полей в телефонном номере для передачи сообщений xx или в телефонном номере для передачи данных xx содержит недопустимое значение. Значение xx равно 1 для первого телефонного номера, 2 для второго и т.д.
03xx	Ошибка в сообщении xx – Номер для передачи сообщений или данных xx превышает максимально допустимую длину. Значение xx равно 1 для первого сообщения, 2 для второго и т.д.
от 0400 до FFFF	Резерв

Светодиоды состояний модемного модуля

Модемный модуль имеет на передней панели 8 светодиодов состояний. Светодиоды состояний описаны в таблице 10–5.

Таблица 10–5. Светодиоды состояний EM 241

Светодиод	Описание
MF	Модуль неисправен - Этот светодиод включен, когда модуль обнаруживает состояние неисправности: <ul style="list-style-type: none"> ■ Отсутствует внешний источник питания 24 В пост. тока ■ Истекло время контроля ввода/вывода ■ Вышел из строя модем ■ Ошибка обмена данными с локальным CPU
MG	Модуль исправен - Этот светодиод включен, когда в модуле отсутствует состояние неисправности. Светодиод "Модуль исправен" мигает, если имеется ошибка в конфигурационной таблице, или пользователь выбрал недопустимый код страны для интерфейса с телефонной линией. Проверьте экран с информацией о ПЛК в STEP 7-Micro/WIN или прочитайте значение в SMW220 (для слота 0 модуля), чтобы получить информацию об ошибке в конфигурации.
OH	Занято - Этот светодиод включен, когда EM 241 активно использует телефонную линию.
NT	Нет тонального вызова - Этот светодиод указывает на наличие ошибочного состояния, он включается, когда EM 241 получил команду передать сообщение, а в телефонной линии нет тонального вызова. Это является ошибочным состоянием только в том случае, если EM 241 настроен на проверку наличия тонального вызова перед набором номера. Светодиод остается включенным в течение примерно 5 секунд после попытки неудачного набора.
RI	Индикатор звонка –Этот светодиод показывает, что EM 241 принимает поступающий вызов.
CD	Обнаружена несущая - Этот светодиод показывает, что установлено соединение с удаленным модемом.
Rx	Прием данных – Этот светодиод мигает, когда модем ведет прием данных.
Tx	Передача данных – Этот светодиод мигает, когда модем ведет передачу данных.

Использование Мастера расширения функций модема для настройки модемного модуля



Расширение модема

Запустите Мастер расширения функций модема из меню Tools [Инструментальные средства] в STEP 7-Micro/WIN или из соответствующего раздела навигационной панели.

Для использования этого мастера проект должен быть скомпилирован и должен быть включен режим символической адресации. Если ваша программа еще не скомпилирована, сделайте это сейчас.

1. На первом экране Мастера расширения функций модема выберите опцию Configure an EM 241 Modem module [Сконфигурировать модемный модуль EM 241] и щелкните на Next> [Дальше >].
2. Мастеру расширения функций модема необходимо расположение модемного модуля относительно CPU S7–200, чтобы сгенерировать правильный программный код. Щелкните на кнопке Read Modules [Читать модули], чтобы автоматически считать положения интеллектуальных модулей, присоединенных к CPU. Модули расширения нумеруются последовательно, начиная с нуля. Дважды щелкните на модемном модуле, который вы хотите сконфигурировать, или установите положение модемного модуля в поле Module Position [Положение модуля]. Щелкните на Next> [Дальше >].

Для CPU S7–200 с версией ПЗУ, меньшей 1.2, интеллектуальный модуль необходимо устанавливать рядом с CPU, чтобы Мастер расширения функций модема мог сконфигурировать этот модуль.

3. Экран парольной защиты дает возможность активизировать парольную защиту для модемного модуля и назначить для модуля пароль длиной от 1 до 8 символов. Этот пароль не зависит от пароля CPU S7–200. Когда модуль защищен паролем, то всякий, кто пытается соединиться с CPU S7–200 через модемный модуль, получает требование сообщить правильный пароль. Если необходимо, выберите парольную защиту и введите пароль. Щелкните на Next> [Дальше >].

4. Модемный модуль поддерживает два протокола связи: протокол PPI (для обмена данными со STEP 7-Micro/WIN) и протокол Modbus RTU. Выбор протокола зависит от типа устройства, которое используется в качестве удаленного партнера по обмену данными. Эта настройка управляет протоколом связи, используемым, когда модемный модуль отвечает на вызов и когда он инициирует передачу данных CPU. Выберите соответствующий протокол и щелкните на Next> [Дальше >].
5. Вы можете настроить модуль на передачу цифровых и текстовых сообщений на пейджеры или коротких сообщений (SMS) на сотовые телефоны. Отметьте триггерную кнопку Enable messaging [Разблокировать передачу сообщений] и щелкните на кнопке Configure Messaging... [Сконфигурировать передачу сообщений...], чтобы определить сообщения и телефонные номера адресатов.
6. При конфигурировании сообщения, подлежащего передаче на пейджер или сотовый телефон, вы должны определить сообщение и телефонный номер. Выберите закладку Messages [Сообщения] на экране Configure Messaging [Конфигурирование передачи сообщений] и щелкните на кнопке New Message [Новое сообщение]. Введите текст сообщения и укажите значения данных CPU, которые должны быть вставлены в сообщение. Чтобы вставить значение данных CPU в сообщение, поместите курсор в то место, куда должны быть вставлены данные, и щелкните на кнопке Insert Data... [Вставить данные...]. Укажите адрес, по которому находится значение данных CPU (напр., VW100), формат отображения (напр., Signed Integer [Целое со знаком]) и количество цифр слева и справа от десятичной точки. Вы можете также указать, каким должен быть разделитель целой и дробной части – точкой или запятой.
 - Сообщения для цифровой пейджерной связи ограничиваются цифрами от 0 до 9, буквами A, B, C и D и астериском (*). Максимально допустимая длина цифрового сообщения пейджинговой связи различна у различных провайдеров.
 - Текстовые сообщения могут иметь длину до 119 символов и содержать любые алфавитно-цифровые символы.
 - Текстовые сообщения могут содержать любое количество встроенных переменных.
 - Встроенные переменные могут быть из областей памяти V, M, SM, I, Q, S, T, C или AI в присоединенном CPU.
 - Шестнадцатеричные данные отображаются с ведущими символами '16#'. Количество символов в значении зависит от размера переменной. Например, VW100 отображается как 16#0123.
 - Количество символов слева от десятичного разделителя должно быть достаточно большим, чтобы отображать ожидаемый диапазон значений, включая знак отрицания, если значение данных является целым числом со знаком или числом с плавающей точкой.
 - Если формат данных является целым, а количество цифр справа от десятичной точки не равно нулю, то целое значение отображается как масштабированное целое число. Например, если VW100 = 1234 и справа от десятичной точки имеется две цифры, то данные отображаются как '12.34'.
 - Если значение данных больше, чем может быть отображено в поле указанного размера, то модемный модуль в позициях для всех символов помещает символ #.
7. Телефонные номера конфигурируются выбором закладки Phone Numbers [Телефонные номера] на экране Configure Messaging [Конфигурирование передачи сообщений]. Щелкните на кнопке New Phone Number... [Новый телефонный номер...], чтобы добавить новый телефонный номер. Когда телефонный номер сконфигурирован, его нужно добавить в проект. Выделите этот телефонный номер в столбце Available Phone Numbers [Доступные телефонные номера] и щелкните на стрелке, показывающей вправо, чтобы добавить телефонный номер в текущий проект. После того как вы добавили телефонный номер в текущий проект, вы можете его выбрать и присвоить ему символическое имя для использования в своей программе.

Телефонный номер состоит из нескольких полей, которые различаются в зависимости вида передачи сообщений, выбранного пользователем.

 - Выбор протокола для передачи сообщений (Messaging Protocol) указывает модемному модулю, какой протокол необходимо использовать для передачи сообщения провайдеру службы передачи сообщений. Цифровые пейджеры поддерживают только цифровой протокол. Службы текстовой пейджинговой связи обычно требуют протокола TAP (Telelocator Alphanumeric Protocol [Алфавитно-цифровой протокол Телелокатор]). Провайдеры службы коротких сообщений (SMS) поддерживают протокол TAP или UCP (Universal Computer Protocol [Универсальный компьютерный протокол]). Имеются три различных службы UCP, обычно используемые для передачи SMS-сообщений. Большинство провайдеров поддерживает команды 1 и 51. Проверьте, какой протокол и какие команды поддерживает ваш провайдер SMS-сообщений.

- Поле Description [Описание] дает вам возможность добавить к телефонному номеру текстовое описание.
 - Поле Phone Number [Телефонный номер] – это номер телефона провайдера службы передачи сообщений. Для текстовых сообщений это телефонный номер модемной линии, которую провайдер использует для приема текстовых сообщений. Для цифровой пейджинговой связи это телефонный номер самого пейджера. Модемный модуль позволяет ввести в поле для телефонного номера не более 40 символов. В телефонных номерах, которые модемный модуль использует для набора, допускается использование следующих символов:

0 – 9	допустимы с телефонной клавиатуры
A, B, C, D, *, #	цифры двухтонального многочастотного набора (DTMF) (только при тональном наборе))
,	остановка набора на 2 секунды
!	команда модему генерировать сигнал занятости
@	ожидание 5 секунд молчания
W	ожидание тонального вызова перед продолжением
()	игнорируются (могут использоваться для форматирования телефонного номера)
 - В поле Specific Pager ID or Cell Phone Number [Идентификатор определенного пейджера или номер сотового телефона] вводится номер пейджера или номер сотового телефона приемника сообщения. Этот номер не должен содержать никаких символов кроме цифр от 0 до 9. Допустимо не более 20 символов.
 - Поле Password [Пароль] для TAP-сообщений является необязательным. Некоторые провайдеры требуют ввода пароля, но обычно это поле должно оставаться пустым. Модемный модуль допускает пароль длиной до 15 символов.
 - Поле Originating Phone Number [Исходящий телефонный номер] позволяет идентифицировать модемный модуль в SMS-сообщениях. Это поле требуется некоторыми провайдерами служб, использующих команды UCP. Некоторые провайдеры могут требовать в этом поле минимального количества символов. Модемный модуль допускает до 15 символов.
 - Поле Modem Standard [Стандарт модема] предназначено для случаев, когда модемный модуль и модем провайдера не могут найти общий стандарт для модема. По умолчанию V.34 (33,6 кБод).
 - Поля Data Format [Формат данных] позволяют настраивать биты данных и способ проверки четности, используемые модемом при передаче сообщения провайдеру. TAP обычно использует 7 битов данных и проверку на четность, но некоторые провайдеры используют 8 битов данных и не проверяют четность. UCP всегда использует 8 битов данных без проверки на четность. Запросите у провайдера, какие настройки нужно использовать.
8. Вы можете настроить модемный модуль на передачу данных другому CPU S7–200 (если был выбран протокол PPI) или на передачу данных устройству Modbus (если был выбран протокол Modbus). Отметьте триггерную кнопку Enable CPU data transfers [Разблокировать передачу данных CPU] и щелкните на кнопке Configure CPU-to... [Настроить CPU на...], чтобы определить передаваемые данные и телефонные номера удаленных устройств.
9. Если вы установили передачу данных от CPU к CPU или от CPU к Modbus, то вы должны определить данные, подлежащие передаче, и телефонный номер удаленного устройства. Выберите закладку Data Transfers [Передачи данных] в экране Configure Data Transfers [Конфигурирование передач данных] и щелкните на кнопке New Transfer [Новая передача]. Передача данных состоит из чтения данных из удаленного устройства, записи данных в удаленное устройство или из чтения данных из удаленного устройства и записи данных в это устройство. Если выбраны чтение и запись, то сначала выполняется чтение, а затем запись.
- В каждом сеансе чтения или записи может быть передано до 100 слов. Передача данных должна осуществляться в память переменных в локальном CPU или из нее. Мастер всегда описывает адреса памяти в удаленном устройстве, как если бы это удаленное устройство было CPU S7–200. Если удаленное устройство является устройством Modbus, то передача осуществляется в регистры временного хранения устройства Modbus или из них (адрес 04xxxx). Эквивалентный адрес Modbus (xxxx) определяется следующим образом:
- Адрес Modbus = 1 + (адрес в памяти переменных / 2)
 Адрес в памяти переменных = (адрес Modbus – 1) * 2

10. Закладка Phone Numbers [Телефонные номера] в экране Configure CPU Data Transfers [Конфигурирование передач данных CPU] дает возможность определить телефонные номера для передач данных из CPU в CPU или из CPU в устройство Modbus. Щелкните на кнопке New Phone Number... [Новый телефонный номер...], чтобы добавить новый телефонный номер. После того как телефонный номер сконфигурирован, он должен быть добавлен в проект. Выделите этот телефонный номер в столбце Available Phone Numbers [Доступные телефонные номера] и щелкните на стрелке, направленной вправо, чтобы добавить телефонный номер в текущий проект. После того как вы добавили телефонный номер в текущий проект, вы можете его выбрать и присвоить ему символическое имя для использования в своей программе.
Поля Description [Описание] и Phone Number [Телефонный номер] не отличаются от описанных выше для передачи сообщений. Поле Password [Пароль] необходимо, если удаленное устройство является модемным модулем, у которого активизирована парольная защита. В поле Password в локальном модемном модуле должен быть установлен пароль удаленного модемного модуля. Локальный модемный модуль передает этот пароль, когда он запрашивается удаленным модемным модулем.
11. Функция обратного вызова заставляет модемный модуль автоматически отсоединяться и набирать заранее определенный телефонный номер после получения вызова из удаленного STEP 7-Micro/WIN. Отметьте триггерную кнопку Enable callback [Разблокировать обратный вызов] и щелкните на кнопке Configure Callback... [Сконфигурировать обратный вызов...], чтобы ввести телефонные номера для обратного вызова. Щелкните на Next> [Дальше>].
12. Экран Configure Callback... [Сконфигурировать обратный вызов...] дает возможность ввести телефонные номера, которые модемный модуль использует, когда он отвечает на поступающий вызов. Активизируйте опцию 'Enable callbacks to only specified phone numbers [Разблокировать обратные вызовы только для указанных телефонных номеров]', если телефонные номера для обратного вызова должны быть заранее определены. Если модемный модуль должен принимать любые телефонные номера, сообщаемые абонентом (чтобы взять на себя расходы на соединение), то активизируйте опцию 'Enable callbacks to any phone number' [Разблокировать обратные вызовы для всех телефонных номеров]..
Если для обратного вызова разрешены только определенные телефонные номера, щелкните на кнопке New Phone Number [Новый телефонный номер], чтобы добавить телефонные номера для обратного вызова. Экран Callback Properties [Свойства обратного вызова] дает возможность вводить заранее определенные телефонные номера для обратного вызова и описание для номера обратного вызова. Введенный здесь номер обратного вызова – это телефонный номер, который модемный модуль использует для набора, когда он осуществляет обратный вызов. Этот телефонный номер должен включать в себя все цифры, необходимые для соединения с внешней линией, символы паузы для ожидания соединения с внешней линией, соединения на большое расстояние и т.д.
После ввода нового телефонного номера для обратного вызова он должен быть добавлен к проекту. Выделите этот телефонный номер в столбце Available Callback Phone Numbers [Доступные телефонные номера для обратного вызова] и щелкните на стрелке, направленной вправо, чтобы добавить телефонный номер в текущий проект.
13. Вы можете установить количество попыток набора номера, которые модемный модуль делает при передаче сообщения или во время передачи данных. Модемный модуль сообщает программе пользователя об ошибке только после того, как все попытки набрать номер и отправить сообщение оказались безуспешными.
У некоторых телефонных линий отсутствует сигнал тонального набора, если телефонная трубка снята. Обычно модемный модуль сообщает программе пользователя об ошибке, если отсутствует сигнал тонального набора, когда модемный модуль получает команду послать сообщение или выполнить обратный вызов. Чтобы разрешить набор номера по линии, где нет сигнала тонального набора, отметьте триггерную кнопку Enable Dialing Without Dial Tone [Разблокировать набор номера без сигнала тонального набора].
14. Мастер расширения функций модема создает конфигурационный блок для модемного модуля и требует от пользователя, чтобы он ввел начальный адрес в памяти, где хранятся конфигурационные данные модемного модуля. Конфигурационный блок модемного модуля хранится в памяти переменных в CPU. STEP 7-Micro/WIN записывает конфигурационный блок в блок данных проекта. Размер конфигурационного блока зависит от количества сконфигурированных сообщений и телефонных номеров. Вы можете выбрать адрес в памяти переменных, где вы хотите хранить конфигурационный блок, или щелкните на кнопке Suggest Address [Предложить адрес], если вы хотите, чтобы мастер предложил вам неиспользуемый блок подходящего размера в памяти переменных. Щелкните на Next> [Дальше>].

15. Последний шаг при конфигурировании модемного модуля состоит в указании адреса в памяти выходов для командного байта модемного модуля. Вы можете определить этот адрес, подсчитав количество выходных байтов, используемых модулями с цифровыми выходами, установленными на S7-200 перед модемным модулем. Щелкните на Next> [Далее>].
16. Теперь Мастер расширения функций модема генерирует компоненты проекта для выбранной вами конфигурации (программный блок и блок данных) и создает код, доступный для использования в вашей программе. Последний экран мастера отображает необходимые вам для конфигурации компоненты проекта. Вы должны загрузить конфигурационный блок модемного модуля (блок данных) и программный блок в CPU S7-200.

Обзор команд и ограничений модема

Мастер расширения функций модема облегчает управление модемным модулем путем создания однозначных подпрограмм, основанных на расположении модуля и опциях конфигурации, которые вы выбрали. Каждой команде предшествует префикс «MODx_», где x указывает на местоположение модуля.

Требования к использованию команд для модемного модуля EM 241

При использовании команд модемного модуля примите во внимание следующие требования:

- Команды для модемного модуля используют три подпрограммы.
- Команды для модемного модуля увеличивают количество памяти, необходимой для вашей программы, не более чем на 370 байт. Если вы удаляете неиспользуемую команду, то вы можете перезапустить Мастер расширения функций модема, чтобы в случае необходимости вновь создать эту команду.
- Вы должны обратить внимание на то, чтобы в каждый данный момент времени была активна только одна команда.
- Эти команды не могут использоваться в программах обработки прерываний.
- Модемный модуль считывает информацию из конфигурационной таблицы, при включении и после каждого перехода из состояния STOP в RUN. Все изменения, которые ваша программа делает в конфигурационной таблице, становятся известными модулю только после изменения режима или при следующем включении.

Использование команд для модемного модуля EM 241

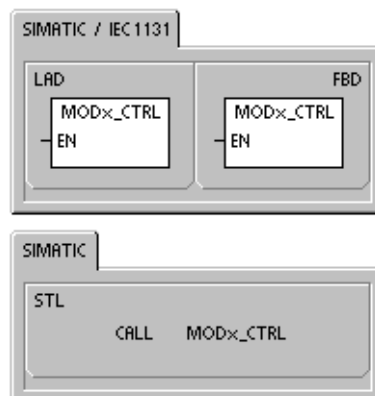
Для использования команд для модемного модуля в своей программе для S7-200 действуйте следующим образом:

1. С помощью Мастера расширения функций модема создайте конфигурационную таблицу модемного модуля.
2. Вставьте в свою программу команду MODx_CTRL и используйте контакт SM0.0 для выполнения ее в каждом цикле.
3. Вставьте команду MODx_MSG для каждого сообщения, которое вы хотите послать.
4. Вставьте команду MODx_XFR для каждой передачи данных.

Команды для модемного модуля

Команда MODx_CTRL

Команда MODx_CTRL (Управление) используется для активизации и инициализации модемного модуля. Эта команда должна вызываться в каждом цикле и должна применяться в проекте только один раз.



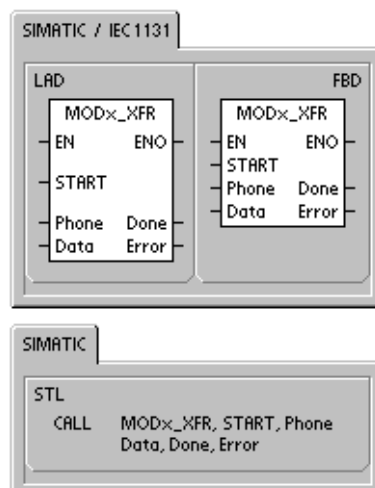
Команда MODx_XFR

Команда MODx_XFR (Передача данных) используется, чтобы заставить модемный модуль прочитать или записать данные в другой CPU S7-200 или устройство Modbus. Эта команда требует от 20 до 30 секунд от момента подачи сигнала на вход START до установки бита Done [Готово].

Бит EN должен быть включен, чтобы разблокировать команду для модуля, и должен оставаться включенным, пока не будет установлен бит Done, сигнализирующий о завершении процесса. Команда XFR передается модемному модулю в каждом цикле, когда включен вход START и модуль при этом не занят. Вход START может быть включен через элемент распознавания фронта, что позволяет послать только одну команду.

Параметр Phone [Телефон] – это один из телефонных номеров, предназначенных для передачи данных. Вы можете использовать здесь символическое имя, которое вы присвоили каждому телефонному номеру для передачи данных при определении номеров с помощью Мастера расширения функций модема.

Параметр Data [Данные] – это номер одной из определенных передач данных. Вы можете использовать здесь символическое имя, которое вы присвоили передаче данных с помощью Мастера расширения функций модема.



Done – это бит, который включается, когда модемный модуль завершает передачу данных.

Error – это байт, содержащий результат передачи данных. В таблице 10–4 определены возможные ошибочные состояния, которые могут появиться при исполнении этой команды.

Таблица 10–6. Параметры для команды MODx_XFR

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала
Phone, Data	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Команда MODx_MSG

Команда MODx_MSG (Передать сообщение) используется для передачи из модемного модуля сообщений для пейджинговой связи или коротких SMS-сообщений. Эта команда требует от 20 до 30 секунд от момента подачи сигнала на вход START до установки бита Done [Готово].

Бит EN должен быть включен, чтобы разблокировать команду для модуля, и должен оставаться включенным, пока не будет установлен бит Done, сигнализирующий о завершении процесса. Команда MSG передается модемному модулю в каждом цикле, когда включен вход START и модуль при этом не занят. Вход START может быть включен через элемент распознавания фронта, что позволяет послать только одну команду.

Параметр Phone [Телефон] – это один из телефонных номеров. Вы можете использовать здесь символическое имя, которое вы присвоили каждому телефонному номеру для передачи сообщений при определении номеров с помощью Мастера расширения функций модема.

Параметр Msg – это номер одного из определенных сообщений. Вы можете использовать здесь символическое имя, которое вы присвоили сообщению при определении этого сообщения с помощью Мастера расширения функций модема.

Done – это бит, который включается, когда модемный модуль завершает передачу сообщения провайдеру.

Error – это байт, содержащий результат этого запроса к модулю. В таблице 10–8 определены возможные ошибочные состояния, которые могут появиться при исполнении этой команды.

Таблица 10–7. Параметры для команды MODx_MSG

Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала
Phone, Msg	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

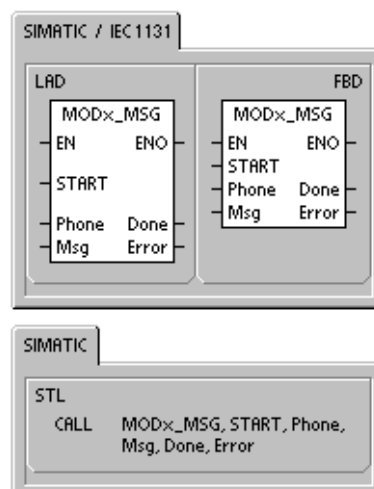


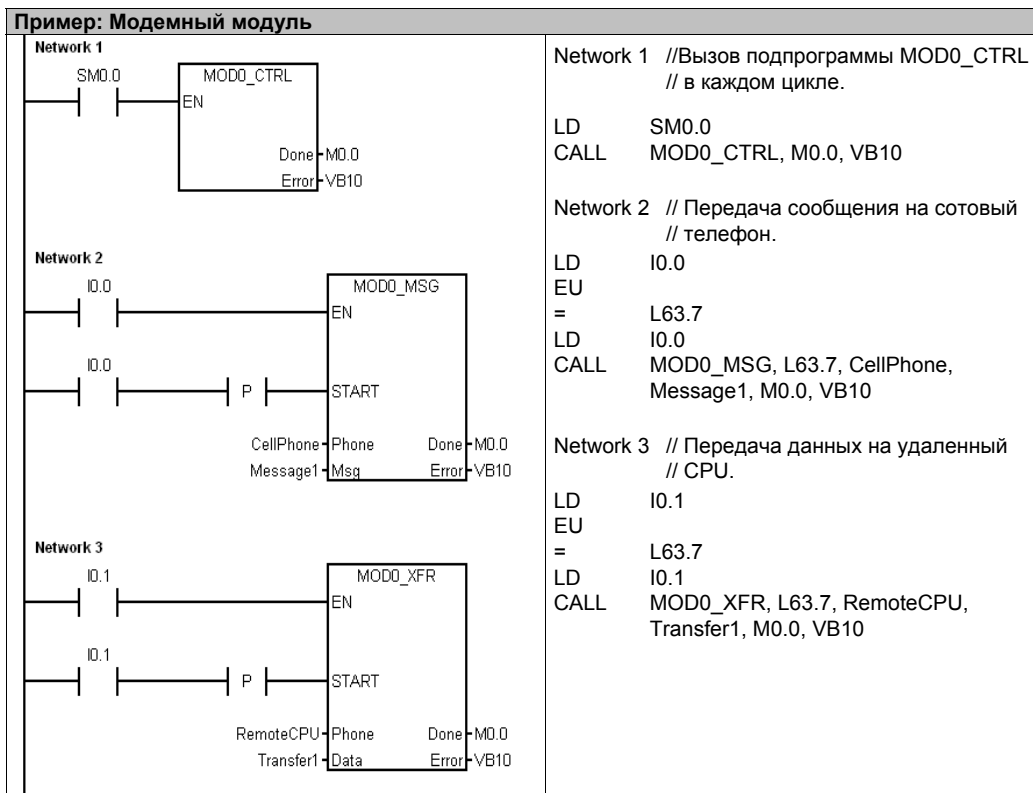
Таблица 10–8. Значения ошибок, возвращаемые командами MODx_MSG и MODx_XFR

Ошибка	Описание
0	Нет ошибки
Ошибки телефонной линии	
1	Отсутствует сигнал тонального вызова
2	Линия занята
3	Ошибка набора номера
4	Нет ответа
5	Превышение времени установления соединения (нет соединения в течение 1 минуты)
6	Соединение прервано или неизвестный ответ
Ошибки в команде	
7	Сообщение для цифровой пейджинговой связи содержит недопустимые цифры
8	Телефонный номер (вход Phone) вне допустимой области
9	Передача сообщения или данных (вход Msg или Data) вне допустимой области
10	Ошибка в текстовом сообщении или в сообщении при передаче данных
11	Ошибка в телефонном номере для передачи сообщений или передачи данных
12	Операция недопустима (напр., попытка установки в ноль)
Ошибки провайдера	
13	Нет ответа (истечение времени ожидания) от службы передачи сообщений
14	Служба передачи сообщений отключена по неизвестной причине
15	Сообщение, прерванное пользователем (деактивизирован бит команды)
TAP – ошибки сообщений пейджинговой связи и SMS-сообщений, возвращаемые провайдером	
16	Удаленный разрыв соединения (провайдер прервал сеанс)
17	Регистрация не принята службой передачи сообщений (неправильный пароль)
18	Блок не принят службой передачи сообщений (неверная контрольная сумма или ошибка передачи)
19	Блок не принят службой передачи сообщений (причина неизвестна)
UCP – ошибки SMS-сообщений, возвращаемые провайдером	
20	Неизвестная ошибка
21	Ошибка контрольной суммы
22	Синтаксическая ошибка
23	Операция не поддерживается системой (недопустимая команда)
24	Операция не разрешена в данное время
25	Активна блокировка вызова (черный список)
26	Недопустимый адрес вызывающего абонента
27	Ошибка контроля прав допуска
28	Ошибка кода легитимизации
29	GA не действителен
30	Повторение недопустимо
31	Код легитимизации для повторения, ошибка
32	Приоритетный вызов недопустим
33	Код легитимизации для приоритетного вызова, ошибка
34	Срочное сообщение недопустимо
35	Код легитимизации для срочного сообщения, ошибка
36	Оплата вызова за свой счет недопустима
37	Код легитимизации для оплаты вызова за свой счет, ошибка

Таблица 10–8. Значения ошибок, возвращаемые командами MODx_MSG и MODx_XFR, продолжение

Ошибка	Описание
UCP – ошибки SMS-сообщений, возвращаемые провайдером (продолжение)	
38	Доставка с задержкой недопустима
39	Новый AC недействителен
40	Новый код легитимизации недопустим
41	Стандартный текст недействителен
42	Интервал времени недействителен
43	Вид сообщений, не поддерживаемый системой
44	Длина сообщения слишком велика
45	Запрошенный стандартный текст недействителен
46	Тип сообщения недействителен для пейджеров
47	Сообщение не найдено в SMSC
48	Резерв
49	Резерв
50	Абонент повесил телефонную трубку
51	Группа факсов не поддерживается
52	Сообщения по факсу не поддерживаются
Ошибки передачи данных	
53	Истечение времени для сообщения (нет ответа от удаленного устройства)
54	Удаленный CPU занят процессом загрузки
55	Ошибка доступа (неверная область памяти, недопустимый тип данных)
56	Коммуникационная ошибка (неизвестный ответ)
57	Ошибка контрольной суммы или CRC в ответе
58	Удаленный EM 241 настроен на обратный вызов (недопустимо)
59	Удаленный EM 241 отверг переданный пароль
60 – 127	Резерв
Ошибки применения команд	
128	Невозможно обработать этот запрос. Модемный модуль занят другим запросом, или не было импульса на входе START для этого запроса.
129	Ошибка модемного модуля: <ul style="list-style-type: none"> ■ Местоположение модемного модуля или адрес в памяти выходов Q, сконфигурированные с помощью мастера расширения функций модема, не совпадает с текущим положением или адресом в памяти ■ См. SMB8 –SMB21 (идентификатор модуля ввода/вывода и регистр ошибок)

Пример программы для модемного модуля



CPU S7–200, поддерживающие интеллектуальные модули

Модемный модуль – это интеллектуальный модуль расширения, спроектированный для работы с CPU S7–200, приведенными в таблице 10–9.

Таблица 10–9. Совместимость модемного модуля EM 241 с CPU S7–200

CPU	Описание
CPU 222 начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224 начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP начиная с версии 2.00	CPU 224XP DC/DC/DC и CPU 224XP AC/DC/Relay
CPU 226 начиная с версии 1.00	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

Биты специальной памяти для модемного модуля

Для каждого интеллектуального модуля выделено пятьдесят байтов специальной памяти (SM) в зависимости от его физического положения на шине расширения ввода/вывода. Когда обнаруживается ошибочное состояние или изменение состояния, модуль показывает это путем обновления содержания адресов SM, соответствующих местоположению модуля. Если это первый модуль, то для сообщения информации об ошибочном состоянии или изменении состояния он обновляет SMB200 – SMB249. Если это второй модуль, то он обновляет SMB250 – SMB299 и т.д. См. таблицу 10–10.

Таблица 10–10. Байты специальной памяти SMB200 – SMB549

Байты специальной памяти SMB200 – SMB549						
Интеллект. модуль в слоте 0	Интеллект. модуль в слоте 1	Интеллект. модуль в слоте 2	Интеллект. модуль в слоте 3	Интеллект. модуль в слоте 4	Интеллект. модуль в слоте 5	Интеллект. модуль в слоте 6
SMB200 – SMB249	SMB250 – SMB299	SMB300 – SMB349	SMB350 – SMB399	SMB400 – SMB449	SMB450 – SMB499	SMB500 – SMB549

В таблице 10–11 представлена область данных специальной памяти, выделенная для модемного модуля. Эта область определена, как если бы речь шла об интеллектуальном модуле, расположенном в слоте 0 системы ввода/вывода.

Таблица 10–11. Адреса специальной памяти для модемного модуля EM 241

Адрес SM- памяти	Описание																																				
SMB200 – SMB215	Имя модуля (16 символов ASCII). SMB200 это первый символ. «EM241 Modem [Модем EM241]»																																				
SMB216 – SMB219	Номер версии программного обеспечения (4 символа ASCII). SMB216 это первый символ.																																				
SMW220	Код ошибки 0000 – Нет ошибки 0001 – Нет напряжения 0002 – Вышел из строя модем 0003 – Отсутствует идентификатор конфигурационного блока 0004 – Конфигурационный блок вне области 0005 – Ошибка конфигурации 0006 – Неверный код страны 0007 – Телефонный номер слишком велик 0008 – слишком длинное сообщение 0009 – 00FF – Резерв 01xx – Ошибка в номере обратного вызова xx 02xx – Ошибка в номере пейджера xx 03xx – Ошибка в номере сообщения xx 0400 – FFFF – Резерв																																				
SMB222	Состояние модуля – отражает состояние светодиодов <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td></td> <td style="text-align: center;">F</td> <td style="text-align: center;">G</td> <td style="text-align: center;">H</td> <td style="text-align: center;">T</td> <td style="text-align: center;">R</td> <td style="text-align: center;">C</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> </table> </div> <table style="width: 100%; border: none;"> <tr> <td style="width: 33%;">F – EM_FAULT</td> <td style="width: 33%;">0 – нет ошибки</td> <td style="width: 33%;">1 – ошибка</td> </tr> <tr> <td>G – EM_GOOD</td> <td>0 – модуль неисправен</td> <td>1 – модуль исправен</td> </tr> <tr> <td>H – OFF_HOOK</td> <td>0 – свободно,</td> <td>1 – занято</td> </tr> <tr> <td>T – NO_DIALTONE</td> <td>0 – есть тональный сигнал</td> <td>1 – нет тонального сигнала</td> </tr> <tr> <td>R – RING</td> <td>0 – нет звонка</td> <td>1 – телефон звенит</td> </tr> <tr> <td>C – CONNECT</td> <td>0 – нет соединения</td> <td>1 – есть соединение</td> </tr> </table>	MSB	7	6	5	4	3	2	1	LSB		F	G	H	T	R	C	0	0	F – EM_FAULT	0 – нет ошибки	1 – ошибка	G – EM_GOOD	0 – модуль неисправен	1 – модуль исправен	H – OFF_HOOK	0 – свободно,	1 – занято	T – NO_DIALTONE	0 – есть тональный сигнал	1 – нет тонального сигнала	R – RING	0 – нет звонка	1 – телефон звенит	C – CONNECT	0 – нет соединения	1 – есть соединение
MSB	7	6	5	4	3	2	1	LSB																													
	F	G	H	T	R	C	0	0																													
F – EM_FAULT	0 – нет ошибки	1 – ошибка																																			
G – EM_GOOD	0 – модуль неисправен	1 – модуль исправен																																			
H – OFF_HOOK	0 – свободно,	1 – занято																																			
T – NO_DIALTONE	0 – есть тональный сигнал	1 – нет тонального сигнала																																			
R – RING	0 – нет звонка	1 – телефон звенит																																			
C – CONNECT	0 – нет соединения	1 – есть соединение																																			
SMB223	Код страны, установленный переключателями (десятичное значение)																																				
SMW224	Скорость передачи, при которой было установлено соединение (десятичное значение без знака).																																				
SMB226	Результат команды пользователя <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td></td> <td style="text-align: center;">D</td> <td style="text-align: center;">0</td> <td style="text-align: center;">ERROR</td> <td style="text-align: center;">0</td> </tr> </table> </div> D - бит завершения операции; 0 – операция выполняется 1 – операция завершена ERROR: описание кода ошибки, см. таблицу 10-8	MSB	7	6	5	LSB		D	0	ERROR	0																										
MSB	7	6	5	LSB																																	
	D	0	ERROR	0																																	
SMB227	Выбор телефонного номера – Этот байт указывает, который телефонный номер для передачи сообщений должен использоваться при передаче сообщения. Допустимые значения от 1 до 250.																																				
SMB228	Выбор сообщения - Этот байт указывает, какое сообщение должно быть послано. Допустимые значения от 1 до 250.																																				
SMB229 – SMB244	Резерв																																				
SMB245	Смещение относительно первого байта выходов (Q), используемое в качестве командного интерфейса для этого модуля. CPU сообщает это смещение для удобства пользователя, модулю оно не нужно.																																				
SMD246	Указатель на конфигурационную таблицу для модемного модуля в памяти переменных. Если указатель указывает на область, отличную от памяти переменных, то его значение не принимается, и модуль продолжает проверять этот адрес, ожидая ненулевого значения указателя.																																				

Таблица 10–12. Конфигурационная таблица для модемного модуля, продолжение

Блок телефонных номеров для обратного вызова (не обязателен)	
Байтовое смещение	Описание
24	Телефонный номер обратного вызова 1 – Строка, представляющая первый телефонный номер, который имеет право на доступ к модемному модулю EM 241 посредством обратного вызова. Каждому телефонному номеру для обратного вызова должно быть выделено одно и то же количество места, которое указано в поле для длины телефонного номера обратного вызова (смещение 6 в конфигурационном блоке).
24+ длина номера для обратного вызова	Телефонный номер обратного вызова 2
:	:
:	Телефонный номер обратного вызова n
Блок телефонных номеров для передачи сообщений (не обязателен)	
Байтовое смещение	Описание
M	Телефонный номер для передачи сообщений 1 - Строка, представляющая телефонный номер для передачи сообщений, протокол и опции для набора. Каждому телефонному номеру должно быть выделено одно и то же количество места, которое указано в поле для длины телефонного номера для передачи сообщений (смещение 7 в конфигурационном блоке). Формат телефонных номеров для передачи сообщений описан ниже
M + длина номера для передачи сообщений	Телефонный номер для передачи сообщений 2
:	:
:	Телефонный номер для передачи сообщений n
Блок сообщений (не обязателен)	
Байтовое смещение	Описание
N	Смещение в памяти переменных (относительно VB0) для первого сообщения (2 байта)
N+2	Длина сообщения 1
N+3	Длина сообщения 2
:	:
:	Длина сообщения n
P	Сообщение 1 - Строка (макс. 120 байтов), представляющая первое сообщение. Эта строка включает в себя текст и встроенные переменные или определяет передачу данных CPU. Формат текстовых сообщений и формат для передачи данных CPU описаны ниже.
P + длина сообщения 1	Сообщение 2
:	:
:	Сообщение n

Модемный модуль снова считывает конфигурационную таблицу в следующих случаях:

- В течение пяти секунд после каждого перехода CPU S7–200 из STOP в RUN (если модем не находится в данный момент в режиме online)
- Каждые пять секунд, пока не будет найдена допустимая конфигурация (если модем не находится в данный момент в режиме online)
- Каждый раз, когда модем переходит из режима online в offline

Формат телефонных номеров для передачи сообщений

Телефонный номер для передачи сообщений – это структура, которая содержит информацию, необходимую модемному модулю для передачи сообщения. Телефонный номер для передачи сообщений представляет собой строку ASCII с ведущим байтом, указывающим длину, за которым следуют символы ASCII. Максимальная длина телефонного номера для передачи сообщений равна 120 байтам (куда входит и байт, указывающий длину).

Телефонный номер для передачи сообщений содержит до 6 полей, разделенных косой чертой (/). Две косых черты подряд указывают на пустое (нулевое) поле. В пустых полях в модемном модуле устанавливаются значения по умолчанию.

Формат: <Телефонный номер>/<Идентификатор>/<Пароль>/<Протокол>/<Стандарт>/<Формат>

Поле для телефонного номера содержит телефонный номер, который модемный модуль набирает при передаче сообщения. Если отправляемое сообщение является текстом или коротким сообщением (SMS), то это телефонный номер провайдера службы передачи сообщений. Если сообщение предназначено для абонента цифровой пейджинговой связи, то это поле является телефонным номером пейджера. Если сообщение используется для передачи данных CPU, то это телефонный номер удаленного устройства. Максимальное число символов в этом поле равно 40.

Идентификатор – это номер пейджера или сотового телефона. Это поле должно состоять только из цифр от 0 до 9. Если производится передача данных CPU, то это поле используется для сообщения адреса удаленного устройства. В этом поле разрешено до 20 символов.

Поле пароля указывает пароль для сообщений, посылаемых через TAP, если ввода пароля требует провайдер. Для сообщений, передаваемых через UCP, это поле используется в качестве исходящего адреса или телефонного номера. Если сообщение используется для передачи данных CPU другому модемному модулю, то в этом поле может быть указан пароль удаленного модемного модуля. Пароль может иметь длину до 15 символов.

Поле протокола содержит один символ ASCII, который сообщает модемному модулю, как он должен форматировать и передавать сообщение. Допустимы следующие значения:

- 1 – протокол цифровой пейджинговой связи (по умолчанию)
- 2 – TAP
- 3 – Команда UCP 1
- 4 – Команда UCP 30
- 5 – Команда UCP 51
- 6 – Передача данных CPU

Поле стандарта заставляет модемный модуль использовать определенный стандарт модема. Поле стандарта содержит один символ ASCII. Допустимы следующие значения:

- 1 - Bell 103
- 2 - Bell 212
- 3 - V.21
- 4 - V.22
- 5 - V.22 bit
- 6 - V.23c
- 7 - V.32
- 8 - V.32 bit
- 9 - V.34 (по умолчанию)

Поле формата состоит из трех символов ASCII, которые указывают количество битов данных и наличие контроля четности при передаче сообщения. Это поле не используется, если в качестве протокола установлена цифровая пейджинговая связь. Допустимы только следующие две настройки:

- 8N1 - битов данных, нет проверки четности, один стоповый бит (по умолчанию)
- 7E1 - 7 битов данных, проверка на четность, один стоповый бит

Формат текстовых сообщений

Формат текстовых сообщений определяет формат для текстовой пейджинговой связи или коротких сообщений (SMS). Эти типы сообщений могут содержать текст и вложенные переменные. Текстовое сообщение – это строка ASCII с ведущим байтом для указания длины, за которым следуют символы ASCII. Максимальная длина текстового сообщения равна 120 байтам (включая байт для указания длины).

Формат: <Текст><Переменная>< Текст><Переменная >...

Текстовое поле состоит из символов ASCII.

Поле переменной определяет значение вложенных данных, которые модемный модуль считывает из локального CPU, форматирует их и помещает в сообщение. Для отметки начала и конца поля переменной используется символ процента (%). Адрес и левые поля отделяются двоеточием. Разделителем левых и правых полей может быть точка или запятая, которая используется в качестве десятичной точки в форматированной переменной. Синтаксис для поля переменной:

%Адрес:Формат Слева.Справа%

Поле адреса указывает адрес, тип данных и размер вложенных данных (напр., VD100, VW50, MB20 или T10). Разрешены следующие типы данных: I, Q, M, SM, V, T (только слово), C (только слово) и AI (только слово). Допускаются размеры: байт, слово и двойное слово.

Поле "Слева" определяет количество цифр, отображаемых слева от десятичной точки. Это значение должно быть достаточно большим, чтобы обрабатывать ожидаемый диапазон вложенных переменных, включая знак минус, если необходимо. Если поле "Слева" является нулем, то число отображается с ведущим нулем. Допустимый диапазон для поля "Слева" от 0 до 10.

Поле "Справа" определяет количество цифр, отображаемых справа от десятичной точки. Нули справа от десятичной точки отображаются всегда. Если поле "Справа" является нулем, то число отображается без десятичной точки. Допустимый диапазон для поля "Справа" от 0 до 10.

Поле "Формат" указывает формат отображения вложенной переменной. Для поля "Формат" допустимы следующие символы:

i – целое со знаком
u – целое без знака
h – шестнадцатеричное значение
f – число с плавающей точкой/вещественное

Пример: «Температура = %VW100:3.1i% Давление = %VD200:4.3f%»

Формат сообщений для передачи данных CPU

Передача данных CPU от CPU к CPU или от CPU к Modbus, задается в формате сообщений для передачи данных CPU. Сообщение для передачи данных CPU – это строка ASCII, которая может задавать любое количество передач данных между устройствами, ограниченное сверху лишь максимальной длиной сообщения в 120 байтов (119 символов плюс байт, указывающий длину). Для разделения различных передач данных между собой может быть использован символ пробела ASCII, но это не обязательно. Все передачи данных осуществляются за одно соединение. Передачи данных осуществляются в последовательности, определенной в сообщении. Если в передаче данных обнаруживается ошибка, то соединение с удаленным устройством завершается, и последующие транзакции не обрабатываются.

Если задана операция чтения, то из удаленного устройства считывается указанное количество слов, начиная с удаленного адреса, а затем они записываются в память переменных в локальном CPU, начиная с локального адреса.

Если задана операция записи, то указанное количество слов считывается из локального CPU, начиная с локального адреса, а затем они записываются в удаленное устройство, начиная с удаленного адреса.

Формат: <Операция>=<Количество>, <Локальный адрес>, <Удаленный адрес>

Поле "Операция" состоит из одного символа ASCII и определяет тип передачи.

R – Чтение данных из удаленного устройства

W – Запись данных в удаленное устройство

Поле "Количество" определяет количество передаваемых слов. Допустимый диапазон для этого поля от 1 до 100 слов.

Поле "Локальный адрес" указывает адрес в памяти переменных локального CPU для передачи данных (напр., VW100).

Поле "Удаленный адрес" указывает адрес в удаленном устройстве для передачи данных (напр., VW500). Этот адрес всегда указывается как адрес в памяти переменных, даже если данные передаются в устройство Modbus. Если удаленным устройством является устройство Modbus, то преобразование между адресом в памяти переменных и адресом Modbus производится следующим образом:

Адрес Modbus = 1 + (Адрес в памяти переменных / 2)

Адрес в памяти переменных = (Адрес Modbus – 1) * 2

Пример: R=20,VW100, VW200 W=50,VW500,VW1000 R=100,VW1000,VW2000

Управление приводом MicroMaster с помощью библиотеки протокола USS

11

Использование библиотек команд STEP 7-Micro/WIN упрощает управление приводами MicroMaster за счет включения предварительно скомпонованных подпрограмм и программ обработки прерываний, специально разработанных для обмена данными с приводами по протоколу USS. При помощи команд USS вы можете управлять физическим приводом и параметрами чтения/записи для привода.

Эти команды можно найти в папке Libraries [Библиотеки] дерева команд STEP 7-Micro/WIN. Когда вы выбираете команду протокола USS, автоматически добавляются одна или более связанных с ней подпрограмм (от USS1 до USS7).

Библиотеки фирмы Siemens продаются на отдельном компакт-диске с дополнением к STEP 7-Micro/WIN: библиотека команд, номер для заказа 6ES7 830-2BC00-0YX0. Если вы купили и установили версию 1.1 библиотеки фирмы Siemens, то любое последующее обновление STEP 7-Micro/WIN V3.2x и V4.0, которое вы устанавливаете, автоматически заменит новой версией и ваши библиотеки без дополнительной оплаты (если библиотеки были дополнены или изменены).

В этой главе

Требования для использования протокола USS	334
Расчет времени, необходимого для обмена данными с приводом	334
Использование команд USS	335
Команды для протокола USS	336
Примеры программ для протокола USS	343
Коды ошибок выполнения команд USS	344
Подключение и наладка привода MicroMaster серии 3	344
Подключение и наладка привода MicroMaster серии 4	347

Требования для использования протокола USS

Библиотеки команд STEP 7-Micro/WIN предлагают для поддержки протокола USS 14 подпрограмм, 3 программы обработки прерываний и 8 команд. Команды USS используют следующие ресурсы S7-200:

- Посредством инициализации протокола USS порт 0 готовится к коммуникациям USS.
Чтобы выбрать для порта 0 протокол USS или протокол PPI, используется команда USS_INIT. (USS – это ссылка на протокол USS для приводов SIMOTION MicroMaster.) После выбора протокола USS для обмена данными с приводами вы не можете использовать порт 0 ни для каких других целей, включая обмен данными со STEP 7-Micro/WIN.
При разработке программы для приложения с протоколом USS вам следует использовать CPU 224XP, CPU 226 или модуль EM 277 PROFIBUS-DP, подключенные к плате PROFIBUS CP в вашем компьютере. Этот второй коммуникационный порт позволяет STEP 7-Micro/WIN контролировать приложение, когда работает протокол USS.
- Команды USS влияют на все ячейки SM, относящиеся к свободно программируемому обмену данными через порт 0.
- Команды USS используют 14 подпрограмм и 3 программы обработки прерываний.
- Команды USS увеличивают объем памяти, необходимой для вашей программы на величину до 3600 байт. В зависимости от конкретно используемых команд USS программы поддержки этих команд могут увеличить программу управления минимум на 2300 байт и максимум на 3600 байт.
- Переменные для команд USS требуют блока V-памяти объемом 400 байт. Начальный адрес для этого блока назначается пользователем и резервируется для переменных USS.
- Некоторые из команд USS требуют также 16-байтного коммуникационного буфера. В качестве параметра для такой команды указывается начальный адрес в памяти переменных для этого буфера. Для каждого экземпляра команд USS рекомендуется назначать уникальный буфер.
- При выполнении расчетов команды USS используют аккумуляторы AC0 – AC3. Вы тоже можете использовать эти аккумуляторы в своей программе, но значения в этих аккумуляторах будут изменяться командами USS.
- Команды USS нельзя использовать в программе обработки прерываний.



Совет

Если вы хотите работу порта 0 снова настроить на протокол PPI, чтобы можно было обмениваться данными со STEP 7-Micro/WIN, используйте другую команду USS_INIT для переназначения порта 0.

Можно также установить переключатель режимов работы на S7-200 в STOP. При этом происходит сброс параметров для порта 0. Обратите внимание на то, что прекращение обмена данными с приводами приводит к остановке приводов.

Расчет времени, необходимого для обмена данными с приводом

Обмен данными с приводом осуществляется асинхронно по отношению к циклу S7-200. За время одной коммуникационной транзакции с приводом обычно завершается несколько циклов S7-200. Для определения необходимого количества времени важны следующие факторы: количество имеющихся приводов, скорость передачи и время цикла S7-200.

Некоторые приводы требуют более длительных задержек при использовании команд с обращением к параметрам. Количество времени, необходимого для доступа к параметру, зависит от вида привода и параметра, к которому производится обращение.

После того как команда USS_INIT назначила порт 0 для использования протокола USS, S7-200 регулярно опрашивает все активные приводы с интервалами, показанными в табл. 11-1. Чтобы эта задача могла быть выполнена, вы должны установить для каждого привода параметр time-out [истечение времени ожидания].

Таблица 11-1. Времена обмена данными

Скорость передачи	Время между опросами активных приводов (при отсутствии активных команд с обращением к параметрам)
1200	240 мс (максимум) * количество приводов
2400	130 мс (максимум) * количество приводов
4800	75 мс (максимум) * количество приводов
9600	50 мс (максимум) * количество приводов
19200	35 мс (максимум) * количество приводов
38400	30 мс (максимум) * количество приводов
57600	25 мс (максимум) * количество приводов
115200	25 мс (максимум) * количество приводов



Совет

В каждый данный момент времени может быть активна только одна команда USS_RPM_x или USS_WPM_x. Выход Done каждой команды должен сообщать о завершении прежде, чем логика программы пользователя запустит новую команду.

Для каждого привода используйте только одну команду USS_CTRL.

Использование команд USS

При использовании команд протокола USS в своей программе для контроллера S7-200 действуйте следующим образом:

1. Вставьте в свою программу команду USS_INIT и выполните эту команду только в одном цикле. С помощью команды USS_INIT вы можете инициализировать или изменить коммуникационные параметры USS.
Когда вы вставляете команду USS_INIT, в вашу программу автоматически добавляется несколько скрытых подпрограмм и программ обработки прерываний.
2. Поместите в свою программу только по одной команде USS_CTRL для каждого активного привода.
Вы можете добавлять столько команд USS_RPM_x и USS_WPM_x, сколько необходимо, но только одна из них может быть активной в каждый конкретный момент времени.
3. Выделите память переменных для библиотечных команд, щелкнув правой кнопкой мыши (для отображения контекстного меню) на символе программного блока (Program Block) в дереве команд.
Выберите пункт Library Memory [Память для библиотеки], чтобы отобразить диалоговое окно Library Memory Allocation [Выделение памяти для библиотеки].
4. Настройте параметры приводов, чтобы согласовать их с используемыми в программе скоростью передачи и адресом.
5. Подключите коммуникационный кабель между S7-200 и приводами.
Обеспечьте, чтобы вся аппаратура управления, например, S7-200, подключенная к приводу, была соединена коротким толстым кабелем с той же самой точкой заземления или нейтральной точкой звезды, что и привод.

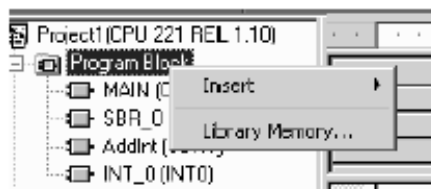


Рис. 11-1. Выделение памяти переменных для библиотеки команд

Пояснение к рисунку: Insert – вставить Library Memory – память для библиотеки

Предостережение

Соединение друг с другом оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель. Эти нежелательные токи могут вызвать ошибки в передаче данных или повреждение оборудования.

Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое соединено при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано.

Экран должен быть соединен с массой или с клеммой 1 в 9-контактном разъеме. Рекомендуется соединить клемму 2-0V на приводе MicroMaster с массой.

Команды для протокола USS

Команда USS_INIT

Команда USS_INIT используется для того, чтобы деблокировать и инициализировать или блокировать обмен данными с приводом MicroMaster. Команда USS_INIT должна быть выполнена без ошибок, прежде чем можно будет использовать любую другую команду USS. Эта команда завершается, и сразу устанавливается бит Done [Готово] перед переходом к следующей команде.

Команда выполняется в каждом цикле обработки программы, когда включен вход EN.

Команда USS_INIT должна выполняться ровно один раз для каждого изменения состояния обмена данными. Поэтому вход EN должен включаться импульсом через элемент обнаружения фронта сигнала. Для изменения параметров инициализации выполните новую команду USS_INIT.

Значение параметра Mode выбирает протокол обмена данными: входное значение, равное 1, назначает порт 0 протоколу USS и активизирует этот протокол, а входное значение, равное 0, назначает порт 0 протоколу PPI и деактивирует протокол USS.

Параметр Baud устанавливает скорость передачи на 1200, 2400, 4800, 9600, 19200, 38400, 57600 или 115200 Бод. Скорости передачи 57600 и 115200 поддерживаются CPU S7-200, начиная с версии 1.2.

Параметр Active указывает, какие приводы являются активными. Некоторые приводы поддерживают только адреса от 0 до 30.

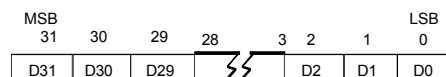
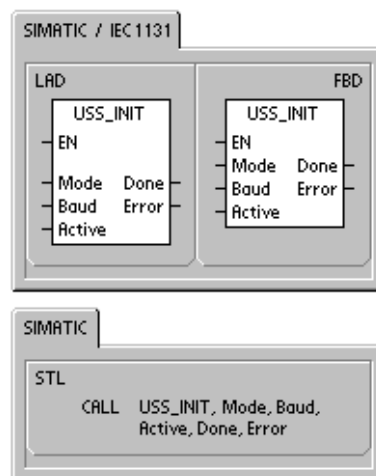
Таблица 11-2. Параметры для команды USS_INIT

Входы/выходы	Тип данных	Операнды
Mode	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD
Baud, Active	DWORD	VD, ID, QD, MD, SD, SMD, LD, константа, AC *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

На рис. 11-2 показаны описание и формат входа активного привода. Любой привод, отмеченный как активный (Active), автоматически опрашивается в фоновом режиме работы для того, чтобы управлять приводом, собирать данные о состоянии и предотвращать простои последовательного канала связи в приводе.

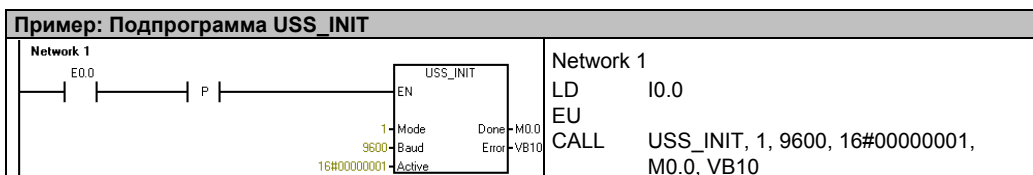
Для вычисления времени между опросами состояния обратитесь к таблице 11-1.

Когда команда USS_INIT завершается, включается выход Done. Выходной байт Error содержит результат выполнения команды. Таблица 11-6 определяет возможные ошибочные состояния, которые могут возникнуть в результате выполнения команды.



- D0 Бит активности привода 0:
0 – привод не активен, 1 – привод активен
- D1 Бит активности привода 1:
0 – привод не активен, 1 – привод активен
- ...

Рис. 11-2. Формат параметра для активного привода



Команда USS_CTRL

Команда USS_CTRL используется для управления активным приводом MicroMaster. Команда USS_CTRL помещает выбранные команды в коммуникационный буфер, который затем передается адресованному приводу (параметр Drive), если этот привод был выбран в параметре Active команды USS_INIT.

Каждому приводу должна быть поставлена в соответствие только одна команда USS_CTRL.

Некоторые приводы сообщают свою скорость вращения только как положительную величину. Если скорость отрицательна, то привод дает для скорости положительное значение, но обращает бит D_Dir (направление).

Бит EN должен быть включен, чтобы активизировать команду USS_CTRL. Эта команда всегда должна быть активизирована.

Бит RUN (RUN/STOP) указывает, включается привод (1) или выключается (0). Когда бит RUN включается, привод MicroMaster получает команду начать вращение с заданной скоростью в заданном направлении. Чтобы привести в действие привод, нужно выполнить следующее:

- Привод (Drive) должен быть выбран как активный (Active) в USS_INIT.
- OFF2 и OFF3 должны быть установлены в 0.
- Fault и Inhibit должны быть равны 0.

Когда бит RUN выключается, приводу MicroMaster передается команда линейно понижать скорость до останова двигателя. Бит OFF2 используется для того, чтобы разрешить приводу MicroMaster вращаться по инерции до останова. Бит OFF3 используется, чтобы подать приводу MicroMaster команду остановиться быстро.

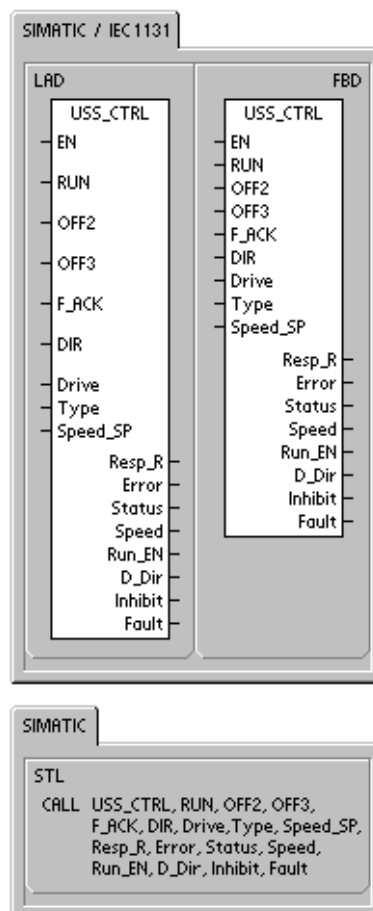
Бит Resp_R (получен ответ) подтверждает получение ответа от привода. Все активные приводы опрашиваются для получения самой последней информации о состоянии привода. Каждый раз, когда S7-200 получает ответ от привода, бит Resp_R включается на один цикл, и все последующие значения обновляются.

Бит F_ACK (Fault Acknowledge [квитирование неисправности]) используется для того, чтобы квитировать неисправность привода. Привод сбрасывает сигнал неисправности (Fault), когда F_ACK изменяется с 0 на 1.

Бит DIR (направление) указывает, в каком направлении привод должен вращаться.

Таблица 11-3. Параметры команды USS_CTRL

Входы/выходы	Типы данных	Операнды
RUN, OFF 2, OFF 3, F_ACK, DIR	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала
Resp_R, Run_EN, D_Dir, Inhibit, Fault	BOOL	I, Q, M, S, SM, T, C, V, L
Drive, Type	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, константа
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD
Status	WORD	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
Speed_SP	REAL	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD, константа
Speed	REAL	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD



Вход Drive (адрес привода) является адресом привода MicroMaster, которому должна передаваться команда USS_CTRL. Действительными являются адреса от 0 до 31

Вход Type (тип привода) выбирает тип привода. Для привода MicroMaster 3 (или ранее) установите Type в 0. Для привода MicroMaster 4 установите Type в 1.

Speed_SP (уставка скорости) – это скорость привода, заданная как процент от полной скорости. Отрицательные значения Speed_SP заставляют привод изменить направление его вращения на противоположное. Диапазон: от –200.0% до 200.0%

Error – это байт ошибки, содержащий результат самого последнего коммуникационного запроса к приводу. Таблица 11–6 определяет возможные ошибочные состояния, которые могут возникать в результате выполнения команды.

Status – это необработанный значение слова состояния, возвращаемое приводом. Рис. 11–3 показывает биты состояния для стандартного слова состояния и главной обратной связи.

Speed – это скорость привода, выраженная в процентах от полной скорости. Диапазон: от –200,0% до 200,0%.

Run_EN (RUN enable [активизация RUN]) показывает, работает привод (1) или остановлен (0).

D_Dir показывает направление вращения привода.

Inhibit показывает состояние бита запрета в приводе (0 - не запрещен, 1 - запрещен). Для сброса бита запрета, нужно выключить бит Fault, и входы RUN, OFF2 и OFF3 тоже должны быть выключены.

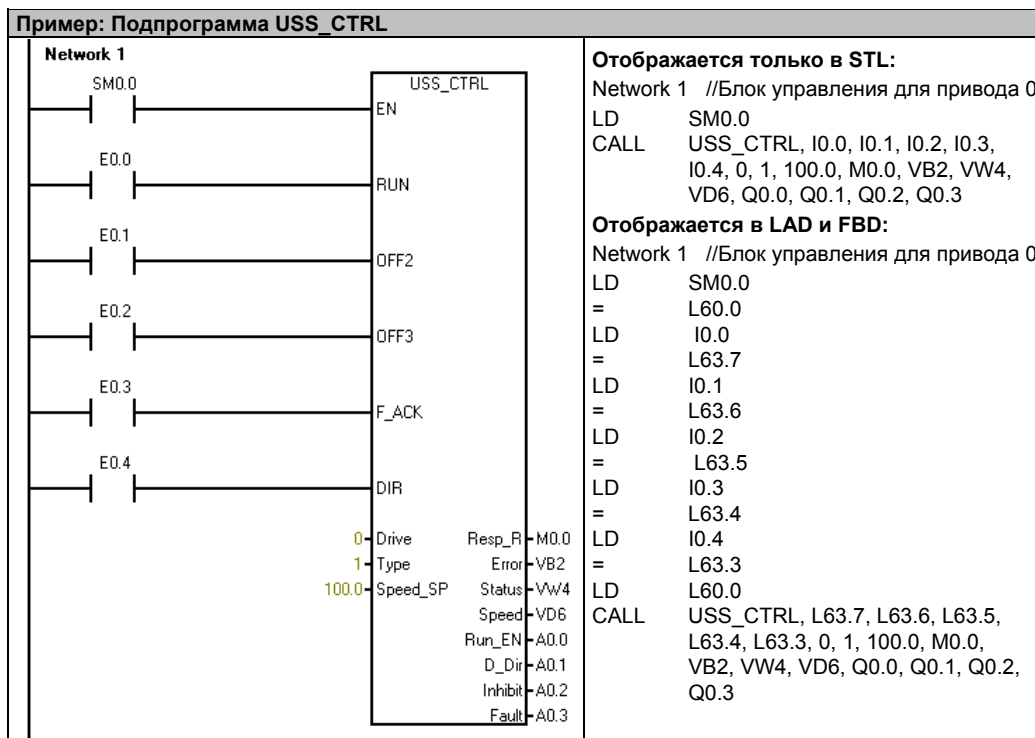
Fault показывает состояние бита неисправности (0 – нет неисправности, 1 - неисправность). Привод отображает код неисправности. (Обратитесь к справочному руководству для вашего привода). Чтобы сбросить бит Fault, устраните причину неисправности и включите бит F_ACK.



Рис. 11–3. Биты состояния для стандартного слова состояния для MicroMaster 3 и главной обратной связи



Рис. 11–4. Биты состояния для стандартного слова состояния для MicroMaster 4 и главной обратной связи



Команда USS_RPM_x

Имеется три команды чтения для протокола USS:

- Команда USS_RPM_W считывает не имеющий знака параметр в формате слова.
- Команда USS_RPM_D считывает не имеющий знака параметр в формате двойного слова.
- Команда USS_RPM_R считывает параметр в формате числа с плавающей точкой.

В каждый конкретный момент времени может быть активна только одна команда чтения (USS_RPM_x) или записи (USS_WPM_x).

Транзакции USS_RPM_x завершаются, когда привод MicroMaster подтверждает получение команды или когда регистрируется ошибочное состояние. Пока этот процесс ожидает ответа, продолжает выполняться логическое сканирование программы.

Бит EN, чтобы деблокировать передачу запроса, должен быть включен и оставаться включенным до тех пор, пока не установится бит Done, сообщая о завершении процесса. Например, запрос USS_RPM_x передается приводу MicroMaster в каждом цикле обработки программы, когда вход XMT_REQ включен. Поэтому вход XMT_REQ должен включаться импульсно через элемент обнаружения фронта сигнала, вызывающий передачу одного запроса для каждого положительного фронта на входе EN.

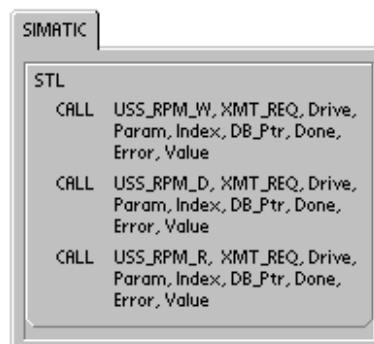
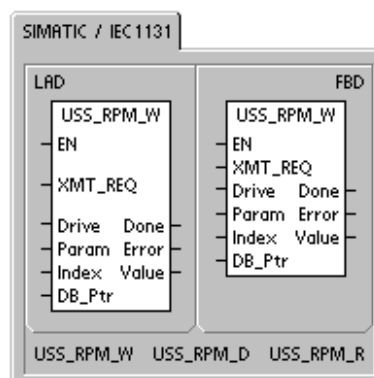
Вход Drive – это адрес привода MicroMaster, которому должна передаваться команда USS_RPM_x. Допустимыми адресами отдельных приводов являются адреса от 0 до 31.

Param является номером параметра. Index – это значение индекса параметра, который должен быть считан. Value – это возвращаемое значение параметра. На вход DB_Ptr должен подаваться адрес 16-байтового буфера. Этот буфер используется командой USS_RPM_x для сохранения результатов команды, поданной на привод MicroMaster.

Когда команда USS_RPM_x завершается, выход Done включается, и выходной байт Error и выход Value содержат результат выполнения команды. Таблица 11–6 определяет возможные ошибочные состояния, которые могут возникать в результате выполнения команды. Выходы Error и Value недействительны, пока не включен выход Done.

Таблица 11–4. Допустимые операнды для команд USS_RPM_x

Входы/выходы	Тип данных	Операнды
XMT_REQ	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала, обусловленный элементом обнаружения положительного фронта сигнала
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, константа
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, константа
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



Команда USS_WPM_x

Имеется три команды записи для протокола USS:

- USS_WPM_W записывает не имеющий знака параметр в формате слова.
- USS_WPM_D записывает не имеющий знака параметр в формате двойного слова.
- USS_WPM_R записывает параметр в формате числа с плавающей точкой.

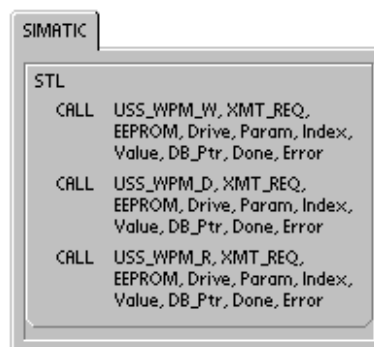
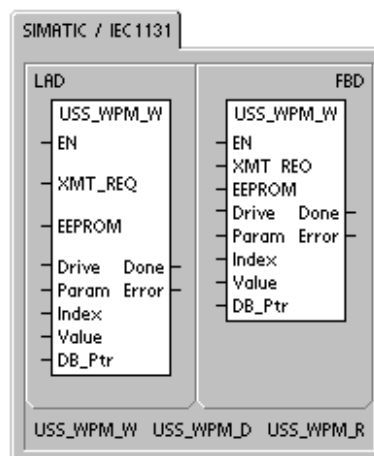
В каждый конкретный момент времени может быть активна только одна команда чтения (USS_RPM_x) или записи (USS_WPM_x).

Транзакции USS_WPM_x завершаются, когда привод MicroMaster подтверждает получение команды или когда регистрируется ошибочное состояние. Пока этот процесс ожидает ответа, продолжает выполняться логическое сканирование программы.

Бит EN, чтобы деблокировать передачу запроса, должен быть включен и оставаться включенным до тех пор, пока не установится бит Done, сообщая о завершении процесса. Например, запрос USS_WPM_x передается приводу MicroMaster в каждом цикле обработки программы, когда вход XMT_REQ включен. Поэтому вход XMT_REQ должен включаться импульсно через элемент обнаружения фронта сигнала, вызывающий передачу одного запроса для каждого положительного фронта на входе EN.

Вход Drive – это адрес привода MicroMaster, которому должна передаваться команда USS_WPM_x.

Допустимыми адресами отдельных приводов являются адреса от 0 до 31.



Param является номером параметра. Index – это значение индекса параметра, который должен быть записан. Value – это значение параметра, которое должно быть записано в ОЗУ в приводе. У приводов MicroMaster 3 вы можете также записать это значение в ЭСППЗУ привода в зависимости от того, как вы сконфигурировали P971 (управление памятью ЭСППЗУ).

На вход DB_Ptr должен подаваться адрес 16-байтового буфера. Этот буфер используется командой USS_WPM_x для сохранения результатов команды, поданной на привод MicroMaster.

Когда команда USS_WPM_x завершается, выход Done включается, и выходной байт Error содержит результат выполнения команды. Таблица 11–6 определяет возможные ошибочные состояния, которые могут возникнуть в результате выполнения команды.

Когда включен вход EEPROM [ЭСППЗУ], команда ведет запись в ОЗУ и в ЭСППЗУ привода. Когда этот вход выключен, команда ведет запись только в ОЗУ привода. Так как MicroMaster 3 не поддерживает эту функцию, то вы должны обеспечить, чтобы этот вход был выключен, если вы хотите использовать эту команду с приводом MicroMaster 3.

Таблица 11–5. Допустимые операнды для команд USS_WPM_x

Входы/выходы	Тип данных	Операнды
XMT_REQ	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала, обусловленный элементом обнаружения положительного фронта сигнала
EEPROM	BOOL	I, Q, M, S, SM, T, C, V, L, поток сигнала
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, константа
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, константа
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Предостережение

Когда вы используете команду USS_WPM_x для обновления набора параметров в ЭСППЗУ привода, вы должны гарантировать, что не превышаете максимальное количество циклов записи (приблизительно 50 000), установленное для ЭСППЗУ.

Превышение максимального количества циклов записи приводит к искажению запоминаемых данных и последующей потере данных. Количество циклов чтения не ограничивается.

Если требуются частые записи в параметры привода, то вы должны сначала обнулять параметр управления памятью ЭСППЗУ в приводе (для приводов MicroMaster 3) и выключать вход EEPROM для приводов MicroMaster 4.

Пример: USS_RPM_x и USS_WPM_x

<p>Network 1</p>	<p>Network 1 //Оба контакта должны иметь один и тот же адрес</p> <pre>LD I0.0 = L60.0 LD I0.0 EU = L63.7 LD L60.0 CALL USS_RPM_W, L63.7, 0, 3, 0, &VB100, M0.0, VB10, VW200</pre>
<p>Network 2</p>	<p>Network 2 //Оба контакта должны иметь один и тот же адрес</p> <pre>LD I0.1 = L60.0 LD I0.1 EU = L63.7 LDN SM0.0 = L63.6 LD L60.0 CALL USS_WPM_W, L63.7, L63.6, 0, 971, 0, 1, &VB120, M0.1, VB11</pre>

Примеры программ для протокола USS

Пример: Пример программы с командами USS, которая правильно отображается в STL

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p> <p>Network 4</p>	<pre> Network 1 //Инициализация протокола USS: //В первом цикле разблокировать //протокол USS для порта 0 при скорости //передачи 19200 с активным адресом // привода "0". LD SM0.1 CALL USS_INIT, 1, 19200, 16#00000001, Q0.0, VB1 Network 2 //Управляющие параметры для привода 0 LD SM0.0 CALL USS_CTRL, I0.0, I0.1, I0.2, I0.3, I0.4, 0, 1, 100.0, M0.0, VB2, VW4, VD6, Q0.1, Q0.2, Q0.3, Q0.4 Network 3 //Чтение параметра в формате слова //из привода 0. //Прочитать параметр 5, индекс 0. //1. Сохранить состояние I0.5 во // временной памяти, чтобы этот // сегмент отображался в LAD. //2. Сохранить нарастающий фронт // на I0.5 во временной локальной // памяти, чтобы его можно было // передать в подпрограмму. LD I0.5 = L60.0 LD I0.5 EU = L63.7 LD L60.0 CALL USS_RPM_W, L63.7, 0, 5, 0, &VB20, M0.1, VB10, VW12 Network 4 //Запись параметра в формате слова //в привод 0. //Записать параметр 2000, index 0. LD I0.6 = L60.0 LD I0.6 EU = L63.7 LDN SM0.0 = L63.6 LD L60.0 CALL USS_WPM_R, L63.7, L63.6, 0, 2000, 0, 50.0, &VB40, M0.2, VB14 </pre> <p>Примечание: Этот код STL не компилируется в LAD или FBD.</p>
---	---

Коды ошибок выполнения команд USS

Таблица 11–6. Коды ошибок выполнения команд USS

Коды ошибок	Описание
0	Ошибка нет
1	Привод не ответил
2	В ответе привода была обнаружена ошибка контрольной суммы
3	В ответе привода была обнаружена ошибка контролем четности
4	Ошибка была вызвана помехой из программы пользователя
5	Была предпринята попытка выполнения запрещенной команды
6	Был указан недействительный адрес привода
7	Для протокола USS не был установлен коммуникационный порт
8	Коммуникационный порт занят обработкой команды
9	Вход скорости привода находится вне допустимого диапазона
10	Длина ответа привода неверна
11	Первый символ в ответе привода неверен
12	Символ длины в ответе привода не поддерживается командами USS
13	Ответил не тот привод
14	Адрес, поданный на DB_Ptr, неверен
15	Указанный номер параметра неверен
16	Выбран недопустимый протокол
17	USS активен; изменение не разрешено
18	Установлена недопустимая скорость передачи
19	Нет обмена данными: привод не активен
20	Неверен параметр или неверно значение в ответе привода или содержат код ошибки
21	Было возвращено значение в формате двойного слова вместо запрошенного значения в формате слова
22	Было возвращено значение в формате слова вместо запрошенного значения в формате двойного слова

Подключение и наладка привода MicroMaster серии 3

Подключение привода MicroMaster 3

Для подключения S7–200 к приводу MicroMaster серии 3 (MM3) можно использовать стандартные кабель PROFIBUS и разъемы. О том, как правильно установить смещение и оконечную нагрузку соединительного кабеля см. рис. 11–5.

Предостережение

Соединение друг с другом оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель.

Эти нежелательные токи могут вызвать ошибки в передаче данных или повреждение оборудования.

Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое соединено при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано.

Экран должен быть соединен с массой или с клеммой 1 в 9-контактном разьеме. Рекомендуется соединить клемму 2-0V на приводе MicroMaster с массой.

P092	3	(1200 Бод)
	4	(2400 Бод)
	5	(4800 Бод)
	6	(9600 Бод – по умолчанию)
	7	(19200 Бод)

6. Введите адрес slave-устройства. Каждым приводом (максимум 31) можно управлять через шину. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P091. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемый адрес slave-устройства. Для ввода нажмите клавишу P.

P091= от 0 до 31.

7. Время разгона (не обязательно). Это время в секундах, которое требуется двигателю для разгона до максимальной частоты. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P002. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время линейного разгона. Для ввода нажмите клавишу P.

P002=0-650.00

8. Время торможения (не обязательно). Это время в секундах, которое требуется двигателю для торможения до полного останова. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P003. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время линейного торможения. Для ввода нажмите клавишу P.

P003=0-650.00

9. Время ожидания последовательного канала связи. Это максимальный допустимый промежуток времени между двумя входящими пакетами данных. Это свойство используется для выключения инвертора в случае нарушения связи.

Отсчет времени начинается после получения действительного пакета данных. Если следующий пакет данных не принимается в течение установленного периода времени, то инвертор отключается и отображается код ошибки F008. Установка этого значения в нуль выключает блок управления. Время между опросами состояния привода можно вычислить, используя таблицу 11–1.

Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P093. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время ожидания последовательного канала связи. Для ввода нажмите клавишу P.

P093=0-240 (0 задан по умолчанию; время в секундах)

10. Номинальная системная уставка последовательного канала связи. Это значение может изменяться, но обычно соответствует 50 Гц или 60 Гц, что определяет соответствующее 100% значение для предустановленного значения (PV) или заданного значения (SP). Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P094. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемая номинальная системная уставка последовательного канала связи. Для ввода нажмите клавишу P.

P094=0-400.00

11. Совместимость с USS (не обязательна). Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P095. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий желаемой совместимости с USS. Для ввода нажмите клавишу P.

P095 = 0 0 разрешающая способность 0,1 Гц (значение по умолчанию)
1 разрешающая способность 0,01 Гц

12. Управление памятью ЭСППЗУ (не обязательно). Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P971. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий желаемому управлению памятью ЭСППЗУ. Для ввода нажмите клавишу P.

P971 = 0 Изменения в настройке параметров (включая P971) теряются при снятии питания.
1 (по умолчанию). Изменения в настройке параметров сохраняются в периоды, когда питание снимается.

13. Отображение режима. Нажмите P, чтобы выйти из режима задания параметров.

Подключение и наладка привода MicroMaster серии 4

Подключение привода MicroMaster 4

Для подключения привода MicroMaster серии 4 (MM4) вставьте концы кабеля RS-485 в два зажима, не имеющие винтов, предназначенных для работы USS. Для подключения S7-200 можно использовать стандартный кабель PROFIBUS и разъемы.

Предостережение

Соединение друг с другом оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель.

Эти нежелательные токи могут вызвать ошибки в передаче данных или повреждение оборудования.

Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое соединено при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано.

Экран должен быть соединен с массой или с клеммой 1 в 9-контактном разъеме. Рекомендуется соединить клемму 2-0V на приводе MicroMaster с массой.

Как показано на рис. 11-6, два провода на противоположных концах кабеля RS-485 должны быть вставлены в клеммный блок привода MM4. Для создания кабельного соединения на приводе MM4 удалите крышку (крышки) привода, чтобы получить доступ к клеммным блокам. В руководстве пользователя привода MM4 вы найдете подробное описание того, как снимается крышка (крышки) на вашем конкретном приводе.

Клеммы клеммного блока пронумерованы. Воспользовавшись на стороне S7-200 соединительным штекером PROFIBUS, соедините клемму A кабеля с клеммой 15 привода (для MM420) или с клеммой 30 (MM440). Соедините клемму B кабельного соединителя с клеммой 14 (MM420) или с клеммой 29 (MM440).

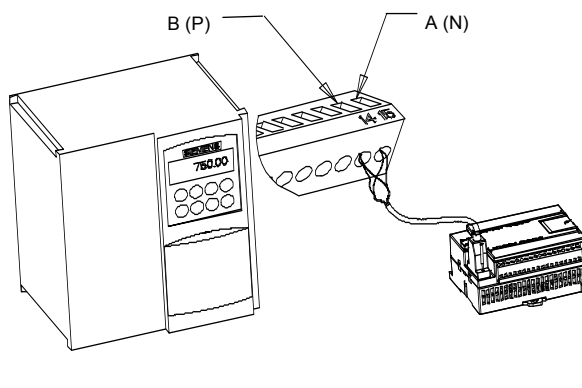


Рис. 11-6. Присоединение клеммного блока MM420

Если S7-200 является завершающим узлом в сети, или если речь идет о двухточечном соединении, необходимо использовать клеммы A1 и B1 (не A2 и B2) штекерного соединителя, так как эти клеммы делают возможной настройку оконечной нагрузки (например, у штекера DP 6ES7 972-0BA40-0X40).

Предостережение

Обратите внимание на то, чтобы крышки привода были надлежащим образом поставлены на свое место перед подачей напряжения на устройство.

Если привод спроектирован как конечный абонент сети, то к соответствующим клеммам должны быть также подключены оконечное сопротивление и сопротивление смещения. На рис. 11-7 показан пример присоединения оконечной нагрузки и смещения для привода MM4.

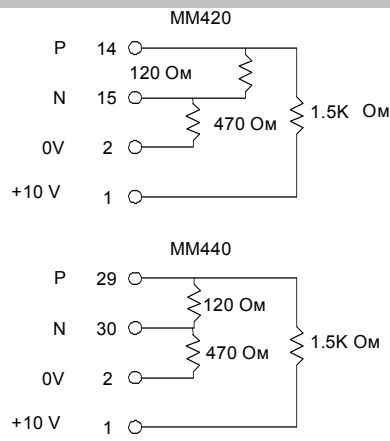


Рис. 11-7. Пример подключения оконечной нагрузки и смещения

Настройка привода MM4

Перед подключением привода к S7–200 вы должны убедиться, что привод имеет указанные ниже системные параметры. Параметры устанавливаются с помощью клавиатуры привода:

1. Восстановите заводскую настройку параметров привода (не обязательно):
 P0010=30
 P0970=1
 Если вы пропускаете этот шаг, то обеспечьте, чтобы для этих параметров были установлены следующие значения:
 USS PZD length [длина]: P2012 Index 0=2
 USS PKW length [длина]: P2013 Index 0=127
2. Разрешите доступ для чтения/записи ко всем параметрам (экспертный режим):
 P0003=3
3. Проверьте параметры настройки двигателя вашего привода:
 P0304 = Номинальное напряжение двигателя (В)
 P0305 = Номинальный ток двигателя (А)
 P0307 = Номинальная мощность двигателя (Вт)
 P0310 = Номинальная частота двигателя (Гц)
 P0311 = Номинальная скорость двигателя (оборотов в минуту)
 Параметры настройки изменяются соответственно используемому двигателю (двигателям).
 Чтобы установить параметры P304, P305, P307, P310 и P311, вы должны сначала установить параметр P010 в 1 (режим быстрого ввода в действие). Когда вы закончите установку этих параметров, установите параметр P010 в 0. Параметры P304, P305, P307, P310 и P311 могут быть изменены только в режиме быстрого ввода в действие.
4. Установите режим управления “локальный/дистанционный”: P0700 Index 0=5
5. Установите заданное значение частоты СОМ-соединения на USS: P1000 Index 0=5
6. Время разгона (не обязательно): P1120= от 0 до 650.00
 Это время в секундах, которое необходимо двигателю для разгона до максимальной частоты.
7. Время торможения (не обязательно): P1121= от 0 до 650.00
 Это время в секундах, которое необходимо двигателю для торможения до полной остановки.
8. Установите опорную частоту для последовательного соединения:
 P2000= от 1 до 650 Гц
9. Установите нормализацию USS: P2009 Index 0=0
10. Установите скорость передачи последовательного интерфейса RS-485:
 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 slave-устройства: P2011 Index 0= от 0 до 31
 Каждым приводом (максимум 31) можно управлять через шину.
12. Время ожидания последовательного канала связи:
 P2014 Index 0= от 0 до 65535 мс
 (0= время ожидания заблокировано)
 Это максимальный допустимый промежуток времени между двумя входящими пакетами данных. Это свойство используется для выключения инвертора в случае нарушения связи. Отсчет времени начинается после получения действительного пакета данных. Если следующий пакет данных не принимается в течение установленного периода времени, то инвертор отключается и отображается код ошибки F0070. Установка этого значения в нуль выключает блок управления. Время между опросами состояния привода можно вычислить, используя таблицу 11–1.
13. Передайте данные из ОЗУ в ЭСППЗУ:
 P0971=1 (начать передачу). Сохраните изменения в настройках параметров в ЭСППЗУ

Использование библиотеки протокола Modbus

12

Библиотека команд STEP 7-Micro/WIN облегчает обмен данными с master-устройствами Modbus путем включения заранее спроектированных подпрограмм и программ обработки прерываний, которые специально разработаны для обмена данными с Modbus. С помощью команд протокола Modbus Slave можно настроить S7-200 так, чтобы он действовал как slave-устройство Modbus RTU и обменивался данными с master-устройствами Modbus.

Вы найдете эти команды в папке Libraries [Библиотеки] дерева команд STEP 7-Micro/WIN. С помощью этих новых команд вы можете заставить S7-200 действовать как slave-устройство Modbus. Когда вы выбираете команду Modbus Slave, к вашему проекту автоматически присоединяется одна или более соответствующих подпрограмм.

Библиотеки фирмы Siemens продаются на отдельном компакт-диске с дополнением к STEP 7-Micro/WIN: библиотека команд, номер для заказа 6ES7 830-2BC00-0YX0. Если вы купили и установили версию 1.1 библиотеки фирмы Siemens, то любое последующее обновление STEP 7-Micro/WIN V3.2x и V4.0, которое вы устанавливаете, автоматически заменит новой версией и ваши библиотеки без дополнительной оплаты (если библиотеки были дополнены или изменены).

В этой главе

Требования для использования протокола Modbus	350
Инициализация и время исполнения протокола Modbus	350
Адресация Modbus	351
Использование команд протокола Modbus Slave	352
Команды протокола Modbus Slave	353

Требования для использования протокола Modbus

Команды протокола Modbus Slave используют следующие ресурсы S7-200:

- Инициализация протокола Modbus Slave подготавливает порт 0 к обмену данными с помощью протокола Modbus Slave.
 Когда порт 0 используется для обмена данными с помощью протокола Modbus Slave, он не может использоваться для других целей, включая обмен данными со STEP 7-Micro/WIN. Команда MBUS_INIT управляет назначением порта 0 протоколу Modbus Slave или PPI.
- Команды протокола Modbus Slave влияют на все адреса специальной памяти (SM), связанные со свободно программируемым обменом данными через порт 0.
- Команды протокола Modbus Slave используют 3 подпрограммы и 2 программы обработки прерываний.
- Команды протокола Modbus Slave требуют 1857 байтов программной памяти для двух команд Modbus Slave и поддерживающих программ.
- Переменные для команды протокола Modbus Slave требуют в памяти переменных блока размером 779 байтов. Начальный адрес этого блока назначается пользователем и резервируется для переменных Modbus.



Совет

Если вы снова хотите настроить порт 0 на работу с PPI, чтобы он мог обмениваться данными со STEP 7-Micro/WIN, еще раз используйте команду MBUS_INIT, чтобы переназначит порт 0.

Вы можете также установить переключатель на S7-200 в положение STOP. Это сбрасывает параметры порта 0.

Инициализация и время исполнения протокола Modbus

Обмен данными с помощью протокола Modbus включает в себя контроль с помощью избыточного циклического кода (cyclic redundancy check, CRC), обеспечивающий целостность передаваемых сообщений. Протокол Modbus Slave использует таблицу заранее рассчитанных значений для уменьшения времени, необходимого для обработки сообщения. Инициализация таблицы CRC требует около 425 миллисекунд. Эта инициализация выполняется внутри подпрограммы MBUS_INIT, и обычно это делается в первом цикле программы пользователя после перехода в режим RUN. Вы отвечаете за сброс контроля времени и сохранение активности выходов (если это необходимо для модулей расширения), если время, необходимое подпрограмме MBUS_INIT и любой другой пользовательской инициализации, превышает 500 миллисекунд контроля времени. Контроль времени модуля вывода сбрасывается путем записи выходы модуля. См. описание команды сброса контроля времени в главе 6.

Время цикла увеличивается, когда подпрограмма MBUS_SLAVE обрабатывает запрос. Так как наибольшая часть времени тратится на расчет CRC Modbus, то время цикла увеличивается примерно на 650 миллисекунд на каждый байт, содержащийся в запросе и в ответе. Максимальное увеличение времени цикла на один запрос или ответ (чтение или запись 120 слов) составляет примерно 165 миллисекунд.

Адресация Modbus

Адреса Modbus обычно записываются как 5- или 6-символьные значения, содержащие тип данных и смещение. Один или два первых символа определяют тип данных, а четыре последних символа указывают надлежащее значение внутри типа данных. Затем master-устройство Modbus отображает эти адреса в соответствующие функции. Командами Modbus Slave поддерживаются следующие адреса:

- от 000001 до 000128 – цифровые выходы, отображаемые в Q0.0 – Q15.7
- от 010001 до 010128 – цифровые входы, отображаемые в I0.0 – I15.7
- от 030001 до 030032 – регистры аналоговых входов, отображаемые в AIW0 – AIW62
- от 040001 до 04xxxx – регистры временного хранения информации, отображаемые в память переменных.

Все адреса Modbus имеют базис, равный 1. В таблице 12–1 показано отображение адресов Modbus в адреса S7–200.

Протокол Modbus Slave позволяет ограничить количество входов, выходов, аналоговых входов и регистров временного хранения информации (памяти переменных), к которым может обращаться master-устройство Modbus. Параметр MaxIQ команды MBUS_INIT указывает максимальное количество цифровых входов или выходов (I или Q), к которым разрешен доступ master-устройству Modbus.

Параметр MaxIQ команды MBUS_INIT указывает максимальное количество цифровых входов или выходов (I или Q), к которым разрешен доступ master-устройству Modbus.

Параметр MaxHold команды MBUS_INIT указывает максимальное количество регистров временного хранения информации (памяти переменных), к которым Modbus разрешен доступ master-устройству Modbus.

Параметр MaxHold команды MBUS_INIT указывает максимальное количество регистров временного хранения информации (памяти переменных), к которым Modbus разрешен доступ master-устройству Modbus.

Таблица 12–1. Отображение адресов Modbus в S7–200

Адрес Modbus	Адрес S7–200
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
...	...
04xxxx	HoldStart+2 x (xxxx-1)

Конфигурирование таблицы символов

После ввода адреса для первого символа таблица автоматически рассчитывает остальные символы и размещает их в таблице.

Вы должны назначить для таблицы, которая занимает 779 байтов, начальный адрес в памяти переменных. Обратите внимание на то, чтобы назначение символов Modbus Slave не пересекалось с памятью переменных, назначенной регистрам временного хранения информации Modbus с помощью параметров HoldStart и MaxHold команды MBUS_INIT. При пересечении областей памяти команда MBUS_INIT выдает ошибку.

Использование команд протокола Modbus Slave

Для использования команд протокола Modbus Slave в своей программе для S7–200 действуйте следующим образом:

1. Вставьте команду MBUS_INIT в свою программу и выполните эту команду только в одном цикле. Команду MBUS_INIT можно использовать для инициализации или для изменения параметров обмена данными протокола Modbus.
Когда вы вставляете команду MBUS_INIT, к вашей программе автоматически добавляется несколько скрытых подпрограмм и программ обработки прерываний.
2. Назначьте начальный адрес для 779 последовательных байтов в памяти переменных для команд протокола Modbus Slave.
3. Поместите в своей программе только одну команду MBUS_SLAVE. Эта команда вызывается в каждом цикле для обслуживания поступающих запросов.
4. Подключите коммуникационный кабель к порту 0 на S7–200 и к master-устройствам Modbus.

Предостережение

Соединение друг с другом оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель. Эти нежелательные токи могут вызвать ошибки в передаче данных или повреждение оборудования.

Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое соединено при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано.

Аккумуляторы (AC0, AC1, AC2, AC3) используются командами Modbus Slave и появляются в списке перекрестных ссылок. Перед выполнением команды Modbus Slave значения, находящиеся в аккумуляторах, сохраняются, а перед завершением этой команды восстанавливаются в аккумуляторах, обеспечивая сохранение всех пользовательских данных в аккумуляторах, когда выполняется команда Modbus Slave.

Команды протокола Modbus Slave поддерживают протокол Modbus RTU. Эти команды используют свободно программируемый обмен данными S7–200 для поддержки наиболее часто используемых функций Modbus. Поддерживаются следующие функции Modbus:

Таблица 12–2. Поддерживаемые функции протокола Modbus Slave

Функция	Описание
1	Чтение состояния одного или нескольких цифровых выходов. Функция 1 возвращает состояние (вкл/выкл) любого количества выходов (Q).
2	Чтение состояния одного или нескольких цифровых входов. Функция 2 возвращает состояние (вкл/выкл) любого количества входов (I).
3	Чтение одного или нескольких регистров временного хранения информации. Функция 3 возвращает содержимое памяти переменных. Регистры временного хранения информации в Modbus – это значения в формате слова, служащие для чтения до 120 слов в одном запросе.
4	Чтение одного или нескольких входных регистров. Функция 4 возвращает значения на аналоговых входах.
5	Запись одного цифрового выхода. Функция 5 устанавливает на указанное значение один цифровой выход. Этот выход не является принудительно установленным, и программа может перезаписать значение, записанное запросом Modbus.
6	Запись одного регистра временного хранения информации. Функция 6 записывает одно значение регистра временного хранения информации в память переменных S7–200.
15	Запись нескольких цифровых выходов. Функция 15 записывает значения нескольких цифровых выходов в регистр образа процесса на выходах Q модуля S7–200. Первый выход должен начинаться на границе байта (например, Q0.0 или Q2.0), а количество записываемых выходов должно быть кратно восьми. Это ограничение для команд протокола Modbus Slave. Эти выходы не являются принудительно установленными, и программа может перезаписать значения, записанные запросом Modbus.
16	Запись нескольких регистров временного хранения информации. Функция 16 записывает значения нескольких регистров временного хранения информации в память переменных S7–200. В одном запросе можно записать до 120 слов.

Команды протокола Modbus Slave

Команда MBUS_INIT

Команда MBUS_INIT используется для активизации и инициализации или деактивизации обмена данных через протокол Modbus. Прежде чем сможет быть применена команда MBUS_SLAVE, должна быть без ошибок выполнена команда MBUS_INIT. При завершении этой команды немедленно устанавливается бит Done [Готово], прежде чем будет выполняться следующая команда.

Команда выполняется в каждом цикле, когда включен вход EN.

Команда MBUS_INIT должна выполняться ровно один раз для каждого изменения состояния обмена данными. Поэтому вход EN должен включаться импульсно с помощью элемента обнаружения фронта или исполняться только в первом цикле.

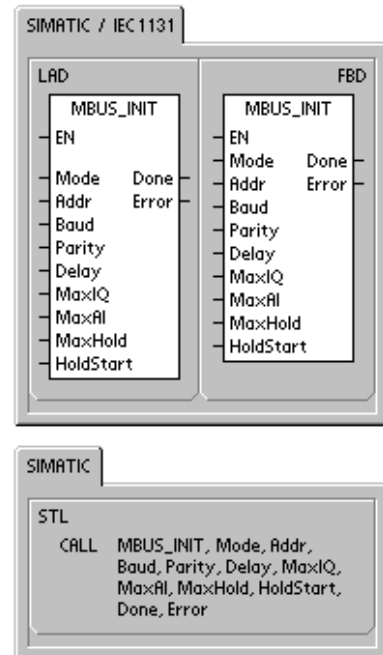
Значение входа Mode [Режим] определяет протокол обмена данными: значение входа, равное 1, назначает порт 0 протоколу Modbus и активизирует этот протокол, а значение входа, равное 0, назначает порт 0 протоколу PPI и деактивизирует протокол Modbus.

Параметр Baud устанавливает скорость передачи на 1200, 2400, 4800, 9600, 19200, 38400, 57600 или 115200 Бод. Скорости передачи 57600 и 115200 поддерживаются CPU S7-200, начиная с версии 1.2.

Параметр Addr устанавливает для адреса значения от 1 до 247 (включительно).

Таблица 12-3. Параметры для команды MBUS_INIT

Входы/выходы	Тип данных	Операнды
Mode, Addr, Parity	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD
Baud, HoldStart	DWORD	VD, ID, QD, MD, SD, SMD, LD, AC, константа, *VD, *AC, *LD
Delay, MaxIQ, MaxAI, MaxHold	WORD	VW, IW, QW, MW, SW, SMW, LW, AC, константа, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



Параметр Parity соответствует контролю четности, используемому master-устройством Modbus. Допустимы следующие значения:

- 0—проверка четности отсутствует
- 1—проверка на нечетность
- 2— проверка на четность

Параметр Delay [Задержка] продлевает обычное для Modbus условие контроля времени ожидания конца сообщения на указанное количество миллисекунд. Обычное значение для этого параметра при работе в подключенной сети должно быть равно 0. Если вы используете модемы с исправлением ошибок, установите задержку на значение от 50 до 100 миллисекунд. Если вы используете радиомодем с расширенной полосой частот, установите задержку на значение от 10 до 100 миллисекунд. Значение параметра Delay может находиться в пределах от 0 до 32767 миллисекунд.

Параметр MaxIQ устанавливает количество входов и выходов, предоставляемых в распоряжение адресам Modbus 00xxxx и 01xxxx, в диапазоне от 0 до 128. Значение 0 блокирует все операции чтения и записи для входов и выходов. Предлагаемое значение для MaxIQ равно 128, что делает возможным обращение ко всем входам и выходам S7-200.

Параметр MaxAI устанавливает количество входных регистров, имеющих размер слова (AI), предоставляемых в распоряжение адресу Modbus 03xxx, в диапазоне от 0 до 32. Значение 0 блокирует чтение аналоговых входов. Предлагаемое значение для MaxAI, допускающее обращение ко всем аналоговым входам S7-200:

- 0 для CPU 221
- 16 для CPU 222
- 32 для CPU 224, CPU 224XP и CPU 226

Параметр MaxHold устанавливает количество имеющих размер слова регистров для временного хранения информации в памяти переменных, предоставляемых в распоряжение адресу Modbus 04xxx. Например, чтобы дать master-устройству возможность обращаться к 2000 байтов памяти переменных, установите MaxHold на значение 1000 слов (регистры временного хранения информации).

Параметр HoldStart – это адрес начала регистров для временного хранения информации в памяти переменных. Это значение обычно устанавливается на VB0, так что параметр HoldStart устанавливается на &VB0 (адрес VB0). В качестве начального адреса для регистров временного хранения информации могут быть указаны и другие адреса в памяти переменных, чтобы дать возможность использования VB0 в каком-нибудь другом месте проекта. Master-устройство Modbus имеет доступ к количеству слов в памяти переменных, определяемому параметром MaxHold, начиная с адреса, указанного в HoldStart.

Когда команда MBUS_INIT завершается, то включается выход Done [Готово]. Выходной байт Error [Ошибка] содержит результат выполнения команды. Таблица 12-5 определяет ошибочные состояния, которые могут возникнуть в результате выполнения команды.

Команда MBUS_SLAVE

Команда MBUS_SLAVE используется для обслуживания запроса от master-устройства Modbus и должна выполняться в каждом цикле, чтобы дать ей возможность контролировать и реагировать на запросы Modbus.

Команда выполняется в каждом цикле, когда включен вход EN.

Команда MBUS_SLAVE не имеет входных параметров.

Выход Done [Готово] включен, когда команда MBUS_SLAVE отвечает на запрос Modbus. Выход Done выключен, если запросы не обрабатывались.

Выход Error [Ошибка] содержит результат выполнения команды. Этот выход действителен, если включен выход Done. Если Done выключен, то параметр Error не изменяется. Таблица 12–5 определяет ошибочные состояния, которые могут возникнуть в результате выполнения команды.

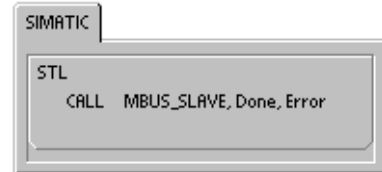
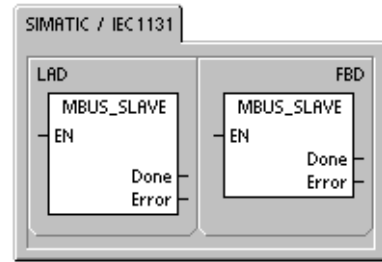
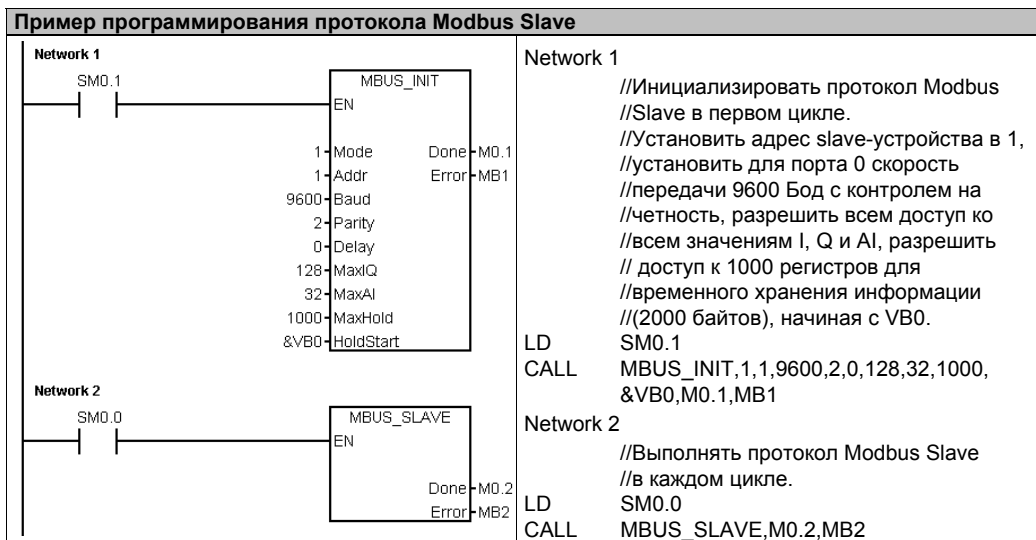


Таблица 12–4. Параметры для команды MBUS_SLAVE

Параметр	Тип данных	Операнды
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Таблица 12–5. Коды ошибок исполнения протокола Modbus Slave

Коды ошибок	Описание
0	Нет ошибки
1	Ошибка области памяти
2	Недопустимая скорость передачи или четность
3	Недопустимый адрес slave-устройства
4	Недопустимое значение для параметра Modbus
5	Регистры для временного хранения информации пересекаются с символами Modbus Slave
6	Ошибка проверки четности при приеме
7	Ошибка CRC при приеме
8	Недопустимый запрос функции/функция не поддерживается
9	Недопустимый адрес памяти в запросе
10	Функция slave-устройства не активизирована



Использование рецептов

13

STEP 7-Micro/Win предоставляет в ваше распоряжение Мастер рецептов, чтобы помочь вам организации рецептов и их определении. Рецепты хранятся не в ПЛК, а в модуле памяти.

В этой главе

Обзор	358
Определение рецептов и терминология	359
Использование Мастера рецептов	359
Команды, создаваемые Мастером рецептов	363

Обзор



Поддержка рецептов встроена в STEP 7-Micro/WIN и ПЛК S7-200. STEP 7-Micro/Win предоставляет в ваше распоряжение Мастер рецептов, чтобы помочь вам организации рецептов и их определении. Рецепты хранятся не в ПЛК, а в модуле памяти.

Все рецепты хранятся в модуле памяти. Поэтому для использования рецептов в ПЛК должен быть вставлен не обязательный модуль памяти на 64 или 256 Кбайт. Дополнительную информацию о модулях памяти вы найдете в Приложении А.

Все рецепты хранятся в модуле памяти. Однако, в память ПЛК считывается только один рецепт, когда программа пользователя обрабатывает это отдельный рецепт. Например, если вы выпекаете печенье, то имеются рецепты для печенья с шоколадной крошкой, сахарного печенья и овсяного печенья. В каждый данный момент времени может выпекаться только один вид печенья, так что должен быть выбран и считан в память ПЛК только один рецепт.

На рис. 13-1 показан процесс изготовления нескольких видов печенья с помощью рецептов. Рецепт для каждого вида печенья хранится в модуле памяти. С помощью текстового дисплея TD 200C оператор выбирает вид печенья, подлежащего изготовлению, а программа пользователя загружает этот рецепт в память.

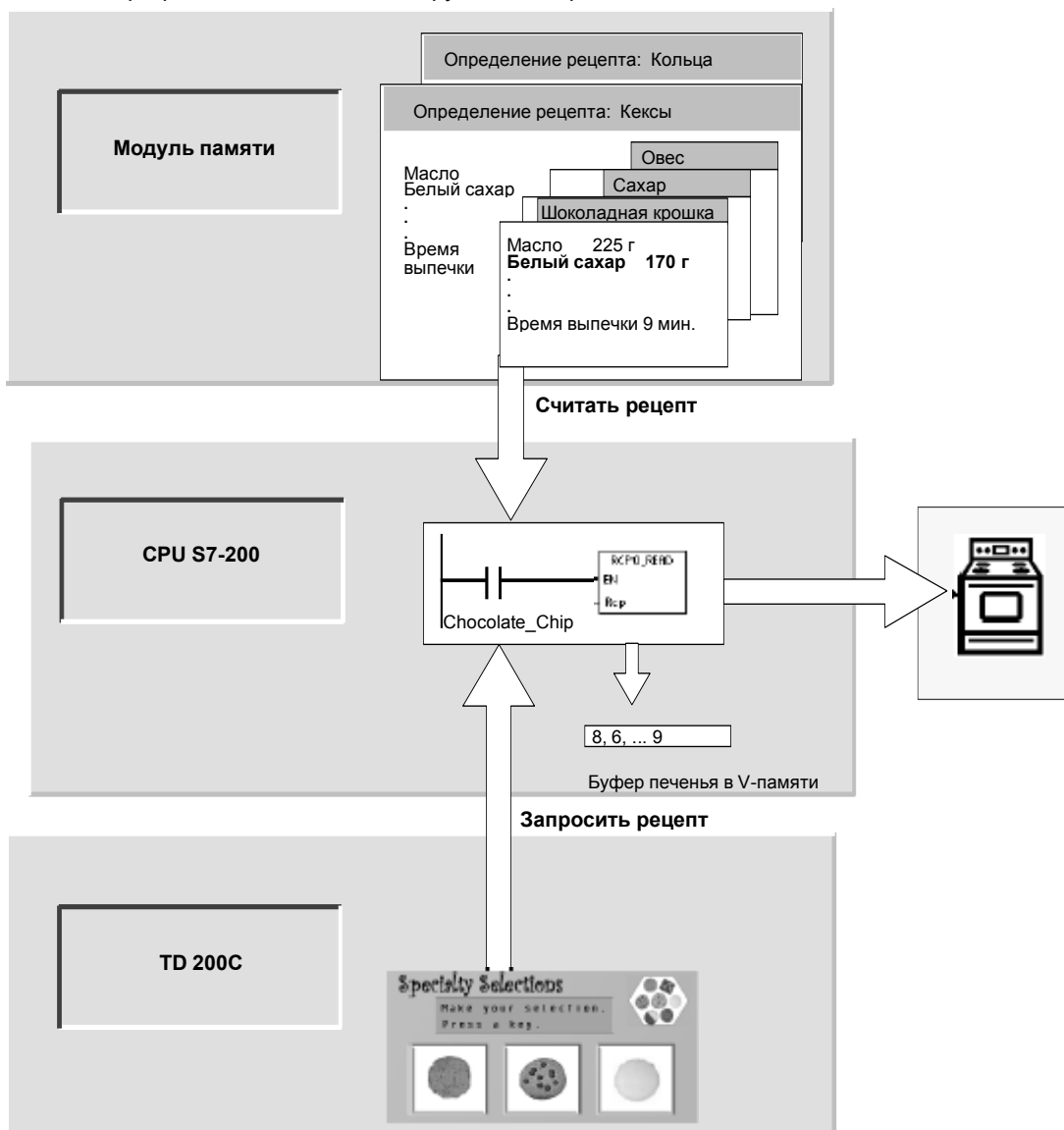


Рис. 13-1. Пример применения рецептов

Определение рецептов и терминология

Для лучшего понимания Мастера рецептов ниже разъясняются следующие определения и термины.

- Конфигурация рецептов – это набор компонентов проекта, созданных Мастером рецептов. Эти компоненты включают в себя подпрограммы с командами, регистры блоков данных и таблицы символов.
- Определение рецептов - это собрание рецептов, обладающих одинаковым набором параметров. Однако значения параметров могут меняться в зависимости от рецепта.
- Рецепт – это набор параметров и их значений, дающий информацию, необходимую для производства продукта или управления процессом.

Например, могут быть созданы различные определения рецептов, например, колечек и печенья. Определение рецептов печенья может содержать много различных рецептов, например, печенье с шоколадной крошкой и сахарное печенье. Пример полей и значений приведен в таблице 13–1.

Таблица 13–1. Пример определения рецептов - печенье

Название поля	Тип данных	С шоколадной крошкой (рецепт 0)	Сахарное (рецепт 1)	Комментарий
Butter [Масло]	Byte	225	225	граммов
White_Sugar [Белый сахар]	Byte	170	340	граммов
Brown_Sugar [Желтый сахарный песок]	Byte	170	0	граммов
Eggs [Яйца]	Byte	2	1	штук
Vanilla [Ваниль]	Byte	1	1	чайных ложек
Flour [Мука]	Byte	500	900	граммов
Baking_Soda [Пищевая сода]	Real	1, 0	0, 5	чайных ложек
Baking_Powder [Разрыхлитель]	Real	0	1, 0	чайных ложек
Salt [Соль]	Real	1, 0	0, 5	чайных ложек
Chocolate_Chips [Шоколадная крошка]	Real	450	0, 0	граммов
Lemon_Peel [Цедра лимона]	Real	0, 0	1, 0	столовых ложек
Cook_Time [Время приготовления]	Real	9, 0	10, 0	минут

Использование Мастера рецептов

Рецепты и их определения создаются с помощью Мастера рецептов. Рецепты хранятся в модуле памяти. Рецепты и их определения могут быть введены непосредственно в Мастер рецептов. В дальнейшем изменения отдельных рецептов могут быть сделаны повторным запуском Мастера рецептов или программированием подпрограммы-команды RCPx_WRITE.

Мастер рецептов создает конфигурацию рецептов, в состав которой входят:

- Таблица символов для каждого определения рецептов. Каждая таблица включает в себя имена символов, совпадающие с названиями полей рецепта. Эти символы определяют адреса в памяти переменных (V), необходимые для доступа к значениям, загруженным в данный момент в память. Каждая таблица содержит также символическую константу для ссылки на отдельные рецепты.
- Регистр блока данных для каждого определения рецептов. Этот регистр определяет начальные значения для адресов в памяти переменных, представленных в таблице символов.
- Подпрограмма-команда RCPx_READ. Эта команда используется для считывания определенного рецепта из модуля памяти в память переменных.
- Подпрограмма-команда RCPx_WRITE. Эта команда используется для записи значений рецепта из памяти переменных в модуль памяти.

Определение рецептов

Для создания рецепта с помощью Мастера рецептов выберите команду меню **Tools > Recipe Wizard [Инструментальные средства > Мастер рецептов]**. Первое диалоговое окно является вводным окном, определяющим основные операции Мастера рецептов. Чтобы начать конфигурирование своих рецептов, щелкните на кнопке Next [Дальше].

Для создания определения рецептов действуйте следующим образом. См. рис. 13–2.

1. Введите имена полей для определения рецептов. Каждое имя становится символом в вашем проекте, как это определено выше.
2. Выберите тип данных из разворачивающегося списка.
3. Введите значение по умолчанию и комментарий для каждого имени. Все новые рецепты, задаваемые в рамках этого определения, будут начинаться с этих значений.
4. Щелкните на Next [Дальше], чтобы создавать и редактировать рецепты для этого определения рецептов.

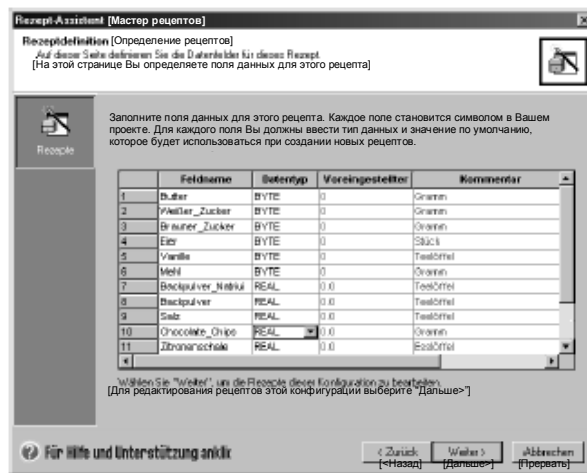


Рис. 13–2. Определение рецептов

Примечание к рисунку: текст таблицы на этом рисунке совпадает с текстом таблицы 13–1.

Для определения всех полей данных в рецепте используйте столько строк, сколько необходимо. Вы можете создать до четырех различных определений рецептов. Количество рецептов для каждого определения ограничено только доступным объемом памяти в вашем модуле памяти.

Создание и редактирование рецептов

В диалоговом окне для создания и редактирования рецептов (Create and Edit Recipes) вы можете создавать отдельные рецепты и вводить для этих рецептов значения. Каждый редактируемый столбец представляет собой один рецепт.

Рецепты можно создавать нажатием кнопки New [Новый]. Каждый рецепт инициализируется значениями по умолчанию, указанными при создании определения рецептов.

Рецепты можно создавать с помощью контекстного меню, появляющегося при нажатии правой кнопки мыши, путем копирования и вставки существующих рецептов. Новые столбцы вставляются слева от текущего положения курсора, включая поле комментариев.

Каждому новому рецепту дается имя по умолчанию, которое включает в себя ссылку на определение рецептов и номер рецепта. Это имя имеет формат DEFx_RCPy.

Для создания и редактирования рецептов действуйте следующим образом. См. рис. 13–3.

1. Щелкните на кнопке Next [Дальше], чтобы перейти в диалоговое окно Create and Edit Recipe [Создание и редактирование рецептов].
2. Выберите кнопку New [Новый], чтобы вставить при необходимости новый рецепт.
3. Переименуйте рецепт, если хотите, чтобы его имя отличалось от установленного по умолчанию.
4. Измените при необходимости значения набора данных каждого рецепта.
5. Щелкните на ОК.

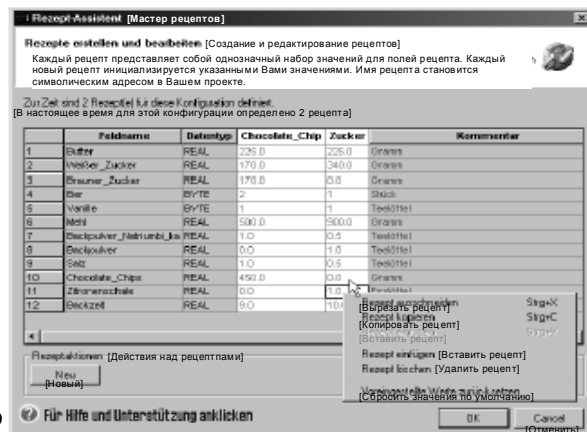


Рис. 13–3. Создание и редактирование рецептов

Выделение памяти

В диалоговом окне для выделения памяти указывается начальный адрес области памяти переменных, где будет храниться рецепт, загруженный из модуля памяти. Вы можете ввести адрес в памяти переменных или предоставить Мастеру рецептов предложить адрес неиспользуемого блока памяти переменных нужного размера.

Для выделения памяти действуйте следующим образом. См. рис. 13–4.

1. Для выбора адреса в памяти переменных, где должен храниться рецепт, щелкните в окне и введите адрес.
2. Чтобы дать возможность Мастеру рецептов выбрать в памяти переменных неиспользуемый блок нужного размера, щелкните на кнопке Suggest Address [Предложить адрес].
3. Щелкните на кнопке Next [Дальше].



Рис. 13–4. Выделение памяти

Компоненты проекта

Диалоговое окно компонентов проекта перечисляет различные компоненты, которые будут добавлены к вашему проекту. См. рис. 13–5.

Щелкните на Finish [Закончить], чтобы закончить работу с Мастером рецептов и добавить эти компоненты.

Каждой конфигурации рецепта может быть присвоено уникальное имя. Это имя будет отображаться в дереве проекта с отдельными конфигурациями, созданными Мастером. К концу этого имени добавляется определение рецепта (RCPx).

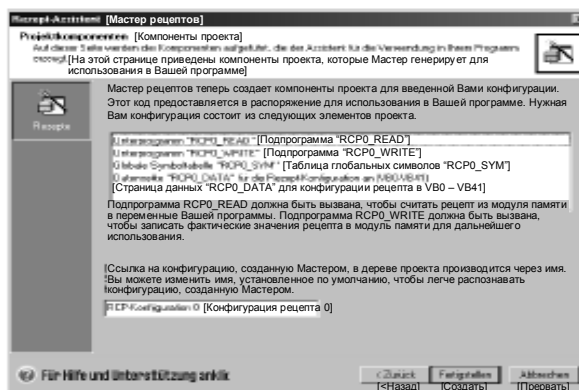


Рис. 13–5. Компоненты проекта

Использование таблицы символов

Таблица символов создается для каждого определения рецептов. Каждая таблица определяет постоянные значения для отдельных рецептов. Эти символы могут использоваться в качестве параметров для команд RCPx_READ и RCPx_WRITE, чтобы задать желаемый рецепт. См. рис. 13–6.

Кроме того, в каждой таблице создаются символические имена для каждого поля рецепта. Вы можете использовать эти символы для доступа к значениям рецепта в памяти переменных.

Symbol	Adresse	Kommentar
Zucker	1	
Chocolate_Chip	1	
Backpulv	V106	Minuten [Минуты]
Zweibeckelab	V104	Stoelowa ложка [Столповая ложка]
Chocolate_Oper	V100	Gramm [Грамм]
Salz	V106	Teeloeel [Чайная ложка]
Backpulv	V102	Teeloeel [Чайная ложка]
Backpulv_Neutralisat	V110	Teeloeel [Чайная ложка]
Mehl	V114	Gramm [Грамм]
Vorle	V113	Teeloeel [Чайная ложка]
Ess	V112	Stueck [Штука]
Swarm_Zucker	V16	Gramm [Грамм]
Wahle_Zucker	V14	Gramm [Грамм]
Butte	V10	Gramm [Грамм]

Рис. 13–6. Таблица символов

Загрузка проектов с конфигурациями рецептов в ПЛК

Для загрузки проекта с конфигурацией рецептов в ПЛК действуйте следующим образом. См. рис. 13–7.

1. Выберите команду меню **File > Download [Файл > Загрузить]**.
2. В диалоговом окне в разделе Options [Возможные варианты] отметьте триггерные кнопки Program Block [Программный блок], Data Block [Блок данных] и [Рецепты].
3. Щелкните на кнопке Download [Загрузить].

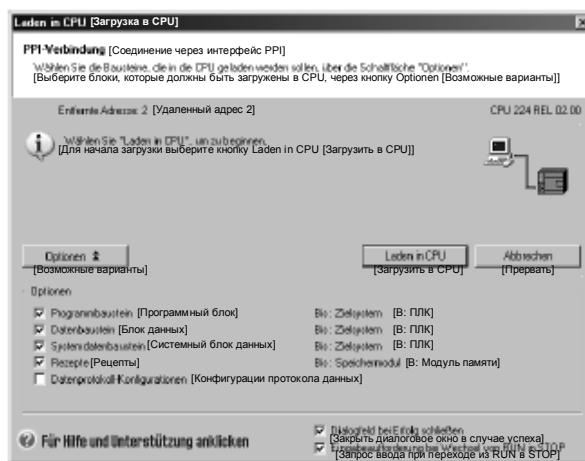


Рис. 13–7. Загрузка проекта с конфигурацией рецептов

Существующие конфигурации рецептов

Для редактирования существующих конфигураций рецептов действуйте следующим образом. См. рис. 13–8.

1. Откройте разворачивающееся поле списка конфигураций и выберите существующую конфигурацию рецепта.
2. Для удаления существующей конфигурации рецепта щелкните на кнопке Delete Configuration [Удалить конфигурацию].

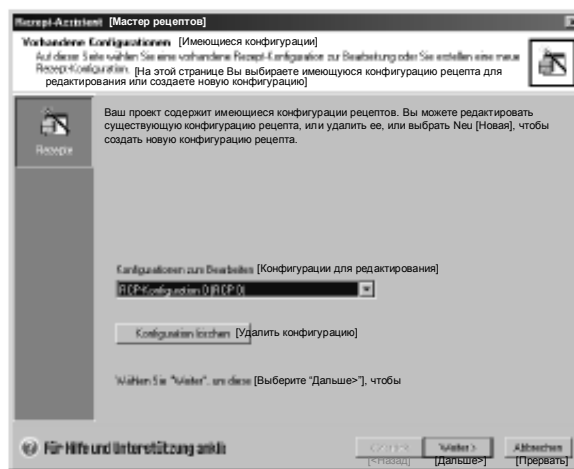


Рис. 13–8. Редактирование существующей конфигурации рецептов

Команды, создаваемые Мастером рецептов

Подпрограмма RCPx_Read

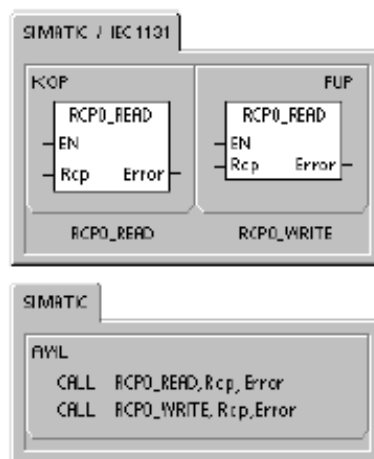
Подпрограмма RCPx_READ создается Мастером рецептов и используется для считывания отдельного рецепта из блока памяти в указанную область в памяти переменных.

Символ x в команде RCPx_READ соответствует определению рецептов, которое содержит рецепт, который вы хотите считать.

Вход EN активизирует исполнение команды при высоком уровне сигнала на этом входе.

Вход Rcp определяет рецепт, который будет загружен из модуля памяти

Выход Error возвращает результат исполнения этой команды. Определения кодов ошибок вы найдете в таблице 13–3.



Подпрограмма RCPx_Write

Подпрограмма RCPx_WRITE создается Мастером рецептов и используется для замены рецепта в модуле памяти содержимым рецепта, содержащегося в памяти переменных.

Символ x в команде RCPx_WRITE соответствует определению рецептов, которое содержит рецепт, который вы хотите заменить.

Вход EN активизирует исполнение команды при высоком уровне сигнала на этом входе.

Вход Rcp определяет рецепт, который будет заменен в модуле памяти.

Выход Error возвращает результат исполнения этой команды. Определения кодов ошибок вы найдете в таблице 13–3.

Таблица 13–2. Допустимые операнды для подпрограммы Recipe

Входы/выходы	Тип данных	Операнды
Rcp	Word	VW, IW, QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD, константа
Error	Byte	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Таблица 13–3. Коды ошибок для команд Recipe

Код ошибки	Описание
0	Нет ошибок
132	Не удался доступ к модулю памяти



Совет

ЭСППЗУ, используемый в модуле памяти, поддерживает ограниченное количество операций записи. Обычно это один миллион циклов записи. Когда это предел достигнут, ЭСПЗУ перестает работать надлежащим образом.

Обратите внимание на то, чтобы активизировать команду RCPx_WRITE не в каждом цикле. Активизация этой команды в каждом цикле приведет к изнашиванию модуля памяти за относительно короткий интервал времени.

Использование протоколов данных

14

STEP 7-Micro/Win предоставляет в ваше распоряжение Мастер протоколирования данных для сохранения результатов измерений процесса в модуле памяти. Перемещение данных процесса в модуль памяти освобождает адреса памяти переменных, которые в противном случае потребовались бы для хранения этих данных.

В этой главе

Обзор	364
Использование Мастера протоколирования данных	365
Команды, создаваемые Мастером протоколирования данных	369

Обзор

Поддержка протоколов данных встроена в STEP 7-Micro/WIN и в ПЛК S7-200. С помощью этой функции вы можете постоянно сохранять записи, содержащие данные процесса, под управлением программы. Эти записи могут содержать также метку времени и дату. Вы можете сконфигурировать до четырех независимых протоколов данных. Формат записи протокола данных определяется в новом Мастере протоколирования данных.

Все протоколы данных хранятся в модуле памяти. Для использования функции протоколирования данных вы должны установить в своем ПЛК необязательный модуль памяти емкостью 64 или 256 Кбайт. Информацию о модулях памяти вы найдете в Приложении А.

Для загрузки содержимого протоколов данных в свой компьютер вы должны использовать проводник S7-200.

Пример применения протокола данных показан на рис. 14-1.

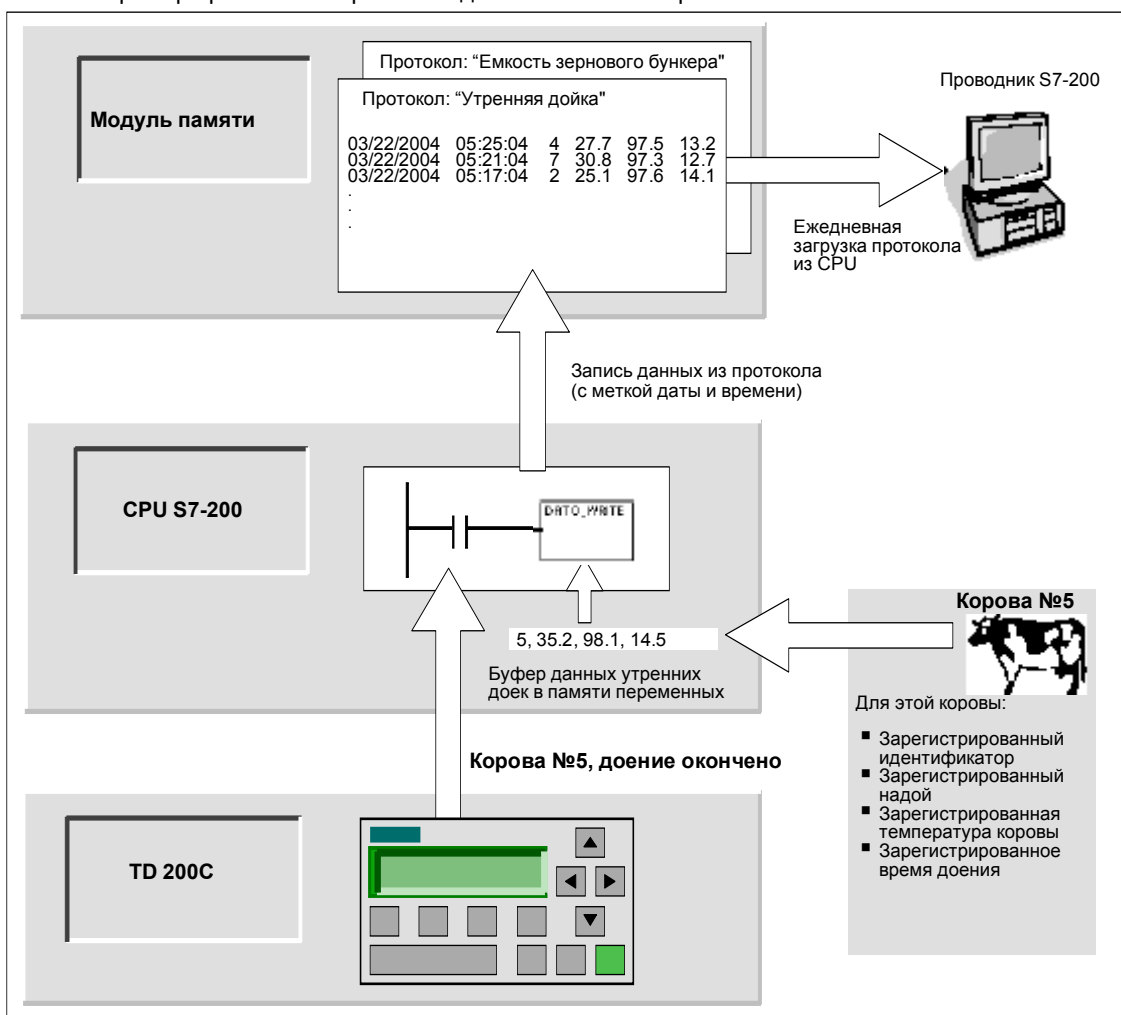


Рис. 14-1. Пример применения протоколирования данных

Определение протокола данных и терминология

Для лучшего понимания работы Мастера протоколирования данных объясняются следующие определения и понятия.

- Протокол данных – это набор записей данных, обычно упорядоченных по дате и времени. Каждая запись представляет некоторое событие в процессе, которое регистрирует набор данных процесса. Организация этих данных определяется в Мастере протоколирования данных.
- Запись данных протокола – это отдельная строка данных, записанных в протокол.

Использование Мастера протоколирования данных



Протокол
данных

В Мастере протоколирования данных можно сконфигурировать до четырех протоколов данных. Мастер протоколирования данных можно использовать для:

- определения формата записи протокола данных
- выбора вариантов протоколирования, например метка времени, метка даты и удаление протокола после загрузки из CPU
- задания максимального количества записей, которые могут храниться в протоколе
- создания кода проекта, используемого для сохранения записей в протоколе данных.

Мастер протоколирования данных создает конфигурацию протокола данных, которая состоит из следующих элементов:

- Таблица символов для каждой конфигурации протоколов данных. Каждая таблица содержит символические имена, которые соответствуют именам полей в протоколе данных. Каждый символ определяет адрес в памяти переменных, по которому сохраняется текущий протокал данных. Каждая таблица содержит также символическую константу для ссылки на отдельные протоколы данных.
- Регистр блока данных для каждой записи протокола данных, в котором каждому полю протокола данных ставится в соответствие адреса в памяти переменных. Ваша программа использует эти адреса памяти переменных для накопления текущего набора данных протокола.
- Подпрограмма-команда DATx_WRITE. Эта команда копирует указанную запись протокола данных из памяти переменных в модуль памяти. Каждое исполнение команды DATx_WRITE добавляет новую запись в протокол данных, хранящийся в модуле памяти.

Опции протокола данных

Вы можете сконфигурировать следующие варианты для протокола данных. См. рис. 14–2.

Метка времени

Вы можете снабдить каждую запись протокола данных меткой времени. Если эта опция активизирована, то CPU автоматически снабжает каждую запись данных меткой времени, когда программа пользователя дает команду на запись протокола данных.

Метка даты

Вы можете снабдить каждую запись протокола данных меткой даты. Если эта опция активизирована, то CPU автоматически снабжает каждую запись данных меткой даты, когда программа пользователя дает команду на запись протокола данных.

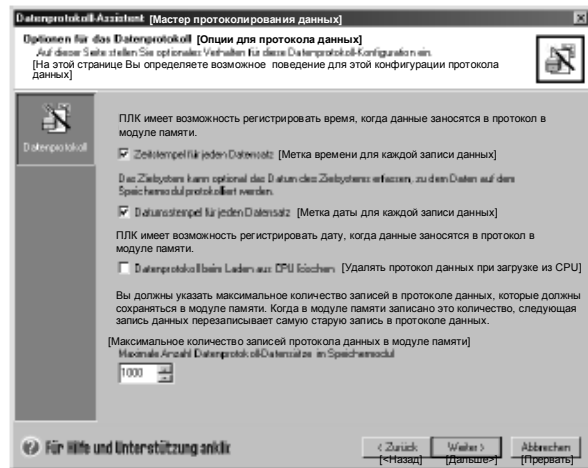


Рис. 14–2. Опции протокола данных

Очистка протокола данных

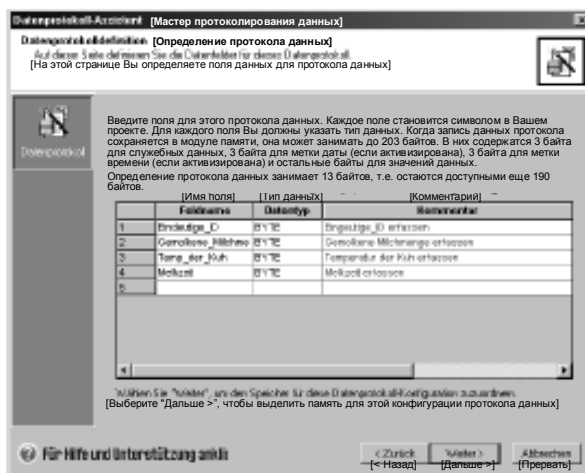
Очистка протокола данных – Вы можете стирать все записи из протокола данных каждый раз, когда он загружается из CPU. Если эта опция установлена, то протокол очищается при каждой его загрузке из CPU.

Протоколы данных реализуются в виде кольцевой очереди (когда протокол полон, новая запись заменяет самую старую). Вы должны указать максимальное количество записей для хранения в протоколе данных. Максимально допустимое количество записей в протоколе данных равно 65 535. Значением по умолчанию для количества записей является 1000.

Определение протокола данных

Вы задаете поля протокола данных, и каждое поле становится символом в вашем проекте. Для каждого поля вы должны указать тип данных. запись протокола данных может содержать от 4 до 203 байтов данных. Для определения полей данных в протоколе данных действуйте следующим образом. См. рис. 14–3.

1. Для ввода имени щелкните на ячейке Field Name [Имя поля]. Это имя становится символом, на который ссылается программа пользователя.
2. Щелкните на ячейке Data Type [Тип данных] и выберите тип данных из разворачивающегося окна списка.
3. Для ввода комментария щелкните на ячейке Comment [Комментарий].
4. Для определения записи используйте столько строк, сколько необходимо.
5. Щелкните на ОК .



Редактирование существующих конфигураций протоколов данных

Для редактирования существующих конфигураций протоколов данных действуйте следующим образом:

1. Откройте ниспадающий список конфигураций и выберите существующую конфигурацию протоколов данных, как показано на рис. 14–4.
2. Для удаления существующей конфигурации протоколов данных щелкните на кнопке Delete Configuration [Удалить конфигурацию].

Вы можете иметь до четырех различных протоколов данных.

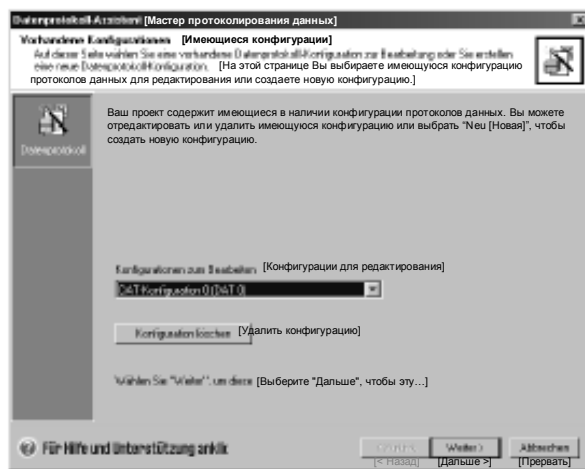


Рис. 14–4. Редактирование существующих конфигураций протоколов данных

Выделение памяти

Мастер протоколирования данных создает блок в области памяти переменных ПЛК. Этот блок является адресом в памяти, где будет создаваться запись протокола данных перед занесением ее в модуль памяти. Вы указываете начальный адрес в памяти переменных, где вы хотите поместить конфигурацию. Вы можете ввести этот адрес в памяти переменных сами или предоставить возможность Мастеру протоколирования данных предложить адрес неиспользуемого блока памяти переменных нужного размера. Размер блока зависит от выбранных вами опций в Мастере протоколирования данных. См. рис. 14–5.

Для выделения памяти действуйте следующим образом:

1. Для выбора адреса в памяти переменных, где должна быть создана запись протокола данных, щелкните в области Suggested Address [Предлагаемый адрес] и введите этот адрес.
2. Чтобы позволить Мастеру протоколирования данных выбрать неиспользуемый блок в памяти переменных нужного размера, щелкните на кнопке Suggest Address [Предложить адрес].
3. Щелкните на кнопке Next [Дальше].

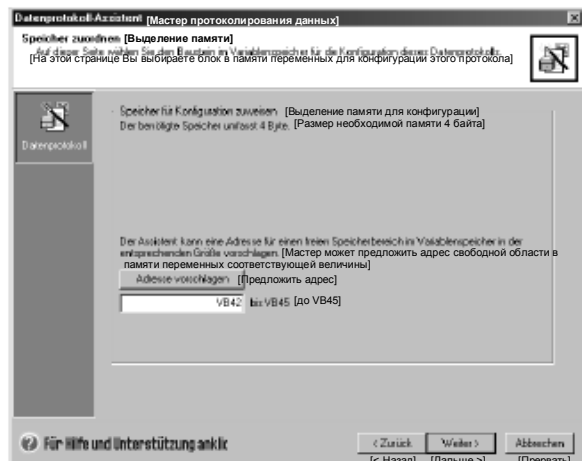


Рис. 14–5. Выделение памяти

Компоненты проекта

В диалоговом окне компонентов проекта приведены различные компоненты, которые будут добавлены к вашему проекту. См. рис. 14–6.

Щелкните на Finish [Закончить], чтобы завершить работу с Мастером протоколирования данных и добавить эти компоненты.

Каждой конфигурации протоколов данных может быть присвоено уникальное имя. Это имя отображается в дереве проекта с отдельными конфигурациями Мастера. Определение протокола данных (DATx) присоединяется к концу этого имени.

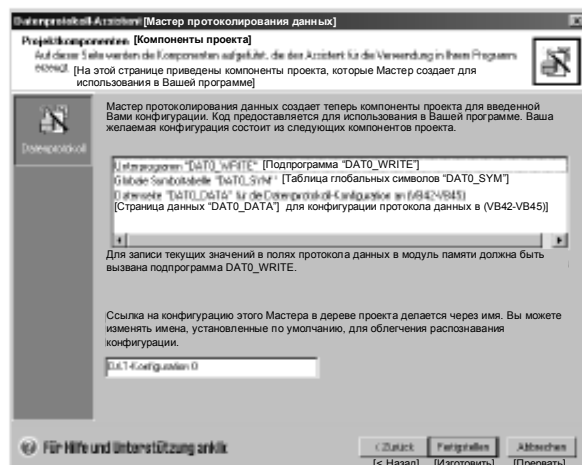


Рис. 14–6. Компоненты проекта

Использование таблицы символов


Таблица символов создается для каждой конфигурации протоколов данных. Каждая таблица определяет постоянные значения для отдельных протоколов данных. Эти символы могут использоваться в качестве параметров для команд DATx_WRITE. Кроме того, в каждой таблице создаются символические имена для каждого поля протокола данных. Вы можете использовать эти символы для доступа к значениям протокола данных в памяти переменных.

Symbol [Символ]	Adresse [Адрес]	Kommentar [Комментарий]
Milchzeit	VB 95	Milchzeit in Tagen [Регистрация времени доения]
Temp_Sym_Prot	VB 41	Temperatur des PUL in °C [Температура коровы]
Global-Sym_Milchmenge	VB 43	Global-Sym Milchmenge in Liter [Величина надоя]
Endebyte_ID	VB 42	Endebyte_ID des PUL [Уникальный идентификатор]

Рис. 14–7. Таблица символов

Загрузка проекта, содержащего конфигурацию протоколов данных

Перед тем, как использовать протокол данных, Вы должны загрузить проект, содержащий конфигурацию протоколов данных, в CPU S7–200. Если проект содержит конфигурацию протоколов данных, то в диалоговом окне для загрузки в CPU опция для конфигураций протоколов данных по умолчанию активизирована.

	<p>Совет</p> <p>При загрузке в ПЛК проекта с конфигурациями протоколов данных все записи протоколов данных, хранящиеся в данный момент в модуле памяти, теряются.</p>
---	--

Для загрузки проекта, содержащего конфигурацию протоколов данных, действуйте следующим образом. См. рис. 14–8.

1. Выберите команду меню **File > Download [Файл > Загрузить]**.
2. В диалоговом окне в разделе Options [Опции] активизируйте триггерную кнопку Data Log Configuration [Конфигурация протоколов данных].
3. Щелкните на кнопке Download [Загрузить].

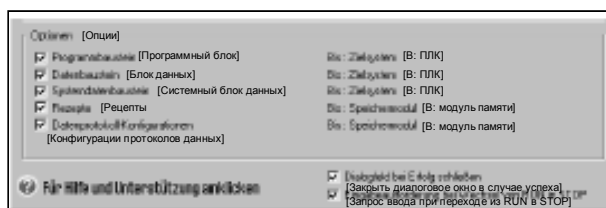


Рис. 14–8. Загрузка проекта с конфигурацией протоколов данных

Использование проводника S7–200

Проводник S7–200 – это приложение, используемое для чтения протокола данных из модуля памяти, а затем сохранения этого протокола в CSV-файле (в файле, в котором значения разделены запятыми).

При каждом чтении протокола данных создается новый файл. Этот файл сохраняется в каталоге Data Log [Протокол данных]. имя файла имеет следующий формат: адрес ПЛК, имя протокола данных, дата и время.

Вы можете выбрать, должно ли приложение, связанное с расширением CSV, автоматически запускаться при успешном считывании протокола данных. Этот выбор можно сделать через контекстное меню правой клавиши мыши для файла протокола данных.

Каталог протоколов данных находится в каталоге, указанном при инсталляции. Если STEP7 не установлен, то по умолчанию каталогом инсталляции является c:\program files\siemens\Microsystems. Если STEP7 установлен, то каталогом инсталляции по умолчанию является c:\siemens\Microsystem.

Для чтения протокола данных действуйте следующим образом:

1. Откройте проводник Windows. Должна автоматически появиться папка My S7–200 Network [Моя сеть S7–200].
2. Выберите папку My S7–200 Network.
3. Выберите папку соответствующего ПЛК S7–200.
4. Выберите папку модуля памяти
5. Найдите нужный файл конфигурации протоколов данных. Эти файлы называются DAT Configuration x (DATx).
6. Вызовите правой кнопкой мыши контекстное меню, а затем выберите Upload [Загрузить из CPU].

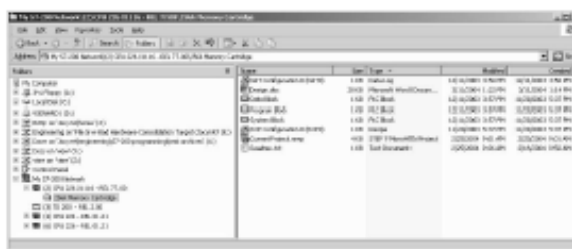


Рис. 14–9. Использование проводника S7–200

Команды, создаваемые Мастером протоколирования данных

Мастер протоколирования данных добавляет в ваш проект одну подпрограмму-команду.

Подпрограмма DATx_WRITE

Подпрограмма DATx_WRITE записывает текущие значения полей протокола данных в модуль памяти. DATx_WRITE добавляет одну запись к запротоколированным данным в модуле памяти. Вызов этой подпрограммы осуществляется следующим образом.

Если этой команде не удастся правильно обратиться к модулю памяти, то возвращается ошибка 132.

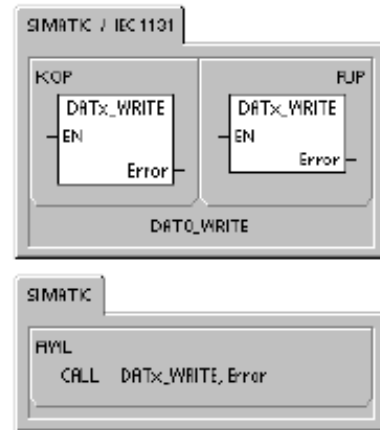


Таблица 14–1. Параметры подпрограммы DATx_WRITE

Входы/выходы	Тип данных	Операнды
Error	Byte	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



Совет

ЭСППЗУ, используемое в модуле памяти, поддерживает ограниченное количество операций записи. Обычно оно составляет один миллион циклов записи. Когда этот предел достигнут, ЭСПЗУ перестает работать надлежащим образом.

Обратите внимание на то, чтобы команда DATx_WRITE не выполнялась в каждом цикле обработки программы. Активизация этой команды в каждом цикле приводит к износу модуля памяти за относительно короткий интервал времени.

Автоматическая настройка PID и панель управления настройкой PID 15

Функция автоматической настройки PID встроена в ПЛК S7-200, а в STEP 7-Micro/WIN теперь добавлена панель управления настройкой PID. Вместе эти две функции существенно расширяют полезность и облегчают использование функции PID, предоставляемой в распоряжение микроконтроллерами серии S7-200.

Автоматическая настройка может быть инициирована посредством программы пользователя через панель оператора или через панель управления настройкой PID. PID-регуляторы могут автоматически настраиваться друг за другом или, в случае необходимости, все восемь регуляторов могут настраиваться одновременно. Функция автоматической настройки вычисляет предлагаемые значения настройки (близкие к оптимальным) для коэффициента усиления, времени воздействия по интегралу (сброса) и времени воздействия по производной (темпа). Кроме того, вы можете выбрать настройку для быстрой, средней, медленной и очень медленной реакции вашего регулятора.

С помощью панели управления настройкой PID вы можете инициировать процесс автоматической настройки, прервать этот процесс и наблюдать результаты в графической форме. Панель управления отображает все ошибочные состояния и предупреждения, которые могут быть сгенерированы. Она позволяет также применить коэффициент усиления, время воздействия по интегралу и время воздействия по производной, рассчитанные функцией автоматической настройки.

В этой главе

Автоматическая настройка PID	374
Расширенная таблица контура регулирования	374
Предпосылки	377
Автоматическое определение гистерезиса и отклонения	377
Последовательность действий для автоматической настройки	378
Исключительные ситуации	379
Указания относительно выхода регулируемой переменной за пределы допустимого диапазона (код результата 3)	379
Панель управления настройкой PID	380

Автоматическая настройка PID

Введение

Алгоритм автоматической настройки, используемый в S7-200, основан на методе, называемом релейной обратной связью, предложенном К. Й. Острёмом (Åström) и Т. Хегглундом (Hägglund) в 1984 году. За последние двадцать лет использовалась в различных отраслях промышленности.

Принцип релейной обратной связи состоит в генерировании небольших, но непрерывных колебаний в устойчивом, в противном случае, процессе. На основе наблюдаемых периода колебаний и изменений амплитуды регулируемых переменных определяются предельная частота и предельный коэффициент усиления процесса. Затем с помощью предельного коэффициента усиления и предельной частоты функция автоматической настройки предлагает значения для настройки коэффициента усиления, времен воздействия по интегралу и по производной.

Предлагаемые значения зависят от вашего выбора для скорости реакции регулятора для вашего процесса. Вы можете выбрать быструю, среднюю, медленную и очень медленную реакцию. В зависимости от вашего процесса быстрая реакция может привести к перерегулированию и соответствует в этом случае настройке со слабым затуханием. Средняя скорость реакции может находиться на границе возникновения перерегулирования и соответствует настройке с критически затуханием. Медленная реакция, возможно, не приведет к возникновению перерегулирования и соответствует настройке с сильным демпфированием. Очень медленная реакция, возможно, не приведет к возникновению перерегулирования и соответствует настройке с очень сильным демпфированием.

Кроме предложения значений настройки, функция автоматической настройки может автоматически определять значения для гистерезиса и пиковых значений отклонения регулируемой величины. Эти параметры используются для уменьшения влияния помех в процессе и ограничивают амплитуду непрерывных колебаний, генерируемых функцией автоматической настройки.

Функция автоматической настройки может определять предлагаемые значения настройки для P-, PI-, PD- и PID-регуляторов как прямого, так и обратного действия.

Назначение функции автоматической настройки состоит в определении набора параметров настройки, дающих разумное приближение к оптимальным значениям для вашего регулятора. Если вы начнете с предложенных значений настройки, то затем вы можете выполнить тонкую настройку и действительно оптимизировать свой процесс.

Расширенная таблица контура регулирования

Команда PID для S7-200 связана с таблицей контура регулирования, которая содержит параметры регулятора. Первоначально эта таблица имела длину 36 байтов. С добавлением функции автоматической настройки PID таблица контура регулирования была расширена и теперь имеет длину 80 байтов. Расширенная таблица контура регулирования представлена в таблицах 15-1 и 15-2.

Если вы используете панель управления настройкой PID, то обработка таблицы для контура регулирования выполняется панелью управления. Если вам нужно сделать автоматическую настройку доступной на панели оператора, то ваша программа должна организовать взаимодействие между оператором и таблицей контура регулирования, чтобы инициировать и наблюдать процесс автоматической настройки, а затем применить предлагаемые значения настройки.

Таблица 15–1. Таблица контура регулирования

Смещение	Поле	Формат	Тип	Описание
0	Регулируемая переменная (PV_n)	REAL	In	Содержит регулируемую переменную, которая должна быть масштабирована в диапазоне от 0.0 до 1.0.
4	Заданное значение (SP_n)	REAL	In	Содержит заданное значение, которое должно быть масштабировано в диапазоне от 0.0 до 1.0.
8	Управляющее воздействие (M_n)	REAL	In/Out	Содержит расчетное управляющее воздействие, масштабированное в диапазоне от 0.0 до 1.0.
12	Коэффициент усиления (K_C)	REAL	In	Содержит коэффициент усиления, который является коэффициентом пропорциональности. Он может быть положительным или отрицательным числом.
16	Интервал опроса (T_s)	REAL	In	Содержит интервал опроса в секундах. Должен быть положительным числом.
20	Время воздействия по интегралу, или сброс (T_I)	REAL	In	Содержит время воздействия по интегралу, или сброс, в минутах.
24	Время воздействия по производной, или темп (T_D)	REAL	In	Содержит время воздействия по производной, или темп, в минутах.
28	Интегральная сумма/смещение (MX)	REAL	In/Out	Содержит смещение, или интегральную сумму, значение между 0.0 и 1.0.
32	Предыдущее значение регулируемой переменной (PV_{n-1})	REAL	In/Out	Содержит значение регулируемой переменной, сохраненное после последнего исполнения команды PID.
36	Идентификатор расширенной таблицы PID контура регулирования	ASCII	Constant	'PIDA' (расширенная таблица PID, версия A): константа ASCII
40	Управление автонстройкой (ACNTL)	BYTE	In	См. табл. 15–2
41	Состояние автонстройкой (ASTAT)	BYTE	Out	См. табл. 15–2
42	Результат автонстройкой (ARES)	BYTE	In/Out	См. табл. 15–2
43	Конфигурация автонстройкой (ACNFG)	BYTE	In	См. табл. 15–2
44	Отклонение (DEV)	REAL	In	Нормализованная величина максимальной амплитуды колебаний регулируемой переменной (диапазон: от 0,025 до 0,25).
48	Гистерезис (HYS)	REAL	In	Нормализованная величина гистерезиса регулируемой переменной, используемая для определения пересечений нуля (диапазон: от 0,005 до 0,1). Если отношение DEV к HYS меньше 4, то при автоматической настройке появится предупреждение.
52	Начальный шаг управляющего воздействия (STEP)	REAL	In	Нормализованная величина изменения шага в значении управляющего воздействия, используемая для возбуждения колебаний в регулируемой величине (диапазон: от 0,05 до 0,4)
56	Время контроля (WDOG)	REAL	In	Максимально допустимое время между пересечениями нуля в секундах (диапазон: от 60 до 7200)
60	Предлагаемый коэффициент усиления (AT K_C)	REAL	Out	Предлагаемый коэффициент усиления регулятора, определяемый в процессе автоматической настройки.
64	Предлагаемое время воздействия по интегралу (AT T_I)	REAL	Out	Предлагаемое время воздействия по интегралу, определяемое в процессе автоматической настройки.
68	Предлагаемое время воздействия по производной (AT T_D)	REAL	Out	Предлагаемое время воздействия по производной, определяемое в процессе автоматической настройки.
72	Фактическая величина шага (ASTEP)	REAL	Out	Нормализованная величина шага управляющего воздействия, определяемая в процессе автоматической настройки.
76	Фактический гистерезис (AHYS)	REAL	Out	Нормализованное значение гистерезиса регулируемой переменной, определяемое в процессе автоматической настройки.

Таблица 15–2. Расширенное описание полей управления и состояния

Поле	Описание						
Управление автонастройкой (ACNTL) Вход - байт	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">EN</td> </tr> </table> <p>EN - 1 для запуска автонастройки; 0 для прерывания автонастройки</p>	MSB	LSB	7	0	0	EN
MSB	LSB						
7	0						
0	EN						
Состояние автонастройки (ASTAT) Выход - байт	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">W0</td> <td style="text-align: center;">IP</td> </tr> </table> <p>W0 - Предупреждение: Настройка отклонения не превышает в четыре раза настройку гистерезиса. W1 - Предупреждение: Несогласованные отклонения процесса могут привести к неправильной настройке значения шага управляющего воздействия. W2 - Предупреждение: Фактическое среднее отклонение не превышает в четыре раза настройку гистерезиса. АН - Идет расчет автоматического гистерезиса: 0 – расчет не выполняется 1 - расчет выполняется IP - Автонастройка выполняется: 0 - не выполняется 1 - выполняется</p> <p>Каждый раз, когда запускается последовательность автоматической настройки, ПЛК сбрасывает биты предупреждений и устанавливает бит выполнения. После завершения автоматической настройки ПЛК сбрасывает бит выполнения.</p>	MSB	LSB	7	0	W0	IP
MSB	LSB						
7	0						
W0	IP						
Результат автонастройки (ARES) Вход/выход - байт	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">D</td> <td style="text-align: center;">Код результата</td> </tr> </table> <p>D - Бит завершения: 0 – автоматическая настройка не завершена 1 - автоматическая настройка завершена Перед запуском автоматической настройки должен быть установлен в 0</p> <p>Код результата: 00 - нормальное завершение (предлагаемые значения настройки доступны) 01 - прервано пользователем 02 - прервано, превышение времени контроля при ожидании пересечения нуля 03 - прервано, регулируемая переменная (PV) вне диапазона 04 - прервано, превышено максимальное значение гистерезиса 05 - прервано, обнаружено недопустимое значение для конфигурации 06 - прервано, обнаружена числовая ошибка 07 - прервано, команда PID исполнена при отсутствии потока сигнала (контур регулирования в ручном режиме) 08 - прервано, автоматическая настройка допустима только для P-, PI-, PD- и PID-регуляторов 09 – 7F - резерв</p>	MSB	LSB	7	0	D	Код результата
MSB	LSB						
7	0						
D	Код результата						
Конфигурация автонастройки (ACNFG) Вход - байт	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">HS</td> </tr> </table> <p>R1 R0 Динамическая реакция 0 0 Быстрая реакция 0 1 Средняя реакция 1 0 Медленная реакция 1 1 Очень медленная реакция DS - Настройка отклонения: 0 – использовать значение отклонения из таблицы контура регулирования 1 – определять значение отклонения автоматически HS - Настройка гистерезиса: 0 - использовать значение гистерезиса из таблицы контура регулирования 1 - определять значение гистерезиса автоматически</p>	MSB	LSB	7	0	0	HS
MSB	LSB						
7	0						
0	HS						

Предпосылки

Контур регулирования, который вы хотите настраивать автоматически, должен находиться в автоматическом режиме. Управляющее воздействие должно устанавливаться путем исполнения команды PID. Автоматическая настройка терпит неудачу, если регулятор работает в ручном режиме.

Перед инициализацией автоматической настройки ваш процесс должен быть приведен в устойчивое состояние, которое означает, что регулируемая переменная достигла заданного значения (или, для P-регулятора, постоянной разности между регулируемой переменной и заданным значением) и что регулируемая переменная не изменяется скачками.

В идеале, значение управляющего воздействия при запуске автоматической настройки должно находиться в центральной части диапазона управления. Процедура автоматической настройки вызывает в процессе колебания, выполняя малые шаговые изменения управляющего воздействия. Если управляющее воздействие находится вблизи от одного из краев диапазона управления, то шаговые изменения, вызванные процедурой автоматической настройки, могут привести управляющее воздействие к попытке выхода за пределы допустимого диапазона.

Если бы это случилось, это могло бы привести к возникновению ошибочного состояния в автоматической настройке и, конечно же, к определению предлагаемых значений, не являющихся близкими к оптимальным.

Автоматическое определение гистерезиса и отклонения

Параметр гистерезис задает отклонение (положительное или отрицательное) от заданного значения, которое регулируемой переменной (PV) разрешается делать, не заставляя релейное управляющее устройство изменять управляющее воздействие. Эта величина используется для минимизации влияния помех в сигнале PV, чтобы более точно определять собственную частоту колебаний процесса.

Если выбирается автоматическое определение величины гистерезиса, то функция автоматической настройки выполняет последовательность действий по определению гистерезиса. Эта последовательность включает в себя опрос регулируемой переменной через некоторый интервал времени. Затем на основе результатов опроса вычисляется стандартное отклонение.

Чтобы получить статистически значимые результаты, должно быть выполнено не менее 100 опросов. В контуре регулирования с интервалом опроса 200 мс получение 100 опросов занимает 20 секунд. В контурах регулирования с более длительным интервалом опроса это потребует более длительного времени. Хотя 100 опросов могут быть получены за менее, чем 20 секунд в контурах регулирования с интервалами опроса меньше 200 мс, последовательность действий для определения гистерезиса всегда занимает не менее 20 секунд.

После выполнения всех опросов рассчитывается стандартное отклонение для этого набора опросов. Величина гистерезиса определяется как двукратное стандартное отклонение. Рассчитанное значение гистерезиса записывается в поле для фактического гистерезиса (AHYS) таблицы контура регулирования.



Совет

Когда производится автоматическое определение гистерезиса, нормальные расчеты для команды PID не выполняются. Поэтому безусловно необходимо, чтобы перед инициализацией автоматической настройки процесс находился в устойчивом состоянии. Это даст лучший результат для значения гистерезиса и обеспечит сохранение управляемости процесса во время автоматического определения величины гистерезиса.

Параметр отклонение определяет желаемый размах колебаний регулируемой переменной около заданного значения. Если вы выбираете автоматическое определение этой величины, то желаемое отклонение регулируемой переменной вычисляется умножением величины гистерезиса на 4,5. Управляющее воздействие устанавливается пропорционально, чтобы вызвать в процессе колебания с этой амплитудой во время автоматической настройки.

Последовательность действий для автоматической настройки

Автоматическая настройка начинается после определения величин гистерезиса и отклонения. Процесс настройки начинается, когда к управляющему воздействию прилагается начальное шаговое изменение.

Изменение управляющего воздействия вызывает соответствующее изменение значения регулируемой переменной. Когда изменение управляющего воздействия приводит к такому отклонению регулируемой переменной от заданного значения, которое выходит за границы гистерезиса, функция автоматической настройки обнаруживает переход через нулевое значение. При каждом переходе через ноль функция автоматической настройки заставляет управляющее воздействие изменяться в противоположном направлении.

Функция автоматической настройки продолжает опрашивать регулируемую переменную, ожидая следующего перехода через ноль. Для завершения последовательности требуется двенадцать переходов через ноль. Величина наблюдаемых значений регулируемой переменной от пика к пику (ошибка пиковых значений) и темп, с которым происходят переходы через ноль, непосредственно связаны с динамикой процесса.

В начале процесса автоматической настройки значение шага управляющего воздействия однократно настраивается таким образом, чтобы вызвать последующие колебания регулируемой переменной, наиболее соответствующие желаемой величине отклонения. После того как эта настройка выполнена, новая величина шага управляющего воздействия записывается в поле для фактической величины шага (ASTEP) таблицы контура регулирования.

Последовательность действий для автоматической настройки завершается с ошибкой, если время между пересечениями нуля превышает установленное для этих событий время контроля. По умолчанию время контроля устанавливается равным двум часам.

На рис. 15–1 показано поведение управляющего воздействия и регулируемой переменной во время автоматической настройки в контуре прямого регулирования. Для инициализации и наблюдения за процессом настройки была использована панель управления настройкой PID.

Обратите внимание, как функция автоматической настройки переключает управляющее воздействие, чтобы заставить процесс (как свидетельствует значение регулируемой переменной) выполнять малые колебания. Частота и амплитуда колебаний регулируемой переменной дают информацию о коэффициенте усиления и собственной частоте процесса.

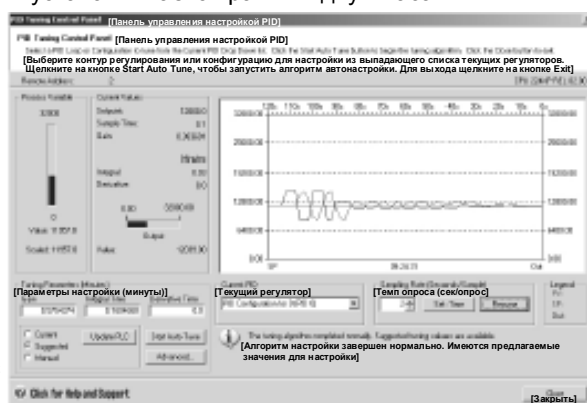


Рис. 15–1. Автоматическая настройка в контуре прямого регулирования

Пояснения к рисунку: Remote Address – удаленный адрес; Process Variable – регулируемая переменная; Value – значение; Scaled – масштабированное; Current Values – текущие значения; Setpoint – заданное значение; Sample Time – интервал опроса; Gain – коэффициент усиления; Derivative – производная; Integral Time – время воздействия по интегралу; Derivative Time – время воздействия по производной; Suggested – предлагаемые; Manual – устанавливаемые вручную; Update PLC – обновить ПЛК; Start Auto Tune – запустить автонастройку; Set Time – установить время; Resume – возобновить.

На основе информации, собранной во время автоматической настройки о частоте и коэффициенте усиления процесса, рассчитываются наибольшее усиление и наибольшая частота. Исходя из этих значений, рассчитываются предлагаемые величины коэффициента усиления, времени воздействия по интегралу и времени воздействия по производной.



Совет

Какие значения для настройки рассчитывает функция автоматической настройки, определяется типом вашего регулятора. Например, для PI-регулятора, функция автоматической настройки рассчитывает коэффициент усиления и время воздействия по интегралу, а предлагаемое время воздействия по производной будет равно 0.0 (нет воздействия по производной).

Как только автоматическая настройка завершается, управляющее воздействие контура регулирования возвращается к своему начальному значению. В следующем цикле регулирования будут выполняться обычные расчеты для команды PID.

Исключительные ситуации

В процессе настройки могут быть сгенерированы три вида предупреждений. Информация об этих предупреждениях содержится в трех битах поля ASTAT таблицы контура регулирования и, будучи установленными, эти биты остаются установленными, пока не будет инициирована следующая последовательность действий для автоматической настройки.

- Предупреждение 0 генерируется, если значение отклонения не превышает как минимум в четыре раза величину гистерезиса. Эта проверка выполняется, когда величина гистерезиса фактически известна, что зависит от настроек при автоматическом определении гистерезиса.
- Предупреждение 1 генерируется, если имеется более, чем 8-кратная разница между двумя значениями пиковой ошибки, полученными в течение первых 2,5 циклов процедуры автоматической настройки.
- Предупреждение 2 генерируется, если измеренное среднее значение пиковой ошибки не превышает как минимум в четыре раза величину гистерезиса.

Кроме предупреждений, возможно появление нескольких ошибочных состояний. В таблице 15–3 приведены ошибочные состояния вместе с описанием причины каждой ошибки.

Таблица 15–3. Ошибочные состояния при выполнении настройки

Код результата (в ARES)	Состояние
01 прервано пользователем	Бит EN сброшен во время выполнения настройки
02 прервано из-за превышения времени контроля между пересечениями нуля	Длительность половины цикла превышает интервал контроля проходов через ноль
03 прервано из-за выхода процесса за пределы допустимого диапазона	Регулируемая переменная выходит за пределы допустимого диапазона: <ul style="list-style-type: none"> • во время процесс автоматического определения величины гистерезиса, или • дважды до четвертого пересечения нуля, или • после четвертого пересечения нуля
04 прервано за превышения гистерезисом максимально допустимого значения	Величина гистерезиса, определенная пользователем или автоматически, > максимума
05 прервано из-за недопустимого значения конфигурации	Следующие ошибки диапазона: <ul style="list-style-type: none"> • начальное значение управляющего воздействия < 0.0 или > 1.0 • значение отклонения, заданное пользователем <= величине гистерезиса или > максимума • начальный шаг изменения управляющего воздействия <= 0.0 или > максимума • время контроля пересечения нуля < минимума • значение интервала опроса в таблице контура регулирования отрицательно
06 прервано из-за числовой ошибки	Обнаружено недопустимое число с плавающей точкой или деление на ноль
07 команда PID выполнена при отсутствии потока сигнала (ручной режим)	Команда PID, выполненная при отсутствии потока сигнала, когда выполняется или запрошена автоматическая настройка
08 автоматическая настройка допускается только для P-, PI-, PD- и PID-регуляторов	Регулятор не является P-, PI-, PD- или PID-регулятором

Указания относительно выхода регулируемой переменной за пределы допустимого диапазона (код результата 3)

Функция автоматической настройки считает, что регулируемая переменная находится в допустимом диапазоне, если ее значение больше 0.0 и меньше 1.0.

Если обнаружено, что регулируемая переменная вышла за пределы допустимого диапазона во время автоматического определения величины гистерезиса, то настройка немедленно прекращается с сообщением об ошибке выхода за пределы допустимого диапазона.

Если обнаружено, что регулируемая переменная вышла за пределы допустимого диапазона между началом автоматической настройки и четвертым переходом через ноль, то значение шага изменения управляющего воздействия сокращается на половину, а процесс настройки запускается сначала. Если обнаружен второй выход регулируемой переменной за пределы допустимого диапазона после четвертого пересечения нуля вслед за перезапуском, то настройка прекращается с сообщением об ошибке выхода за пределы допустимого диапазона.

Если после четвертого перехода через ноль обнаруживается, что регулируемая переменная находится за пределами допустимого диапазона, то настройка прекращается с сообщением об ошибке выхода за пределы допустимого диапазона.

Панель управления настройкой PID

STEP 7-Micro/WIN включает в себя панель управления настройкой PID, которая позволяет графически отображать поведение вашего PID-регулятора. Кроме того, дает вам возможность инициировать последовательность действий для автоматической настройки, прервать эту последовательность и применить предлагаемые значения для настройки или ввести ваши собственные значения.

Для использования панели управления вы должны обмениваться данными с ПЛК S7-200, а в ПЛК должна находиться конфигурация для PID-регулятора, сгенерированная Мастером. ПЛК должен находиться в режиме RUN, чтобы панель управления могла отображать функционирование PID-регулятора. На рис. 15-2 показана стандартная панель управления.

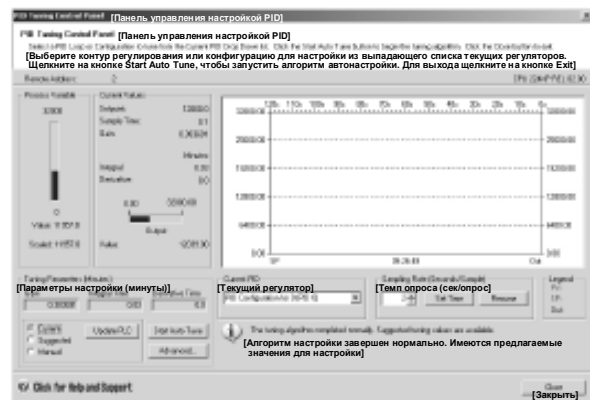


Рис. 15-2. Панель управления настройкой PID
Пояснения к рисунку: см. рис. 15-1.

Панель управления отображает адрес станции (Remote Address [Удаленный адрес]) целевого ПЛК в верхней левой части экрана. В правой верхней части экрана отображаются тип ПЛК и номер версии. Под полем для удаленного адреса находится представление регулируемой переменной (PV) в виде столбиковой диаграммы, а также ее масштабированное (Scaled) и немасштабированное значения (Value). Справа от столбиковой диаграммы расположена область текущих значений (Current Values).

В области текущих значений представлены заданное значение (Setpoint, SP), интервал опроса Sample Time), коэффициент усиления (Gain), время воздействия по интегралу (Integral time) и время воздействия по производной. Величина управляющего воздействия (Output) отображается в горизонтальной столбиковой диаграмме вместе с его числовым значением. Справа от области текущих значений находится графическое отображение.

Графическое отображение показывает закодированные цветами представления PV, SP и управляющего воздействия в функции времени. Вертикальная шкала в левой части графика соответствует значениям PV и SP, а вертикальная шкала в правой части графика соответствует управляющему воздействию.

В нижней левой части экрана находится область параметров настройки в минутах (Tuning Parameters). Внутри этой области отображаются значения коэффициента усиления (Gain), времени воздействия по интегралу (Integral Time) и времени воздействия по производной (Derivative Time). Селективные кнопки показывают, какие значения коэффициента усиления, времени воздействия по интегралу и времени воздействия по производной отображаются - текущие (Current), предлагаемые (Suggested) или установленные вручную (Manual). Вы можете щелкнуть на селективной кнопке, чтобы отобразить один из трех источников этих значений. Для изменения параметров настройки щелкните на селективной кнопке Manual.

Вы можете использовать кнопку Update PLC [Обновить ПЛК], чтобы передать отображенные значения коэффициента усиления, времени воздействия по интегралу и времени воздействия по производной в ПЛК для наблюдаемого PID-регулятора. Вы можете использовать кнопку Start Auto Tune [Запустить автоматическую настройку] для инициализации последовательности автоматической настройки. После запуска автоматической настройки кнопка Start Auto Tune становится кнопкой Stop Auto Tune [Остановить автоматическую настройку].

Непосредственно под графическим изображением находится область для выбора текущего PID-регулятора (Current PID) с разворачивающимся меню, которое дает вам возможность выбрать желаемый PID-регулятор, за которым вы хотите наблюдать с помощью панели управления.

В области Sampling Rate [Темп опроса] вы можете установить темп опроса для графического отображения от 1 до 480 секунд на опрос. Вы можете изменить темп опроса, а затем применить это изменение с помощью кнопки Set Time [Установить время]. Шкала времени графика автоматически настраивается на наилучшее отображение данных при новом темпе опроса.

Вы можете заморозить график, нажав кнопку Pause. Для возобновления опроса данных с выбранным интервалом нажмите кнопку Resume [Возобновить]. Для стирания графика выберите Clear [Очистить], нажав правую кнопку мыши внутри графика.

Справа от области параметров графика находится Legend [Условные обозначения], которые определяют цвета, которые используются для отображения значений PV, SP и управляющего воздействия.

Непосредственно под областью для выбора текущего регулятора (Current PID) находится область, которая используется для отображения информации, относящейся к выполняемой операции.

Кнопка Advanced ... [Больше ...] в области настройки параметров (Tuning Parameters) дает возможность более точной установки параметров для процесса автоматической настройки. Диалоговое окно расширенной настройки показано на рис. 15-3.

В этом диалоговом окне вы можете отметить триггерную кнопку, которая заставит функцию автоматической настройки автоматически определять значения для гистерезиса и отклонения (настройка по умолчанию) или вы можете ввести значения для этих полей, которые минимизируют помехи в вашем процессе во время процедуры автоматической настройки.

В области Other Options [Другие возможности] вы можете определить начальную величину шага управляющего воздействия и ввести интервал контроля времени пересечения нуля.

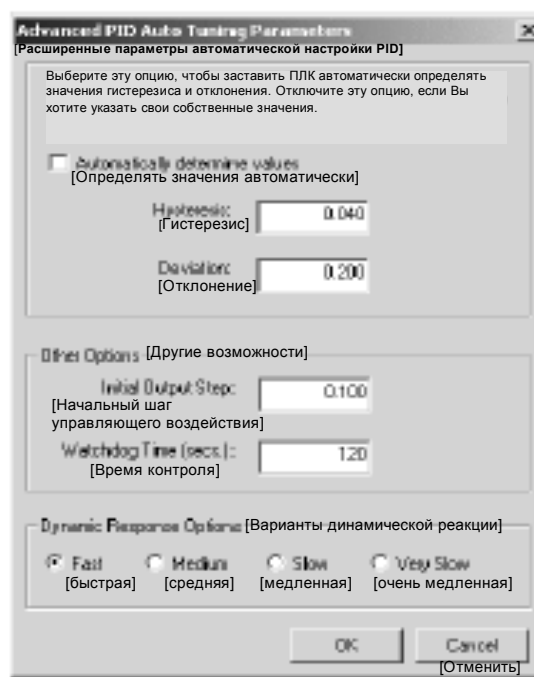


Рис. 15-3. Расширенные параметры

В области Dynamic Response Options [Варианты динамической реакции] щелкните на селективной кнопке, которая соответствует виду реакции контура регулирования, которую вы хотите иметь в своем процессе. В зависимости от вашего процесса быстрая реакция может привести к перерегулированию и соответствует в этом случае настройке со слабым затуханием. Средняя скорость реакции может находиться на границе возникновения перерегулирования и соответствует настройке с критически затуханием. Медленная реакция, возможно, не приведет к возникновению перерегулирования и соответствует настройке с сильным демпфированием. Очень медленная реакция, возможно, не приведет к возникновению перерегулирования и соответствует настройке с очень сильным демпфированием.

Сделав желаемые настройки, щелкните на ОК, чтобы вернуться в основное диалоговое окно панели управления настройкой PID.

После того как вы завершите автоматическую настройку и передадите предложенные параметры в ПЛК, вы можете использовать панель управления для наблюдения реакции вашего регулятора на ступенчатое изменение заданного значения. На рис. 15–4 показана реакция регулятора на изменение заданного значения (с 12000 до 14000) с первоначальными параметрами настройки (перед запуском автоматической настройки).

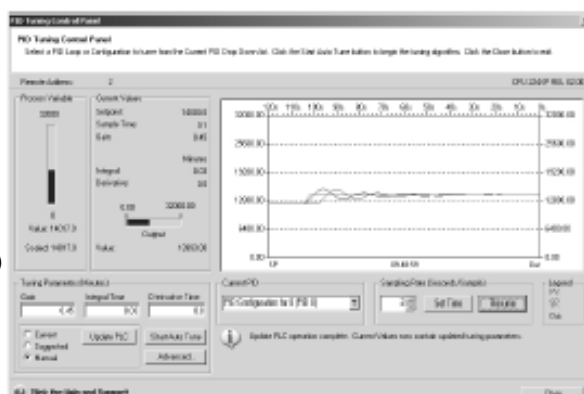


Рис. 15–4. Реакция на изменение заданного значения

Обратите внимание на перерегулирование и длительные последующие затухающие колебания процесса при использовании для настройки первоначальных параметров.

На рис. 15–5 показана реакция регулятора на такое же изменение заданного значения (с 12000 до 14000) после применения значений, определенных функцией автоматической настройки для быстрой реакции. Обратите внимание, что в этом процессе нет перерегулирования, но имеется несколько последующих колебаний. Если вы хотите устранить эти колебания за счет скорости реакции, вам нужно выбрать среднюю или медленную реакцию и перезапустить процесс автоматической настройки.



Рис. 15–5. Реакция после автоматической настройки

Если вы получили хорошую начальную точку для настройки параметров вашего контура регулирования, то вы можете использовать панель управления для тонкой настройки параметров. Затем вы можете наблюдать реакцию регулятора на изменения заданного значения. Таким образом, вы можете выполнить тонкую настройку для получения оптимальной реакции в своем приложении.

Технические данные

A

В этой главе

Общие технические данные	384
Технические данные CPU	387
Технические данные цифровых модулей расширения	396
Технические данные аналоговых модулей расширения	402
Технические данные модулей расширения для термпар и термометров сопротивления (RTD)	413
Технические данные модуля EM 277 PROFIBUS-DP	425
Технические данные модемного модуля EM 241	437
Технические данные модуля позиционирования EM 253	439
Технические данные модуля Ethernet (CP 243-1)	445
Технические данные модуля Internet (CP 243-1 IT)	447
Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2)	450
Дополнительные съемные модули	452
Кабель с разъемами для модулей расширения	453
Кабель RS-232/PPI Multi-Master и кабель USB/PPI Multi-Master	454
Имитаторы входов	458

Общие технические данные

Соответствие стандартам

Национальные и международные стандарты, перечисленные ниже, использовались для определения рабочих характеристик и проведения испытаний семейства продуктов S7-200. Таблица А–1 определяет конкретное соответствие этим стандартам.

- Директива Европейского сообщества (СЕ) по устройствам низкого напряжения 73/23/ЕЕС
EN 61131-2: Программируемые контроллеры – Требования к оборудованию
- Директива Европейского сообщества (СЕ) по электромагнитной совместимости 89/336/ЕЕС
Стандарты по электромагнитному излучению
EN 61000-6-3: Жилая, коммерческая зоны и легкая промышленность
EN 61000-6-4: Промышленная среда
Стандарты по электромагнитной помехоустойчивости
EN 61000-6-2: Промышленная среда
- Underwriters Laboratories [Лаборатории страхователей], Inc.: UL 508, зарегистрировано (Промышленная аппаратура управления)
Регистрационный номер E75310
- Canadian Standards Association [Канадская ассоциация стандартов]: CSA C22.2 номер 142, сертифицировано (Оборудование для управления процессами)
- Factory Mutual Research [Заводские совместные исследования]: номер класса 3600, номер класса 3611, FM-класс I, раздел 2, группы опасных помещений А, В, С и D, T4A и класс I, зона 2, IIC, T4.
- Директива Европейского сообщества по взрывоопасным атмосферам (ATEX) 94/9/ЕС
EN 60079-0 Общие требования
EN 50020 Искробезопасность 'i'
EN 60079-15 Род защиты 'n'

Сертификация на соответствие директиве ATEX 94/9/ЕС к моменту публикации данного документа не была завершена. За последней информацией обратитесь в своему местному представителю фирмы Siemens.



Совет

Серия продуктов SIMATIC S7–200 удовлетворяет стандарту CSA. Логотип cULus показывает, что S7–200 был проверен и сертифицирован Лабораториями страхователей (Underwriters Laboratories, UL) на соответствие стандартам UL 508 и CSA 22.2 No. 142.

Удостоверения о допуске к эксплуатации на море

На продукты S7–200 регулярно подаются заявки в определенные агентства на получение допусков к эксплуатации относительно определенных рынков и приложений. В этой таблице приведены агентства и номера сертификатов соответствия на продукты S7–200. Большинство продуктов S7–200, представленных в данном руководстве, имеют специальные допуски этих агентств. Если вам нужен список последних допусков к эксплуатации относительно определенных номеров для заказа, обратитесь к своему местному представителю фирмы Siemens.

Агентство	Номер сертификата
Lloyds Register of Shipping [Судовой регистр Ллойда] (LRS)	99 / 20018(E1)
American Bureau of Shipping [Американское судовое бюро] (ABS)	01-HG20020-PDA
Germanischer Lloyd [Германский Ллойд] (GL)	12 045 - 98 HH
Det Norske Veritas [Норвежский Веритас] (DNV)	A-8862
Bureau Veritas [Бюро Веритас] (BV)	09051 / B0BV
Nippon Kaiji Kyokai [Ниппон Кэйдзи Кёкай, Япония] (NK)	A-534
Polski Rejestr [Польский реестр]	TE/1246/883241/99

Срок службы реле

На рис. А–1 показаны типовые эксплуатационные данные, предоставленные поставщиками реле. Фактическая характеристика может отличаться в зависимости от вашего конкретного приложения.

Внешний защитный контур, подобранный в соответствии с нагрузкой, продлит срок службы контактов.

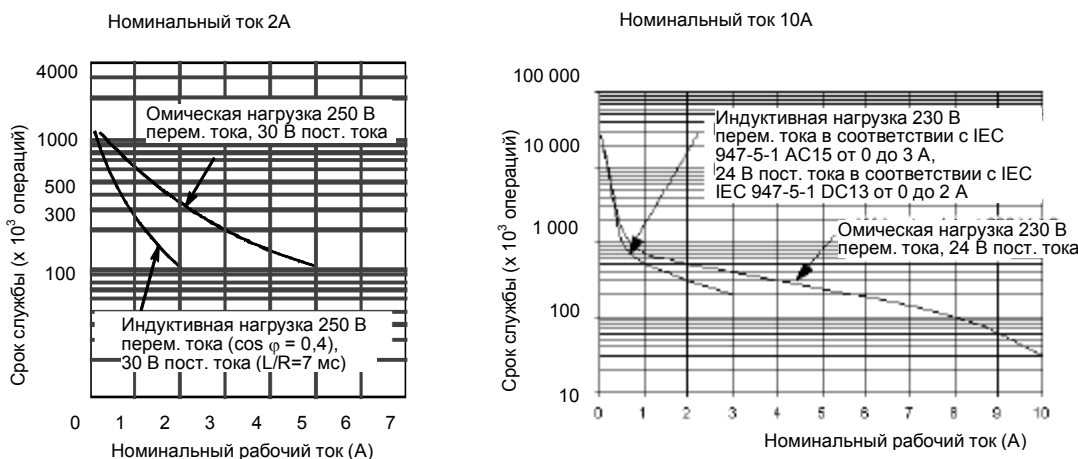


Рис. А–1. Срок службы реле

Технические данные

Все CPU S7–200 и модули расширения соответствуют техническим данным, перечисленным в таблице А–1.

Указание

Если механический контакт включает выходное напряжение для CPU S7–200 или цифрового модуля расширения, то он посылает сигнал «1» на цифровые выходы в течение примерно 50 микросекунд. Вы должны учитывать это, особенно если вы используете устройство, которое реагирует на короткие импульсы.

Таблица А–1. Технические данные

Условия окружающей среды — Транспортировка и хранение	
EN 60068-2-2, испытание Bb, сухое тепло и EN 60068-2-1, испытание Ab, холод	от –40° С до +70° С
EN 60068-2-30, испытание Db, влажное тепло	от 25° С до 55° С, влажность 95%
EN 60068-2-14, испытание Na, температурный удар	от –40° С до +70° С время пребывания 3 часа, 2 цикла
EN 60068-2-31, опрокидывание	100 мм, 4 падения, без упаковки
EN 60068-2-32, свободное падение	1 м, 5 раз, в упаковке для перевозки
Условия окружающей среды – Эксплуатация	
Диапазон температур окружающей среды (подача воздуха на 25 мм ниже модуля)	от 0° С до 55° С при горизонтальной установке, от 0° С до 45° С при вертикальной установке Влажность 95% без конденсации
Атмосферное давление	от 1080 до 795 гПа (соответствует высоте от -1000 до 2000 м)
Концентрация загрязняющих веществ	SO ₂ : < 0,5 0/00; H ₂ S: < 0,1 0/00; RH < 60% без конденсации
EN 60068-2-14, испытание Nb, изменение температуры	от 5° С до 55° С, 3° С/мин
EN 60068-2-27, механический удар	15 G, импульс 11 мс, 6 ударов по каждой из 3 осей
EN 60068-2-6, синусоидальные колебания	Монтаж в распределительном шкафу: 0,30 мм от 10 до 57 Гц; 2 G от 57 до 150 Гц Монтаж на профильной шине: 0,15 мм от 10 до 57 Гц; 1 G от 57 до 150 Гц 10 проходов по каждой оси, 1 октава в минуту
EN 60529, механическая защита IP20	Защита от прикосновения к высокому напряжению, обнаруживаемому стандартными щупами. Внешняя защита требуется для предохранения от пыли, грязи, воды и инородных предметов диаметром менее 12,5 мм.

Таблица А–1. Технические данные, продолжение

Электромагнитная совместимость – Помехоустойчивость согласно EN61000-6-2¹	
EN 61000-4-2, электростатический разряд	Воздушный разряд 8 кВ относительно всех поверхностей и коммуникационного порта, контактный разряд 4 кВ относительно незащищенных проводящих поверхностей
EN 61000-4-3, излучаемое электромагнитное поле	10 В/м, 80-1000 МГц и от 1,4 до 2,0 ГГц, амплитудная модуляция 80% при 1 кГц
EN 61000-4-4, быстрые переходные помехи	2 кВ, 5 кГц при наличии сети связи с системами питания переменного и постоянного тока 2 кВ, 5 кГц при наличии соединительной клеммы с входами-выходами 1 кВ, 5 кГц при наличии соединительной клеммы с коммуникационным оборудованием
EN 61000-4-5, устойчивость к перенапряжениям	Блок питания: 2 кВ асимметрично, 1 кВ симметрично входы/выходы 1 кВ симметрично (цели 24 В постоянного тока требуют внешней защиты от перенапряжений)
EN 61000-4-6, помехи, поступающие по проводам	от 0,15 до 80 МГц, 10 В _{эфф} , 80% амплитудная модуляция при 1 кГц
EN 61000-4-11, провалы напряжения, краткие перебои и изменения напряжения	Уменьшение >95% в течение 8,3 мс, 83 мс, 833 мс и 4167 мс
VDE 0160, непериодическое перенапряжение	В линии 85 В переменного тока, фазовый угол 90°, подача 390 В (пиковое), импульс 1,3 мс В линии 180 В переменного тока, фазовый угол 90°, подача 750 В (пиковое), импульс 1,3 мс
Электромагнитная совместимость — Создаваемые помехи (по проводам и путем излучения) по EN 61000-6-3² и EN 61000-6-4	
EN 55011, класс А, группа 1, по проводам ¹ 0,15 МГц - 0,5 МГц 0,5 МГц - 5 МГц 5 МГц - 30 МГц	< 79 дБ (мкВ) квазипиковое; < 66 дБ (мкВ) среднее < 73 дБ (мкВ) квазипиковое; < 60 дБ (мкВ) среднее < 73 дБ (мкВ) квазипиковое; < 60 дБ (мкВ) среднее
EN 55011, класс А, группа 1, излучаемое ¹ 30 МГц - 230 МГц 230 МГц - 1 ГГц	40 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 10 м 47 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 10 м
EN 55011, класс В, группа 1, по проводам ² 0,15 МГц - 0,5 МГц 0,5 МГц - 5 МГц 5 МГц - 30 МГц	< 66 дБ (мкВ) квазипик., уменьш. с логарифм. частоты до 56 дБ (мкВ); < 56 дБ (мкВ) среднее, уменьш. с логарифмом частоты до 46 дБ (мкВ) < 56 дБ (мкВ) квазипиковое; < 46 дБ (мкВ) среднее < 60 дБ (мкВ) квазипиковое; < 50 дБ (мкВ) среднее
EN 55011, класс В, группа 1, излучаемое ² 30 МГц - 230 кГц 230 МГц - 1 ГГц	30 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 10 м 37 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 10 м
Высоковольтное испытание изоляции	
Цепи с номинальным напряж. 24 В/5 В Цепи 115/230 В относительно земли Цепи 115/230 В относит. цепей 115/230 В Цепи 230 В относит. цепей 24 В/5 В Цепи 115 В относит. цепей 24 В/5 В	500 В переменного тока (границы оптической развязки) 1500 В переменного тока 1500 В переменного тока 1500 В переменного тока 1500 В переменного тока
<p>1 Блок должен устанавливаться на заземленной металлической раме с присоединением "земли" S7-200 непосредственно к металлу монтажной арматуры. Кабели прокладываются вдоль металлических опор.</p> <p>2 Блок должен устанавливаться в заземленный металлический корпус. Входная линия электроснабжения переменного тока должна оборудоваться фильтром EPCOS B84115-E-A30 или эквивалентным фильтром, максимальная длина проводов от фильтров до S7-200 равна 25 см. Провода источника питания 24 В постоянного тока и датчиков должны быть экранированными.</p>	

Технические данные CPU

Таблица А–2. Номера для заказа CPU

Номер для заказа	Модель CPU	Блок питания (номинальное напряжение)	Цифровые входы	Цифровые выходы	Коммуникационные порты	Аналоговые входы	Аналоговые выходы	Съемный клеммный блок
6ES7 211-0AA23-0XB0	CPU 221	24 В пост. тока	6 x 24 В пост. тока	4 x 24 В пост. тока	1	Нет	Нет	Нет
6ES7 211-0BA23-0XB0	CPU 221	от 120 до 240 В перем. тока	6 x 24 В пост. тока	4 x реле	1	Нет	Нет	Нет
6ES7 212-1AB23-0XB0	CPU 222	24 В пост. тока	8 x 24 В пост. тока	6 x 24 В пост. тока	1	Нет	Нет	Нет
6ES7 212-1BB23-0XB0	CPU 222	от 120 до 240 В перем. тока	8 x 24 В пост. тока	6 x реле	1	Нет	Нет	Нет
6ES7 214-1AD23-0XB0	CPU 224	24 В пост. тока	14 x 24 В пост. тока	10 x 24 В пост. тока	1	Нет	Нет	Да
6ES7 214-1BD23-0XB0	CPU 224	от 120 до 240 В перем. тока	14 x 24 В пост. тока	10 x реле	1	Нет	Нет	Да
6ES7 214-2AD23-0XB0	CPU 224XP	24 В пост. тока	14 x 24 В пост. тока	10 x 24 В пост. тока	2	2	1	Да
6ES7 214-2AS23-0XB0	CPU 224XPsi	24 В пост. тока	14 x 24 В пост. тока	10 x 24 В пост. тока	2	2	1	Да
6ES7 214-2BD23-0XB0	CPU 224XP	от 120 до 240 В перем. тока	14 x 24 В пост. тока	10 x реле	2	2	1	Да
6ES7 216-2AD23-0XB0	CPU 226	24 В пост. тока	24 x 24 В пост. тока	16 x 24 В пост. тока	2	Нет	Нет	Да
6ES7 216-2BD23-0XB0	CPU 226	от 120 до 240 В перем. тока	24 x 24 В пост. тока	16 x реле	2	Нет	Нет	Да

Таблица А–3. Общие технические данные CPU

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Имеющееся напряжение пост. тока	
					+5 В	+24 В ¹
6ES7 211-0AA23-0XB0	CPU 221 DC/DC/DC, 6 входов/ 4 выхода	90 x 80 x 62	270 г	3 Вт	0 мА	180 мА
6ES7 211-0BA23-0XB0	CPU 221 AC/DC/Relay, 6 входов/ 4 реле	90 x 80 x 62	310 г	6 Вт	0 мА	180 мА
6ES7 212-1AB23-0XB0	CPU 222 DC/DC/DC, 8 входов/ 8 выходов	90 x 80 x 62	270 г	5 Вт	340 мА	180 мА
6ES7 212-1BB23-0XB0	CPU 222 AC/DC/Relay, 8 входов/ 6 реле	90 x 80 x 62	310 г	7 Вт	340 мА	180 мА
6ES7 214-1AD23-0XB0	CPU 224 DC/DC/DC, 14 входов/ 10 выходов	120.5 x 80 x 62	360 г	7 Вт	660 мА	280 мА
6ES7 214-1BD23-0XB0	CPU 224 AC/DC/Relay, 14 входов/ 10 реле	120.5 x 80 x 62	410 г	10 Вт	660 мА	280 мА
6ES7 214-2AD23-0XB0	CPU 224XP DC/DC/DC, 14 входов/10 выходов	140 x 80 x 62	390 г	8 Вт	660 мА	280 мА
6ES7 214-2AS23-0XB0	CPU 224XPsi DC/DC/DC, 14 входов/10 выходов	140 x 80 x 62	390 г	8 Вт	660 мА	280 мА
6ES7 214-2BD23-0XB0	CPU 224XP AC/DC/Relay 14 входов/10 реле	140 x 80 x 62	440 г	11 Вт	660 мА	280 мА
6ES7 216-2AD23-0XB0	CPU 226 DC/DC/DC 24 входа/18 выходов	196 x 80 x 62	550 г	11 Вт	1000 мА	400 мА
6ES7 216-2BD23-0XB0	CPU 226 AC/DC/Relay 24 входа/16 реле	196 x 80 x 62	660 г	17 Вт	1000 мА	400 мА

¹ Это напряжение питания датчиков 24 В пост. тока, имеющееся в распоряжении после того, как учтены потребности в питании катушек внутренних реле и коммуникационных портов.

Таблица А–4. Технические данные CPU

	CPU 221	CPU 222	CPU 224	CPU 224XP CPU 224XPSi	CPU 226
Память					
Размер программы пользователя с редактированием в режиме RUN без редактирования в режиме RUN	4096 байт 4096 байт		8192 байт 12288 байт	12288 байт 16384 байт	16384 байт 24576 байт
Данные пользователя	2048 байт		8192 байт	10240 байт	10240 байт
Буферизация (мощный конденсатор) (возможна батарейка)	Тип. 50 час. (мин. 8 час. при 40°C) Тип. 200 дней		Тип. 100 час. (мин. 70 час. при 40°C) Тип. 200 дней	Тип. 100 час. (мин. 70 час. при 40°C) Тип. 200 дней	
Входы/выходы					
Цифровые входы и выходы	6 входов/ 4 выхода	8 входов/6 выходов	14 входов/10 выходов	14 входов/ 10 выходов	24 входа/16 выходов
Аналоговые входы и выходы	нет			2 входа/1 выход	нет
Цифровые входы и выходы (образ процесса)	256 (128 входов/128 выходов)				
Аналоговые входы и выходы (образ процесса)	нет	32 (16 входов и 16 выходов)	64 (32 входа и 32 выхода)		
Макс. количество модулей расширения	нет	2 модуля ¹	7 модулей ¹		
Макс. количество интеллектуальных модулей	нет	2 модуля ¹	7 модулей ¹		
Входы для регистрации импульсов	6	8	14		24
Скоростные счетчики однофазные	Всего 4 счетчика 4 при 30 кГц		Всего 6 счетчиков 6 при 30 кГц	Всего 6 счетчиков 4 при 30 кГц 2 при 200 кГц	Всего 6 счетчиков 6 при 30 кГц
двухфазные	2 при 20 кГц		4 при 20 кГц	3 при 20 кГц 1 при 100 кГц	4 при 20 кГц
Импульсные выходы	2 при 20 кГц (только выходы постоянного тока)			2 при 100 кГц (только выходы постоянного тока)	2 при 20 кГц (только выходы постоянного тока)
Общие данные					
Таймеры	Всего таймеров 256; 4 таймера (1 мс); 16 таймеров (10 мс); 236 таймеров (100 мс)				
Счетчики	256 (с буферизацией от конденсатора большой емкости или батарейки)				
Биты внутренней памяти. Сохраняются при потере питания	256 (с буферизацией от конденсатора большой емкости или батарейки) 112 (сохраняются в ЭСППЗУ)				
Прерывания, управляемые временем	2 с разрешением 1 мс				
Прерывания по фронту сигнала	4 по нарастающему и/или 4 по убывающему фронту				
Аналоговый потенциометр	1 с разрешением 8 битов		2 с разрешением 8 битов		
Скорость выполнения булевых операций	0,22 мкс на команду				
Часы реального времени	Дополнительный съемный модуль		Встроенные		
Дополнительные съемные модули	Память, батарейка и часы реального времени		Память и батарейка		
Встроенные средства для обмена данными					
Порты (с ограничением мощности)	1 порт RS-485			2 порта RS-485	
Скорости передачи PPI, DP/T	9,6; 19,2; 187,5 кБод				
Скорости передачи для свободно программируемого обмена данными	от 1,2 кБод до 115,2 кБод				
Макс. длина кабеля на сегмент	С гальванически развязанным повторителем: 1000 м до 187,5 кБод, 1200 м до 38,4 кБод Без повторителя без гальванической развязки: 50 м				
Макс. количество станций	32 на сегмент, 126 на сеть				
Макс. количество master-устройств	32				
Двухточечное соединение (режим master-устройства PPI)	Да (NETR/NETW)				
Соединения MPI	Всего 4, 2 зарезервированы (1 для устройства программирования и 1 для панели оператора)				

1 Вы должны рассчитать свой баланс мощностей, чтобы определить, какую мощность (или ток) CPU S7-200 может поставить для вашей конфигурации. Если баланс мощностей CPU превышен, то вы не сможете подключить максимальное количество модулей. В этом приложении вы найдете потребности мощности CPU и модулей расширения, а в Приложении В вы сможете рассчитать свой баланс мощностей.

Таблица А–5. Данные о мощности CPU

Постоянный ток		Переменный ток	
Входная мощность			
Входное напряжение	от 20,4 до 28,8 В пост. тока		от 85 до 264 В перем. тока (от 47 до 63 Гц)
Входной ток	только CPU при 24 В пост. тока	Макс. нагрузка при 24 В пост. тока	только CPU
CPU 221	80 мА	450 мА	30/15 мА при 120/240 В перем. тока
CPU 222	85 мА	500 мА	40/20 мА при 120/240 В перем. тока
CPU 224	110 мА	700 мА	60/30 мА при 120/240 В перем. тока
CPU 224XP	120 мА	900 мА	70/35 мА при 120/240 В перем. тока
CPU 224XPsi	120 мА	900 мА	-
CPU 226	150 мА	1050 мА	80/40 мА при 120/240 В перем. тока
Толчок тока при включении	12 А при 28,8 В пост. тока		20 А при 264 В перем. тока
Электрическая развязка (полевых устройств с логикой)	Нет развязки		1500 В перем. тока
Время задержки (при потере питания)	10 мс при 24 В пост. тока		20/80 мс при 120/240 В перем. тока
Предохранитель (не заменяемый)	3 А, 250 В Slow Blow		2 А, 250 В Slow Blow
Питание датчиков 24 В пост. тока			
Напряжение датчиков (с ограничением мощности)	L+ минус 5 В		от 20,4 до 28,8 В пост. тока
Ограничение тока	Пиковый ток 1,5 А, тепловая граница, не приводящая к разрушению (номинальная нагрузка, см. таблицу А–3)		
Пульсации/помехи	Наводки от входного напряжения		Менее 1 В от пика к пику
Электрическая развязка (датчиков с логикой)	Нет развязки		

Таблица А–6. Технические данные цифровых входов CPU

Общие данные	Вход 24 В пост. тока (CPU 221, CPU 222, CPU 224, CPU 226)	Вход 24 В пост. тока (CPU 224XP)
Тип	Обычная полярность/обратная полярность (IEC тип 1 – обычная полярность)	Обычная полярность/обратная полярность (IEC тип 1 – обычная полярность, кроме входов I0.3 – I0.5)
Номинальное напряжение	Тип. 24 В пост. тока при 4 мА	Тип. 24 В пост. тока при 4 мА
Макс. допустимое длительное напряжение	30 В пост. тока	
Бросок напряжения	35 В пост. тока в течение 0,5 с	
Логика 1 (мин.)	15 В пост. тока при 2,5 мА	15 В пост. тока при 2,5 мА (I0.0 – I0.2 и I0.6 – I1.5) 4 В пост. тока при 8 мА (I0.3 – I0.5)
Логика 0 (макс.)	5 В пост. тока при 1 мА	5 В пост. тока при 1 мА (I0.0 – I0.2 и I0.6 – I1.5) 1 В пост. тока при 1 мА (I0.3 – I0.5)
Входная задержка	Настраивается (от 0,2 до 12,8 мс)	
Присоединение 2-проводного датчика близости (Вего)		
Допустимый ток утечки (макс.)	1 мА	
Электрическая развязка (полевых устройств с логикой)	Да	
Оптическая (гальваническая)	500 В перем. тока в течение 1 минуты	
Потенциально развязанные группы	См. схему соединений	
Входная частота скоростных счетчиков (HSC)		
Входы HSC	Уровень логики 1	Однофазный счетчик
Все HSC	от 15 до 30 В пост. тока	Двухфазный счетчик
Все HSC	от 15 до 26 В пост. тока	10 кГц
HС4, HС5 только на CPU 224XP и CPU 224XPsi	> 4 В пост. тока	20 кГц
Одновременно включенные входы	Все	100 кГц
Длина кабеля (макс.)	Все	
экранированный	Только CPU 224XP AC/DC/RELAY:	
не экранированный	Все при 55° С с входами пост. тока при 26 пост. тока макс. Все при 50° С с входами пост. тока при 30 В пост. тока макс.	
	500 м для обычных входов, 50 м для входов скоростных счетчиков ¹	
	300 м для обычных входов	

¹ Для входов скоростных счетчиков рекомендуется экранированная витая пара.

Таблица А–7. Технические данные цифровых выходов CPU

Общие данные	Выход 24 В пост. тока (CPU 221, CPU 222, CPU 224, CPU 226)	Выход 24 В пост. тока (CPU 224XP)	Релейный выход
Тип	Канальный полевой униполярный МОП-транзистор ¹ (нормальная полярность)		Слаботочный контакт
Номинальное напряжение	24 В пост. тока	24 В пост. тока	24 В пост. тока или 250 В перем. тока
Диапазон напряжений	от 20,4 до 28,8 В пост. тока	от 5 до 28,8 В пост. тока (Q0.0 – Q0.4) от 20,4 до 28,8 В пост. тока (Q0.5 – Q1.1)	от 5 до 30 В пост. тока или от 5 до 250 В перем. тока
Бросок тока (макс.)	8 А в течение 100 мс		5 А в течение 4 с при относительной длительности 10%
Логика 1 (мин.)	20 В пост. тока при максимальном токе	L+ минус 0,4 В при макс. токе	-
Логика 0 (макс.)	0,1 В пост. тока с нагрузкой 10 кОм		-
Номинальный ток на один выход (макс.)	0,75 А		2,0 А
Номинальный ток на провод (макс.)	6 А	3,75 А	10 А
Ток утечки (макс.)	10 мкА		-
Ламповая нагрузка (макс.)	5 Вт		30 Вт пост. тока; 200 Вт перем. тока ^{2, 4}
Индуктивное напряжение на клеммах	L+ минус 48 В пост. тока, мощность потерь 1 Вт		-
Сопротивление в состоянии «включено» (контакт)	Тип. 0,3 Ом (0,6 Ом макс.)		0,2 Ом (макс., когда новый)
Электрическая развязка	500 В перем. тока в течение 1 минуты		-
Оптическая (гальваническая, полевых устройств с логикой)	-		1500 В перем. тока в течение 1 минуты
Логика с контактом	-		100 МОм
Сопротивление (логика с контактом)	-		
Потенциально развязанные группы	См. схему соединений		См. схему соединений
Задержка (макс.) выкл.-вкл. (мс)	2 мкс (Q0.0, Q0.1), 15 мкс (все остальные)	0,5 мкс (Q0.0, Q0.1), 15 мкс (все остальные)	-
вкл.-выкл. (мс)	10 мкс (Q0.0, Q0.1), 130 мкс (все остальные)	1,5 мкс (Q0.0, Q0.1), 130 мкс (все остальные)	-
Процессы переключения	-		10 мс
Частота следования импульсов (макс.)	20 кГц ² (Q0.0 и Q0.1)	100 кГц ² (Q0.0 и Q0.1)	1 Гц
Срок службы механической части, циклов переключения	-	-	10 000 000 (без нагрузки)
Срок службы контактов	-	-	100 000 (номинальная нагрузка)
Число одновременно включенных выходов	Все при 55° С (горизонтальный монтаж), все при 45° С (вертикальный монтаж)		
Параллельное включение двух выходов	Да, только выходы одной группы		Нет
Длина кабеля (макс.) экранированный не экранированный	500 м 150 м		

¹ Когда механический контакт включает выходное напряжение для CPU S7–200 или цифрового модуля расширения, он посылает сигнал "1" на цифровые выходы в течение примерно 50 микросекунд. Вы должны учитывать это, особенно если вы используете устройства, реагирующие на короткие импульсы.

² В зависимости от вашего приемника импульсов и кабеля дополнительное внешнее нагрузочное сопротивление (при, по крайней мере, 10% номинального тока) может улучшить качество импульсного сигнала и помехоустойчивости.

³ Срок службы реле с ламповой нагрузкой может сократиться на 75%, если не будут предприняты шаги по уменьшению броска тока при включении до величины, не превышающей номинальный бросок тока для выхода.

⁴ Номинальная мощность ламповой нагрузки приведена для номинального напряжения. Уменьшайте номинальную мощность пропорционально переключаемому напряжению (например, 120 В перем. тока - 100 Вт).

Таблица А–8. Технические данные аналогового входа CPU 224XP и CPU 224XPsi

Общие данные	Аналоговый вход (CPU 224XP, CPU 224XPsi)
Число входов	2
Вид аналогового входа	Однотактный вход
Диапазон напряжений	±10 В
Формат слова данных, полный диапазон	от -32 000 до +32 000
Входной импеданс для постоянного тока	>100 КОм
Максимальное входное напряжение	30 В пост. тока
Разрешение	11 битов плюс 1 знаковый бит
Значение LSB	4,88 мВ
Электрическая развязка	нет
Точность	
Худший случай от 0° до 55° С	±2,5% полного диапазона
Тип. 25° С	±1,0% полного диапазона
Повторяемость	±0,05% полного диапазона
Время аналого-цифрового преобразования	125 мс
Тип преобразования	Sigma-Delta
Реакция на ступенчатое воздействие	250 мс макс.
Подавление помех	Тип. -20 дБ при 50 Гц

Таблица А–9. Технические данные аналогового выхода CPU 224XP и CPU 224XPsi

Общие данные	Аналоговый выход (CPU 224XP, CPU 224XPsi)
Число выходов	1
Диапазон сигнала	
Напряжение	от 0 до 10 В (при ограничении мощности)
Ток	от 0 до 20 мА (при ограничении мощности)
Формат слова данных, полный диапазон	от 0 до +32767
Формат слова данных, полный диапазон	от 0 до +32000
Разрешение, полный диапазон	12 битов
Значение LSB	
Напряжение	2,44 мВ
Ток	4,88 мкА
Электрическая развязка	нет
Точность	
Худший случай, от 0° до 55° С	
Вывод напряжения	± 2% полного диапазона
Вывод тока	± 3% полного диапазона
Тип. 25° С	
Вывод напряжения	± 1% полного диапазона
Вывод тока	± 1% полного диапазона
Время установления сигнала	
Вывод напряжения	< 50 мкс
Вывод тока	< 100 мкс
Максимальный вывод	
Вывод напряжения	≥ мин. 5000 Ом
Вывод тока	≤ мин. 500 Ом

Схемы соединений

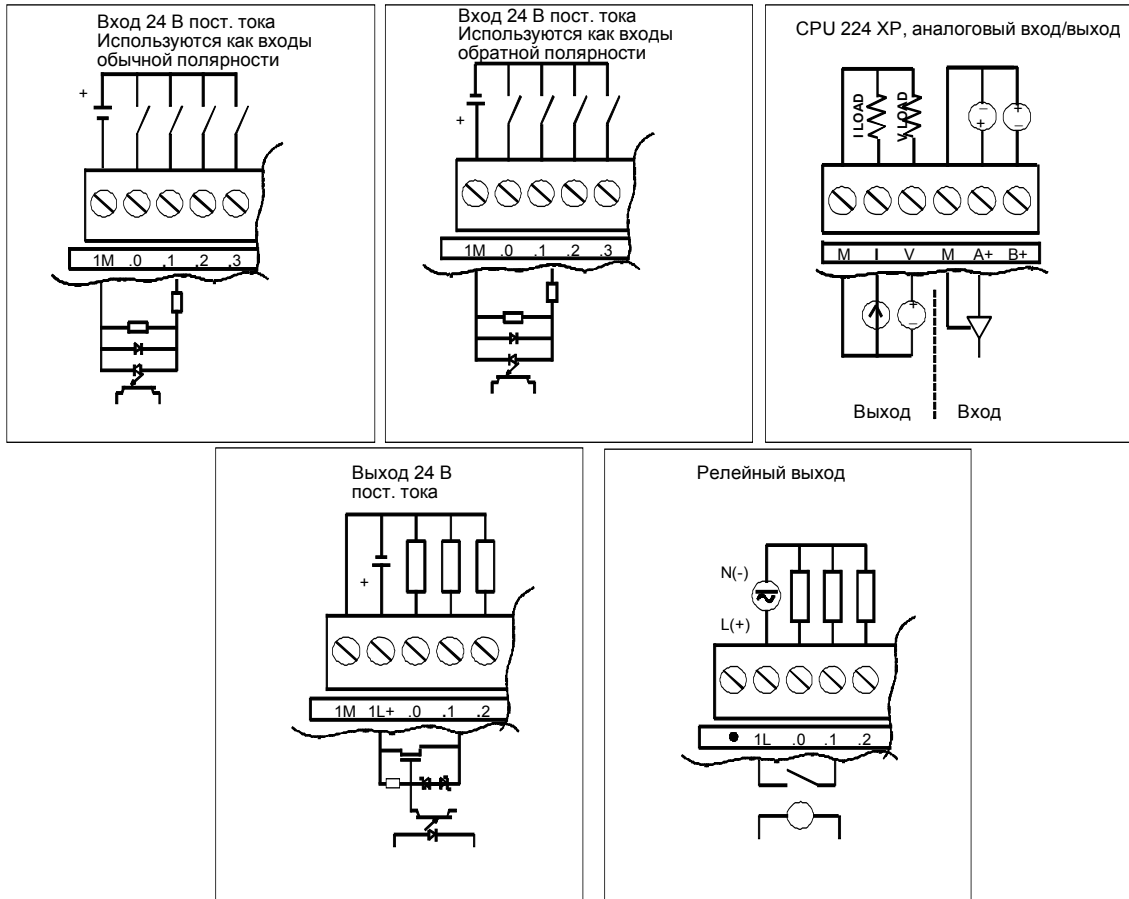


Рис. А-2. Входы и выходы CPU

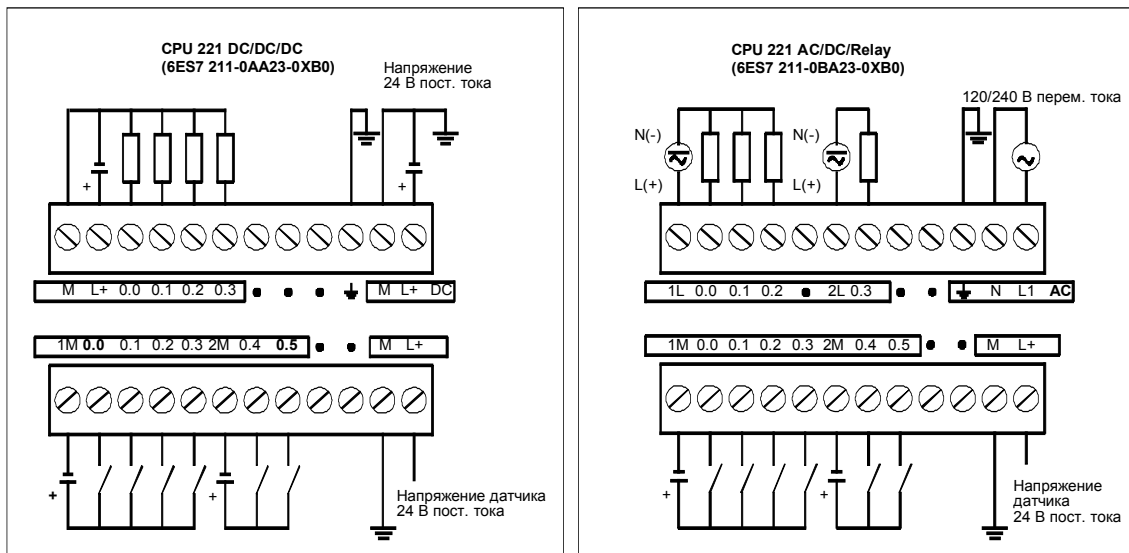


Рис. А-3. Схемы соединений CPU 221

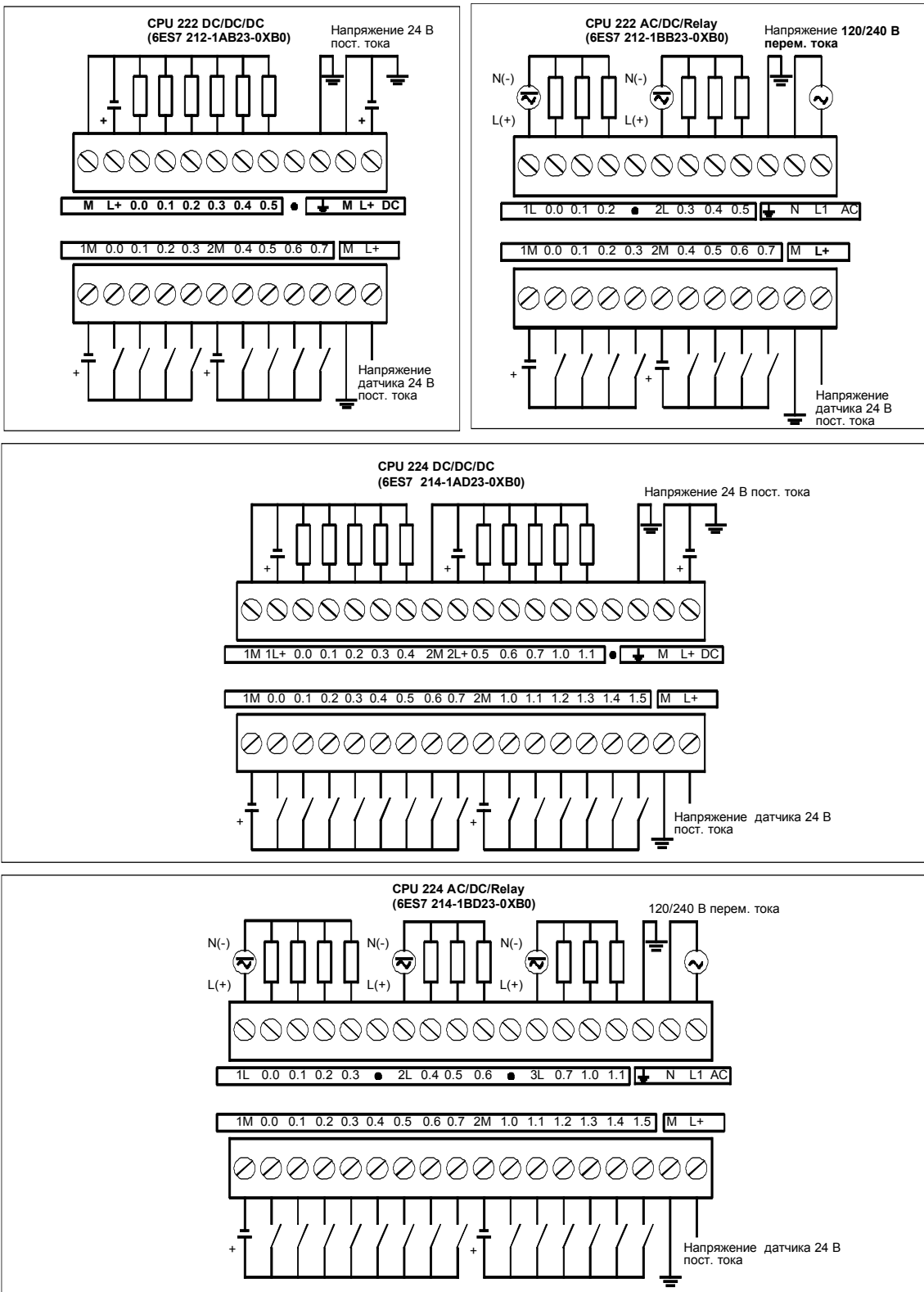


Рис. А-4. Схемы соединений CPU 222 и CPU 224

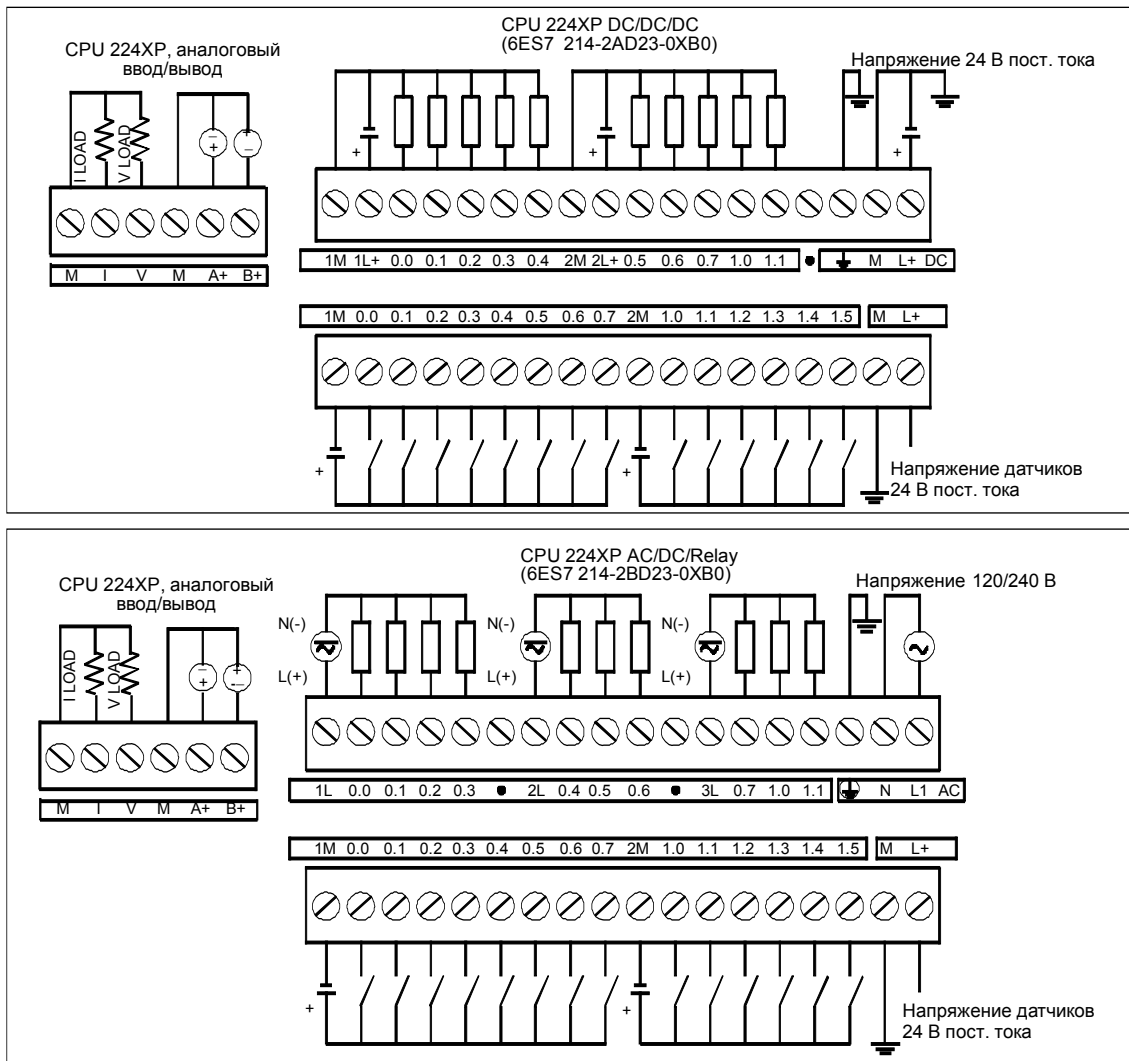


Рис. А-5. Схемы соединений CPU 224XP

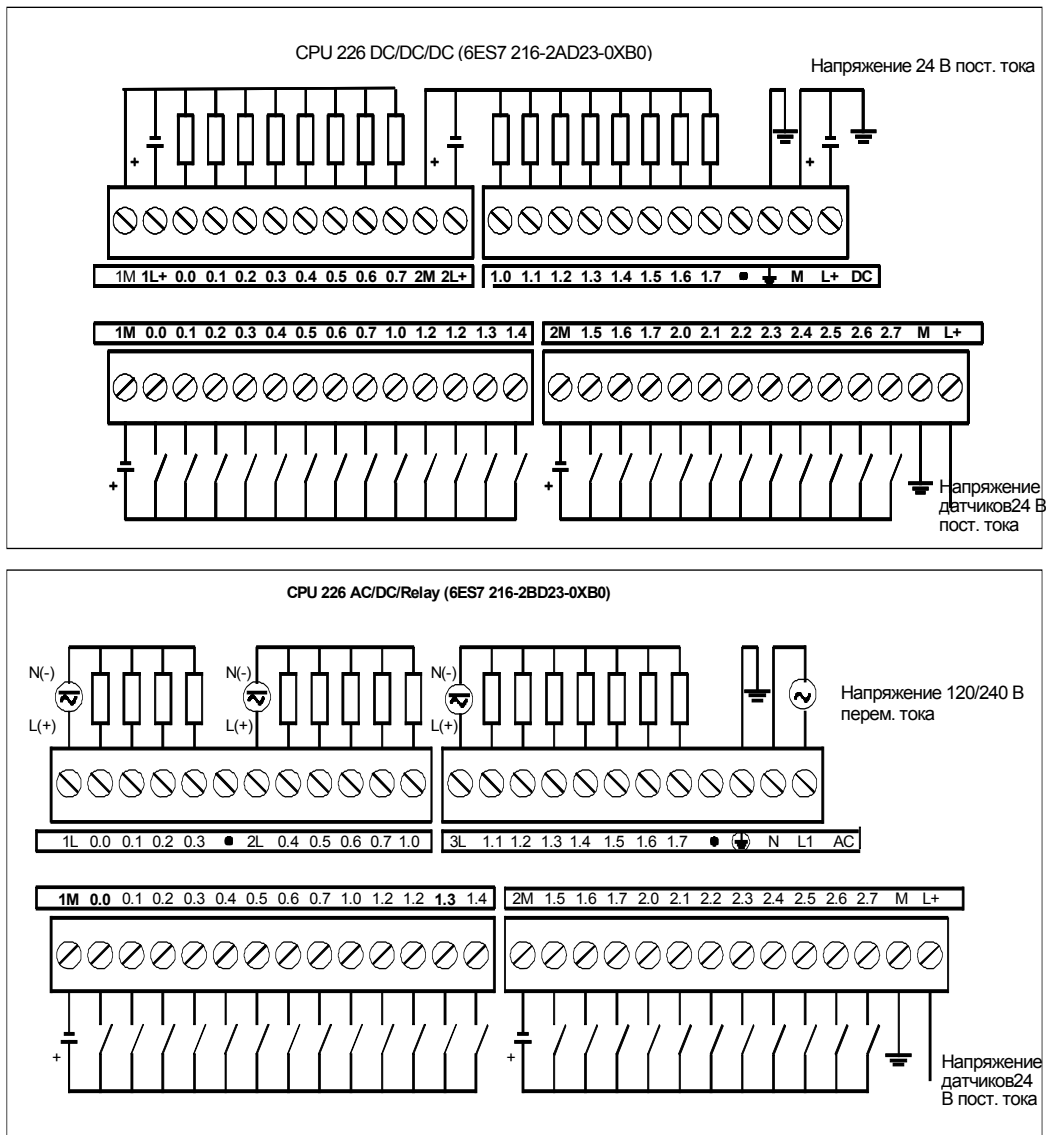


Рис. А–6. Схемы соединений CPU 226

Таблица А–10. Назначение контактов коммуникационного порта S7–200 (при ограничении мощности)

Разъем	Номер контакта	Сигнал PROFIBUS	Порт 0/Порт 1
	1	Экран	Заземление
	2	Обратный провод 24 В	Общий провод логики
	3	RS–485, сигнал В	RS–485, сигнал В
	4	Запрос на передачу	RTS (TTL)
	5	Обратный провод 5 В	Общий провод логики
	6	+5 В	+5 В, последовательное сопротивление 100 Ом
	7	+24 В	+24 В
	8	RS–485, сигнал А	RS–485, сигнал А
	9	Не используется	10-битовый протокол (вход)
	Корпус разъема	Экран	Заземление

Технические данные цифровых модулей расширения

Таблица А–11. Номера для заказа цифровых модулей расширения

Номер для заказа	Модуль расширения	Цифровые входы	Цифровые выходы	Съемный клеммный блок
6ES7 221-1BF22-0XA0	EM 221, цифровой ввод, 8 x 24 В пост. тока	8 x 24 В пост. тока	-	Да
6ES7 221-1EF22-0XA0	EM 221, цифровой ввод, 8 x 120/230 В перем. тока	8 x 120/230 В перем. тока	-	Да
6ES7 221-1BH22-0XA0	EM 221, цифровой ввод, 16 x 24 В пост. тока	16 x 24 В пост. тока	-	Да
6ES7 222-1BD22-0XA0	EM 222, цифровой вывод, 4 x 24 В пост. тока-5А	-	4 x 24 В пост. тока-5А	Да
6ES7 222-1HD22-0XA0	EM 222, цифровой вывод, 4 x реле-10А	-	4 x реле-10А	Да
6ES7 222-1BF22-0XA0	EM 222, цифровой вывод, 8 x 24 В пост. тока	-	8 x 24 В пост. тока-0,75А	Да
6ES7 222-1HF22-0XA0	EM 222, цифровой вывод, 8 x реле	-	8 x реле-10А	Да
6ES7 222-1EF22-0XA0	EM 222, цифровой вывод, 8 x 120/230 VAC	-	8 x 120/230 VAC	Да
6ES7 223-1BF22-0XA0	EM 223, 24 В пост. тока, цифровые входы и выходы, 4 входа/4 выхода	4 x 24 В пост. тока	4 x 24 В пост. тока-0,75А	Да
6ES7 223-1HF22-0XA0	EM 223, 24 В пост. тока, цифровые входы и выходы, 4 входа/4 релейных выходов	4 x 24 В пост. тока	4 x реле -2А	Да
6ES7 223-1BH22-0XA0	EM 223, 24 В пост. тока, цифровые входы и выходы, 8 входов/8 выходов	8 x 24 В пост. тока	8 x 24 В пост. тока-0,75А	Да
6ES7 223-1PH22-0XA0	EM 223 24 В пост. тока цифровые входы и выходы, 8 входов/8 релейных выходов	8 x 24 В пост. тока	8 x реле -2А	Да
6ES7 223-1BL22-0XA0	EM 223 24 В пост. тока цифровые входы и выходы, 16 входов/16 выходов	16 x 24 В пост. тока	16 x 24 В пост. тока-0,75А	Да
6ES7 223-1PL22-0XA0	EM 223 24 В пост. тока цифровые входы и выходы, 16 входов/16 релейных выходов	16 x 24 В пост. тока	16 x реле -2А	Да

Таблица А–12. Общие технические данные цифровых модулей расширения

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потеря	Потребность в напряжении пост. тока	
					+5 В пост. тока	+24 В пост. тока
6ES7 221-1BF22-0XA0	EM 221 DI 8 x 24 VDC	46 x 80 x 62	150 г	2 Вт	30 мА	ВКЛ:4 мА/вход
6ES7 221-1EF22-0XA0	EM 221 DI 8 x 120/230 VAC	71,2 x 80 x 62	160 г	3 Вт	30 мА	-
6ES7 221-1BH22-0XA0	EM 221 DI 16 x 24 VDC	71,2 x 80 x 62	160 г	3 Вт	70 мА	ВКЛ:4 мА/вход
6ES7 222-1BD22-0XA0	EM 222 DO 4 x 24 VDC-5А	46 x 80 x 62	120 г	3 Вт	40 мА	-
6ES7 222-1HD22-0XA0	EM 222 DO 4 x Relay-10А	46 x 80 x 62	150 г	4 Вт	30 мА	ВКЛ:20 мА/выход
6ES7 222-1BF22-0XA0	EM 222 DO 8 x 24 VDC	46 x 80 x 62	150 г	2 Вт	50 мА	-
6ES7 222-1HF22-0XA0	EM 222 DO 8 x Relay	46 x 80 x 62	170 г	2 Вт	40 мА	ВКЛ:9 мА/выход
6ES7 222-1EF22-0XA0	EM 222 DO 8 x 120/230 VAC	71,2 x 80 x 62	165 г	4 Вт	110 мА	-
6ES7 223-1BF22-0XA0	EM 223 24 VDC 4 In/4 Out	46 x 80 x 62	160 г	2 Вт	40 мА	ВКЛ:4 мА/вход
6ES7 223-1HF22-0XA0	EM 223 24 VDC 4 In/4 Relay	46 x 80 x 62	170 г	2 Вт	40 мА	ВКЛ:9 мА/выход, 4 мА/вход
6ES7 223-1BH22-0XA0	EM 223 24 VDC 8 In/8 Out	71,2 x 80 x 62	200 г	3 Вт	80 мА	-
6ES7 223-1PH22-0XA0	EM 223 24 VDC 8 In/8 Relay	71,2 x 80 x 62	300 г	3 Вт	80 мА	ВКЛ:9 мА/выход, 4 мА/вход
6ES7 223-1BL22-0XA0	EM 223 24 VDC 16 In/16 Out	137,3 x 80 x 62	360 г	6 Вт	160 мА	-
6ES7 223-1PL22-0XA0	EM 223 24 VDC 16 In/16 Relay	137,3 x 80 x 62	400 г	6 Вт	150 мА	ВКЛ:9 мА/выход, 4 мА/вход

Таблица А–13. Технические данные входов цифровых модулей расширения		
Общие данные	Вход 24 В пост. тока	Вход 120/230 В перем. тока (от 47 до 63 Гц)
Тип	Обычная полярность/обратная полярность (IEC тип 1 – обычная полярность)	IEC Тип I
Номинальное напряжение	24 В пост. тока при 4 мА	120 В перем. тока при 6 мА или 230 В перем. тока при 9 мА (расчет)
Максимально допустимое длительное напряжение	30 В пост. тока	264 В перем. тока
Бросок напряжения (макс.)	35 В пост. тока в течение 0,5 с	-
Логика 1 (мин.)	15 В пост. тока при 2,5 мА	79 В перем. тока при 2,5 мА
Логика 0 (макс.)	5 В пост. тока при 1 мА	20 В перем. тока или 1 мА перем. тока
Входная задержка (макс.)	4,5 мс	15 мс
Присоединение 2-проводного датчика близости (Вего) Допустимый ток утечки (макс.)	1 мА	1 мА перем. тока
Электрическая развязка Оптическая (гальваническая, полевых устройств с логикой) Потенциально развязанные группы	500 В перем. тока в течение 1 минуты См. схему соединений	1500 В перем. тока в течение 1 минуты 1 вход
Одновременно включенные входы	Все при 55° С (горизонтальный монтаж), все при 45° С (вертикальный монтаж)	
Длина кабеля (макс.) экранированный не экранированный	500 м 300 м	500 м 300 м

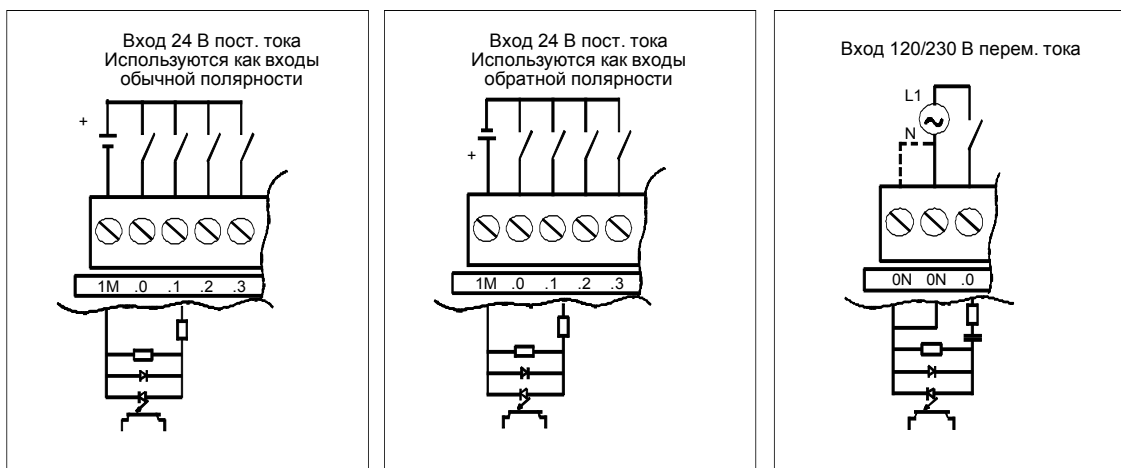


Рис. А–7. Входы цифровых модулей расширения S7–200

Таблица А–14. Технические данные выходов цифровых модулей расширения

Общие данные	Выход 24 В пост. тока		Релейный выход		Выход 120/230 В перем. тока
	0,75 А	5 А	2 А	10 А	
Тип	Канальный полевой униполярный МОП-транзистор ¹ (обратная полярность)		Слаботочный контакт		Триак, включение при переходе через ноль ²
Номинальное напряжение	24 В пост. тока		24 В пост. тока или 250 В перем. тока		120/230 В перем. тока
Диапазон напряжений	от 20,4 до 28,8 В пост. тока		от 5 до 30 В пост. тока или от 5 до 250 В перем. тока	от 12 до 30 В пост. тока или от 12 до 250 В перем. тока	от 4 до 264 В перем. тока (от 47 до 63 Гц)
Диапазон напряжений для катушки (выхода) 24 В пост. тока	-		от 20,4 до 28,8 В пост. тока		-
Бросок тока (макс.)	8 А в течение 100 мс	30 А	5 А в течение 4 с при относительной длительности 10%	15 А в течение 4 с при относительной длительности 10%	5 А эфф. в течение 2 циклов перем. тока
Логика 1 (мин.)	20 В пост. тока		-		L1 (-0,9 В эфф.)
Логика 0 (макс.)	0,1 В пост. тока при нагрузке 10 КОм	0,2 В пост. тока при нагрузке 5КОм	-		-
Номинальный ток на один выход (макс.)	0,75 А	5 А	2,00 А	Омическая нагрузка 10 А; Индуктивная нагрузка пост. тока 2 А; Индуктивная нагрузка перем. тока 3 А	0,5 А перем. тока
Номинальный ток на провод (макс.)	6 А	5 А	8 А	10 А	0,5 А перем. тока
Ток утечки (макс.)	10 мкА	30 мкА	-		1,1 мА _{эфф} при 132 В перем. тока и 1,8 мА _{эфф} при 264 В перем. тока
Ламповая нагрузка (макс.)	5 Вт	50 Вт	30 Вт пост. тока / 200 Вт перем. тока ^{6,7}	100 Вт пост. тока / 1000 Вт перем. тока	60 Вт
Индуктивное напряжение на клеммах	L+ минус 48 В	L+ минус 47 В ⁴	-		-
Сопротивление в состоянии "включено" (контакт)	Тип. 0,3 Ом (макс. 0,6 Ом)	макс. 0,05 Ом	макс. 0,2 Ом, когда новый	макс. 0,1 Ом, когда новый	макс. 410 Ом, когда ток нагрузки меньше 0,05 А
Электрическая развязка Оптическая (гальваническая, полевых устройств с логикой) катушка с логикой катушка с контактом сопротивление (катушка с контактом) Потенциально развязанные группы	500 В перем. тока в течение 1 минуты - - См. схему соединений		- нет 1500 В перем. тока в течение 1 минуты мин. 100 МОм когда новый См. схему соединений		1500 В перем. тока в течение 1 минуты - - 1 выход
Задержка выкл.-вкл./вкл.-выкл. (макс.) Процессы переключения (макс.)	50 мкс / 200 мкс -	500 мкс -	- 10 мс	- 15 мс	0,2 мс + 1/2 цикла АС -
Частота переключения (макс.)	-		1 Гц		10 Гц
Срок службы механической части, циклов переключения	-		10 000 000 (без нагрузки)	30 000 000 (без нагрузки)	-
Срок службы контактов	-		100 000 (номинальная нагрузка)	30 000 (номинальная нагрузка)	-
Количество одновременно включенных выходов	Все при 55° С (горизонтальный монтаж), все при 45° С (вертикальный монтаж)		Все при 55° С (горизонтальный монтаж) при макс. токе модуля 20А Все при 45° С (вертикальный монтаж) макс. токе модуля 20А ⁸ Все при 40° С (горизонтальный монтаж) при 10А на выход		Все при 55° С (горизонтальный монтаж), все при 45° С (вертикальный монтаж)
Параллельное включение двух выходов	Да, только выходы одной группы		Нет		Нет
Длина кабеля (макс.) экранированный не экранированный	500 м 150 м		500 м 150 м		500 м 150 м

1 Когда механический контакт включает выходное напряжение для CPU S7-200 или цифрового модуля расширения, он посылает сигнал "1" на цифровые выходы в течение примерно 50 микросекунд. Вы должны учитывать это, особенно если вы используете устройства, реагирующие на короткие импульсы.

2 Когда механический контакт включает выходное напряжение для модуля расширения переменного тока, он посылает сигнал "1" на выходы переменного тока в течение примерно 1/2 цикла переменного тока. Вы должны учитывать это.

3 Из-за схемы, обеспечивающей включение при переходе через ноль, ток нагрузки должен быть полнопериодным, а не однополупериодным переменным током. Минимальный ток нагрузки равен 0,05 А перем. тока. При токе нагрузки от 5 до 50 мА им можно управлять, но имеет место дополнительное падение напряжения из-за последовательно включенного сопротивления 410 Ом.

4 Если выход перегревается из-за чрезмерно частых переключений индуктивной нагрузки или ненормальных условий работы, то выход может выключиться или получить повреждение. Выход может перегреться или быть поврежден, если при выключении индуктивной нагрузки он подвергается воздействию энергии более 0,7 Джоуля. Для устранения этого ограничения параллельно выходу можно включить гасящую цепочку, описанную в главе 3. Эти компоненты должны быть

- заранее надлежащим образом рассчитаны для данного приложения.
- 5 EM 222 DO 4 x Relay имеет другую оценку FM, чем другие модули S7-200. Этот модуль имеет оценку T4, а не T4A для FM-класса I, раздел 2, группы опасных помещений A, B, C и D.
 - 6 Срок службы реле с ламповой нагрузкой может сократиться на 75%, если не будут предприняты шаги по уменьшению броска тока при включении до величины, не превышающей номинальный бросок тока для выхода.
 - 7 Номинальная мощность ламповой нагрузки приведена для номинального напряжения. Уменьшайте номинальную мощность пропорционально переключаемому напряжению (например, 120 В перем. тока - 100 Вт).

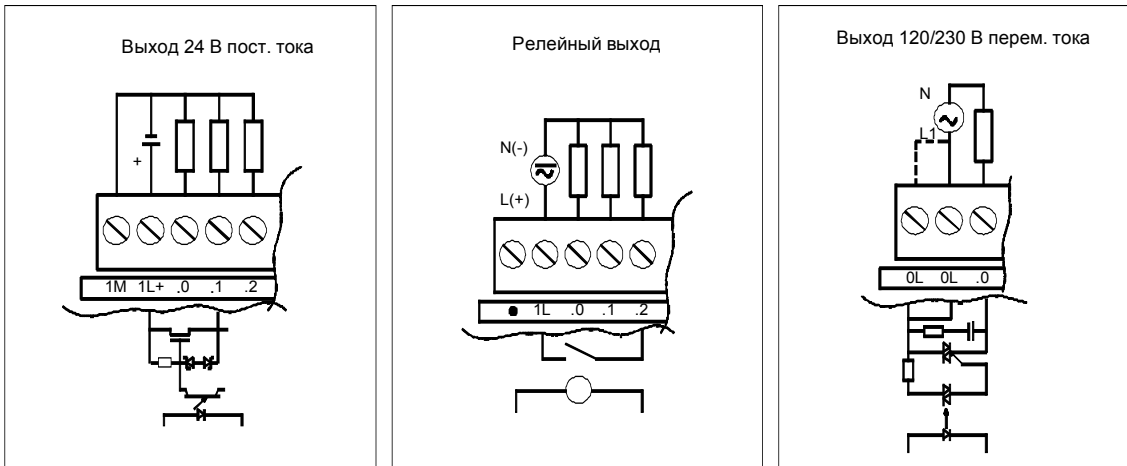


Рис. А-8. Выходы цифровых модулей расширения S7-200

Схемы соединений

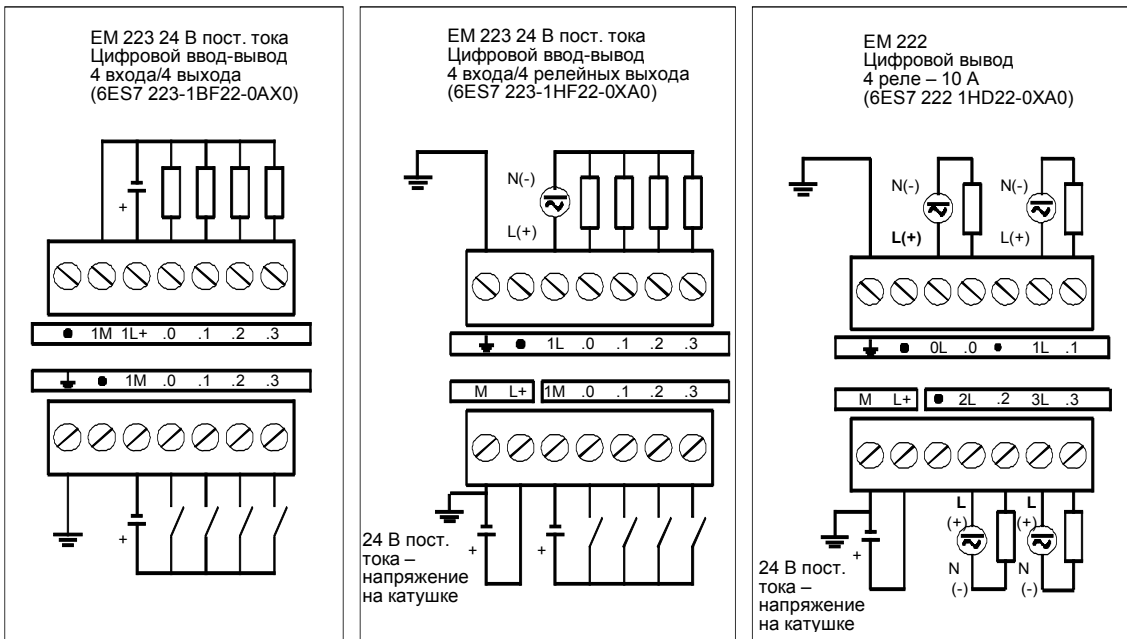


Рис. А-9. Схемы соединений для модулей расширения EM222 и EM223

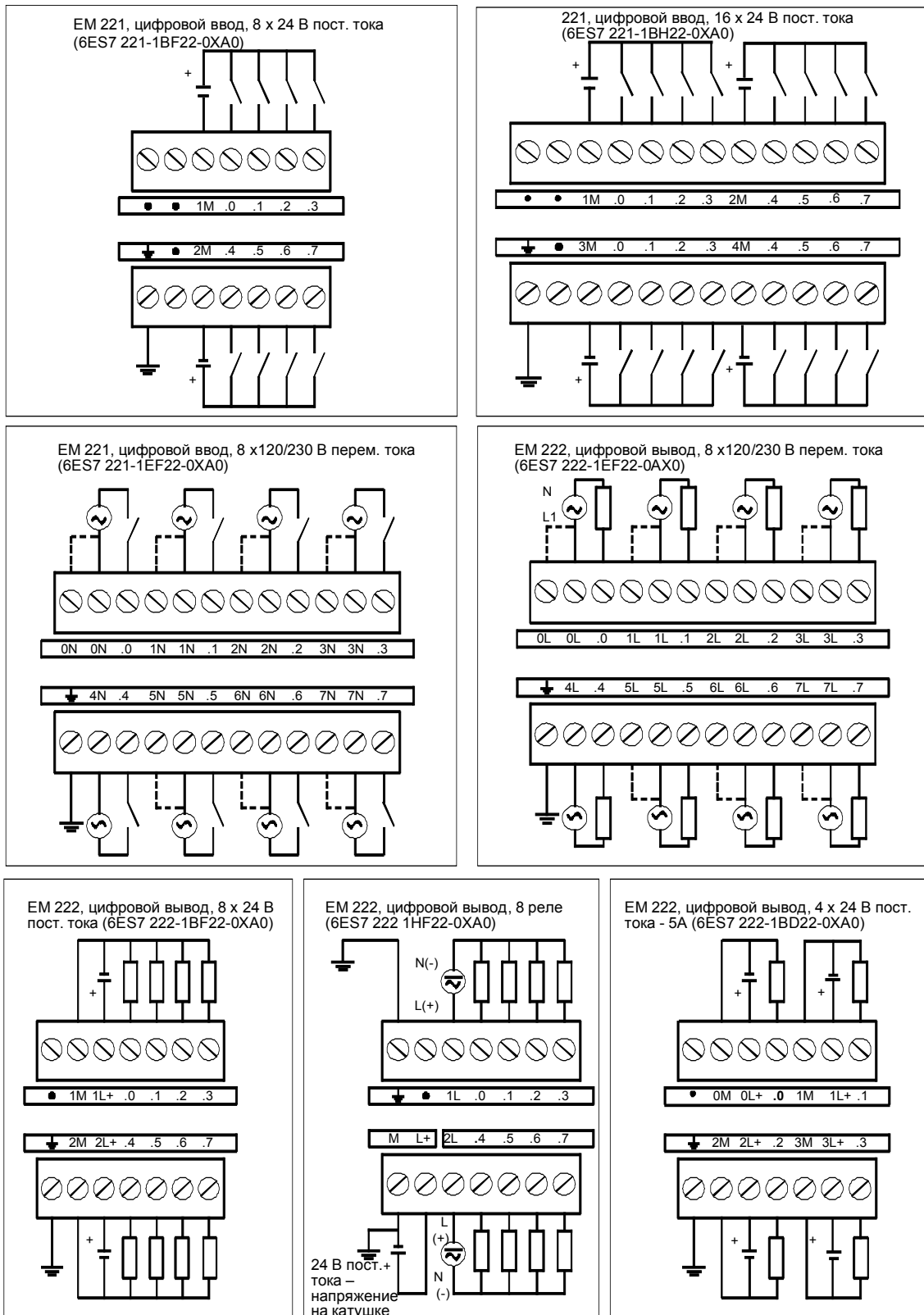


Рис. А-10. Схемы соединений для модулей расширения EM 221 и EM 222

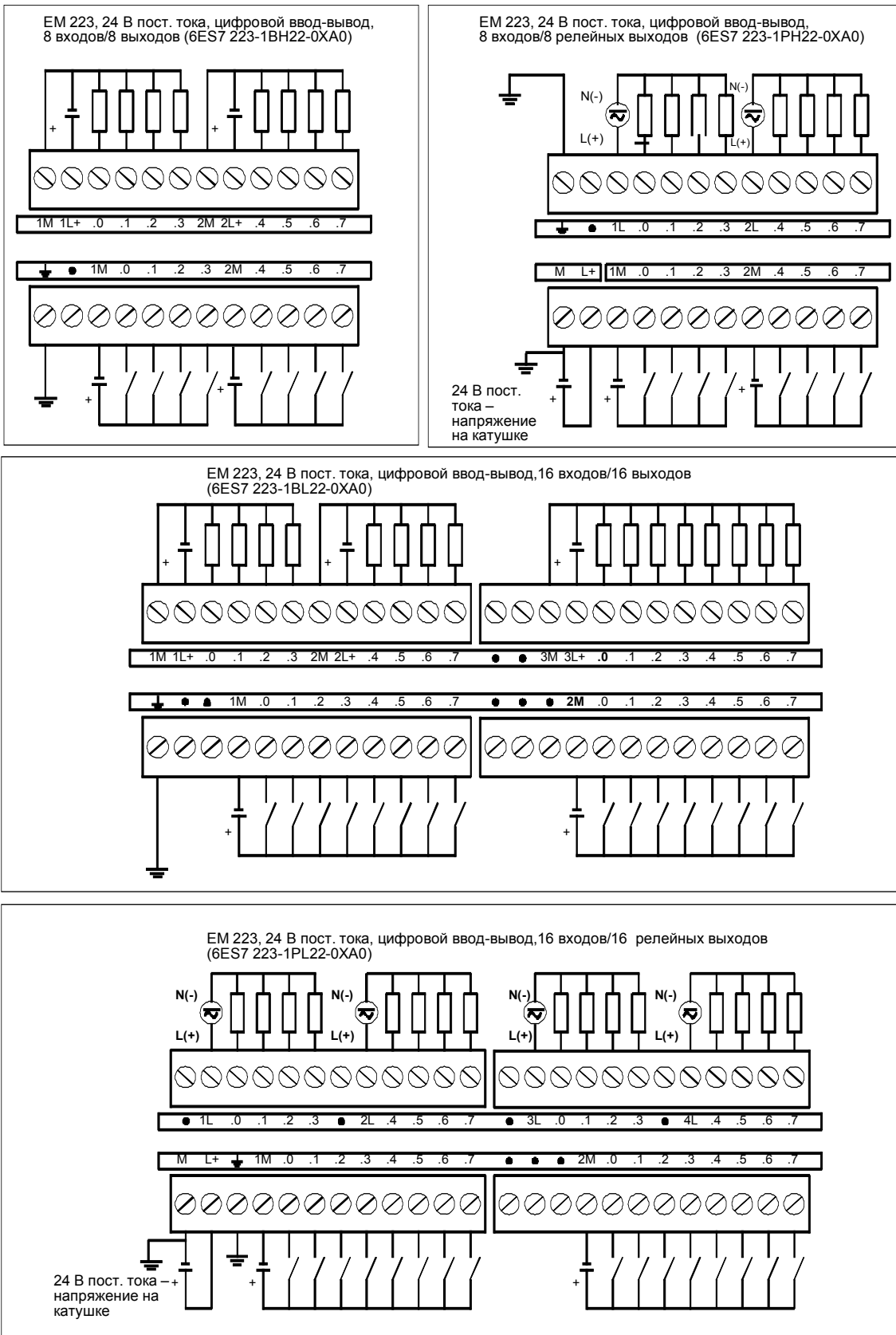


Рис. А–11. Схемы соединений для модулей расширения EM 223

Технические данные аналоговых модулей расширения

Таблица А–15. Номера для заказа аналоговых модулей расширения

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6ES7 231-0HC22-0XA0	EM 231, аналоговый ввод, 4 входа	4	-	Нет
6ES7 231-0HF22-0XA0	EM 231, аналоговый ввод, 8 входов	8	-	Нет
6ES7 232-0HB22-0XA0	EM 232, аналоговый вывод, 2 выхода	-	2	Нет
6ES7 232-0HD22-0XA0	EM 232, аналоговый вывод, 4 выхода	-	4	Нет
6ES7 235-0KD22-0XA0	EM 235, аналоговый ввод/вывод, 4 входа/1 выход	4	1 ¹	Нет

1 CPU резервирует 2 аналоговых выхода для этого модуля.

Таблица А–16. Общие технические данные аналоговых модулей расширения

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В	+24 В
6ES7 231-0HC22-0XA0	EM 231, аналоговый ввод, 4 входа	71,2 x 80 x 62	183 г	2 Вт	20 мА	60 мА
6ES7 231-0HF22-0XA0	EM 231, аналоговый ввод, 8 входов	71,2 x 80 x 62	190 г	2 Вт	20 мА	60 мА
6ES7 232-0HB22-0XA0	EM 232, аналоговый вывод, 2 выхода	46 x 80 x 62	148 г	2 Вт	20 мА	70 мА (с двумя выходами при 20 мА)
6ES7 232-0HD22-0XA0	EM 232, аналоговый вывод, 4 выхода	71,2 x 80 x 62	190 г	2 Вт	20 мА	100 мА (с двумя выходами при 20 мА)
6ES7 235-0KD22-0XA0	EM 235, аналоговый ввод/вывод, 4 входа/1 выход	71,2 x 80 x 62	186 г	2 Вт	30 мА	60 мА (с выходом при 20 мА)

Таблица А–17. Технические данные входов аналоговых модулей расширения

Общие данные	6ES7 231-0HC22-0XA0 6ES7 235-0KD22-0XA0	6ES7 231-0HF22-0XA0
	Формат слова данных Биполярный, полный диапазон Униполярный, полный диапазон	(См. рис. А–14) от -32000 до +32000 от 0 до 32000
Входной импеданс для постоянного тока	≥2 МОм, потенциальный вход 250 Ом, токовый вход	> 2 МОм потенциальный вход 250 Ом, токовый вход
Затухание входного фильтра	-3 дБ при 3,1 кГц	-3 дБ при 3,1 кГц
Максимальное входное напряжение	30 В пост. тока	30 В пост. тока
Максимальный входной ток	32 мА	32 мА
Разрешение Биполярный диапазон Униполярный диапазон	11 битов плюс 1 знаковый бит 12 битов	
Электрическая развязка (полевых устройств с логикой)	нет	нет
Тип входа	Дифференциальный	Дифференциальный по напряжению, 2 канала настраиваются на токовый сигнал
Диапазоны входных сигналов Напряжение Ток	Настраиваемый, доступные диапазоны см. в таблице А–20 от 0 до 20 мА	Каналы от 0 до 7: 0-10В, 0-5В, ±2,5В Каналы 6 и 7 от 0 до 20 мА
Разрешающая способность входа	см. таблицу А–20	см. таблицу А–21
Время аналого-цифрового преобразования	< 250 мкс	< 250 мкс
Переходная характеристика аналогового входа (реакция на единичный скачок)	от 1,5 мс до 95%	от 1,5 мс до 95%
Подавление синфазной помехи	40 дБ, от постоянного тока до 60 Гц	40 дБ, от постоянного тока до 60 Гц
Напряжение синфазной помехи	Напряжение сигнала плюс напряжение синфазной помехи должно быть ≤ ±12 В	Напряжение сигнала плюс напряжение синфазной помехи должно быть ≤ ±12 В
Диапазон питающего напряжения 24 В пост. тока	от 20,4 до 28,8 В пост. тока (класс 2, при ограничении мощности, или напряжение питания датчиков от ПЛК)	

Таблица А–18. Технические данные выходов аналоговых модулей расширения

Общие данные	6ES7 232-0NB22-0XA0 6ES7 232-0ND22-0XA0 6ES7 235-0KD22-0XA0
Электрическая развязка (полевых устройств с логикой)	нет
Диапазон сигнала Вывод напряжения Вывод тока	± 10 В от 0 до 20 мА
Разрешение, полное Напряжение Ток	12 битов плюс знаковый бит 11 битов
Формат слова данных Напряжение Ток	-3200 от 0 до +32000 от 0 до +32000
Точность Худший случай, от 0° до 55° С Вывод напряжения Вывод тока Тип., 25° С Потенциальный выход Токовый выход	$\pm 2\%$ полного диапазона $\pm 2\%$ полного диапазона $\pm 0,5\%$ полного диапазона $\pm 0,5\%$ полного диапазона
Время установления Потенциальный выход Токовый выход	100 мкс 2 мс
Сопротивление Потенциальный выход Токовый выход	Минимальное 5000 Ом Максимальное 500 Ом
Диапазон питающего напряжения 24 В пост. тока	от 20,4 до 28,8 В пост. тока (класс 2, при ограничении мощности, или напряжение питания датчиков от ПЛК)

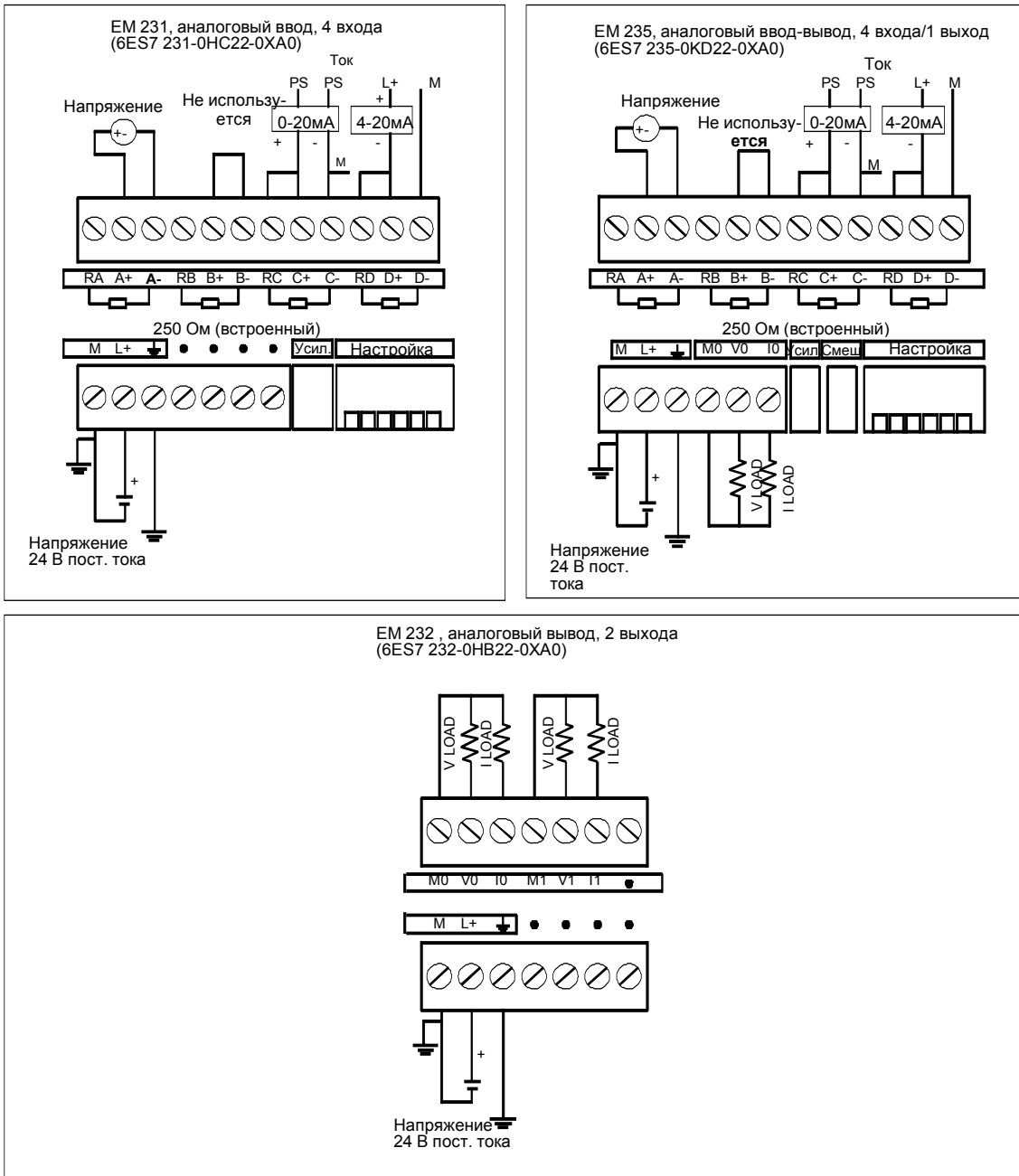


Рис. А-12. Схемы соединений для аналоговых модулей расширения

Светодиодные индикаторы аналоговых модулей

Светодиодные индикаторы аналоговых модулей представлены в таблице А–19.

Таблица А–19. Светодиодные индикаторы аналоговых модулей

Светодиодный индикатор	Включен	Выключен
Имеется питающее напряжение 24 В пост. тока	Нет неисправностей	Нет напряжения 24 В пост. тока



Совет

Информация о состоянии напряжения у пользователя содержится также в битах специальной памяти (SM). Дополнительную информацию вы найдете в Приложении D, SMB8 – SMB21, регистры идентификатора и ошибок модуля ввода-вывода.

Калибровка входов

Калибровочные потенциометры воздействуют на каскад измерительного усилителя, который следует за аналоговым мультиплексором (см. схему соединений для входов EM 231 на рис. А–15 и EM 235 на рис. А–16). Поэтому калибровка влияет на все входные каналы пользователя. Разброс значений компонентов входных цепей, предшествующих аналоговому мультиплексору, вызывает небольшие различия в отсчетах каналов, связанных с одним и тем же входным сигналом, даже после калибровки.

Чтобы удовлетворить техническим требованиям, содержащимся в этих спецификациях, необходимо разблокировать фильтры всех аналоговых входов модуля. Для вычисления среднего значения выберите не менее 64 отсчетов.

Для калибровки входа действуйте следующим образом.

1. Выключите питание модуля. Выберите желаемый входной диапазон.
2. Включите питание CPU и модуля. Предоставьте модулю возможность стабилизироваться в течение 15 минут.
3. Используя измерительный преобразователь, источник напряжения или источник тока, подайте нулевой сигнал на одну из входных клемм.
4. Считайте значение, сообщаемое CPU соответствующим входным каналом.
5. Регулируйте потенциометр OFFSET [смещение], пока не станет считываться нуль или желаемое цифровое значение данных.
6. Включите максимальный сигнал на одну из входных клемм. Считайте значение, передаваемое в CPU.
7. Регулируйте потенциометр GAIN [усиление], пока не станет считываться значение 32000 или желаемое цифровое значение данных.
8. Повторяйте калибровку OFFSET [смещение] и GAIN [смещение] по мере необходимости.

Местоположение органов калибровки и конфигурирования EM 231 и EM 235

Как показано на рис. А–13, калибровочный потенциометр и конфигурационные двухпозиционные переключатели расположены в правой нижней части клеммного блока модуля.

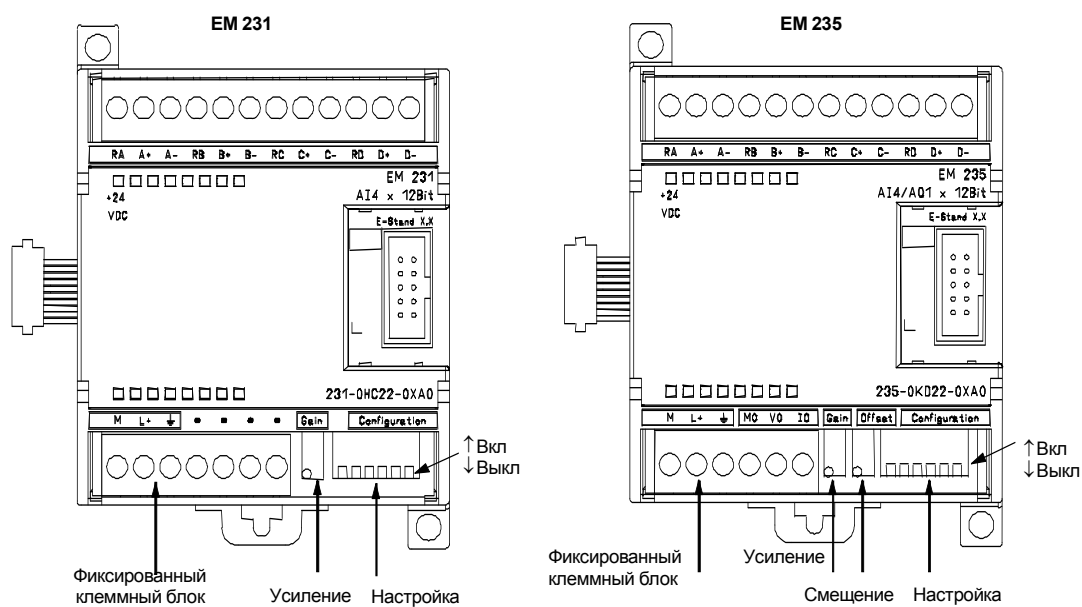


Рис. А–13. Расположение калибровочного потенциометра и конфигурационных двухпозиционных переключателей у EM 231 и EM 235

Настройка EM 231

Таблица А–20 показывает, как настраивать модуль EM 231, используя настроечные двухпозиционные переключатели. Переключатели 1, 2 и 3 выбирают диапазон аналогового входного сигнала. Все входы настраиваются на один и тот же диапазон аналогового входного сигнала. В этой таблице ON означает “замкнуто”, а OFF означает “разомкнуто”. Положения переключателей считываются только при включенном напряжении.

Таблица А–20. Таблица положений настроечных переключателей EM 231 с 4 входами при выборе диапазона аналогового входного сигнала

Униполярный			Полный диапазон входного сигнала	Разрешение
SW1	SW2	SW3		
ON	OFF	ON	от 0 до 10 В	2,5 мВ
	ON	OFF	от 0 до 5 В	1,25 мВ
			от 0 до 20 мА	5 мкА
Биполярный			Полный диапазон входного сигнала	Разрешение
SW1	SW2	SW3		
OFF	OFF	ON	±5 В	2,5 мВ
	ON	OFF	± 2,5 В	1,25 мВ

Для модулей EM 231 с 8 входами, переключатели 3,4,5 выбирают диапазон. Переключатели 1 и 2 переключают 6 и 7 каналы на измерение тока.

Таблица А–21. Таблица положений настроечных переключателей EM 231 с 8 входами при выборе диапазона аналогового входного сигнала

Униполярный			Полный диапазон входного сигнала	Разрешение
SW3	SW4	SW5		
ON	OFF	ON	от 0 до 10 В	2,5 мВ
	ON	OFF	от 0 до 5 В	1,25 мВ
			от 0 до 20 мА	5 мкА
Биполярный			Полный диапазон входного сигнала	Разрешение
SW3	SW4	SW5		
OFF	OFF	ON	±5 В	2,5 мВ
	ON	OFF	± 2,5 В	1,25 мВ

Настройка EM 235

Таблица А–21 показывает, как настроить модуль EM 235, используя настроечные двухпозиционные переключатели. Переключатели 1 – 6 выбирают диапазон аналогового входного сигнала и разрешающую способность. Все входы настраиваются на один и тот же диапазон аналогового входного сигнала и формат. Таблица А–21 показывает, как выбрать тип сигнала “униполярный/биполярный” (переключатель 6), усиление (переключатели 4 и 5) и ослабление (переключатели 1, 2 и 3). В этих таблицах ON означает “замкнуто”, а OFF означает “разомкнуто”.

Таблица А–22. Таблица положений настроечных переключателей EM 235 при выборе диапазона аналогового входного сигнала и разрешающей способности

Униполярный						Полный диапазон входного сигнала	Разрешение
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	от 0 до 50 мВ	12,5 мкВ
OFF	ON	OFF	ON	OFF	ON	от 0 до 100 мВ	25 мкВ
ON	OFF	OFF	OFF	ON	ON	от 0 до 500 мВ	125 мкВ
OFF	ON	OFF	OFF	ON	ON	от 0 до 1 В	250 мкВ
ON	OFF	OFF	OFF	OFF	ON	от 0 до 5 В	1,25 мВ
ON	OFF	OFF	OFF	OFF	ON	от 0 до 20 мА	5 мкА
OFF	ON	OFF	OFF	OFF	ON	от 0 до 10 В	2,5 мВ
Биполярный						Полный диапазон входного сигнала	Разрешение
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	+25 мВ	12,5 мкВ
OFF	ON	OFF	ON	OFF	OFF	+50 мВ	25 мкВ
OFF	OFF	ON	ON	OFF	OFF	+100 мВ	50 мкВ
ON	OFF	OFF	OFF	ON	OFF	+250 мВ	125 мкВ
OFF	ON	OFF	OFF	ON	OFF	+500 мВ	250 мВ
OFF	OFF	ON	OFF	ON	OFF	+1 В	500 мкВ
ON	OFF	OFF	OFF	OFF	OFF	+2,5 В	1,25 мВ
OFF	ON	OFF	OFF	OFF	OFF	+5 В	2,5 мВ
OFF	OFF	ON	OFF	OFF	OFF	+10 В	5 мВ

Формат слова данных для входов EM 231 и EM 235

На рис. А–14 показано размещение 12-разрядного значения данных в слове аналоговых входов CPU.

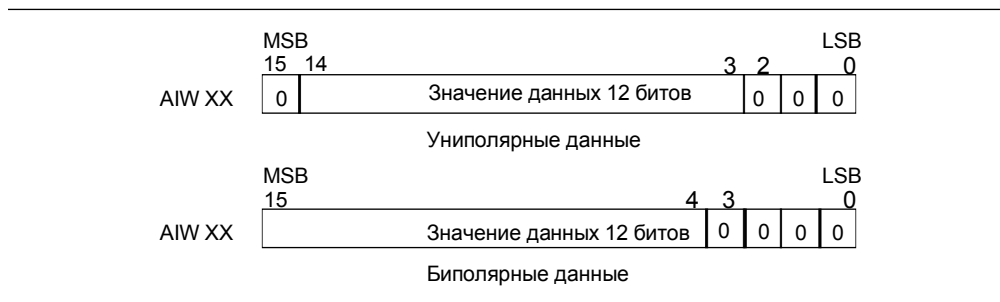


Рис. А–14. Формат слова данных для входов EM 231 и EM 235



Совет

12 битов данных аналого-цифрового преобразователя (АЦП) выравниваются в формате слова данных по левому краю. MSB – это знаковый бит: ноль указывает на положительное значение слова данных.

В униполярном формате три конечных нуля заставляют слово данных изменяться на восемь единиц при каждом изменении значения АЦП на одну единицу.

В биполярном формате четыре конечных нуля заставляют слово данных изменяться на шестнадцать единиц при каждом изменении значения АЦП на одну единицу.

Схема соединений для входов EM 231 и EM 235

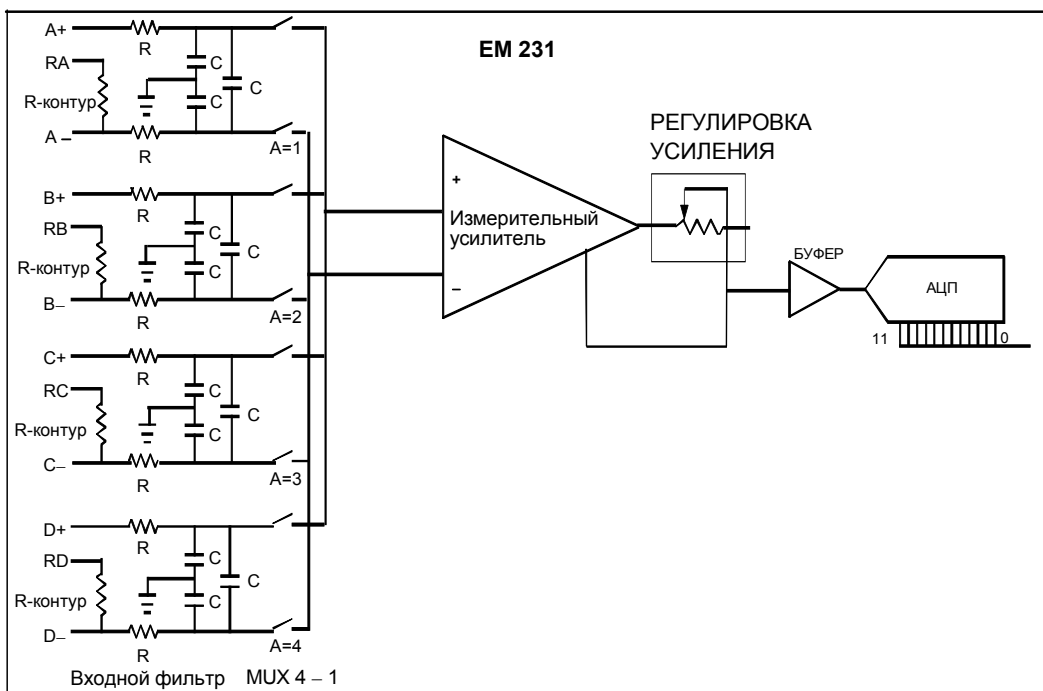


Рис. А–15. Схема соединений для входов EM 231

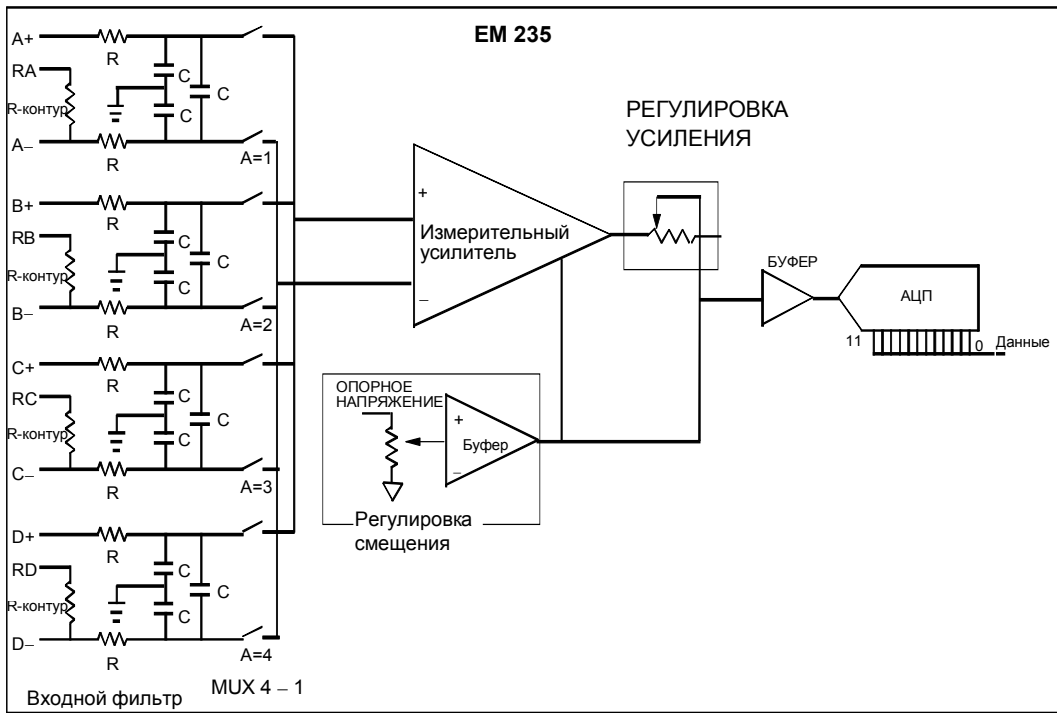


Рис. А-16. Схема соединений для входов EM 235

Формат слова данных для выходов EM 232 и EM 235

На рис. А-17 показано размещение 12-разрядного значения данных в слове аналоговых выходов CPU.

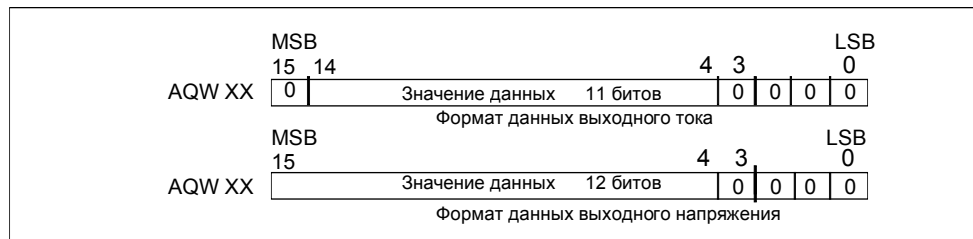


Рис.

А-17. Формат слова данных для выходов EM 232 и EM 235

Совет

12 битов данных цифро-аналогового преобразователя (ЦАП) выравниваются по левому краю в формате выходного слова данных. MSB – это знаковый бит: нуль указывает на положительное значение слова данных. Четыре концевых нуля усекаются перед загрузкой в регистры ЦАП. Эти биты не влияют на значение выходного сигнала.

Схема соединений для выходов EM 232 и EM 235

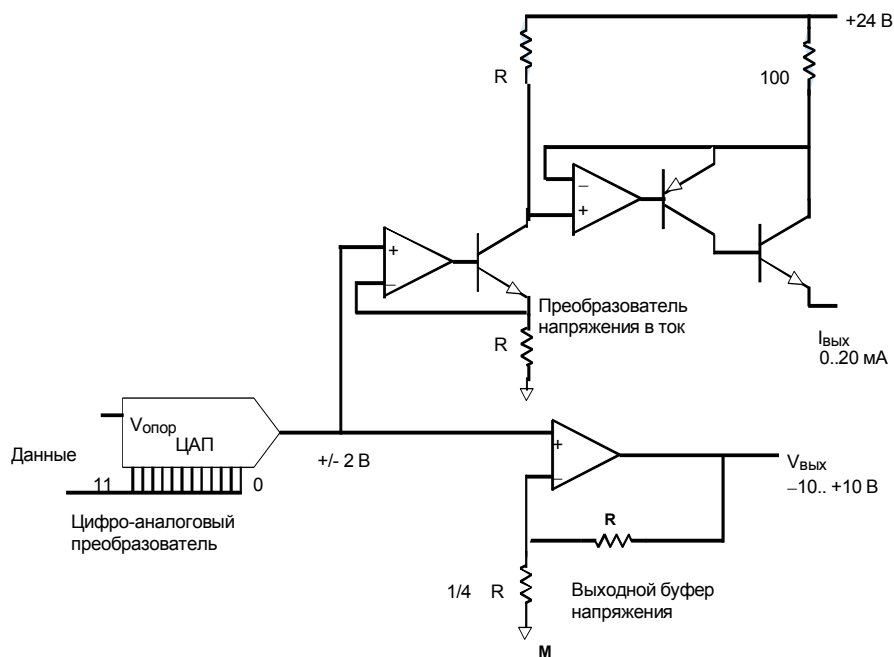


Рис. А–18. Схема соединений для выходов EM 232 и EM 235

Рекомендации по монтажу

Чтобы гарантировать точность и повторяемость, используйте следующие рекомендации:

- Обратите внимание на то, чтобы источник питания датчиков 24 В постоянного тока был свободным от помех и стабильным.
- Используйте для датчиков как можно более короткие провода.
- Используйте в качестве проводов датчиков экранированные витые пары.
- Заканчивайте экран только в месте расположения датчика.
- Замыкайте накоротко входы всех неиспользуемых каналов, как показано на рис. А–18.
- Избегайте изгиба проводов под острым углом.
- Используйте для прокладки проводов кабельные каналы.
- Избегайте размещения сигнальных проводов параллельно высоковольтным проводам. Если два провода должны пересекаться, то организуйте пересечение под прямым углом.
- Обеспечьте, чтобы входные сигналы были в пределах технических требований к напряжению синфазной помехи, путем гальванической развязки входных сигналов или выбирая в качестве опорной точки для них общий провод внешнего напряжения 24 В аналогового модуля.



Совет

Не рекомендуется использовать модули расширения EM 231 и EM 235 с термопарами.

Работа с аналоговым модулем ввода: точность и повторяемость

Модули аналогового ввода EM 231 и EM 235 являются недорогими, скоростными 12-разрядными модулями аналогового ввода. Эти модули способны преобразовывать аналоговый входной сигнал в соответствующее ему цифровое значение в течение 149 мкс. Преобразование входного аналогового сигнала выполняется каждый раз, когда ваша программа обращается к аналоговому входу. Эти времена должны прибавляться к основному времени выполнения команды, используемой для обращения к аналоговому входу.

EM 231 и EM 235 предоставляют необработанное цифровое значение (без линейаризации и фильтрации), которое соответствует аналоговому напряжению или току, присутствующему на входных клеммах модуля. Так как эти модули являются скоростными, они могут отслеживать быстрые изменения аналогового входного сигнала (включая внутренние и внешние помехи).

Вызываемые помехами изменения в отсчетах постоянного или медленно изменяющегося аналогового входного сигнала можно минимизировать путем усреднения по множеству отсчетов. По мере увеличения количества отсчетов, используемых при вычислении среднего значения, наблюдается соответствующее замедление реакции на изменения входного сигнала.

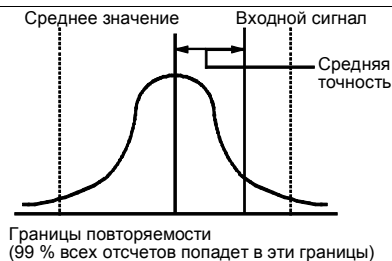


Рис. А–19. Определения точности

На рис. А–19 в графической форме показаны границы 99-процентной повторяемости, среднее значение отдельных отсчетов и средняя точность.

Технические требования к повторяемости описывают изменения в модуле от отсчета к отсчету при неизменном входном сигнале. Технические требования к повторяемости определяют границы интервала, в который будут попадать 99 % отсчетов. Повторяемость изображена на этом рисунке в форме колоколообразной кривой.

Технические требования к средней точности описывают среднее значение погрешности (разность между средним значением отдельных отсчетов и точным значением фактического аналогового входного сигнала).

Таблица А–22 дает технические требования к повторяемости и среднюю точность в той мере, в какой это касается каждого из настраиваемых диапазонов.

Определение данных для аналоговых модулей

- Точность: отклонение от ожидаемого значения на определенном входе или выходе
- Разрешающая способность: влияние изменения младшего байта на выходе.

Таблица А–22. Технические данные EM 231 и EM 235

Полный входной диапазон	Повторяемость ¹		Средняя точность ^{1,2,3,4}	
	% от полного диапазона	Отсчеты	% от полного диапазона	Отсчеты
Технические данные EM 231				
от 0 до 5 В	± 0,075%	± 24	± 0,1%	± 32
от 0 до 20 мА			± 0,05%	
от 0 до 10 В		± 48	± 0,05%	
± 2,5 В				
± 5 В				
Технические данные EM 235				
от 0 до 50 мВ	± 0,075%	± 24	± 0,25%	± 80
от 0 до 100 мВ			± 0,2%	± 64
от 0 до 500 мВ			± 0,05%	± 16
от 0 до 1 В				
от 0 до 5 В				
от 0 до 20 мА				
от 0 до 10 В	± 0,075%	± 48	± 0,25%	± 160
± 25 мВ			± 0,2%	± 128
± 50 мВ			± 0,1%	± 64
± 100 мВ				
± 250 мВ				
± 500 мВ			± 0,05%	± 32
± 1 В				
± 2,5 В				
± 5 В				
± 10 В				

- ¹ Измерения, сделанные после калибровки выбранного входного диапазона.
- ² Погрешность смещения сигнала вблизи нуля аналогового входа не скорректирована и не учитывается в данных о точности.
- ³ Вследствие конечного времени установления аналогового мультиплексора имеет место погрешность преобразования при переносе из канала в канал. Максимальная погрешность переноса равна 0,1 % разности между каналами.
- ⁴ Средняя точность включает влияние нелинейности и дрейфа от 0 до 55 °С.

Технические данные модулей расширения для термопар и термометров сопротивления (RTD)

Таблица А–24. Номера для заказа модулей для термопар и термометров сопротивления (RTD)

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6ES7 231-7PD22-0XA0	EM 231, аналоговый ввод, термопара, 4 входа	4 термопары	-	Нет
6ES7 231-7PF22-0XA0	EM 231, аналоговый ввод, термопара, 8 входов	8 термопар	-	Нет
6ES7 231-7PB22-0XA0	EM 231, аналоговый ввод RTD, 2 входа	2 RTD	-	Нет
6ES7 231-7PC22-0XA0	EM 231, аналоговый ввод RTD, 4 входа	4 RTD	-	Нет

Таблица А–25. Общие технические данные модулей для термопар и термометров сопротивления (RTD)

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В	+24 В
6ES7 231-7PD22-0XA0	EM 231, аналоговый ввод, термопара, 4 входа	71,2 x 80 x 62	210 г	1,8 Вт	87mA	60 mA
6ES7 231-7PF22-0XA0	EM 231, аналоговый ввод, термопара, 8 входов	71,2 x 80 x 62	210 г	1,8 Вт	87mA	60 mA
6ES7 231-7PB22-0XA0	EM 231, аналоговый ввод, RTD, 2 входа	71,2 x 80 x 62	210 г	1,8 Вт	87 mA	60 mA
6ES7 231-7PC22-0XA0	EM 231, аналоговый ввод RTD, 4 входа	71,2 x 80 x 62	210 г	1,8 Вт	87mA	60 mA

Таблица А–26. Технические данные модулей для термопар и термометров сопротивления (RTD)

Общие данные	6ES7 231-7PD22-0XA0	6ES7 231-7PF22-0XA0	6ES7 231-7PB22-0XA0	6ES7 231-7PC22-0XA0
	Термопара 4 входа	Термопара 8 входов	RTD, 2 входа	RTD, 4 входа
Электрическая развязка Полевые устройства с логикой Полевые устройства с 24 В пост. тока 24 В пост. тока с логикой	500 В перем. тока 500 В перем. тока 500 В перем. тока		500 В перем. тока 500 В перем. тока 500 В перем. тока	
Диапазон синфазной помехи на входе (входной канал относительно входного канала)	120 В перем. тока		0	
Подавление синфазной помехи	> 120 дБ при 120 В перем. тока		> 120 дБ при 120 В перем. тока	
Тип входа	Незаземленная термопара		RTD с "землей" модуля в качестве опорной точки	
Диапазоны входных сигналов ¹	Типы термопар (выбрать один на модуль) S, T, R, E, N, K, J Диапазон напряжений : +/- 80 мВ		Типы термопар (выбрать один на модуль): платина (Pt), медь (Cu), никель (Ni), или сопротивление Имеющиеся типы RTD вы найдете в таблице.	
Разрешающая способность входа Температура Напряжение Сопротивление	0,1° C / 0,1° F 15 битов плюс знак -		0,1° C / 0,1° F - 15 битов плюс знак	
Принцип измерения	Sigma-delta		Sigma-delta	
Время обновления модуля: все каналы	405 мс	810 мс	405 мс (700 мс для Pt10000)	810 мс (1400 мс для Pt10000)
Длина проводов	Максимум 100 метров до датчика		Максимум 100 метров до датчика	
Сопротивление проводной петли	Макс. 100 Ом		Макс. 20 Ом, 2,7 Ом для Cu	Макс. 20 Ом, 2,7 Ом для Cu RTD
Подавление помех	85 дБ при 50 Гц/60 Гц/ 400 Гц		85 дБ при 50 Гц/60 Гц/400 Гц	
Формат слова данных	Напряжение: от -27648 до + 27648		Сопротивление: от 0 до +27648	
Максимальная мощность потерь датчиков	-		1 мВт	
Полное входное сопротивление	≥ 1 МОм		≥ 10 МОм	
Максимальное входное напряжение	30 В пост. тока		30 В пост. тока (датчик), 5 В пост. тока (источник)	30 В пост. тока (датчик)
Затухание входного фильтра	-3 дБ при 21 кГц		-3 дБ при 3,6 кГц	-3 дБ при 21 кГц
Основная погрешность	0,1% от всей шкалы (напряжение)		0,1% от всей шкалы (сопротивление)	
Повторяемость	0,05% от всей шкалы		0,05% от всей шкалы	
Погрешность холодного спая	±1,5 ° C		-	
Диапазон питающего напряжения 24 В пост. тока	от 20,4 до 28,8 В пост. тока		от 20,4 до 28,8 В пост. тока	

¹ Установленный входной диапазон (температуры, напряжения или сопротивления) действителен для всех каналов модуля.

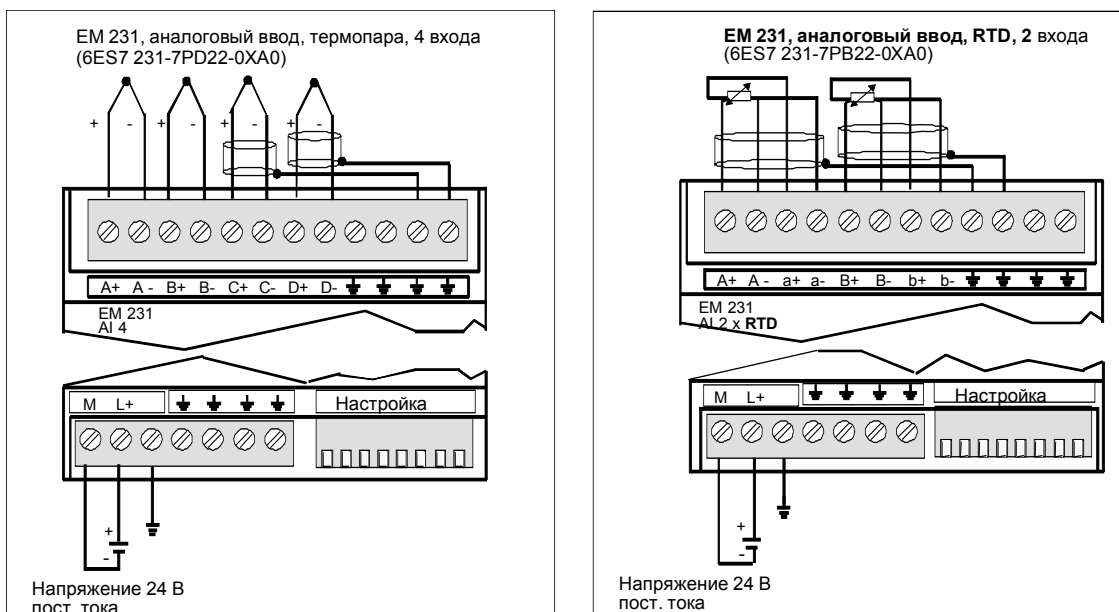


Рис. А–20. Обозначение клемм у модулей EM 231 Thermocouple и EM 231 RTD

Совместимость

Модули RTD и Thermocouple предназначены для работы с CPU 222, CPU 224, CPU 224XP и CPU 226.



Совет

Модули RTD и Thermocouple рассчитаны на максимальную производительность при установке в среде с устойчивой температурой.

В частности, модуль EM 231 Thermocouple имеет специальную схему компенсации холодного спая, которая измеряет температуру в точках подключения модуля и вносит необходимые изменения в измерение, чтобы компенсировать разность между опорной температурой и температурой модуля. Если температура окружающей среды быстро изменяется в области, где установлен модуль EM 231 Thermocouple, то вносится дополнительная погрешность.

Чтобы добиться максимальной точности и повторяемости, фирма Siemens рекомендует устанавливать S7–200 RTD и Thermocouple в местах с устойчивой температурой окружающей среды.

Помехоустойчивость

Для повышения помехоустойчивости используйте экранированные провода. Если входной канал термопары не используется, то неиспользуемый канал следует закоротить или включить его параллельно другому каналу.

Модуль EM 231 Thermocouple

Модуль EM 231 Thermocouple предоставляет семейству S7-200 удобный интерфейс с электрической развязкой для термопар семи типов: J, K, E, N, S, T и R. Он позволяет подключать S7-200 к аналоговым сигналам низкого уровня в диапазоне ± 80 мВ. Все термопары, присоединенные к модулю, должны быть одного типа.

Основные сведения о термопарах

Термопары образуются всякий раз, когда два разнородных металла электрически соединяются друг с другом. Генерируется напряжение, пропорциональное температуре точки спая. Это напряжение мало; один микровольт может представлять много градусов. Измерение напряжения от термопары, компенсация для дополнительных точек соединения и последующая линейаризация результата составляют основу измерения температуры при помощи термопар.

Когда вы подключаете термопару к модулю EM 231 Thermocouple, к модулю присоединяются два разнородных металлических провода в сигнальном разъеме модуля. Место, где два разнородных провода соединяются друг с другом, образует термопару датчика.

Еще две термопары образуются там, где два разнородных провода присоединяются к сигнальному разъему. Температура соединительного блока порождает напряжение, которое прибавляется к напряжению от термопары датчика. Если это напряжение не компенсируется, то сообщаемая температура отличается от температуры датчика.


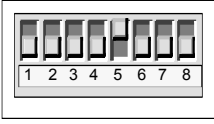
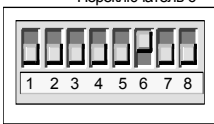
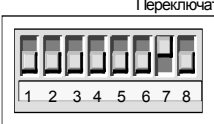
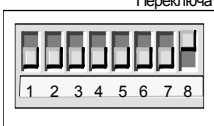
Для компенсации термопар разъема используется компенсатор холодного спая. Таблицы термопар основываются на температуре холодного спая, обычно равной нулю по шкале Цельсия. Компенсатор холодного спая модуля компенсирует разъем до нуля по шкале Цельсия. Компенсатор холодного спая восстанавливает напряжение, добавляемое термопарами разъема. Температура модуля измеряется внутри. Эта температура преобразуется в значение, прибавляемое к результату преобразования датчика. Скорректированный результат преобразования датчика затем линейаризуется с помощью таблиц для термопар.


Настройка модуля EM 231 Thermocouple


Настроечные двухпозиционные переключатели, расположенные в нижней части модуля, позволяют выбрать тип термопары, контроль обрыва проводов, шкалу измерения температуры и компенсацию холодного спая. Чтобы настройка двухпозиционных переключателей вступила в силу, нужно выключить, а затем включить ПЛК и/или источник питания потребителей 24 В.

Двухпозиционный переключатель 4 зарезервирован для будущего использования. Установите переключатель 4 в положение 0 (вниз, off). В таблице А–26 показаны другие установки двухпозиционных переключателей.

Таблица А–27. Двухпозиционные переключатели для настройки модуля EM 231 Thermocouple

Переключатели 1,2,3	Тип термопары	Установка	Описание
<p>Переключатели 1, 2, 3</p>  <p>↑ 1 - Вкл ↓ 0 - Выкл</p> <p>* Установите переключатель 4 в положение 0 (вниз)</p>	J (по умолчанию)	000	<p>Переключатели 1 – 3 устанавливают тип термопары (или режим mV) для всех каналов модуля. Например, для термопары типа E</p> <p>Переключатель 1 = 0, Переключатель 2 = 1, Переключатель 3 = 1.</p>
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
+/-80 мВ	111		
Переключатель 5	Проверка на обрыв провода – направление	Установка	Описание
<p>Переключатель 5</p>  <p>↑ 1 - Вкл ↓ 0 - Выкл</p>	Вверх по шкале (+3276,7 градуса)	0	<p>0 указывает положительное направление при обрыве провода, 1 указывает отрицательное направление при обрыве провода</p>
	Вниз по шкале (-3276,8 градуса)	1	
Переключатель 6	Проверка на обрыв провода – активизация	Установка	Описание
<p>Переключатель 6</p>  <p>↑ 1 - Вкл ↓ 0 - Выкл</p>	Активизирована	0	<p>Обнаружение обрыва провода выполняется посредством подпитки входных клемм током 25 мкА. Переключатель активизации проверки на обрыв провода разблокирует или блокирует источник тока. Проверка попадания в диапазон обрыва провода выполняется всегда, даже когда источник тока заблокирован. Модуль EM 231 Thermocouple обнаруживает обрыв провода, если входной сигнал по абсолютной величине превышает значение примерно 200 мВ. Когда обнаружен обрыв провода, показание модуля устанавливается равным значению, выбранному при настройке проверки на обрыв провода.</p>
	Деактивизирована	1	
Переключатель 7	Шкала температур	Установка	Описание
<p>Переключатель 7</p>  <p>↑ 1 - Вкл ↓ 0 - Выкл</p>	Цельсий (°C)	0	<p>Модуль EM 231 Thermocouple может сообщать температуру в градусах Цельсия или Фаренгейта. Преобразование градусов Цельсия в градусы Фаренгейта выполняется внутри модуля.</p>
	Фаренгейт (°F)	1	
Переключатель 8	Холодный спай	Установка	Описание
<p>Переключатель 8</p>  <p>↑ 1 - Вкл ↓ 0 - Выкл</p>	Компенсация холодного спая активизирована	0	<p>При использовании термопар нужно активизировать компенсацию холодного спая. Если компенсация холодного спая не активизирована, то результаты преобразования модуля будут ошибочными из-за напряжения, генерируемого там, где провод термопары подключается к разъему модуля. Холодный спай автоматически блокируется, если вы выбираете диапазон ±80 мВ.</p>
	Компенсация холодного спая деактивизирована	1	

	<p>Совет</p> <ul style="list-style-type: none"> ■ Источник тока контроля обрыва провода может создавать помехи сигналам от некоторых низковольтных источников, таких как имитаторы термодпар. ■ Входные напряжения, превышающие примерно ± 200 мВ, активизируют обнаружение обрыва провода даже тогда, когда источник тока для контроля обрыва провода заблокирован.
---	--

	<p>Совет</p> <ul style="list-style-type: none"> ■ При изменении температуры окружающей среды погрешность модуля может превышать приведенные технические данные. ■ При превышении диапазона температур окружающей среды модуля компенсация холодного спая модуля может быть ошибочной.
---	--

Использование термопары: индикаторы состояния

Модуль EM 231 Thermocouple снабжает ПЛК словами данных, отображающими температуру или состояния ошибки. Биты состояния показывают ошибку диапазона и отказа блока питания пользователя или модуля. Светодиоды отображают состояние модуля. Ваша программа должна иметь логику, предназначенную для того, чтобы обнаруживать состояния ошибки и реагировать на них в соответствии с приложением. В таблице А–27 показаны индикаторы состояния EM 231 Thermocouple.


Таблица А–28. Индикаторы состояния EM 231 Thermocouple

Ошибочное состояние	Данные канала	Светодиод SF Красный	Светодиод 24 V Зеленый	Бит состояния для диапазона ¹	Неисправность питания 24 В пост. тока ²
Нет ошибок	Данные преобразования	ВЫКЛ	ВКЛ	0	0
Отсутствует питание 24 В	32766	ВЫКЛ	ВЫКЛ	0	1
Контроль обрыва провода и источник тока активизированы	-32768/32767	Мигает	ВКЛ	1	0
Входной сигнал за пределами диапазона	-32768/32767	Мигает	ВКЛ	1	0
Диагностическая ошибка ³	0000	ВКЛ	ВЫКЛ	0	Примечание ³

¹ Бит состояния для диапазона – это бит 3 в байте регистра ошибок модуля (SMB9 для модуля 1, SMB11 для модуля 2 и т.д.)

² Бит состояния для неисправности питания 24 В – это бит 2 в байте регистра ошибок модуля (SMB 9, SMB 11 и т.д. См. Приложение D)

³ Диагностические ошибки вызывают ошибку настройки модуля. Бит состояния для неисправности питания 24 В может быть установлен или не установлен перед ошибкой настройки модуля.

	<p>Совет</p> <p>Формат данных канала представляет собой поразрядное дополнение до двух, 16-разрядные слова. Температура представляется в единицах, равных 0,1 градуса. (Например, если измеренная температура равна 100,2 градуса, то сообщаемые данные содержат 1002.) Данные о напряжении приводятся к шкале 27648. Например, сообщение о напряжении –60,0 мВ содержит значение –20736 ($= -60 \text{ мВ} / 80 \text{ мВ} * 27648$).</p>
---	--

Все четыре (восемь) канала обновляются каждые 405 (810) миллисекунд, если ПЛК выполнил считывание данных. Если ПЛК не считывает данные в течение времени одного обновления, то модуль сообщает старые данные до тех пор, пока не произойдет следующее обновление модуля после считывания ПЛК. Чтобы данные канала всегда сохранялись в актуальном состоянии, рекомендуется, чтобы программа ПЛК считывала данные, по крайней мере, с такой же частотой, как частота обновления модуля.


	<p>Совет</p> <p>При использовании модуля EM 231 Thermocouple нужно отключить аналоговую фильтрацию в ПЛК. Аналоговая фильтрация может препятствовать своевременному обнаружению состояний ошибки.</p>
---	--

Таблица А–29. Диапазоны температур (°C) и точность для различных типов термопар

Слово данных (1 цифра = 0,1 °C)		Тип J	Тип K	Тип T	Тип E	Типы R, S	Тип N	±80 мВ	
Десятич.	16-ричн.								
32767	7FFF	>1200,0 °C	>1372,0 °C	>400,0 °C	>1000,0 °C	>1768,0 °C	>1300,0 °C	>94,071 мВ	OF
↑	↑							↑	↑
32511	7EFF							94,071 мВ	OR
:	:							80,0029 мВ	
27649	6C01							80 мВ	
27648	6C00								
:	:								
17680	4510								
:	:								
13720	3598		↑						
:	:								
13000	32C8		↑						
:	:								
12000	2EE0	1200,0 °C							
:	:								
10000	2710								
:	:								
4000	0FA0								
:	:								
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1 °C	400,0 °C	0,1 °C	0,0029 мВ	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,1 °C	0,0 °C	0,0 мВ	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	0,0 °C	-0,1 °C	-0,0029 мВ	
:	:								
-500	FE0C	-150,0 °C							
-1500	FA24								
:	:								
-2000	F830	отрицат. перегрузка							
:	:								
-2100	F7CC	-210,0 °C							
:	:								
-2550	F60A								
:	:								
-2700	F574								
:	:								
-27648	9400								
-27649	93FF								
:	:								
-32512	8100								
↓	↓								
-32768	8000	<-210,0 °C	<-270,0 °C	<-270,0 °C	<-270,0 °C	<-50,0 °C	<-270,0 °C	<-94,071 мВ	UF
Точность по всему диапазону		±0,1%	±0,3%	±0,6%	±0,1%	±0,6%	±0,1%	±0,1%	
Точность (номинальный диапазон без холодного спая)		±1,5 °C	±1,7 °C	±1,4 °C	±1,3 °C	±3,7 °C	±1,6 °C	±0,10%	
Погрешность холодного спая		±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	-/-	
*OF = Overflow (переполнение); OR = Overrange (перегрузка); NR = Nominal range (номинальный диапазон); UR = Underrange (отрицательная перегрузка); UF = Underflow (отрицательное переполнение)									
↑ показывает, что все аналоговые величины, большие этой и не превышающие порога обрыва провода, выдают значение, соответствующее переполнению, 32767 (0x7FFF).									
↓ показывает, что все аналоговые величины, меньшие этой, но большие порога обрыва провода, выдают значение, соответствующее отрицательному переполнению, -32768 (0x8000).									

Таблица А–30. Диапазоны температур (°F) для различных типов термопар

Слово данных (1 цифра = 0,1°F)		Тип J	Тип K	Тип T	Тип E	Тип R, S	Тип N	±80 мВ	
Десятич.	16-ричн.								
32767	7FFF	>2192,0 °F	>2502,0 °F	>752,0 °F	>1832,0 °F	>3214,0 °F	>2372,0 °F	>94,071 мВ	OF
↑	↑					↑		↑	↑
32511	7EFF							94,071 мВ	OR
								80,0029 мВ	
32140	7D90					3214,0 °F			
27649	6C01					2764,8 °F			
27648	6C00		↑					80 мВ	NR
:	:								
25020	61B8		2502,0 °F перегрузка						
:	:								
23720	5CA8	↑	2372,0 °F				↑	2372,0 °F	
:	:								
21920	55A0	2192,0 °F			↑				
:	:								
18320	4790			↑	1832,0 °F				
:	:								
7520	1D60			752,0 °F		752,0 °F			
:	:								
320	0140					отрицат. перегрузка	32,0 °F		
:	:								
1	0001	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,0029 мВ	
0	0000	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 мВ	
-1	FFFF	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,0029 мВ	
:	:								
-580	FDBC					-58,0 °F			
:	:								
-2380	F6B4	-238,0 °F							
:	:								
-3280	F330	отрицат. перегрузка	-328,0 °F						
:	:								
-3460	F27C	-346,0 °F							
:	:		отрицат. перегрузка						
-4270	EF52			-427,0 °F	-427,0 °F				
:	:			отрицат. перегрузка	отрицат. перегрузка				
-4540	EE44	↓	-454,0 °F	-454,0 °F	-454,0 °F				
:	:								
-27648	9400		↓	↓	↓			-80 мВ	
-27649	93FF							-80,0029 мВ	OR
:	:							-94,071 мВ	
-32512	8100								
↓	↓								
-3268	8000	<-346,0 °F	<-454,0 °F	<-454,0 °F	<-454,0 °F	<-58,0 °F	<-454,0 °F	<-94,07 мВ	UF

*OF = Overflow (переполнение); OR = Overrange (перегрузка); NR = Nominal range (номинальный диапазон);

UR = Underrange (отрицательная перегрузка); UF = Underflow (отрицательное переполнение)

↑ показывает, что все аналоговые величины, большие этой и не превышающие порога обрыва провода, выдают значение, соответствующее переполнению, 32767 (0x7FFF).

↓ показывает, что все аналоговые величины, меньшие этой, но большие порога обрыва провода, выдают значение, соответствующее отрицательному переполнению, -32768 (0x8000).

Модуль EM 231 RTD

Модуль EM 231 RTD предоставляет семейству S7-200 удобный интерфейс с несколькими разными термометрами сопротивления (RTD). Это позволяет S7-200 также выполнять измерения в трех различных диапазонах сопротивлений. Оба RTD, подключенные к модулю, должны быть одного типа.

Настройка модуля EM 231 RTD

Двухпозиционные переключатели позволяют устанавливать тип RTD, конфигурацию подключения, температурную шкалу и направление перегорания. Двухпозиционные переключатели расположены в нижней части модуля, как показано на рис. А–21. Чтобы настройка двухпозиционных переключателей вступила в силу, вам нужно выключить, а затем включить ПЛК и/или источник питания потребителей 24 В.

Выберите тип RTD, устанавливая двухпозиционные переключатели 1, 2, 3, 4, 5 и 6 в соответствии с RTD, как показано в таблице А–31 и А–32. Другие установки двухпозиционных переключателей показаны в таблице А–33.



Рис. А–21. Двухпозиционные переключатели для модуля EM 231 RTD

Таблица А–31. Выбор типа RTD: двухпозиционные переключатели (SW) с 1 по 6 для модулей с 4 входами

Тип RTD и альфа ¹	SW1	SW2	SW3	SW4	SW5	SW6	Тип RTD и альфа ¹	SW1	SW2	SW3	SW4	SW5	SW6
100 Ом Pt 0,003850 (по умолчанию)	0	0	0	0	0	0	100 Ом Pt 0,00302	1	0	0	0	0	0
200 Ом Pt 0,003850	0	0	0	0	1	0	200 Ом Pt 0,003902	1	0	0	0	1	0
500 Ом Pt 0,003850	0	0	0	1	0	0	500 Ом Pt 0,003902	1	0	0	1	0	0
1000 Ом Pt 0,003850	0	0	0	1	1	0	1000 Ом Pt 0,003902	1	0	0	1	1	0
100 Ом Pt 0,003920	0	0	1	0	0	0	ЗАМЕНИТЕЛЬ	1	0	1	0	0	0
200 Ом Pt 0,003920	0	0	1	0	1	0	100 Ом Ni 0,00672	1	0	1	0	1	0
500 Ом Pt 0,003920	0	0	1	1	0	0	120 Ом Ni 0,00672	1	0	1	1	0	0
1000 Ом Pt 0,003920	0	0	1	1	1	0	1000 Ом Ni 0,00672	1	0	1	1	1	0
100 Ом Pt 0,00385055	0	1	0	0	0	0	100 Ом Ni 0,006178	1	1	0	0	0	0
200 Ом Pt 0,00385055	0	1	0	0	1	0	120 Ом Ni 0,006178	1	1	0	0	1	0
500 Ом Pt 0,00385055	0	1	0	1	0	0	1000 Ом Ni 0,006178	1	1	0	1	0	0
1000 Ом Pt 0,00385055	0	1	0	1	1	0	10000 Ом Pt 0,003850	1	1	0	1	1	0
100 Ом Pt 0,003916	0	1	1	0	0	0	10 Ом Cu 0,004270	1	1	1	0	0	0
200 Ом Pt 0,003916	0	1	1	0	1	0	150 Ом FS Сопротивление	1	1	1	0	1	0
500 Ом Pt 0,003916	0	1	1	1	0	0	300 Ом FS Сопротивление	1	1	1	1	0	0
1000 Ом Pt 0,003916	0	1	1	1	1	0	600 Ом PHYS Сопротивление	1	1	1	1	1	0
ГОСТ 50 Ом Pt 0,00385055	0	0	0	0	1	1	ГОСТ 50 Ом Cu 0,00426	0	1	0	1	1	1
ГОСТ 100 Ом Pt 0,00385055	0	0	0	1	0	1	ГОСТ 100 Ом Cu 0,00426	0	1	1	0	0	1
ГОСТ 500 Ом Pt 0,00385055	0	0	0	1	1	1	ГОСТ 500 Ом Cu 0,00426	0	1	1	0	1	1

ГОСТ 10 Ом Pt 0,003910	0	0	1	0	0	1	ГОСТ 10 Ом Cu 0,00428	0	1	1	1	0	1
ГОСТ 50 Ом Pt 0,003910	0	0	1	0	1	1	ГОСТ 50 Ом Cu 0,00428	0	1	1	1	1	1
ГОСТ 100 Ом Pt 0,003910	0	0	1	1	0	1	ГОСТ 100 Ом Cu 0,00428	1	0	0	0	0	1
ГОСТ 500 Ом Pt 0,003910	0	0	1	1	1	1	ГОСТ 500 Ом Cu 0,00428	1	0	0	0	1	1
ГОСТ 1000 Ом Pt 0,003910	0	1	0	0	0	1	Резервный	1	0	0	1	0	1
LG-Ni 1000 Ом Pt 0,005000	0	1	0	0	1	1							

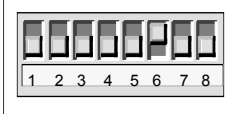
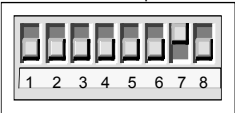
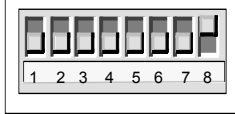
¹ Все RTD должны представлять 0°C при приведенных в таблице значениях сопротивления, кроме Cu 10 Ом. Cu 10 Ом представляет 25°C при 10 омах и 0°C при 9,035 Ом.

Таблица А-32. Выбор типа RTD: двухпозиционные переключатели (SW) с 1 по 5 для модулей с 2 входами

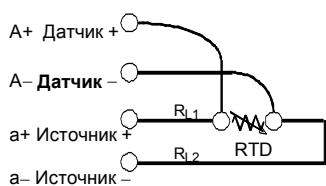
Тип RTD и альфа ¹	SW1	SW2	SW3	SW4	SW5	Тип RTD и альфа ¹	SW1	SW2	SW3	SW4	SW5
100 Ом Pt 0,003850 (по умолчанию)	0	0	0	0	0	100 Ом Pt 0,00302	1	0	0	0	0
200 Ом Pt 0,003850	0	0	0	0	1	200 Ом Pt 0,003902	1	0	0	0	1
500 Ом Pt 0,003850	0	0	0	1	0	500 Ом Pt 0,003902	1	0	0	1	0
1000 Ом Pt 0,003850	0	0	0	1	1	1000 Ом Pt 0,003902	1	0	0	1	1
100 Ом Pt 0,003920	0	0	1	0	0	ЗАМЕНИТЕЛЬ	1	0	1	0	0
200 Ом Pt 0,003920	0	0	1	0	1	100 Ом Ni 0,00672	1	0	1	0	1
500 Ом Pt 0,003920	0	0	1	1	0	120 Ом Ni 0,00672	1	0	1	1	0
1000 Ом Pt 0,003920	0	0	1	1	1	1000 Ом Ni 0,00672	1	0	1	1	1
100 Ом Pt 0,00385055	0	1	0	0	0	100 Ом Ni 0,006178	1	1	0	0	0
200 Ом Pt 0,00385055	0	1	0	0	1	120 Ом Ni 0,006178	1	1	0	0	1
500 Ом Pt 0,00385055	0	1	0	1	0	1000 Ом Ni 0,006178	1	1	0	1	0
1000 Ом Pt 0,00385055	0	1	0	1	1	10000 Ом Pt 0,003850	1	1	0	1	1
100 Ом Pt 0,003916	0	1	1	0	0	10 Ом Cu 0,004270	1	1	1	0	0
200 Ом Pt 0,003916	0	1	1	0	1	150 Ом FS Сопротивление	1	1	1	0	1
500 Ом Pt 0,003916	0	1	1	1	0	300 Ом FS Сопротивление	1	1	1	1	0
1000 Ом Pt 0,003916	0	1	1	1	1	600 Ом PHYS Сопротивление	1	1	1	1	1

¹ Все RTD должны представлять 0°C при приведенных в таблице значениях сопротивления, кроме Cu 10 Ом. Cu 10 Ом представляет 25°C при 10 омах и 0°C при 9,035 Ом.

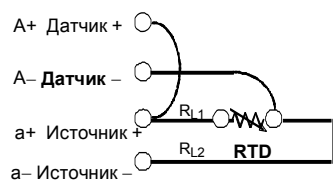
Таблица А–33. Установка двухпозиционных переключателей RTD

Переключатель 6	Проверка на обрыв проводов/Выход за пределы диапазона	Установка	Описание
<p>Переключатель 6</p>  <p>Настройка ↑ 1 - Вкл ↓ 0 - Выкл</p>	Вверх по шкале (+3276,7 градусов)	0	Указывает положительное направление при обрыве провода или выходе за пределы допустимого диапазона
	Вниз по шкале (-3276,8 градусов)	1	Указывает отрицательное направление при обрыве провода или выходе за пределы допустимого диапазона
Переключатель 7	Шкала температур	Установка	Описание
<p>Переключатель 7</p>  <p>Настройка ↑ 1 - Вкл ↓ 0 - Выкл</p>	Цельсий (°C)	0	Модуль RTD может сообщать температуру в градусах Цельсия или Фаренгейта. Преобразование градусов Цельсия в градусы Фаренгейта выполняется внутри модуля.
	Фаренгейт (°F)	1	
Переключатель 8	Схема подключения	Установка	Описание
<p>Переключатель 8</p>  <p>Настройка ↑ 1 - Вкл ↓ 0 - Выкл</p>	3–проводная	0	Вы можете подключить модуль RTD к датчику тремя способами (показанными на рисунке). Самым точным является 4-проводное подключение. Наименее точным является 2-проводное подключение, которое рекомендуется только тогда, когда в вашем приложении можно игнорировать погрешности вследствие монтажа проводов.
	2–проводная или 4–проводная	1	

4-проводное подключение RTD
(самое точное)



3-проводное подключение RTD



2-проводное подключение RTD



Примечание: R_{L1} – сопротивление провода от клеммы a+ до RTD
 R_{L2} – сопротивление провода от клеммы a- до RTD

Рис. А–22. Подключение RTD к датчику 4, 3 и 2 проводами

Индикаторы состояния EM 231 RTD

Модуль RTD снабжает ПЛК словами данных, отображающими температуру или ошибочные состояния. Биты состояния показывают ошибку диапазона и отказ блока питания пользователя или модуля. Светодиоды отображают состояние модуля. Ваша программа должна иметь логику, предназначенную для того, чтобы обнаруживать ошибочные состояния и реагировать на них в соответствии с приложением. Таблица А–32 показывает индикаторы состояния, предоставляемые модулем EM 231 RTD.


	<p>Совет</p> <p>Формат данных канала представляет собой поразрядное дополнение до двух, 16-разрядные слова. Температура представляется в единицах, равных 0,1 градуса. (Например, если измеренная температура равна 100,2 градуса, то сообщаемые данные содержат 1002.) Данные о сопротивлении приводятся к шкале 27648. Например, сообщение о сопротивлении величиной 75 % от полной шкалы содержит значение 20736. ($225 \text{ Ом}/300 \text{ Ом} * 27648 = 20736$)</p>
---	--

Таблица А–34. Индикаторы состояния EM 231 RTD


Ошибочное состояние	Данные канала	Светодиод SF Красный	Светодиод 24 V Зеленый	Бит состояния для диапазона ¹	Неисправность питания 24 В пост. тока ²
Нет ошибок	Данные преобразования	ВЫКЛ	ВКЛ	0	0
Отсутствует питание 24 В	32766	ВЫКЛ	ВЫКЛ	0	1
SW распознает обрыв провода	-32768/32767	Мигает	ВКЛ	1	0
Входной сигнал за пределами диапазона	-32768/32767	Мигает	ВКЛ	1	0
Диагностическая ошибка ³	0000	ВКЛ	ВЫКЛ	0	Примечание ³

¹ Бит состояния для диапазона – это бит 3 в байте регистра ошибок модуля (SMB9 для модуля 1, SMB11 для модуля 2 и т.д.)


² Бит состояния для неисправности питания 24 В – это бит 2 в байте регистра ошибок модуля (SMB 9, SMB 11 и т.д. обратитесь к Приложению D)

³ Диагностические ошибки вызывают ошибку настройки модуля. Бит состояния для неисправности питания 24 В может быть установлен или не установлен перед ошибкой настройки модуля.

Данные 2 (4) канала обновляются каждые 405 (810) миллисекунд, если ПЛК выполнил считывание данных. Если ПЛК не считывает данные в течение времени одного обновления, то модуль сообщает старые данные до тех пор, пока не произойдет следующее обновление модуля после считывания ПЛК. Чтобы данные канала всегда сохранялись в актуальном состоянии, рекомендуется, чтобы программа ПЛК считывала данные, по крайней мере, с такой же частотой, как частота обновления модуля.

	<p>Совет</p> <p>При использовании модуля RTD нужно отключить аналоговую фильтрацию в ПЛК. Аналоговая фильтрация может препятствовать своевременному обнаружению состояний ошибки.</p>
---	--

Проверка на обрыв провода выполняется программным обеспечением внутри модуля RTD. О состояниях "Входы вне допустимого диапазона" и "Обнаружен обрыв провода" сигнализируется установкой бита состояния диапазона в SMB и установкой данных канала на верхнем или нижнем конце диапазона в зависимости от положения переключателя. Обнаружение обрыва провода занимает, как минимум, три цикла сканирования модуля и может длиться больше в зависимости от того, в каком проводе (проводах) имеет место обрыв. Обрыв провода "Источник+" и/или провода "Источник–" обнаруживается за минимальное время. Обнаружение обрыва провода "Датчик+" и/или провода "Датчик–" может занимать 5 секунд и более. Линии датчиков, имеющие обрыв, могут случайным образом представлять достоверные данные вперемежку с обнаруживаемыми обрывами провода, особенно в среде с электрическими помехами. Электрические помехи могут также продлевать время, требуемое для обнаружения состояния обрыва провода. Рекомендуется, чтобы в прикладной программе после того, как были сообщены достоверные данные, фиксировались индикации обрыва провода и выхода сигнала за пределы допустимого диапазона.

	<p>Совет</p> <p>Если вы не используете входной канал RTD, то вы можете подключить к этому каналу вместо RTD резистор, чтобы светодиод SF не мигал при обнаружении обрыва провода. Этот резистор должен иметь такой же номинал, как и RTD. Например, используйте 100 Ом вместо RTD PT100.</p>
---	---

Диапазоны модуля EM 231 RTD

Температурные диапазоны и точность для каждого типа модулей EM 231 RTD представлены в таблицах А–35 и А–36.

Таблица А–35. Диапазоны температур (°C) для различных типов RTD

Decimal	Hex	Pt10000	Pt100, Pt200, Pt500, Pt1000, & GOST 0.003850	GOST 0.003910 Pt10, Pt50, Pt100, Pt500	Ni100, Ni120, Ni1000 LG-Ni1000	Ni100 GOST 0.006170	Cu 10 0.00427	GOST 0.00426 Cu 10 Cu 50 Cu 100 Cu 500	GOST 0.00428 Cu 10 Cu 50 Cu 100 Cu 500	
32767	7FFF									
32768	7FFE									
32511	7EFF									
27649	6C01									
27648	6C00									
25000	61A8									
18000	4650									
15000	3A98									
12950	3296									
11000	2AF8									
10000	2710									
8500	2134									
6000	1770									
3120	0C30									
2950	0B86									
2600	0A26									
2500	09C4									
2400	960									
2124	84C									
2000	7D0									
1800	708									
1	0001									
0	0000									
-1	FFFF									
500	FE0C									
-600	FDA8									
-1050	FBE6									
-2000	F830									
-2400	F6A0									
-2430	F682									
-2600	F5D8									
-273.2	F554									
-6000	E890									
-10500	D6FC									
-12000	D120									
-32767	8001									
-32768	8000									
ACCURACY OVER FULL SPAN		±0.4%	±0.1%	±0.5%	±0.2%	±0.5%	±0.2%	±0.3%	±0.3%	
ACCURACY IN NOMINAL RANGE		±4° C	±1° C	±1° C ¹	±0.6° C	±42.6° C	±1° C	±1° C	±1° C	

Технические данные модуля EM 277 PROFIBUS-DP

Таблица А–35. Номер для заказа модуля EM 277 PROFIBUS-DP

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	-	-	Нет

Таблица А–36. Общие технические данные модуля EM 277 PROFIBUS-DP

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В	+24 В
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	71 x 80 x 62	175 г	2,5 Вт	150 мА	См. ниже

Таблица А–37. Технические данные модуля EM 277 PROFIBUS-DP

Общие данные	6ES7 277-0AA22-0XA0
Количество портов (при ограничении мощности)	1
Электрический интерфейс	RS-485
Скорости передачи для PROFIBUS-DP/MPI (устанавливаются автоматически)	9,6; 19,2; 45,45; 93,75; 187,5 и 500 кБод; 1, 1,5; 3; 6 и 12 МБод
Протоколы	PROFIBUS-DP slave и MPI slave
Длина кабеля	
до 93,75 кБод	1200 м
187,5 кБод	1000 м
500 кБод	400 м
от 1 до 1,5 МБод	200 м
от 3 до 12 МБод	100 м
Сетевые возможности	
Устанавливаемые адреса станций	от 0 до 99 (устанавливаются поворотными переключателями)
Максимальное количество станций на сегмент	32
Максимальное количество станций на сеть	126, до 99 станций EM 277
Соединения MPI	Всего 6, 2 зарезервированы (1 для PG и 1 для OP)
Требования к блоку питания входов 24 В пост. тока	
Диапазон напряжений	от 20,4 до 28,8 В пост. тока (класс 2, при ограничении мощности или от блока питания датчиков из ПЛК)
Максимальный ток только модуль с активным портом плюс 90 мА нагрузки порта 5 В плюс 120 мА нагрузки порта 24 В	30 мА 60 мА 180 мА
Пульсации/паразитные токи (<10 МГц)	<1 В от минимума к минимуму (максимум)
Электрическая развязка (полевые устройства с логикой) ¹	500 В перем. тока в течение 1 минуты
Блок питания 5 В пост. тока на коммуникационный порт	
Максимальный ток на порт	90 мА
Электрическая развязка (24 В пост. тока с логикой)	500 В перем. тока в течение 1 минуты
Блок питания 24 В пост. тока на коммуникационный порт	
Диапазон напряжений	от 20,4 до 28,8 В пост. тока
Максимальный ток на порт	120 мА
Ограничение тока	от 0,7 до 2,4 А
Электрическая развязка	Нет развязки, та же самая схема, что и для входа 24 В пост. тока

¹ Блок питания 24 В пост. тока не снабжает питанием логику модуля. Блок питания 24 В пост. тока предоставляет питание для коммуникационного порта.

CPU S7-200, поддерживающие интеллектуальные модули

Slave-модуль EM 277 PROFIBUS-DP – это интеллектуальный модуль расширения, предназначенный для работы с ПЛК S7-200, которые представлены в таблице А-38.

Таблица А-38. Совместимость модуля EM 277 PROFIBUS-DP с CPU S7-200

CPU	Описание
CPU 222, начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224, начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP, начиная с версии 2.0	CPU 224XP DC/DC/DC и CPU 224XP AC/DC/Relay
CPU 226, начиная с версии 1.10	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

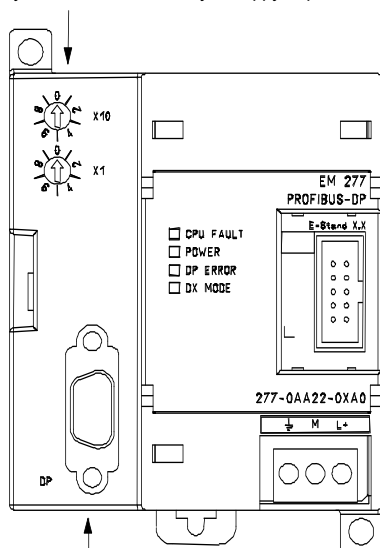
Переключатели адресов и светодиоды

Переключатели адресов и светодиоды состояния расположены на передней панели модуля, как показано на рис. А-23. Показано также распределение контактов порта для slave-устройства DP. Описание светодиодов состояния вы найдете в таблице А-42.

Вид спереди EM 277 PROFIBUS-DP

Переключатели адресов:

- x10 устанавливает старшую цифру адреса
- x1 устанавливает младшую цифру адреса



Назначение контактов 9-контактного миниатюрного D-образного разъема

№ контакта	Описание
1	Масса, соединенная с корпусом разъема
2	Обратный провод 24 В (то же, что М на клеммном блоке)
3	Потенциально развязанный сигнал В (RxD/TxD+)
4	Потенциально развязанный запрос на передачу (уровень TTL)
5	Потенциально развязанный обратный провод 5 В
6	Потенциально развязанные 5 В (макс. 90 мА)
7	+24 В (макс. 120 мА, с диодом для защиты от обратной полярности)
8	Потенциально развязанный сигнал А (RxD/TxD-)
9	Не подключен

Примечание: Потенциально развязанный означает наличие изоляции, выдерживающей напряжение 500 В, относительно цифровой логики и входного напряжения 24 В.

Разъем slave-устройства DP

Рис. А-23. EM 277 PROFIBUS-DP

Стандартный обмен данными с децентрализованной периферией (DP)

PROFIBUS-DP (или DP Standard) – это протокол обмена данными с удаленными входами-выходами, определенный Европейским стандартом EN 50170. Устройства, соответствующие этому стандарту, совместимы, даже если они изготовлены разными компаниями. DP означает децентрализованную периферию, то есть удаленные входы-выходы. PROFIBUS означает Process Field Bus [полевая шина процесса].

В модуле EM 277 PROFIBUS-DP реализован протокол DP Standard, определенный для slave-устройств в следующих стандартах на протоколы обмена данными:

- EN 50 170 (PROFIBUS) В модуле EM 277 PROFIBUS-DP реализован протокол DP Standard, определенный для slave-устройств в следующих стандартах на протоколы обмена данными.
- EN 50 170 (DP Standard) описывает скоростной циклический обмен данными между master- и slave-устройствами DP. Этот стандарт определяет процедуры конфигурирования и параметризации, объясняет, как функционирует циклический обмен данными с децентрализованной периферией, и перечисляет поддерживаемые возможности диагностики.

DP-master конфигурируется для того, чтобы знать адреса, типы slave-устройств и любую информацию о назначении параметров, которую требуют slave-устройства. Master-устройству также сообщается, где размещать данные, которые считываются из slave-устройств (входы), и где получить данные, чтобы записывать в slave-устройства (выходы). DP-master организует сеть, а затем инициализирует ее slave-устройства DP. DP-master записывает информацию о назначении параметров и конфигурацию входов-выходов в slave-устройство. Затем master считывает из slave-устройства диагностику для проверки того, что DP-slave принял параметры и конфигурацию входов-выходов. Затем master начинает обмен данными со slave-устройством. Каждая транзакция со slave-устройством записывает выходы и считывает входы. Режим обмена данными продолжается неограниченное время. Slave-устройства могут уведомлять master-устройство о том, что имеет место исключительная ситуация, и тогда master считывает из slave-устройства диагностическую информацию.

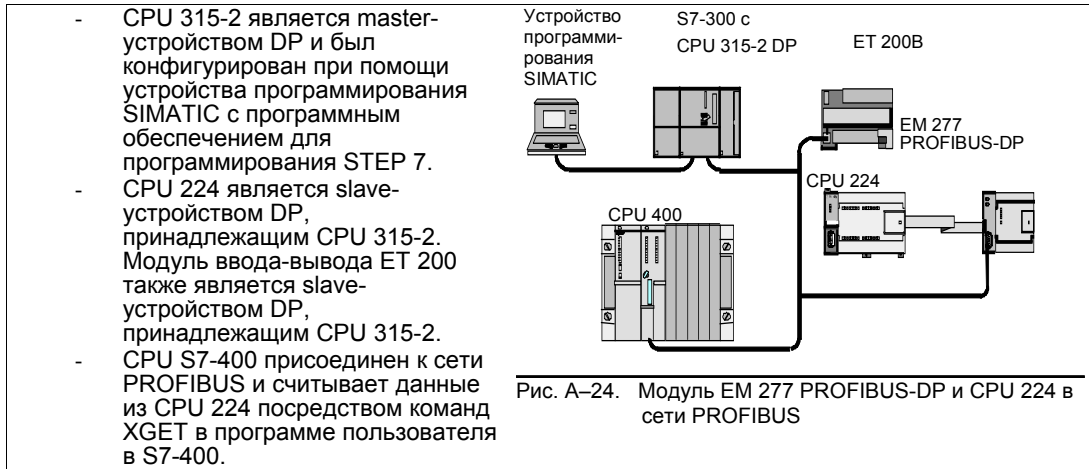
Если DP-master записал параметры и конфигурацию входов-выходов в DP-slave и DP-slave принял параметры и конфигурацию из master-устройства, то теперь master владеет этим slave-устройством. DP-slave принимает запросы на запись только от того master-устройства, которое владеет им. Другие master-устройства в сети могут считывать входы и выходы slave-устройства, но они не могут записывать что-либо в это slave-устройство.

Использование EM 277 для подключения S7–200 к сети в качестве slave-устройства DP

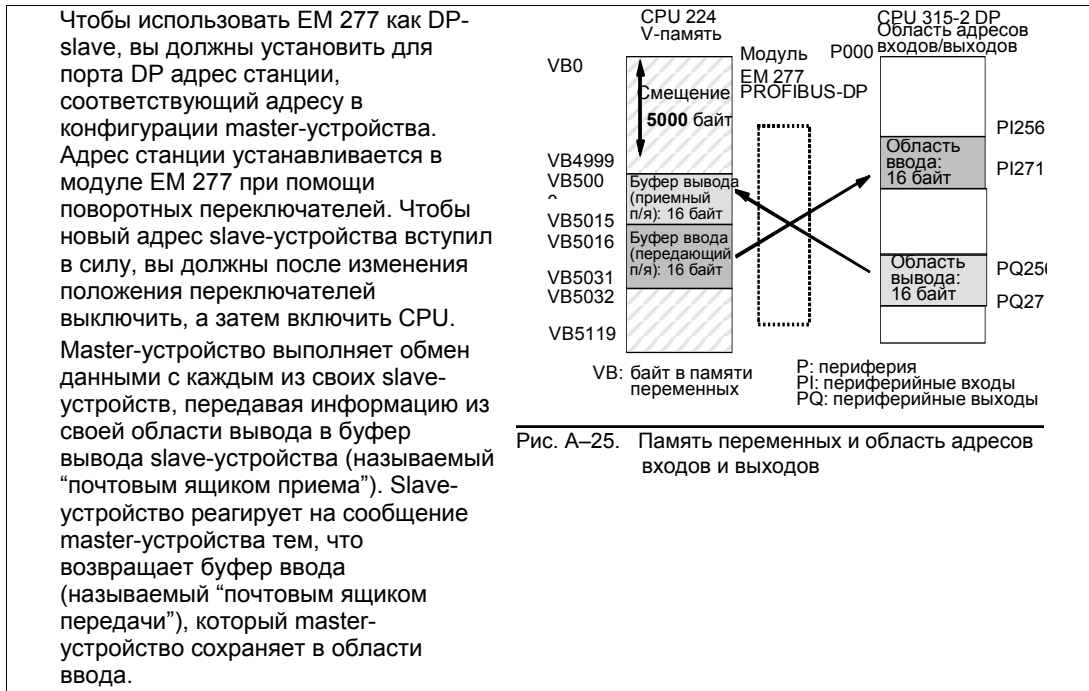
CPU S7-200 может подключаться к сети PROFIBUS-DP через slave-модуль расширения EM 277 PROFIBUS-DP. EM 277 подключается к CPU S7-200 через шину последовательного ввода-вывода. Сеть PROFIBUS подключается к модулю EM 277 PROFIBUS-DP через его коммуникационный порт DP. Этот порт работает с любой скоростью передачи данных PROFIBUS в диапазоне от 9600 Бод до 12 МБод. Поддерживаемые скорости передачи вы найдете в технических данных модуля EM 277 PROFIBUS-DP.

Модуль EM 277 в качестве slave-устройства DP принимает из master-устройства несколько различных конфигураций входов-выходов, позволяющих вам подстраивать количество передаваемых данных, чтобы удовлетворить требования приложения. В отличие от многих устройств DP, модуль EM 277 передает не только данные входов-выходов. Входы, значения счетчиков, значения таймеров и другие вычисляемые значения могут передаваться в master-устройство посредством пересылки данных вначале в память переменных в CPU S7-200. Аналогично, данные из master-устройства сохраняются в памяти переменных в CPU S7-200 и могут пересылаться в другие области данных.

Порт DP модуля EM 277 PROFIBUS-DP может быть закреплен за master-устройством DP в сети и, тем не менее, производить обмен данными в качестве slave-устройства MPI с другими master-устройствами, такими как устройства программирования SIMATIC или CPU S7-300/S7-400 в той же самой сети. На рис. А–24 показана сеть PROFIBUS с CPU 224 и модулем EM 277 PROFIBUS-DP.



Конфигурирование



На рис. А–25 показан пример памяти переменных и области адресов входов и выходов master-устройства PROFIBUS-DP.

DP-master может конфигурировать EM 277 PROFIBUS-DP так, чтобы он принимал из master-устройства выходные данные и возвращал master-устройству входные данные. Буферы выходных и входных данных находятся в памяти переменных (V-памяти) CPU S7-200. При конфигурировании master-устройства DP вы определяете в рамках параметризации EM 277 адрес байта в V-памяти, где должен начинаться буфер выходных данных. Вы также определяете конфигурацию входов-выходов как количество выходных данных, записываемых в CPU S7-200, и количество входных данных, поступающих обратно из CPU S7-200. EM 277 из конфигурации входов-выходов определяет размер буферов ввода и вывода. DP-master записывает информацию о назначении параметров и конфигурацию входов-выходов в модуль EM 277 PROFIBUS DP. Затем EM 277 передает в CPU S7-200 адрес V-памяти и длину входных и выходных данных.

На рис. А–25 показана модель V-памяти CPU 224 и области адресов входов-выходов CPU master-устройства DP. В этом примере DP-master определил конфигурацию входов-выходов из 16 выходных байтов и 16 входных байтов и смещение 5000 в памяти переменных. Длины буферов вывода и ввода в CPU 224 (определяемые из конфигурации входов-выходов) равны 16 байтам. Буфер выходных данных начинается в ячейке V5000; буфер ввода следует непосредственно за буфером вывода и начинается в ячейке V5016. Выходные данные (от master-устройства) помещаются в V-память, начиная с ячейки V5000. Входные данные (для master-устройства) берутся из V-памяти, начиная с ячейки V5016.



Совет

Если вы работаете с блоком данных (согласованные данные) длиной в три байта или блоками данных длиной более четырех байтов, то вы должны использовать SFC14, чтобы считывать входы slave-устройства DP, и SFC15, чтобы обращаться к выходам slave-устройства DP. Дополнительную информацию вы найдете в руководстве «Системное программное обеспечение для S7-300 и S7-400. Справочное руководство по системным и стандартным функциям».

В таблице А–39 перечислены конфигурации, поддерживаемые модулем EM 277 PROFIBUS-DP. Заданная по умолчанию конфигурация для модуля EM 277 составляет два слова ввода и два слова вывода.

Таблица А–39. Конфигурационные возможности EM 277

Конфигурация	Входы в Master-устройство	Входы из Master-устройства	Целостность данных
1	1 слово	1 слово	Целостность слов
2	2 слова	2 слова	
3	4 слова	4 слова	
4	8 слов	8 слов	
5	16 слов	16 слов	
6	32 слова	32 слова	
7	8 слов	2 слова	
8	16 слов	4 слова	
9	32 слова	8 слов	
10	2 слова	8 слов	
11	4 слова	16 слов	
12	8 слов	32 слова	
13	2 байта	2 байта	Целостность байтов
14	8 байтов	8 байтов	
15	32 байта	32 байта	
16	64 байта	64 байта	
17	4 байта	4 байта	Целостность буфера
18	8 байтов	8 байтов	
19	12 байтов	12 байтов	
20	16 байтов	16 байтов	

Местоположение буферов ввода и вывода можно конфигурировать в любом месте V-памяти CPU S7-200. Заданный по умолчанию адрес буферов ввода и вывода равен VB0. Местоположение буферов ввода и вывода – это часть информации о назначении параметров, которую master записывает в CPU S7-200. Master-устройство нужно конфигурировать так, чтобы оно распознавало свои slave-устройства и записывало требуемые параметры и конфигурацию входов-выходов в каждое из своих slave-устройств.

Для конфигурирования master-устройства DP используются следующие инструментальные средства:

- Для master-устройств SIMATIC S5 используйте программное обеспечение Windows COM PROFIBUS
- Для master-устройств SIMATIC S7 используйте программное обеспечение для программирования STEP 7
- Для master-устройств SIMATIC 505 используйте COM PROFIBUS и либо TISOFT2, либо SoftShop

Для получения подробной информации об использовании этих пакетов программ для конфигурирования и программирования, обратитесь к руководствам по этим устройствам. Для получения подробной информации о сети PROFIBUS и ее компонентах обратитесь к Системному руководству по децентрализованной периферии ET 200.

Целостность данных



- Целостность буферов следует использовать, когда значения данных являются двойными словами или значениями с плавающей точкой или когда группа значений полностью относится к одному вычислению или объекту.

Целостность данных устанавливается как часть конфигурации входов-выходов в master-устройстве. Выбранный вид целостности данных записывается в slave-устройство как часть инициализации slave-устройства. И DP-master, и DP-slave используют выбранную целостность данных для обеспечения того, что значения данных (байты, слова или буфера) передаются внутри master-устройства или slave-устройства без прерываний. Различные типы целостности показаны на рис. А–26.

Указания к программе пользователя

Если модуль EM 277 PROFIBUS-DP был успешно конфигурирован master-устройством DP, то EM 277 и DP-master входят в режим обмена данными. В режиме обмена данными master записывает выходные данные в модуль EM 277 PROFIBUS-DP, затем модуль EM 277 отвечает, используя самые последние входные данные CPU S7-200. Чтобы предоставлять master-устройству DP самые последние входные данные, модуль EM 277 непрерывно обновляет свои входы от CPU S7-200. Затем модуль передает в CPU S7-200 выходные данные. Выходные данные от master-устройства помещаются в память переменных (буфер вывода), начиная с адреса, который master предоставил во время инициализации. Входные данные для master-устройства берутся из ячеек памяти переменных (буфер ввода), непосредственно следующих за выходными данными.

Выходные данные от master-устройства должны пересылаться программой пользователя в CPU S7-200 из буфера вывода в области данных, где они должны использоваться. Аналогично, входные данные для master-устройства должны пересылаться из различных областей данных в буфер ввода для передачи master-устройству.

Выходные данные от master-устройства DP помещаются в память переменных сразу после выполнения части цикла сканирования, относящейся к программе пользователя. В это же самое время входные данные (для master-устройства) копируются из памяти переменных в EM 277 для передачи master-устройству.

Выходные данные от master-устройства записываются в память переменных только тогда, когда имеются в распоряжении новые данные от master-устройства.

Входные данные для master-устройства передаются master-устройству при следующем обмене данными с master-устройством.

Начальный адрес буферов данных в памяти переменных и размер этих буферов должен быть известен во время создания программы пользователя для CPU S7-200.

Информация о состоянии

Для каждого интеллектуального модуля выделено 50 байтов специальной памяти (SM) в зависимости от его физического размещения. Модуль обновляет адреса SM-памяти в соответствии с его расположением относительно CPU (с учетом других модулей). Если это первый модуль, то он обновляет SMB200 – SMB249. Если это второй модуль, то он обновляет SMB250 – SMB299 и т.д. См. таблицу A–40.

Таблица A–40. Байты специальной памяти SMB200 – SMB549

Байты специальной памяти SMB200 – SMB549						
Интеллектуальный модуль в слоте 0	Интеллектуальный модуль в слоте 1	Интеллектуальный модуль в слоте 2	Интеллектуальный модуль в слоте 3	Интеллектуальный модуль в слоте 4	Интеллектуальный модуль в слоте 5	Интеллектуальный модуль в слоте 6
SMB200 – SMB249	SMB250 – SMB299	SMB300 – SMB349	SMB350 – SMB399	SMB400 – SMB449	SMB450 – SMB499	SMB500 – SMB549

Эти адреса SM-памяти являются значениями по умолчанию, если связь с master-устройством DP не была установлена. После того как master записал параметры и конфигурацию входов-выходов в модуль EM 277 PROFIBUS-DP, эти адреса SM-памяти показывают конфигурацию, установленную master-устройством DP. Перед использованием информации из адресов SM-памяти, указанных в таблице A–41, или данных из буфера в памяти переменных вы должны проверить байт состояния протокола (например, SMB224 для слота 0), чтобы убедиться в том, что EM 277 в данный момент находится в режиме обмена данными с master-устройством.


	<p>Совет Вы не можете конфигурировать размеры и адреса буферов ввода-вывода EM 277 PROFIBUS-DP, выполняя запись в ячейки специальной памяти. Только DP-master может конфигурировать модуль EM 277 PROFIBUS-DP для режима DP.</p>
---	---

Таблица A–41. Байты специальной памяти для EM 277 PROFIBUS-DP

Интеллектуальный модуль в слоте 0	...	Интеллектуальный модуль в слоте 6	Описание																			
SMB200 – SMB215	...	SMB500 – SMB515	Имя модуля (16 символов ASCII) «EM277 ProfibusDP»																			
SMB216 – SMB219	...	SMB516 – SMB519	Номер версии программного обеспечения (4 символа ASCII) xxxx																			
SMW220	...	SMW520	Код ошибки 16#0000 Нет ошибки 16#0001 Нет питания у пользователя 16#0002 – 16#FFFF Резерв																			
SMB222	...	SMB522	Адрес станции slave-устройства DP, установленный переключателями адреса (0 – 99 десятич.)																			
SMB223	...	SMB523	Резерв																			
SMB224	...	SMB524	Байт состояния стандартного протокола DP <table style="margin-left: 40px;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">0 0 0 0 0 0 S1</td> <td style="text-align: center;">S0</td> </tr> </table> <table style="margin-left: 40px;"> <tr> <td>S1</td> <td>S0</td> <td>Описание байта состояния стандартного протокола DP</td> </tr> <tr> <td>0</td> <td>0</td> <td>После включения обмен данными DP не иницирован</td> </tr> <tr> <td>0</td> <td>1</td> <td>Обнаружена ошибка конфигурирования или параметризации</td> </tr> <tr> <td>1</td> <td>0</td> <td>Имеет место режим обмена данными</td> </tr> <tr> <td>1</td> <td>1</td> <td>Потерян режим обмена данными</td> </tr> </table>	MSB	LSB	0 0 0 0 0 0 S1	S0	S1	S0	Описание байта состояния стандартного протокола DP	0	0	После включения обмен данными DP не иницирован	0	1	Обнаружена ошибка конфигурирования или параметризации	1	0	Имеет место режим обмена данными	1	1	Потерян режим обмена данными
MSB	LSB																					
0 0 0 0 0 0 S1	S0																					
S1	S0	Описание байта состояния стандартного протокола DP																				
0	0	После включения обмен данными DP не иницирован																				
0	1	Обнаружена ошибка конфигурирования или параметризации																				
1	0	Имеет место режим обмена данными																				
1	1	Потерян режим обмена данными																				
SMB225	...	SMB525	Стандартный протокол DP – адрес master-устройства, владеющего slave-устройством (от 0 до 126)																			
SMW226	...	SMW526	Стандартный протокол DP – адрес в памяти переменных буфера вывода в виде смещения от VB0.																			
SMB228	...	SMB528	Стандартный протокол DP – количество байтов выходных данных																			
SMB229	...	SMB529	Стандартный протокол DP – количество байтов входных данных																			
SMB230 – SMB249	...	SMB530 – SMB549	Резерв – стираются при запуске																			

Примечание: Ячейки специальной (SM) памяти обновляются каждый раз, когда slave-модуль DP принимает информацию о конфигурации или параметризации. Эти ячейки обновляются даже тогда, когда обнаруживается ошибка конфигурации или параметризации. Эти ячейки очищаются при каждом запуске.

Светодиодные индикаторы состояния EM 277 PROFIBUS-DP

Модуль EM 277 PROFIBUS-DP имеет четыре светодиода состояния на лицевой панели для отображения рабочего состояния порта DP:

- После включения CPU S7-200 светодиод DX MODE [режим DX] остается выключенным до тех пор, пока не предпринимается попытка связи на основе DP.
- Если связь на основе DP успешно инициализирована (модуль EM 277 PROFIBUS-DP вошел в режим обмена данными с master-устройством), то светодиод DX MODE загорается зеленым светом и остается включенным до тех пор, пока не произойдет выход из режима обмена данными.
- Если связь на основе DP теряется, что заставляет модуль EM 277 выйти из режима обмена данными, то светодиод DX MODE выключается, и включается светодиод DP ERROR [ошибка DP] красного цвета. Это состояние сохраняется до тех пор, пока не будет выключен CPU S7-200 или возобновлен обмен данными.
- Если имеется ошибка в информации о конфигурации входов-выходов или параметрах, которую DP-master записывает в модуль EM 277, то светодиод DP ERROR красного цвета мигает.
- Если напряжение 24 В постоянного тока не снабжается питанием, то светодиод POWER [питание] будет выключен.

В таблице А-42 приведены сигналы состояния, отображаемые светодиодами состояния EM 277.

Таблица А-42. Светодиоды состояния модуля EM 277 PROFIBUS-DP

Светодиод	Выключен	Красный	Мигающий красный	Зеленый
CPU FAULT [Отказ CPU]	Модуль исправен	Внутренний отказ модуля	--	--
POWER [Питание]	Нет питания 24 В пост. тока	--	--	Питание 24 В пост. тока исправно
DP ERROR [Ошибка DP]	Нет ошибок	Произошел выход из режима обмена данными	Ошибка параметризации/конфигурации	--
DX MODE [Режим обмена данными]	Не находится в режиме обмена данными	--	--	Находится в режиме обмена данными

Примечание: Когда модуль EM 277 PROFIBUS-DP используется исключительно как slave-устройство MPI, включается только зеленый светодиод Power [питание].

Дополнительные особенности конфигурации

Модуль EM 277 PROFIBUS-DP может использоваться как коммуникационный интерфейс с другими master-устройствами MPI, независимо от того, используется ли он в качестве slave-устройства PROFIBUS-DP. Этот модуль может обеспечивать подключение S7-300/400 к S7-200 с помощью функций XGET/XPUT в S7-300/400. Для обмена данными с S7-200 через модуль EM 277 PROFIBUS-DP могут применяться STEP 7-Micro/WIN и сетевая плата (например, CP5611), использующая набор параметров MPI или PROFIBUS, устройство OP или TD 200 (версии 2.0 или выше, заказной номер 6ES7 272-0AA20-0YA0).

Кроме master-устройства DP, к модулю EM 277 PROFIBUS-DP может быть сделано не более шести подключений (шесть устройств). Одно соединение резервируется для устройства программирования (PG) и одно резервируется для панели оператора (OP). Остальные четыре соединения могут использоваться любым master-устройством MPI. Чтобы модуль EM 277 PROFIBUS-DP обменивался данными с несколькими master-устройствами, все master-устройства должны работать с одинаковой скоростью передачи. Одна из возможных конфигураций сети представлена на рис. А-27.

Когда модуль EM 277 PROFIBUS-DP используется для обмена данными на основе MPI, MPI-master должен использовать адрес станции модуля для всех сообщений, передаваемых в S7-200, к которому подключен модуль. Сообщения MPI, передаваемые модулю EM 277 PROFIBUS-DP, пропускаются в S7-200.

Модуль EM 277 PROFIBUS-DP является slave-модулем и не может использоваться для обмена данными между ПЛК S7-200 с применением функций NETR и NETW. Модуль EM 277 PROFIBUS-DP не может использоваться для обмена данными на основе протокола свободно программируемой связи Freepoint.

Рис. А–28. Распечатка GSD-файла для модуля EM 277 PROFIBUS

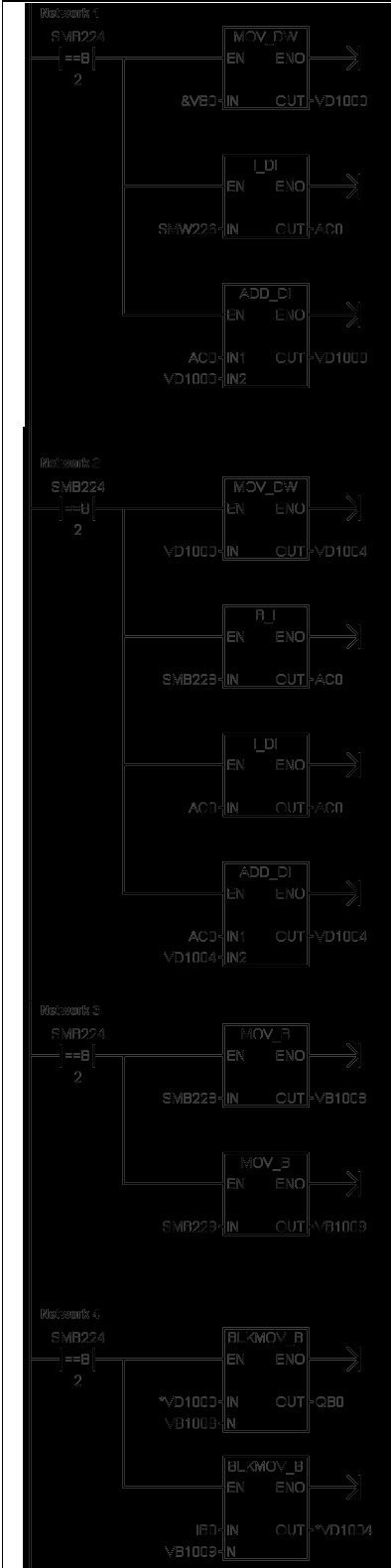
Пример программы для обмена данными DP с CPU

Ниже приведен пример программы в виде списка команд (STL) для модуля PROFIBUS-DP, находящегося в слоте 0, для CPU, использующего информацию порта DP в SM-памяти. Эта программа определяет адреса буферов DP через SMW226, а размеры буферов из SMB228 и SMB229. Эта информация используется в программе для копирования данных из буфера вывода DP в регистр выходов образа процесса CPU. Точно так же данные из регистра входов образа процесса в CPU копируются в буфер ввода в памяти переменных.

В следующем примере программы для модуля, находящегося в слоте 0, конфигурационные данные DP, находящиеся в области специальной (SM) памяти, дают конфигурацию slave-устройства DP. Программа использует следующие данные:

SMW220	Состояние ошибки модуля DP
SMB224	Состояние DP
SMB225	Адрес master-устройства
SMW226	Смещение в памяти переменных для выходов
SMB228	Количество байтов выходных данных
SMB229	Количество байтов входных данных
VD1000	Указатель на выходные данные
VD1004	Указатель на входные данные

Пример обмена данными DP с CPU



Network 1 //Вычислить указатель на выходные данные.
 //При нахождении в режиме обмена данными:
 //1. Буфер вывода – это смещение от VB0
 //2. Преобразовать смещение в V-памяти в
 // двойное целое число
 //3. Чтобы получить указатель на выходные
 // данные, прибавить к адресу.

LDB= SMB224, 2
 MOVD &VB0, VD1000
 ITD SMW226, AC0
 +D AC0, VD1000

Network 2 //Вычислить указатель на входные данные.
 //При нахождении в режиме обмена данными:
 //1. Скопировать указатель на выходные
 // данные
 //2. Получить количество выходных байтов
 //3. Чтобы получить указатель на начало
 // входных данных, прибавить к указателю
 // на выходные данные.

LDB= SMB224, 2
 MOVD VD1000, VD1004
 BTI SMB228, AC0
 ITD AC0, AC0
 +D AC0, VD1004

Network 3 //Установить количество данных, подлежащих
 //копированию.
 //При нахождении в режиме обмена данными:
 //1. Получить количество выходных байтов,
 // подлежащих копированию
 //2. Получить количество входных байтов,
 // подлежащих копированию

LDB= SMB224, 2
 MOVB SMB228, VB1008
 MOVB SMB229, VB1009

Network 4 //Передать выходы master-устройства на
 //выходы CPU. Скопировать входы CPU во
 // входы master- устройства. При нахождении в /
 / режиме обмена данными:
 //1. Скопировать выходы master-устройства
 // на выходы CPU
 //2. Скопировать входы CPU во входы
 // master- устройства.

LDB= SMB224, 2
 BMB *VD1000, QB0, VB1008
 BMB IB0, *VD1004, VB1009

Технические данные модемного модуля EM 241

Таблица А–43. Номер для заказа модемного модуля EM 241

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6ES7 241-1AA22-0XA0	Модемный модуль EM 241	-	8 ¹	Нет

¹ Восемь выходов Q используются как логические управляющие элементы функциями модема, а не для непосредственного управления внешними сигналами.

Таблица А–44. Общие технические данные модемного модуля EM 241

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
6ES7 241-1AA22-0XA0	Модемный модуль EM 241	71,2 x 80 x 62	190 г	2,1 Вт	80 мА	70 мА

Таблица А–44. Технические данные модемного модуля EM 241

Общие данные	6ES7 241-1AA22-0XA0
Телефонное соединение	
Электрическая развязка (телефонной линии с логикой и полевым напряжением)	1500 В перем. тока (гальваническая развязка)
Физическое присоединение	RJ11 (6 позиций, 4 провода)
Стандарты модема	Bell 103, Bell 212, V.21, V.22, V.22 bis, V.23c, V.32, V.32 bis, V.34 (по умолчанию)
Защита от несанкционированного доступа	Пароль Обратный вызов
Набор номера	Импульсный или тональный
Протоколы передачи сообщений	Цифровой TAP (алфавитно-цифровой) Команды UCP 1, 30, 51
Промышленные протоколы	Modbus PPI
Потребности входов в питании 24 В пост. тока	
Диапазон напряжений	от 20,4 до 28,8 В пост. тока
Электрическая развязка (полевое напряжение с логикой)	500 В перем. тока в течение 1 минуты

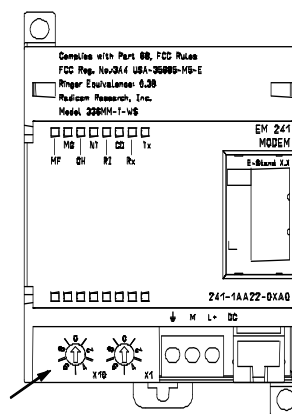
Модемный модуль EM 241 заменяет функцию внешнего модема, подключенного к коммуникационному порту CPU. Если в вашей системе S7–200 установлен модуль EM 241, то для удаленного обмена данными с вашим CPU необходим только персональный компьютер с внешним модемом и STEP 7-Micro/WIN.

Для получения информации о настройке обратитесь к главе 7 "Обмен данными в сети". Информацию о программировании и расширенных функциях этого модуля вы найдете в главе 10 "Создание программы для модемного модуля".

Для настройки модемного модуля EM 241 можно использовать Мастер расширения функций модема (Modem Expansion wizard) STEP 7-Micro/WIN. Дополнительную информацию о Мастере расширения функций модема вы найдете в главе 10.



Модем



Переключатель кода страны

Рис. А–29. Модемный модуль EM 241

CPU S7–200, поддерживающие интеллектуальные модули

Модемный модуль EM 241 – это интеллектуальный модуль расширения, спроектированный для работы с CPU S7–200, показанными в таблице А–46.

Таблица А–46. Совместимость модемного модуля EM 241 с CPU S7–200

CPU	Описание
CPU 222, начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224, начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP, начиная с версии 2.0	CPU 224XP DC/DC/DC и CPU 224XP DC/DC/Relay
CPU 226, начиная с версии 1.00	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

Монтаж EM 241

Для установки EM 241 действуйте следующим образом:

1. Защелкните EM 241 на стандартной профильной шине и вставьте плоский кабель.
2. Подключите питание 24 В пост. тока от источника питания датчиков CPU или внешнего источника и соедините клемму заземления с землей вашей системы.
3. Вставьте телефонный провод в гнездо RJ11.
4. Установите переключатели кода страны в соответствии с таблицей А–47. Чтобы код страны был считан правильно, переключатели следует установить перед подачей питания на CPU.
5. Включите CPU. Должен загореться индикатор MG (Module Good [Модуль в порядке]).

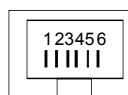
Ваш EM 241 теперь готов к обмену данными.

Таблица А–47. Коды стран, поддерживаемые EM 241

Код	Страна	Стандарт Telecom
01	Австрия	CTR21
02	Бельгия	CTR21
05	Канада	IC CS03
08	Дания	CTR21
09	Финляндия	CTR21
10	Франция	CTR21
11	Германия	CTR21
12	Греция	CTR21
16	Ирландия	CTR21
18	Италия	CTR21
22	Люксембург	CTR21
25	Нидерланды	CTR21
27	Норвегия	CTR21
30	Португалия	CTR21
34	Испания	CTR21
35	Швеция	CTR21
36	Швейцария	CTR21
38	Великобритания	CTR21
39	США	FCC Part 68

Штекер RJ11

На рис. А–30 показан штекер RJ11. Вы можете использовать адаптеры для других стандартных телефонных соединений. Подробную информацию вы найдете в документации к своему адаптерному соединению.



Контакт Описание

3 Звонок
4 Контакт

Допустимо обратное присоединение

Рис. А–30. Вид штекера RJ11

Осторожно

Удары молнии и другие неожиданные высокие напряжения в телефонной линии могут повредить ваш модемный модуль EM 241.

Используйте имеющееся в продаже устройство защиты от перенапряжений в телефонной линии, которые обычно продаются для защиты модемов персональных компьютеров. Эти устройства могут быть повреждены при защите вашего модемного модуля EM 241. Выбирайте устройство защиты от перенапряжений с индикатором, показывающим, работает оно или нет.

Регулярно проверяйте свое устройство защиты от перенапряжений, чтобы убедиться, что ваш модемный модуль EM 241 продолжает оставаться защищенным.

Технические данные модуля позиционирования EM 253

Таблица А–48. Номер для заказа модуля позиционирования EM 253

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6ES7 253-1AA22-0XA0	Модуль позиционирования EM 253	-	8 ¹	Да

¹ Восемь выходов Q используются как логические управляющие элементы функций перемещения, а не для непосредственного управления внешними сигналами.

Таблица А–49. Общие технические данные модуля позиционирования EM 253

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В	+24 В
6ES7 253-1AA22-0XA0	Модуль позиционирования EM 253	71,2 x 80 x 62	0,190 кг	2,5 Вт	190 мА	См. ниже

Таблица А–50. Технические данные модуля позиционирования EM 253

Общие данные	6ES7 253-1AA22-0XA0
Входные характеристики	
Количество входов	5 входов
Тип входа	Обычная/обратная полярность (IEC Тип 1, обычная полярность, кроме ZP)
Входное напряжение Макс. длительно допустимое STP, RPS, LMT+, LMT- ZP Толчок (все входы) Номинальное значение STP, RPS, LMT+, LMT- ZP Логическая «1» (минимум) STP, RPS, LMT+, LMT- ZP Логический «0» (максимум) STP, RPS, LMT+, LMT- ZP	30 В пост. тока макс. 30 В пост. тока при 20 мА 35 В пост. тока в течение 0,5 с 24 В пост. тока при 4 мА, номинал 24 В пост. тока при 15 мА, номинал 15 В пост. тока при 2,5 мА, минимум 3 В пост. тока при 8,0 мА, минимум 5 В пост. тока при 1 мА, максимум 1 В пост. тока при 1 мА, максимум
Электрическая развязка (полевые устройства с логикой) Гальваническая развязка Потенциально развязанные группы	500 В перем. тока в течение 1 минуты 1 вход для STP, RPS и ZP 2 входа для LMT+ и LMT-
Времена входной задержки STP, RPS, LMT+, LMT- ZP (вычисляемая ширина импульса)	от 0,2 до 12,8 мс, выбирается пользователем мин. 2 мкс
Присоединение 2-проводного реле близости (Веро) Допустимый ток утечки	1 мА, максимум
Длина кабеля неэкранированный STP, RPS, LMT+, LMT- ZP экранированный STP, RPS, LMT+, LMT- ZP	30 метров не рекомендуется 100 метров 10 метров
Количество одновременно включенных входов 55 °С	5

Таблица А-50. Технические данные модуля позиционирования EM 253, продолжение

Общие данные	6ES7 253-1AA22-0XA0							
Выходные характеристики								
Количество встроенных выходов Тип выхода P0+, P0-, P1+, P1- P0, P1, DIS, CLR	6 выходов (4 сигнала) Драйвер RS422/485 транзисторные выходы с открытым стоком							
Выходное напряжение Драйверы P0, P1, RS-422, дифференциальное выходное напряжение Разомкнутая цепь тока В диоде оптрона с последовательным сопротивлением 200 Ом Нагрузка 100 Ом Нагрузка 54 Ома P0, P1, DIS, CLR с открытым стоком рекомендуемое напряжение, разомкнутая цепь тока допустимое напряжение, разомкнутая цепь тока Отвод тока Сопротивление во включенном состоянии Ток утечки в выключенном состоянии, 30 В пост. тока Внутренний корректирующий резистор, выходная утечка в T1	тип. 3,5 В мин. 2,8 В мин. 1,5 В мин. 1,0 В 5 В пост. тока, из модуля 30 В пост. тока ¹ макс. 50 мА макс. 15 Ом макс. 10 мкА 3,3 кОм ²							
Выходной ток Количество групп выходов Число одновременно включенных выходов Ток утечки на выход P0, P1, DIS, CLR Защита от перегрузки	1 Все при 55° С (горизонтальный монтаж), все при 45° С (вертикальный монтаж) макс. 10 мкА Нет							
Электрическая развязка (полевых устройств с логикой) Гальваническая развязка	500 В перем. тока в течение 1 минуты							
Задержка вывода DIS, CLR: выкл.-вкл. / вкл.-выкл.	макс. 30 мкс							
Искажение импульса P0, P1, выходы, драйверы RS-422, внешняя нагрузка 100 Ом Выходы P0, P1, транзисторные с открытым стоком, 5 В / внешняя нагрузка 470 Ом	макс. 75 нс макс. 300 нс							
Частота переключения P0+, P0-, P1+, P1-, P0 и P1	200 кГц							
Длина кабеля не экранированный экранированный	не рекомендуется 10 метров							
Блок питания								
Питающее напряжение L+	от 11 до 30 В пост. тока (класс 2, при ограничении мощности или напряжение питания датчиков от ПЛК)							
Питание логики Питающий ток L+ при нагрузке 5 В пост. тока Ток нагрузки 0 мА (без нагрузки) 200 мА (при номинальной нагрузке)	макс. +5 В пост. тока +/- 10%, 200 мА <table border="1"> <thead> <tr> <th>Вход 12 В пост. тока</th> <th>Вход 24 В пост. тока</th> </tr> </thead> <tbody> <tr> <td>120 мА</td> <td>70 мА</td> </tr> <tr> <td>300 мА</td> <td>130 мА</td> </tr> </tbody> </table>		Вход 12 В пост. тока	Вход 24 В пост. тока	120 мА	70 мА	300 мА	130 мА
Вход 12 В пост. тока	Вход 24 В пост. тока							
120 мА	70 мА							
300 мА	130 мА							
Электрическая развязка напряжения на L+ с логикой напряжения на L+ с входами напряжения на L+ с выходами	500 В перем. тока в течение 1 минуты 500 В перем. тока в течение 1 минуты Нет							
Защита от обратной полярности	Вход L+ и выход +5 В защищены диодами. Подача на клемму M положительного напряжения относительно выходных клемм может привести к появлению повреждающего тока.							

¹ Работа транзисторных выходов с открытым стоком при напряжении выше 5 В пост. тока может увеличить излучение радиопомех свыше допустимых границ. Поэтому для вашей системы или проводки могут потребоваться меры защиты от распространения радиочастот.

² В зависимости от вашего приемника импульсов и кабеля дополнительный внешний ограничивающий резистор может улучшить качество импульсного сигнала и помехоустойчивость.

CPU S7–200, поддерживающие интеллектуальные модули

Модуль позиционирования EM 253 – это интеллектуальный модуль расширения, спроектированный для работы с CPU S7–200, показанными в таблице А–51.

Таблица А–51. Совместимость модуля позиционирования EM 253 с CPU S7–200

CPU	Описание
CPU 222, начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224, начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP, начиная с версии 2.0	CPU 224XP DC/DC/DC и CPU 224XP DC/DC/Relay
CPU 226, начиная с версии 1.10	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

Светодиоды состояния модуля позиционирования EM 253

Светодиоды состояния модуля позиционирования приведены в таблице А–52.

Таблица А–52. Светодиоды состояния модуля позиционирования

Встроенный вход или выход	Светодиод	Цвет	Функциональное описание
-	MF	Красный	Горит, когда модуль обнаруживает фатальную ошибку
-	MG	Зеленый	Горит, когда модуль исправен, и мигает с частотой 1 Гц, когда обнаружена конфигурационная ошибка
-	PWR	Зеленый	Горит, когда напряжение 24 В пост. тока подается на клеммы L+ и M модуля
Вход	STP	Зеленый	Горит, когда включен вход останова
Вход	RPS	Зеленый	Горит, когда включен вход переключателя опорной точки
Вход	ZP	Зеленый	Горит, когда включен вход нулевого импульса
Вход	LMT-	Зеленый	Горит, когда включен вход для отрицательного граничного значения
Вход	LMT +	Зеленый	Горит, когда включен вход для положительного граничного значения
Выход	P0	Зеленый	Горит, когда выход P0 находится в импульсном режиме
Выход	P1	Зеленый	Горит, когда выход P1 находится в импульсном режиме или этот выход указывает на положительное перемещение
Выход	DIS	Зеленый	Горит, когда выход DIS активен
Выход	CLR	Зеленый	Горит, когда активен выход для стирания счетчика отклонений

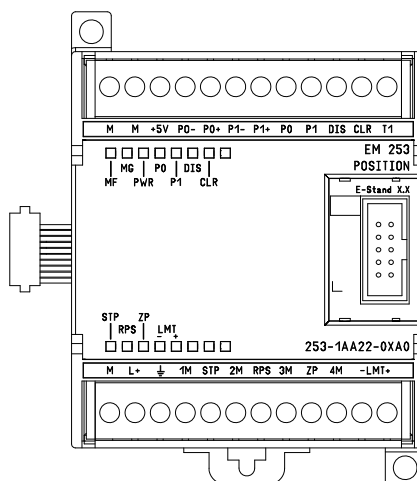


Рис. А–31. Модуль позиционирования EM 253

Схемы соединений

На следующих рисунках клеммы расположены не по порядку. Расположение клемм см. на рис. А-31.

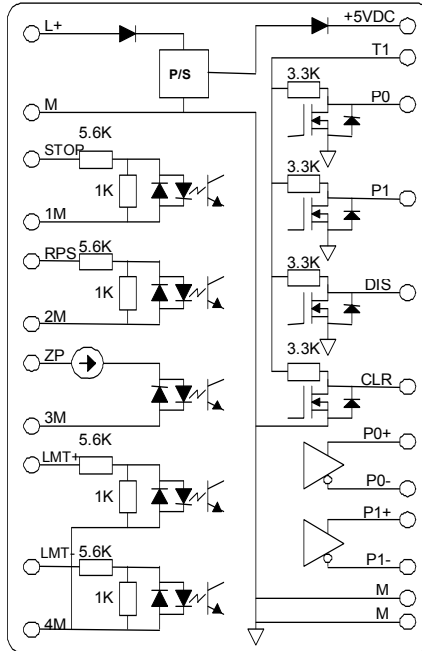
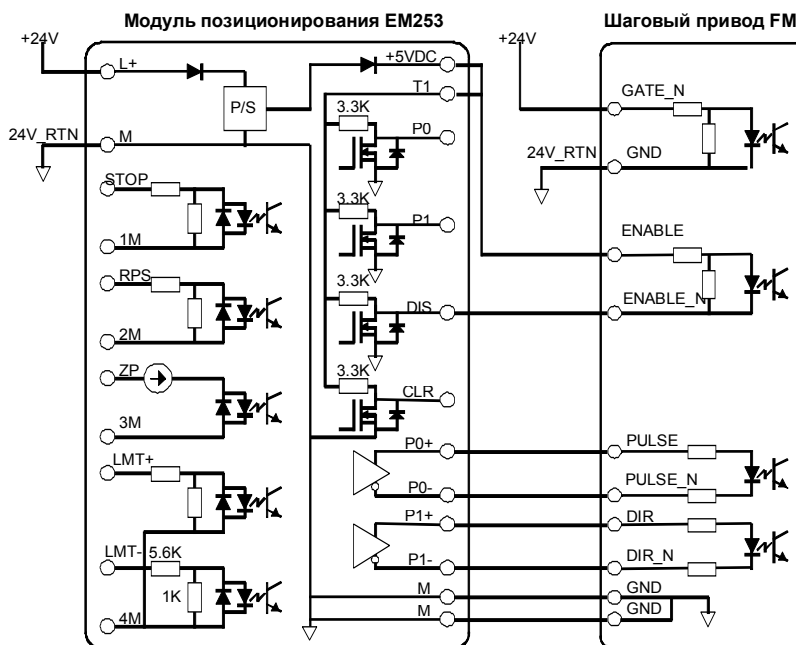


Рис. А-32. Внутренняя схема входов и выходов модуля позиционирования EM 253



Клеммы расположены не по порядку. Расположение клемм см. на рис. А-31.

Рис. А-33. Подключение модуля позиционирования EM 253 к шаговому приводу SIMATIC FM

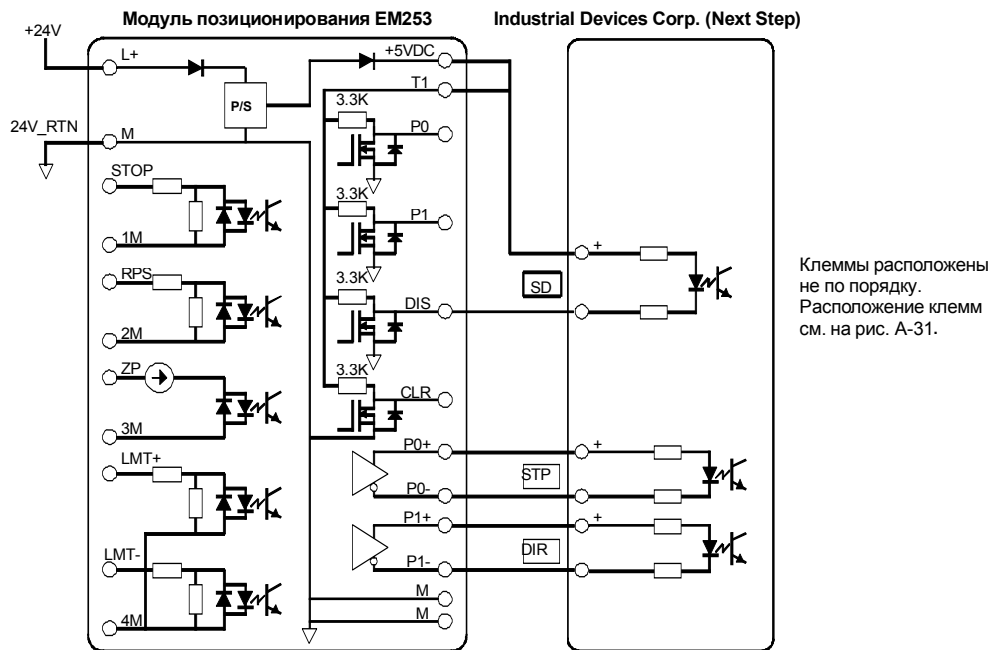


Рис. А-34. Подключение модуля позиционирования EM 253 к Industrial Devices Corp. (Next Step)

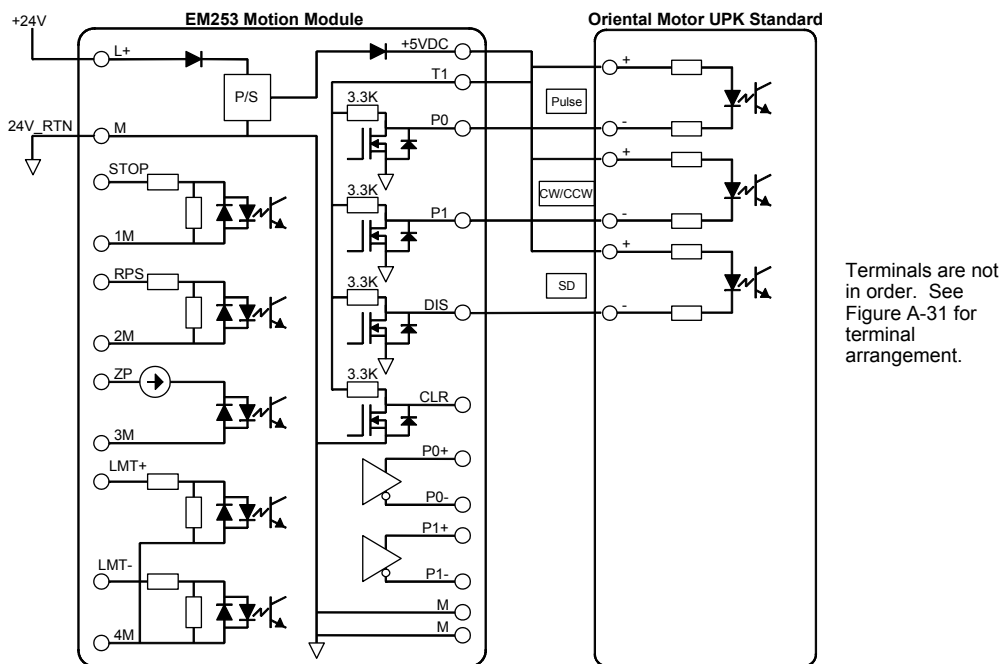
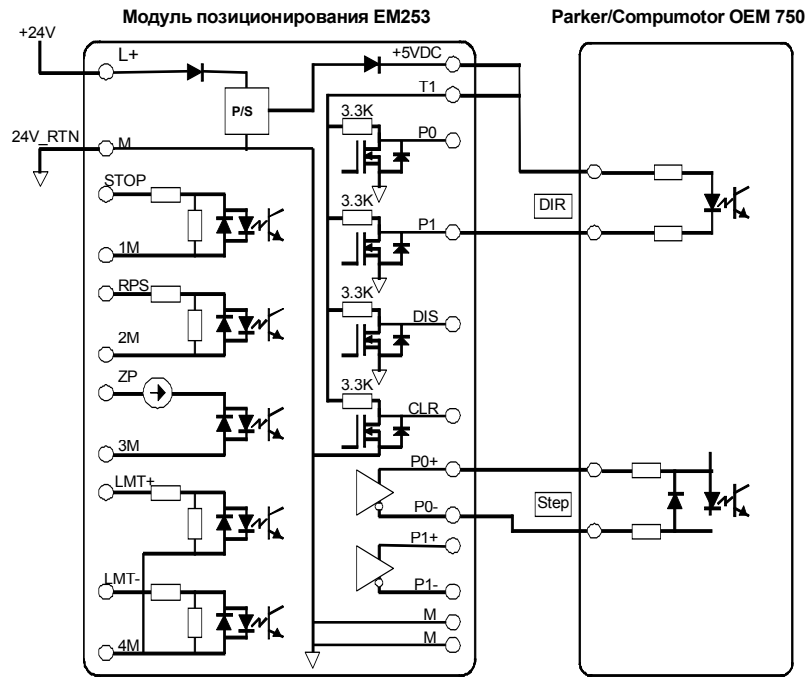


Рис. А-35. Подключение модуля позиционирования EM 253 к Oriental Motor UPK Standard



Клеммы расположены не по порядку. Расположение клемм см. на рис. А-31.

Рис. А-36. Подключение модуля позиционирования EM 253 к Parker/Compumotor OEM 750

Технические данные модуля Ethernet (CP 243-1)

Таблица А-53. Номер для заказа модуля Ethernet (CP 243-1)

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6GK7 243-1EX00-0XE0	Модуль Ethernet (CP 243-1)	-	8 ¹	Нет

¹ Восемь выходов Q модуля расширения используются как логические элементы управления функции Ethernet и не управляют непосредственно никакими внешними сигналами.

Таблица А-54. Общие технические данные модуля Ethernet (CP 243-1)

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В пост.тока	+24 В пост. тока
6GK7 243-1EX00-0XE0	Модуль Ethernet (CP 243-1)	71,2 x 80 x 62	ок. 150 г	1,75 Вт	55 мА	60 мА

Таблица А-55. Технические данные модуля Ethernet (CP 243-1)

Общие данные	6GK7 243-1EX00-0XE0
Скорость передачи	10 Мбит/с и 100 Мбит/с
Размер флэш-памяти	1 Мбайт
Размер синхронного динамического ОЗУ	8 Мбайт
Интерфейс Соединение с Industrial Ethernet (10/100 Мбит/с)	8-контактная розетка RJ45
Входное напряжение	от 20,4 до 28,8 В пост. тока
Максимальное количество соединений	Максимум 8 S7-соединений (XPUT/XGET и READ/WRITE) плюс 1 соединение со STEP 7-Micro/WIN на модуль Ethernet (CP 243-1) ²
Время пуска и перезапуска после сброса	ок. 10 секунд
Количество данных пользователя	Как клиент: до 212 байт для XPUT/XGET Как сервер: до 222 байт для XGET или READ до 212 байт для XPUT или WRITE

² К каждому CPU S7-200 можно подключить только один модуль Ethernet (CP 243-1).

Модуль Ethernet (CP 243-1) – это коммуникационный процессор, используемый для соединения системы S7-200 с сетью Industrial Ethernet (IE). S7-200 может дистанционно настраиваться, программироваться и диагностироваться через Ethernet с помощью STEP 7 Micro/WIN. S7-200 может обмениваться данными с другими контроллерами S7-200, S7-300 и S7-400 через Ethernet. Он может также обмениваться данными с сервером OPC.

Industrial Ethernet спроектирован для работы в промышленности. Он может применяться с шумящей технологией ITP (Industrial Twisted Pair - промышленная витая пара) или с промышленной стандартной технологией TP (Twisted Pair - витая пара). Использование Industrial Ethernet может обеспечить реализацию широкого диапазона конкретных приложений, например, переключателей, скоростного резервирования, быстрого соединения и резервируемых сетей. С помощью модуля Ethernet (CP 243-1) ПЛК S7-200 делается совместимым с широким спектром существующих продуктов, поддерживающих Ethernet.

CPU S7–200, поддерживающие интеллектуальные модули

Модуль Ethernet (CP 243-1) – это интеллектуальный модуль расширения, спроектированный для работы с CPU S7-200, представленными в таблице А–56.

Таблица А–56. Совместимость модуля Ethernet (CP 243-1) с CPU S7–200

CPU	Описание
CPU 222, начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224, начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP, начиная с версии 2.00	CPU 224XP DC/DC/DC и CPU 224XP AC/DC/Relay
CPU 226, начиная с версии 1.10	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

Модуль Ethernet (CP 243-1) поставляется с заранее установленным, уникальным мировым адресом MAC, который не может быть изменен.

Функции

Модуль Ethernet (CP 243-1) независимо обрабатывает передачу данных через Industrial Ethernet.

- Обмен данными основан на TCP/IP
- Для обмена данными между CPU S7–200 и другими системами управления S7 или ПК через Ethernet имеются в распоряжении коммуникационные услуги типа клиент-сервер. Одновременно может эксплуатироваться до восьми соединений.
- Реализация приложений для ПК возможна через интеграцию с сервером S7–OPC
- Модуль Ethernet (CP 243-1) делает возможным прямой доступ программного обеспечения S7–200 STEP 7-Micro/WIN к S7–200 через Ethernet



Ethernet

Настройка

Чтобы настроить модуль Ethernet (CP 243-1) для подключения ПЛК S7–200 к сети Ethernet, можно использовать Мастер Ethernet пакета STEP 7-Micro/WIN. Мастер Ethernet помогает вам определить параметры для модуля Ethernet (CP 243-1), а затем помещает команды настройки в папку вашего проекта. Для запуска Мастера Ethernet выберите команду меню **Tools > Ethernet Wizard [Инструментальные средства > Мастер Ethernet]**. Мастер использует следующую информацию: IP-адрес, маску подсети, адрес шлюза и вид коммуникационного соединения.

Присоединительные элементы

Модуль Ethernet (CP 243-1) имеет следующие присоединительные элементы. Они находятся под передней дверцей.

- Клеммный блок для подключения питающего напряжения 24 В пост. тока и заземления
- 8–контактная розетка RJ45 для подключения к Ethernet
- Штепсельный разъем для шины ввода-вывода
- Встроенный плоский кабель с розеткой для шины ввода-вывода



Рис. А–37. Подключение модуля Ethernet (CP 243-1)

Дополнительная информация

Дополнительную информацию о модуле Ethernet (CP 243-1) вы найдете в техническом руководстве *SIMATIC NET CP 243-1 Communications Processor for Industrial Ethernet [Коммуникационный процессор SIMATIC NET CP 243-1 для Industrial Ethernet]*.

Технические данные модуля Internet (CP 243-1 IT)

Таблица А–57. Номер для заказа модуля Internet (CP 243-1 IT)

Номер для заказа	Модуль расширения	Выходы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6GK7 243-1GX00-OXE0	Модуль Internet (CP 243-1 IT)	-	8 ¹	Нет

¹ Восемь выходов Q модуля расширения используются как логические элементы управления функции IT и не управляют непосредственно никакими внешними сигналами.

Таблица А–58. Общие технические данные модуля Internet (CP 243-1 IT)

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 VDC	+24 VDC
6GK7 243-1GX00-OXE0	Модуль Internet (CP 243-1 IT)	71,2 x 80 x 62	ок. 150 г	1,75 Вт	55 mA	60 mA

Таблица А–59. Технические данные модуля Internet (CP 243-1 IT)

Общие данные	6GK7 243-1GX00-OXE0
Скорость передачи	10 Мбит/с и 100 Мбит/с
Размер флэш-памяти	8 Мбайт для программ ПЗУ модуля Internet (CP 243-1 IT), 8 Мбайт как ОЗУ для файловой системы
Размер синхронного динамического ОЗУ	16 Мбайт
Гарантированный срок службы флэш-памяти для файловой системы	1 миллион операций записи и чтения
Интерфейс Соединение с Industrial Ethernet (10/100 Мбит/с)	8-контактная розетка RJ45
Входное напряжение	от 20,4 до 28,8 В пост. тока
Максимальное количество соединений	Максимум 8 S7-соединений (XPUT/XGET и READ/WRITE) плюс 1 соединение со STEP 7-Micro/WIN на модуль Internet (CP 243-1 IT) ¹
Максимальное количество IT-соединений	1 для FTP-сервера 1 для FTP-клиента 1 для клиента e-mail 4 для HTTP-соединений
Время пуска и перезапуска после сброса	ок. 10 секунд
Количество данных пользователя	Клиент: до 212 байт для XPUT/XGET Сервер: до 222 байт для XGET или READ до 212 байт для XPUT или WRITE
Размер E-mail, максимум	1024 символа
Файловая система: Длина пути, включая размер файла и имена дискетов Длина имени файла Глубина вложения каталогов	макс. 254 символа макс. 99 символов макс. 49
Имеющиеся порты сервера: HTTP Командный канал FTP Каналы данных FTP для FTP-сервера Установка S7-соединений S7-сервер	80 21 310 от 0 до 3199 102 от 3000 до 3008

¹ К каждому CPU S7-200 можно подключить только один модуль Internet (CP 243-1 IT).

Модуль Internet (CP 243-1 IT) – это коммуникационный процессор, используемый для соединения системы S7–200 с сетью Industrial Ethernet (IE). S7–200 может дистанционно настраиваться, программироваться и диагностироваться через Ethernet с помощью STEP 7 Micro/WIN. S7–200 может обмениваться данными с другими контроллерами S7-200, S7–300 и S7–400 через Ethernet. Он может также обмениваться данными с сервером OPC.

Функции IT модуля Internet (CP 243-1 IT) образуют основу для контроля и, если необходимо, также манипулирования системами автоматизации с помощью WEB-браузера из ПК, подключенного к сети. Диагностические сообщения могут быть посланы из системы по электронной почте. С помощью функций IT легко обмениваться целыми файлами с другими компьютерами и системами управления.

Industrial Ethernet – это сеть для уровня управления процессами и уровня ячеек открытой коммуникационной системы SIMATIC NET. Физически Industrial Ethernet – это электрическая сеть связи, базирующаяся на экранированных коаксиальных линиях, витых кабельных парах и оптической сети из волоконно-оптических кабелей. Industrial Ethernet определена Международным стандартом IEEE 802.3.

CPU S7–200, поддерживающие интеллектуальные модули

Модуль Internet (CP 243-1 IT) – это интеллектуальный модуль расширения, спроектированный для работы с CPU S7-200, представленными в таблице А–46.

Таблица А–60. Совместимость модуля Internet (CP 243-1 IT) с CPU S7–200

CPU	Описание
CPU 222, начиная с версии 1.10	CPU 222 DC/DC/DC и CPU 222 AC/DC/Relay
CPU 224, начиная с версии 1.10	CPU 224 DC/DC/DC и CPU 224 AC/DC/Relay
CPU 224XP, начиная с версии 2.00	CPU 224XP DC/DC/DC и CPU 224XP AC/DC/Relay
CPU 226, начиная с версии 1.10	CPU 226 DC/DC/DC и CPU 226 AC/DC/Relay

Модуль Internet (CP 243-1 IT) обладает следующими характеристиками:

- Модуль Internet (CP 243-1 IT) полностью совместим с модулем Ethernet (CP 243-1). Программы пользователя, написанные для модуля Ethernet (CP 243-1), могут исполняться также и на модуле Internet (CP 243-1 IT).

Модуль Internet (CP 243-1 IT) поставляется с заранее установленным, уникальным мировым адресом MAC, который не может быть изменен.



Совет

К CPU S7–200 может быть подключен только один модуль Internet (CP 243-1 IT). Если подключено более одного модуля Internet (CP 243-1 IT), то CPU S7–200 может работать ненадлежащим образом.

Функции

Модуль Internet (CP 243-1 IT) предоставляет в распоряжение следующие функции:

- S7-связь основана на TCP/IP
- IT-связь
- Настройка
- Контроль времени
- Возможность адресации предустановленных адресов MAC (48-битовое значение)



Настройка

Чтобы настроить модуль Internet (CP 243-1 IT) для подключения ПЛК S7–200 к сети Ethernet/Internet, можно использовать Мастер Internet пакета STEP 7-Micro/WIN. Модуль Internet (CP 243-1 IT) имеет дополнительную функцию web-сервер, которая может быть настроена с помощью Мастера Internet Wizard. Для запуска Мастера Internet Wizard выберите команду меню **Tools > Internet Wizard [Инструментальные средства > Мастер Internet]**.

Присоединительные элементы

Модуль Internet (CP 243-1 IT) имеет следующие присоединительные элементы. Они находятся под передней дверцей.

- Клеммный блок для подключения питающего напряжения 24 В пост. тока и заземления
- 8–контактная розетка RJ45 для подключения к Ethernet
- Штепсельный разъем для шины ввода-вывода
- Встроенный плоский кабель с розеткой для шины ввода-вывода

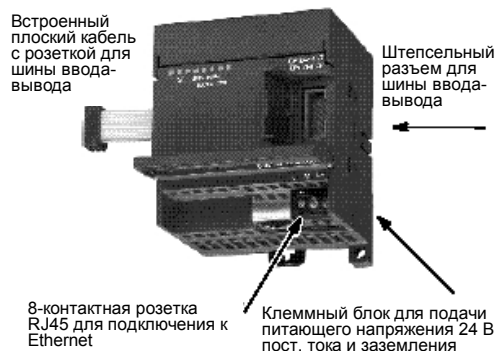


Рис. А–38. Подключение модуля Internet (CP 243-1 IT)

Дополнительная информация

Дополнительную информацию о модуле (CP 243-1 IT) Internet вы найдете в техническом руководстве *SIMATIC NET CP 243-1 IT Communications Processor for Industrial Ethernet and Information Technology [Коммуникационный процессор SIMATIC NET CP 243-1 IT для Industrial Ethernet и информационной технологии]*.

Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2)

Таблица А–61. Номер для заказа модуля AS-Interface (CP 243-2)

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	Съемный клеммный блок
6GK7 243-2AX01-0XA0	Модуль AS-Interface (CP 243-2)	8 цифровых и 8 аналоговых	8 цифровых и 8 аналоговых	Да

Таблица А–62. Общие технические данные модуля AS-Interface (CP 243-2)

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока +5 В из AS-Interface	
6GK7 243-2AX01-0XA0	Модуль AS-Interface (CP 243-2)	71 x 80 x 62	ок. 250 г	3,7 Вт	220 мА	100 мА

Таблица А–63. Технические данные модуля AS-Interface (CP 243-2)

Общие данные	6GK7 243-2AX01-0XA0
Время цикла	5 мс с 31 slave-устройством 10 мс с 62 slave-устройствами AS-I с использованием режима расширенной адресации
Настройка	Кнопка для настройки на передней панели или использование команды для общего конфигурирования (см. описание команд AS-I в руководстве CP 243-2 AS-I Interface Master)
Поддерживаемые профили master-устройства AS-I	M1e
Присоединение к кабелю AS-I	Через клеммный блок S7-200. Допустимая токовая нагрузка из клемм 1 – 3 или 2 – 4 не более 3 А.
Диапазон адресов	Один цифровой модуль с 8 цифровыми входами и 8 цифровыми выходами и один аналоговый модуль с 8 аналоговыми входами и 8 аналоговыми выходами

Характеристики

Вы можете одновременно эксплуатировать на S7-200 до двух модулей AS-Interface, существенно увеличив количество имеющихся в распоряжении цифровых и аналоговых входов и выходов (максимум 124 цифровых входа/124 цифровых выхода на AS-Interface на каждый CP). Времена настройки сокращаются благодаря возможности конфигурирования простым нажатием кнопки. Светодиоды сокращают времена простоя при возникновении ошибок, отображая состояние CP и всех подсоединенных slave-устройств и контролируя основное напряжение модуля AS-Interface.

Модуль AS-Interface имеет следующие характеристики:

- Поддерживает аналоговые модули
- Поддерживает все функции master-устройств и допускает присоединение до 62 slave-устройств AS-Interface
- Светодиоды на передней панели отображают рабочие состояния и наличие присоединенных slave-устройств.
- Светодиоды на передней панели отображают ошибки (включая сбои напряжения, ошибки конфигурирования модуля AS-Interface)
- Две клеммы допускают прямое присоединение кабеля AS-Interface.
- Две кнопки для отображения информации о состоянии slave-устройств, для переключения режима работы и приема существующей конфигурации в качестве установленной (SET).



Для настройки модуля AS-Interface (CP 243-2) можно использовать Мастер AS-i пакета STEP 7-Micro/WIN. Мастер AS-Interface помогает вам использовать данные из сети AS-Interface в вашей конфигурации. Для запуска Мастера AS-Interface выберите команду меню **Tools > AS-i Wizard [Инструментальные средства > Мастер AS-i]**.

Режим

В образе процесса S7-200 модуль AS-Interface занимает один байт цифровых входов (байт состояния), один байт цифровых выходов (управляющий байт), 8 слов аналоговых входов и 8 слов аналоговых выходов. Модуль AS-Interface использует два места для логических модулей. Вы можете использовать байт состояния и управляющий байт для установки режима модуля AS-Interface через программу пользователя. В зависимости от этого режима AS-Interface сохраняет входные и выходные данные slave-устройства AS-Interface, диагностические значения или активизирует вызовы master-устройства (например, изменение адреса slave-устройства) в области аналоговых адресов S7-200.

Все присоединенные slave-устройства AS-Interface могут быть настроены простым нажатием на кнопку. Дальнейшая настройка CP не требуется.

Осторожно

При использовании модуля AS-Interface необходимо заблокировать аналоговую фильтрацию в CPU.

Если аналоговая фильтрация в CPU не заблокирована, то цифровые данные будут разрушены, и ошибочные состояния не будут возвращаться в виде битовых значений в аналоговом слове.

Обратите внимание на то, чтобы аналоговая фильтрация в CPU была деактивизирована.

Функции

CP 243-2 – это master-устройство AS-Interface класса master-устройств M1e, что означает поддержку всех указанных функций. Это делает возможной эксплуатацию на модуле AS-Interface до 31 цифрового slave-устройства посредством двойной адресации (A-B). CP 243-2 имеет два режима работы:

- Стандартный режим: доступ к данным ввода/вывода slave-устройства AS-Interface
- Расширенный режим: вызовы master-устройства (например, запись параметров) или запрос диагностических значений

Присоединительные элементы

Модуль AS-Interface имеет следующие присоединительные элементы:

- Две пары клемм для кабеля модуля AS-Interface (с внутренней перемычкой)
- Одна клемма для функциональной земли

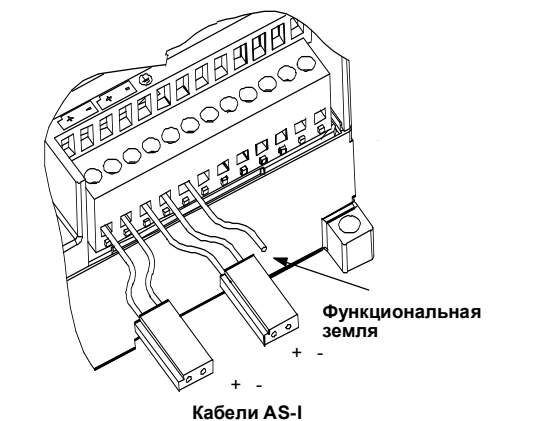


Рис. А-39. Присоединение кабеля к модулю AS-Interface

Клеммы находятся под крышкой передней панели, как показано на рис. А-39.

Осторожно

Нагрузка на контакты модуля AS-Interface не должна превышать 3 А. Если это значение на кабеле модуля AS-Interface превышено, то AS-Interface не должен включаться в контур кабеля AS-I, а должен присоединяться через отдельный кабель (в этом случае используется только одна пара клемм модуля AS-Interface). AS-Interface должен быть присоединен к заземляющему проводу через заземляющую клемму.



Совет

Модуль AS-Interface имеет клемму для функциональной земли. Эта клемма должна быть соединена с линией PE возможно меньшим сопротивлением.

Дополнительная информация

Дальнейшую информацию о master-устройстве CP 243-2 AS-Interface вы найдете в руководстве *SIMATIC NET CP 243-2 AS-Interface Master*.

Дополнительные съёмные модули

Съёмный модуль	Описание	Номер для заказа
Модуль памяти	Модуль памяти, 32К (программа пользователя)	6ES7 291-8GE20-0XA0
Модуль памяти	Модуль памяти, 64К (программа пользователя, рецепт и протокол данных)	6ES7 291-8GF23-0XA0
Модуль памяти	Модуль памяти, 256К (программа пользователя, рецепт и протокол данных)	6ES7 291-8GH23-0XA0
Часы реального времени с батареей	Точность модуля часов: 2 минуты в месяц при 25°C, 7 минут в месяц в диапазоне от 0°C до 55°C	6ES7 297-1AA23-0XA0
Батарейный модуль	Батарейный модуль (время хранения данных): обычно 200 дней	6ES7 291-8BA20-0XA0

Общие технические характеристики		Размеры
Батарейка	3 В, 30 мА·час., Renata CR 1025	
Размеры	9,9 x 2,5 мм	
Тип	Литиевая < 0,6 г	

Модуль памяти

Имеются ограничения для применения модулей памяти в различных вариантах CPU. Модули памяти, запрограммированные определенной моделью CPU, могут прочитаны CPU с таким же или большим номером модели, как показано в таблице А–64:

Таблица А–64. Ограничения в чтении модулей памяти

Модуль памяти запрограммирован в ...	Может быть прочитан ...
CPU 221	CPU 221, CPU 222, CPU 224, CPU 224XP и CPU 226
CPU 222	CPU 222, CPU 224, CPU 224XP и CPU 226
CPU 224	CPU 224, CPU 224XP и CPU 226
CPU 224XP	CPU 224XP и CPU 226
CPU 226	CPU 226

Модули памяти емкостью 64К и 256К спроектированы для работы только с новыми CPU, имеющими приведенный здесь номер для заказа: 6ES7 21х-хх23-0ХВ0. Каждый «х» означает, что эта цифра не имеет значения.

Не рекомендуется использовать модули памяти емкостью 32К (6ES7 291-8GE20-0XA0) в CPU версии «23», так как этот модуль памяти 32К не в состоянии поддерживать новые функциональные возможности CPU. Если вы с помощью CPU «23» сохраняете программу в модуле памяти емкостью 32К, то запрограммированный модуль намеренно делается совместимым с более ранними версиями CPU. Ни одна из дополнительных функций новых CPU не может быть сохранена в модуле памяти емкостью 32К.

Возможно, что у вас есть пользовательские программы, сохраненные в модулях памяти емкостью 32К, первоначально запрограммированных более старыми CPU (версия «20», «21» или «22»). Эти модули могут быть прочитаны новыми CPU с учетом ограничений, приведенных в таблице А–64.

Модуль часов реального времени

Модуль часов реального времени (6ES7 297-1AA23-0XA0) спроектирован для работы только с CPU версии «23». Более ранняя версия модуля часов реального времени (6ES7 297-1AA20-0XA0) ни физически, ни электрически не совместима с CPU версии «23».

Кабель с разъемами для модулей расширения

Общие технические характеристики (6ES7 290-6AA20-0XA0)	
Длина кабеля	0,8 м
Вес	25 г
Тип соединителя	10-контактный ленточный

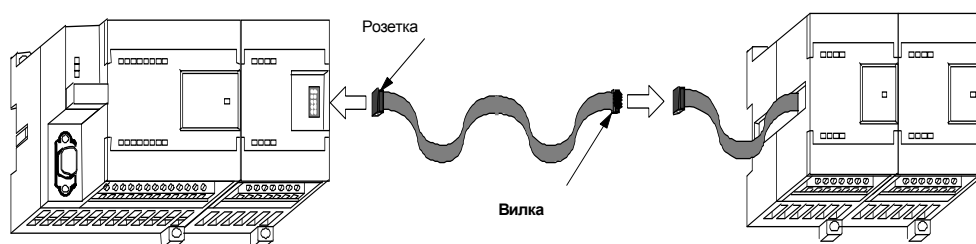


Рис. А–40. Типичный монтаж кабеля с разъемами для модулей расширения



Совет

В цепи CPU/модуль расширения допустим только один кабель расширения.

Кабель RS–232/PPI Multi–Master и кабель USB/PPI Multi–Master

Таблица А–65. Технические данные кабеля RS–232/PPI Multi–Master и кабеля USB/PPI Multi–Master

Описание Номер для заказа	Кабель RS–232/PPI Multi–Master для S7–200 6ES7 901-3CB30-0XA0	Кабель S7–200 USB/PPI Multi–Master для S7–200 6ES7-901-3DB30-0XA0
Общие свойства		
Напряжение питания	от 14,4 до 28,8 В пост. тока	от 14,4 до 28,8 В пост. тока
Питающий ток при номинальном напряжении 24 В	макс. 60 мА, эффективное значение	макс. 50 мА, эффективное значение
Задержка изменения направления: фронт стопового бита RS–232, принимаемый для передачи RS–485, деактивизирован	-	-
Электрическая развязка	RS–485 относительно RS–232: 500 В пост. тока	RS–485 относительно USB: 500 В пост. тока
Электрические свойства со стороны RS–485		
Диапазон напряжения синфазной помехи	от -7 В до +12 В, 1 секунда, 3 В, эффективное значение непрерывно	от -7 В до +12 В, 1 секунда, 3 В, эффективное значение непрерывно
Полное входное сопротивление приемника	мин. 5,4 КОм, включая оконечную нагрузку	мин. 5,4 КОм, включая оконечную нагрузку
Оконечная нагрузка/смещение	10 КОм до +5 В на В, PROFIBUS, контакт 3 10 КОм до GND на А, PROFIBUS, контакт 8	10 КОм до +5 В на В, PROFIBUS, контакт 3 10 КОм до GND на А, PROFIBUS, контакт 8
Порог срабатывания/чувствительность приемника	+/-0,2 В, типовой гистерезис 60 мВ	+/-0,2 В, типовой гистерезис 60 мВ
Дифференциальное выходное напряжение передатчика	мин. 2 В при $R_L=100$ Ом, мин. 1,5 В при $R_L=54$ Ом	мин. 2 В при $R_L=100$ Ом, мин. 1,5 В при $R_L=54$ Ом
Электрические свойства со стороны RS–232		
Полное входное сопротивление приемника	мин. 3 КОм	-
Порог срабатывания/чувствительность приемника	мин. нижний 0,8 В, макс. верхний 2,4 В типовой гистерезис 0,5 В	-
Выходное напряжение передатчика	мин. +/- 5 В при $R_L=3$ КОм	-
Электрические свойства со стороны USB		
Полная скорость (12 МБ/с), интерфейс с человеком (HID)		
Питающее напряжение 5 В	-	макс. 50 мА
Ток в выключенном состоянии	-	макс. 400 μ А

Свойства

Кабель RS–232/PPI Multi–Master для S7–200 настроен на заводе на оптимальную производительность с пакетом программирования STEP 7-Micro/WIN 3.2 Service Pack 4 (или позднее). Заводская настройка для этого кабеля отличается от кабелей PC/PPI. Дальнейшую информацию для настройки кабеля для своего приложения вы найдете на рис. 1.

Вы можете настроить кабель S7–200 RS–232/PPI Multi–Master, чтобы он работал так же, как кабель PC/PPI, и был совместим с любой версией пакета программирования STEP 7-Micro/WIN, установив переключатель 5 в положение PPI/Freerport, а затем выбрав нужную вам скорость передачи.

Кабелю USB для работы необходим пакет программирования STEP 7-Micro/WIN 3.2 Service Pack 4 (или выше).



Совет

Информацию о кабеле PC/PPI вы найдете в редакции 3 руководства *S7-200 Programmable Controller [Программируемый контроллер S7-200]* (6ES7 298-8FA22-8BH0).

Кабель RS-232/PPI Multi-Master для S7-200

Таблица А-66. Кабель RS-232/PPI Multi-Master для S7-200 – Назначение контактов разъема RS-485 относительно локального разъема RS-232

Назначение контактов разъема RS-485		Назначение контактов локального разъема RS-232	
Номер контакта	Описание сигналов	Номер контакта	Описание сигналов
1	Не соединен	1	Обнаружение носителя данных (Data Carrier Detect, DCD) (не используется)
2	Обратный провод 24 В (логическая земля RS-485)	2	Получение данных (Receive Data, RD) (выход кабеля PC/PPI)
3	Сигнал В (RxD/TxD+)	3	Передача данных (Transmit Data, TD) (вход кабеля PC/PPI)
4	RTS (уровень TTL)	4	Терминал готов к передаче данных (Data Terminal Read, DTR) ¹
5	Не соединен	5	Земля (логическая земля RS-232)
6	Не соединен	6	Готовность к работе (Data Set Ready, DSR) ¹
7	Питание 24 В	7	Запрос на передачу (Request To Send, RTS) (не используется)
8	Сигнал А (RxD/TxD-)	8	Готов к передаче (Clear To Send, CTS) (не используется)
9	Выбор протокола	9	Кольцевой индикатор (Ring Indicator, RI) (не используется)

¹ Контакты 4 и 6 подключены внутри.

Таблица А-67. Кабель RS-232/PPI Multi-Master для S7-200 – Назначение контактов разъема RS-485 относительно удаленного разъема RS-232

Назначение контактов разъема RS-485		Назначение контактов удаленного разъема RS-232 ¹	
Номер контакта	Описание сигналов	Номер контакта	Описание сигналов
1	Не соединен	1	Обнаружение носителя данных (Data Carrier Detect, DCD) (не используется)
2	Обратный провод 24 В (логическая земля RS-485)	2	Получение данных (Receive Data, RD) (вход кабеля PC/PPI)
3	Сигнал В (RxD/TxD+)	3	Передача данных (Transmit Data, TD) (выход кабеля PC/PPI)
4	RTS (уровень TTL)	4	Терминал готов к передаче данных (Data Terminal Read, DTR) ²
5	Не соединен	5	Земля (логическая земля RS-232)
6	Не соединен	6	Готовность к работе (Data Set Ready, DSR) ²
7	Питание 24 В	7	Запрос на передачу (Request To Send, RTS) (выход кабеля PC/PPI)
8	Сигнал А (RxD/TxD-)	8	Готов к передаче (Clear To Send, CTS) (не используется)
9	Выбор протокола	9	Кольцевой индикатор (Ring Indicator, RI) (не используется)

¹ Для модемов необходимо преобразование розетка/вилка и 9-/25-контактный разъем.

² Контакты 4 и 6 подключены внутри.

Использование кабеля RS-232/PPI Multi-Master для S7-200 со STEP 7-Micro/WIN в качестве замены кабеля PC/PPI или для режима свободно программируемого обмена данными (Freeport)

Для непосредственного подключения к персональному компьютеру:

- Установите режим PPI/Freeport (переключатель 5=0)
- Установите скорость передачи (переключатели 1, 2, и 3)
- Установите локальный режим (переключатель 6=0). Настройка Local совпадает с настройкой DCE кабеля PC/PPI.
- Установите бит 11 (переключатель 7=0)

Для подключения к модему:

- Установите режим PPI/Freerport (переключатель 5=0)
- Установите скорость передачи (переключатели 1, 2, и 3)
- Установите удаленный режим (переключатель 6=1). Настройка удаленного режима (Remote) совпадает с настройкой DTE кабеля PC/PPI.
- Установите бит 10 или бит 11 (переключатель 7). Эта настройка должна соответствовать настройке для количества битов на символ вашего модема.

Использование кабеля RS-232/PPI Multi-Master для S7-200 со STEP 7-Micro/WIN 3.2 Service Pack 4 (или выше)

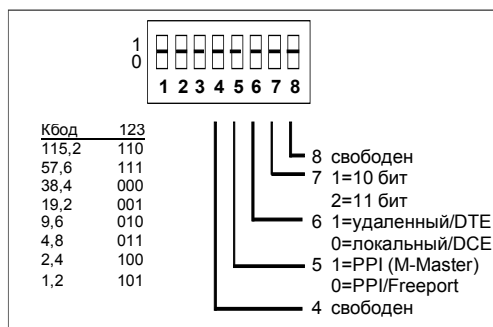
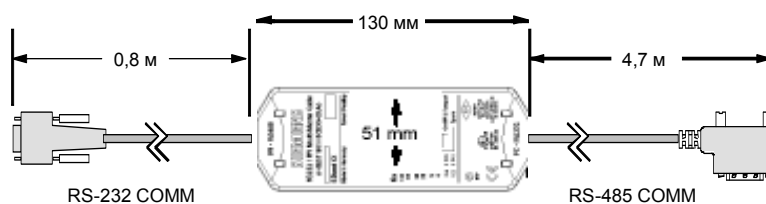
Для непосредственного подключения к персональному компьютеру:

- Установите режим PPI (переключатель 5=1)
- Установите локальный режим (переключатель 6=0)

Для подключения к модему:

- Установите режим PPI (переключатель 5=1)
- Установите удаленный режим (переключатель 6=1)

На рис. А-41 показаны размеры, маркировка и светодиоды кабеля RS-232/PPI Multi-Master для S7-200.



Светодиод	Цвет	Описание
Tx	Зеленый	Индикатор передачи RS-232
Rx	Зеленый	Индикатор приема RS-232
PPI	Зеленый	Индикатор передачи RS-485

Рис. А-41. Размеры, маркировка и светодиоды кабеля RS-232/PPI Multi-Master для S7-200

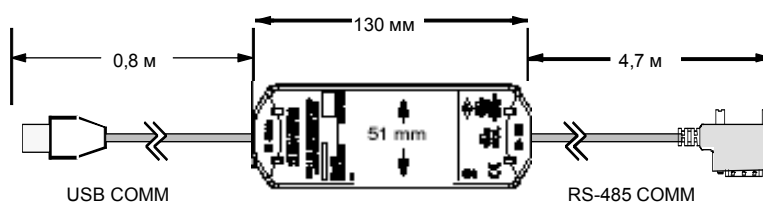
Кабель USB/PPI Multi-Master для S7-200

Для использования кабеля USB должен быть установлен STEP 7-Micro/WIN 3.2 Service Pack 4 (или выше). Рекомендуется использовать кабель USB только с CPU S7-2002x или выше. Кабель USB не поддерживает ни свободно программируемый обмен данными (Freerport), ни загрузку дизайнера TP (TP Designer) в TP070.

Таблица А-68. Кабель USB/PPI Multi-Master для S7-200 – распределение контактов разъема RS-485 относительно разъема USB версии «А»

Назначение контактов разъема RS-485		Назначение контактов разъема USB	
Номер контакта	Описание сигналов	Номер контакта	Описание сигналов
1	Не соединен	1	USB – Data P
2	Обратный провод 24 В (Логическая земля RS-485)	2	USB – Data M
3	Сигнал В (RxD/TxD+)	3	USB 5 В
4	RTS (уровень TTL)	4	Логическая земля USB
5	Не соединен		
6	Не соединен		
7	Питание 24 В		
8	Сигнал А (RxD/TxD-)		
9	Выбор протокола (low = 10 bit)		

На рис. А-42 показаны размеры и светодиоды кабеля USB/PPI Multi-Master для S7-200.



Светодиод	Цвет	Описание
Tx	Зеленый	Индикатор передачи USB
Rx	Зеленый	Индикатор приема USB
PPI	Зеленый	Индикатор передачи RS-485

Рис. А-42. Размеры и светодиоды кабеля USB/PPI Multi-Master для S7-200

Имитаторы входов

Номер для заказа	8-позиционный имитатор 6ES7 274-1XF00-0XA0	14-позиционный имитатор 6ES7 274-1XH00-0XA0	24-позиционный имитатор 6ES7 274-1XK00-0XA0
Размеры (Д x Ш x Г)	61 x 33,5 x 22 мм	91,5 x 35,5 x 22 мм	148,3 x 35,5 x 22 мм
Вес	0,02 кг	0,03 кг	0,04 кг
Количество входов и выходов	8	14	24

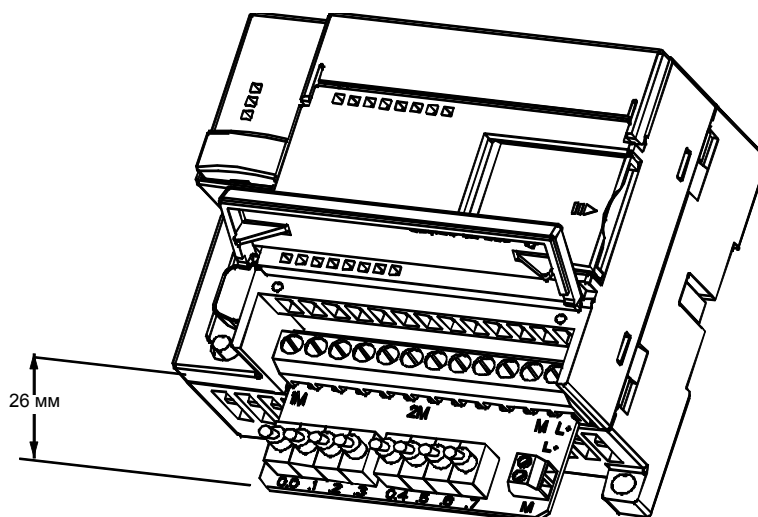


Рис. А-43. Монтаж имитатора входов

Предупреждение

Эти имитаторы входов непригодны для использования во взрывоопасных помещениях, относящихся к категориям опасности Class I DIV 2 и Class I Zone 2. Переключатели представляют потенциальную опасность образования искры.

Не используйте имитаторы входов во взрывоопасных помещениях, относящихся к категориям опасности Class I DIV 2 и Class I Zone 2.

Расчет баланса мощностей

B

CPU S7–200 имеет внутренний источник напряжения, который обеспечивает питанием сам CPU, все модули расширения и других потребителей напряжения 24 В постоянного тока. Следующую информацию вы можете использовать в качестве руководства для определения того, какую мощность (или ток) может предоставить для вашей конфигурации CPU S7–200.

Потребности в мощности

Каждый CPU S7–200 поставляет напряжения 5 и 24 В пост. тока:

- Каждый CPU имеет источник питания датчиков напряжением 24 В пост. тока, который может снабжать этим напряжением встроенные входы и катушки реле модулей расширения. Если потребности в мощности напряжением 24 В пост. тока превышают возможности внутреннего источника питания CPU, то вы можете дополнительно подключить внешний источник напряжением 24 В пост. тока для питания модулей расширения. Вы должны подключить этот источник к входам и катушкам реле вручную.
- Кроме того, CPU предоставляет питание напряжением 5 В пост. тока для, возможно, подключенных модулей расширения. Если потребности в мощности напряжением 5 В пост. тока превышают возможности внутреннего источника питания CPU, то вы должны убирать модули расширения, пока потребности в этом напряжении не окажутся в пределах возможностей внутреннего источника питания CPU.

Технические данные в Приложении А дают информацию о наличных мощностях отдельных CPU и потребностях в мощности модулей расширения.



Совет

Если баланс мощностей CPU нарушен, то к вашему CPU нельзя подключить максимально допустимое для этого CPU количество модулей.



Предупреждение

Включение внешнего источника питания 24 В постоянного тока параллельно с источником постоянного тока для питания датчиков S7–200 может привести к конфликту между двумя источниками питания, так как каждый из них стремится установить свой собственный уровень выходного напряжения.

Результатом этого конфликта может быть сокращение срока службы или немедленный выход из строя одного или обоих источников питания с последующим непредсказуемым поведением ПЛК. Непредсказуемое поведение может привести к гибели или тяжким телесным повреждениям людей и/или повреждению оборудования.

Источник питания датчиков постоянного тока S7–200 и внешний источник питания должны подавать напряжение в разные точки. Между двумя источниками напряжения возможно не более одного соединения.

Пример расчета потребностей в мощности

В таблице В–1 показан расчета потребностей в мощности для S7–200, который включает в себя:

- CPU 224 AC/DC/Relay
- 3 EM 223 8 DC In/8 Relay Out
- 1 EM 221 8 DC In

Эта структура имеет в целом 46 входов и 34 выхода.



Совет

CPU уже выделил мощность, необходимую для обеспечения потребностей катушек внутренних реле. Поэтому вам не нужно включать потребности внутренних реле в расчет баланса мощностей.

CPU S7–200 в этом примере обеспечивает достаточное питание напряжением 5 В пост. тока для модулей расширения, но не обеспечивает достаточного питания напряжением 24 В пост. тока из источника питания датчиков для всех входов и катушек реле модулей расширения. Входам и выходам необходимо 400 мА, а CPU S7–200 предоставляет только 280 мА. Эта структура требует дополнительного источника не менее 120 мА при напряжении 24 В пост. тока для питания всех входов и выходов 24 В пост. тока.

Таблица В–1. Расчет баланса мощностей для конфигурации из примера

Мощность CPU	5 В пост. тока	24 В пост. тока
CPU 224 AC/DC/Relay	660 мА	280 мА

минус

Потребности в мощности системы	5 В пост. тока	24 В пост. тока
CPU 224, 14 входов		14 * 4 мА = 56 мА
3 EM 223, необходимо питание 5 В	3 * 80 мА = 240 мА	
1 EM 221, необходимо питание 5 В	1 * 30 мА = 30 мА	
3 EM 223, по 8 входов каждый		3 * 8 * 4 мА = 96 мА
3 EM 223, по 8 катушек реле каждый		3 * 8 * 9 мА = 216 мА
1 EM 221, 8 входов		8 * 4 мА = 32 мА
Суммарная потребность	270 мА	400 мА

равно

Баланс токов	5 В пост. тока	24 В пост. тока
Общий баланс токов	390 мА	[120 мА]

Расчет ваших потребностей в мощности

Используйте следующую таблицу для определения того, какую мощность (или ток) может поставить CPU S7-200 для вашей конфигурации. Располагаемую мощность CPU своей модели и потребности в мощности ваших модулей расширения вы найдете в Приложении А.

Мощность CPU	5 В пост. тока	24 В пост. тока

минус

Потребности системы	5 В пост. тока	24 В пост. тока
Суммарные потребности		

равно

Баланс токов	5 В пост. тока	24 В пост. тока
Общий баланс токов		

Коды ошибок

C

Информация о кодах ошибок предоставляется в ваше распоряжение, чтобы помочь вам определить, что случилось с вашим модулем CPU S7-200.

В этой главе

Коды фатальных ошибок и сообщения	464
Ошибки этапа выполнения	465
Нарушение правил компиляции	466

Коды фатальных ошибок и сообщения

Фатальные (неустраняемые) ошибки заставляют CPU остановить исполнение вашей программы. В зависимости от серьезности ошибки CPU может оказаться неспособным выполнять некоторые или даже все функции. Целью обработки фатальных ошибок является перевод CPU в безопасное состояние, в котором CPU может отвечать на вопросы о текущей сбойной ситуации.

При обнаружении фатальной ошибки S7-200 выполняет следующие действия:

- переходит в состояние STOP
- включает светодиодные индикаторы SF/DIAG (красный) и светодиод состояния STOP
- выключает выходы

S7-200 остается в этом состоянии, пока ошибка не будет устранена. Чтобы увидеть коды ошибок, выберите из строки главного меню команду **PLC > Information [ПЛК > Информация]**. В таблице С-1 приведен список с описаниями кодов ошибок, которые могут быть считаны из S7-200.

Таблица С-1. Коды фатальных ошибок и сообщения, считываемые из S7-200

Код ошибки	Описание
0000	Нет фатальных ошибок
0001	Ошибка контрольной суммы программы пользователя
0002	Ошибка контрольной суммы скомпилированной программы на LAD
0003	Превышение времени при контроле времени цикла
0004	Неисправность ПЗУ
0005	Ошибка контрольной суммы ПЗУ в программе пользователя
0006	Ошибка контрольной суммы ПЗУ в параметрах конфигурации (SDB0)
0007	Ошибка контрольной суммы ПЗУ в принудительно задаваемых данных
0008	Ошибка контрольной суммы ПЗУ в устанавливаемых по умолчанию значениях таблицы выходов
0009	Ошибка контрольной суммы ПЗУ в данных пользователя, DB1
000A	Неисправность модуля памяти
000B	Ошибка контрольной суммы модуля памяти в программе пользователя.
000C	Ошибка контрольной суммы модуля памяти в параметрах конфигурации (SDB0)
000D	Ошибка контрольной суммы модуля памяти в принудительно задаваемых данных
000E	Ошибка контрольной суммы модуля памяти в устанавливаемых по умолчанию значениях таблицы выходов
000F	Ошибка контрольной суммы модуля памяти в данных пользователя, DB1
0010	Внутренняя ошибка программного обеспечения
0011 ¹	Ошибка при косвенной адресации контакта сравнения
0012 ¹	Недопустимое значение с плавающей точкой для контакта сравнения
0013	Модуль памяти пуст или программа непонятна для этого S7-200
0014 ¹	Ошибка области для контакта сравнения

¹ Ошибки контакта сравнения – это единственные ошибки, которые генерируют как фатальные, так и нефатальные условия ошибки. Нефатальная ошибка генерируется для сохранения адреса ошибки в программе.

Ошибки этапа выполнения

При нормальном выполнении вашей программы могут возникать нефатальные ошибки (например, ошибки адресации). В этом случае S7-200 генерирует код нефатальной ошибки этапа выполнения. В таблице С-2 приведены описания кодов нефатальных ошибок.

Таблица С-2. Ошибки этапа выполнения

Код ошибки	Описание
0000	Фатальная ошибка отсутствует; нет ошибок
0001	Блок HSC разблокирован до выполнения блока HDEF
0002	Конфликт при назначении прерывания от ввода входу, уже назначенному HSC
0003	Конфликт при назначении входов HSC, уже назначенному прерыванию от ввода или другому HSC
0004	Попытка выполнения команды, не допустимой в программе обработки прерывания
0005	Попытка выполнения второй команды HSC/PLS с тем же самым номером до завершения первой (HSC/PLS в программе обработки прерывания конфликтует с HSC/PLS в главной программе)
0006	Ошибка косвенной адресации
0007	Ошибка данных TODW (Записать время суток) или TODR (Прочитать время суток)
0008	Превышен максимальный уровень вложенности подпрограмм пользователя
0009	Одновременное выполнение команд XMT/RCV в порту 0
000A	Попытка повторного определения HSC путем выполнения команды HDEF для того же самого HSC
000B	Одновременное выполнение команд XMT/RCV в порту 1
000C	Отсутствует модуль часов для доступа через TODR, TODW или обмена данными
000D	Попытка повторного определения импульсного выхода, когда он активен
000E	Номер сегмента профиля PTO установлен в 0
000F	Недопустимое числовое значение в контакте сравнения
0010	Команда недопустима в текущем режиме работы PTO
0011	Недопустимый код команды PTO
0012	Недопустимая таблица профилей PTO
0013	Недопустимая таблица PID-регулятора
0091	Ошибка диапазона (в адресной информации): проверьте диапазоны операндов
0092	Ошибка в поле счета в команде (в информации о счете): проверьте максимальное значение счета
0094	Ошибка диапазона при записи в энергонезависимую память в адресной информации
009A	Попытка перейти в режим свободно программируемого обмена данными в прерывании от пользователя
009B	Недопустимый индекс (строка с нулевым значением в начальной позиции)
009F	Отсутствует или не отвечает модуль памяти

Нарушение правил компиляции

При загрузке программы S7-200 ее компилирует. Если S7-200 обнаруживает, что программа нарушает правила компиляции (например, содержит недопустимую команду), то S7-200 прерывает загрузку и генерирует код нефатальной ошибки нарушения правил компиляции. Таблица С-3 дает описания кодов ошибок, которые генерируются при нарушениях правил компиляции.

Таблица С-3. Нарушения правил компиляции

Код ошибки	Ошибки компиляции (нефатальные)
0080	Программа слишком велика для компиляции; уменьшите размер программы
0081	Переполнение стека; разделите сегмент на несколько сегментов.
0082	Недопустимая команда; проверьте мнемонику команды.
0083	Отсутствует MEND или команда не допустима в главной программе: добавьте команду MEND или удалите неправильную команду.
0084	Резерв
0085	Отсутствует FOR; добавьте команду FOR или удалите команду NEXT.
0086	Отсутствует NEXT; добавьте команду NEXT или удалите команду FOR.
0087	Отсутствует метка (LBL, INT, SBR); добавьте подходящую метку.
0088	Отсутствует RET или команда не допустима в подпрограмме: добавьте RET к концу подпрограммы или удалите неправильную команду.
0089	Отсутствует RETI или команда не допустима в программе обработки прерывания: добавьте RETI к концу программы обработки прерывания или удалите неправильную команду.
008A	Резерв
008B	Недопустимая команда перехода JMP в сегмент SCR или из него
008C	Двойная метка (LBL, INT, SBR); переименуйте одну из меток.
008D	Недопустимая метка (LBL, INT, SBR); обеспечьте, чтобы не превышалось допустимое количество меток
0090	Недопустимый параметр; проверьте допустимые параметры для команды.
0091	Ошибка диапазона (в адресной информации); проверьте диапазоны операндов
0092	Ошибка в поле счета в команде (в информации о счете): проверьте максимальное значение счета.
0093	Превышен уровень вложенности FOR/NEXT.
0095	Отсутствует команда LSCR (Загрузить SCR)
0096	Отсутствует команда SCRE (Конец SCR) или недопустимая команда перед командой SCRE
0097	Программа пользователя содержит как нумерованные, так и нумерованные команды EV/ED
0098	Недопустимое редактирование в режиме RUN (предпринята попытка редактирования в программе с нумерованными командами EV/ED)
0099	Слишком много скрытых программных сегментов (команд HIDE)
009B	Недопустимый индекс (строка с нулевым значением в начальной позиции)
009C	Превышена максимальная длина команды
009D	В SDB0 обнаружены недопустимые параметры
009E	Слишком много строк PCALL
009F – 00FF	Резерв

Биты специальной памяти (SM)

D

Биты специальной памяти предоставляют в распоряжение ряд функций состояния и управления, а также служат для организации обмена информацией между S7-200 и вашей программой. Биты специальной памяти могут быть использованы как биты, байты, слова или двойные слова.

В этой главе

SMB0: биты состояния	468
SMB1: биты состояния	468
SMB2: принимаемые символы при свободно программируемом обмене данными	469
SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными	469
SMB4: переполнение очереди ожидания	470
SMB5: состояние входов/выходов	470
SMB6: идентификационный регистр CPU	471
SMB7: Резерв	471
SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода	472
SMW22 – SMW26: времена цикла	473
SMB28 и SMB29: аналоговый потенциометр	473
SMB30 и SMB130: регистры управления свободно программируемым обменом данными	473
SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ)	474
SMB34 и SMB35: регистры интервалов времени для прерываний, управляемых временем	474
SMB36 – SMB65: регистры HSC0, HSC1 и HSC2	475
SMB66 – SMB85: регистры PTO/PWM	476
SMB86 – SMB94 и SMB186 – SMB194: управление приемом сообщений	478
SMW98: ошибки в шине расширения	479
SMB130: регистр управления свободно программируемым обменом данными (см. SMB30)	479
SMB131 – SMB165: регистры HSC3, HSC4 и HSC5	479
SMB166 – SMB185: таблица определения профилей PTO0, PTO1	480
SMB186 – SMB194: управление приемом сообщений (см. SMB86 – SMB94)	481
SMB200 – SMB549: состояние интеллектуальных модулей	481

SMB0: биты состояния

Как описано в таблице D–1, SMB0 содержит восемь битов состояния, которые обновляются S7–200 в конце каждого цикла обработки программы.

Таблица D–1. Байт специальной памяти SMB0 (от SM0.0 до SM0.7)

SM-биты	Описание (только чтение)
SM0.0	Этот бит всегда установлен.
SM0.1	Этот бит устанавливается на время первого цикла обработки программы. Он используется, например, для вызова подпрограммы инициализации.
SM0.2	Этот бит включается на один цикл обработки программы, если были потеряны сохраняемые (реманентные) данные. Этот бит может быть использован как бит ошибки или как механизм для вызова специальной последовательности запуска.
SM0.3	Этот бит включается на один цикл обработки программы, если режим RUN устанавливается при включении питания. Этот бит может быть использован для предоставления установке времени на разогрев перед началом работы.
SM0.4	Этот бит предоставляет в распоряжение генератор тактовых импульсов, которые включены в течение 30 секунд и выключены в течение 30 секунд, с периодом следования импульсов 1 минута. Тем самым вы получаете в распоряжение легкую в использовании задержку или генератор тактовых импульсов с периодом 1 минута.
SM0.5	Этот бит предоставляет в распоряжение генератор тактовых импульсов, которые включены в течение 0,5 секунды и выключены в течение 0,5 секунды, с периодом следования импульсов 1 секунда. Тем самым вы получаете в распоряжение легкую в использовании задержку или генератор тактовых импульсов с периодом 1 секунда.
SM0.6	Этот бит предоставляет в распоряжение генератор импульсов, которые включены в течение одного цикла обработки программы, а затем выключены в течение следующего цикла. Этот бит может быть использован как вход для счетчика циклов.
SM0.7	Этот бит отражает положение переключателя режимов работы (выключен для положения TERM и установлен для положения RUN). Если вы используете этот бит для разблокирования режима свободно программируемого обмена данными, когда переключатель находится в положении RUN, то нормальная связь с устройством программирования может быть разблокирована переключением в положение TERM.

SMB1: биты состояния

Как описано в таблице D–2, SMB1 содержит индикаторы возможных ошибок. Эти биты устанавливаются и сбрасываются командами во время исполнения.

Таблица D–2. Байт специальной памяти SMB1 (от SM1.0 до SM1.7)

SM-биты	Описание (только чтение)
SM1.0	Этот бит устанавливается при исполнении определенных команд, когда результат операции равен нулю.
SM1.1	Этот бит устанавливается при исполнении определенных команд, когда возникает переполнение или когда обнаруживается недопустимое числовое значение.
SM1.2	Этот бит устанавливается, когда арифметическая операция дат отрицательный результат.
SM1.3	Этот бит устанавливается, когда делается попытка деления на нуль.
SM1.4	Этот бит устанавливается, когда команда ввода значения в таблицу пытается переполнить таблицу.
SM1.5	Этот бит устанавливается, когда команда LIFO или FIFO пытается читать из пустой таблицы.
SM1.6	Этот бит устанавливается, когда делается попытка преобразовать в двоичный код величину, не представленную в формате BCD.
SM1.7	Этот бит устанавливается, когда значение в коде ASCII не может быть преобразовано в допустимое шестнадцатеричное значение.

SMB2: принимаемые символы при свободно программируемом обмене данными

SMB2 – это буфер для приема символов в режиме свободно программируемого обмена данными. Как описано в таблице D–3, каждый символ, принимаемый в режиме свободно программируемого обмена данными, помещается по этому адресу для облегчения доступа к нему из программы, написанной на LAD.


	<p>Совет</p> <p>SMB2 и SMB3 используются совместно портом 0 и портом 1. Если прием символа в порту 0 приводит к выполнению программы обработки прерывания, поставленной в соответствие этому событию (прерывающее событие 8), то SMB2 содержит символ, полученный портом 0, а SMB3 содержит результат проверки этого символа на четность. Если прием символа в порту 1 приводит к выполнению программы обработки прерывания, поставленной в соответствие этому событию (прерывающее событие 25), то SMB2 содержит символ, полученный портом 1, а SMB3 содержит результат проверки этого символа на четность.</p>
---	---

Таблица D–3. Байт специальной памяти SMB2

SM-байт	Описание (только чтение)
SMB2	Этот байт содержит каждый символ, принимаемый из порта 0 или порта 1 в режиме свободно программируемого обмена данными.

SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными

SMB3 используется в режиме свободно программируемого обмена данными и содержит бит контроля четности, который устанавливается при обнаружении ошибки четности в принятом символе. Как показано в таблице D–4, SM3.0 устанавливается, когда обнаруживается ошибка четности. Используйте этот бит, чтобы отвергнуть сообщение.

Таблица D–4. Байт специальной памяти SMB3 (от SM3.0 до SM3.7)

SM-биты	Описание (только чтение)
SM3.0	Ошибка четности от порта 0 или порта 1 (0 = нет ошибки; 1 = ошибка обнаружена)
SM3.1 – SM3.7	Резерв

SMB4: переполнение очереди ожидания

Как описано в таблице D–5, SMB4 содержит биты переполнения очереди прерываний, индикатор состояния, показывающий, разблокированы или заблокированы прерывания, и также бит памяти, показывающий, свободен ли передатчик. Биты переполнения очереди показывают, что прерывания происходят чаще, чем они могут быть обработаны, или что прерывания были заблокированы командой глобального блокирования прерываний.

Таблица D–5. Байт специальной памяти SMB4 (от SM4.0 до SM4.7)

SM-биты	Описание (только чтение)
SM4.0 ¹	Этот бит устанавливается, когда переполнена очередь коммуникационных прерываний.
SM4.1 ¹	Этот бит устанавливается, когда переполнена очередь прерываний от ввода.
SM4.2 ¹	Этот бит устанавливается, когда переполнена очередь прерываний, управляемых временем.
SM4.3	Этот бит устанавливается, когда во время выполнения обнаруживается ошибка программирования.
SM4.4	Этот бит отражает состояние глобального разблокирования прерываний. Он устанавливается, когда прерывания разблокированы.
SM4.5	Этот бит устанавливается, когда передатчик не работает (порт 0).
SM4.6	Этот бит устанавливается, когда передатчик не работает (порт 1).
SM4.7	Этот бит устанавливается, когда что-то делается принудительно.

¹ Используйте биты состояния 4.0, 4.1 и 4.2 только в программе обработки прерывания. Эти биты состояния сбрасываются, когда очередь становится пустой, и управление возвращается в главную программу.

SMB5: состояние входов/выходов

Как описано в таблице D–6, SMB5 содержит биты сбойных состояний, обнаруженных в системе ввода-вывода. Эти биты дают обзор обнаруженных ошибок ввода-вывода.

Таблица D–6. Байт специальной памяти SMB5 (от SM5.0 до SM5.7)

SM-биты	Описание (только чтение)
SM5.0	Этот бит устанавливается при наличии любой ошибки ввода-вывода.
SM5.1	Этот бит устанавливается, если к шине ввода-вывода подключено слишком много цифровых входов и выходов.
SM5.2	Этот бит устанавливается, если к шине ввода-вывода подключено слишком много аналоговых входов и выходов.
SM5.3	Этот бит устанавливается, если к шине ввода-вывода подключено слишком много интеллектуальных модулей ввода/вывода.
SM5.4 – SM5.7	Резерв

SMB6: идентификационный регистр CPU

Как описано в таблице D-7, SMB6 – это идентификационный регистр CPU S7-200. Биты от SM6.4 до SM6.7 содержат идентификатор CPU S7-200. Биты от SM6.0 до SM6.3 зарезервированы для использования в будущем.

Таблица D-7. Байт специальной памяти SMB6

SM-биты	Описание (только чтение)	
Формат	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 LSB 0 </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">x</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">x</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">x</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">x</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">r</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">r</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">r</div> <div style="border: 1px solid black; padding: 2px;">r</div> </div>	Идентификационный регистр CPU
SM6.0 – SM6.3	резерв	
SM6.4 – SM6.7	xxxx = 0000 = CPU 222 0010 = CPU 224 0110 = CPU 221 1001 = CPU 226/CPU 226XM	

SMB7: Резерв

SMB7 зарезервирован для использования в будущем.

SMW22 – SMW26: времена цикла

Как описано в таблице D–9, SMW22, SMW24 и SMW26 дают информацию о времени цикла: минимальное время цикла, максимальное время цикла и время последнего цикла в миллисекундах.

Таблица D–9. Слова специальной памяти от SMW22 до SMW26

SM-слово	Описание (только чтение)
SMW22	Время последнего цикла в миллисекундах
SMW24	Минимальное время цикла в миллисекундах с момента перехода в режим RUN
SMW26	Максимальное время цикла в миллисекундах с момента перехода в режим RUN

SMB28 и SMB29: аналоговый потенциометр

Как описано в таблице D–10, SMB28 содержит цифровое значение, представляющее положение аналогового потенциометра 0. SMB29 содержит цифровое значение, представляющее положение аналогового потенциометра 1.

Таблица D–10. Байты специальной памяти SMB28 и SMB29

SM-байт	Описание (только чтение)
SMB28	Этот байт хранит значение, введенное аналоговым потенциометром 0. Это значение обновляется один раз за цикл в STOP/RUN.
SMB29	Этот байт хранит значение, введенное аналоговым потенциометром 1. Это значение обновляется один раз за цикл в STOP/RUN.

SMB30 и SMB130: регистры управления свободно программируемым обменом данными

SMB30 управляет свободно программируемым обменом данными для порта 0; SMB130 управляет свободно программируемым обменом данными для порта 1. Вы можете читать и записывать в SMB30 и SMB130. Как описано в таблице D–11, эти байты настраивают соответствующий коммуникационный порт для работы в режиме свободно программируемого обмена данными и предоставляют возможность выбора протокола свободно программируемого обмена данными или системного протокола.

Таблица D–11. Байт специальной памяти SMB30

Порт 0	Порт 1	Описание								
Формат SMB30	Формат SMB130	Управляющий байт режима свободно программируемого обмена данными <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">MSB</td> <td style="text-align: center;">LSB</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">p</td> <td style="text-align: center;">p</td> <td style="text-align: center;">d</td> <td style="text-align: center;">b b b m m</td> </tr> </table> </div>	7	MSB	LSB	0	p	p	d	b b b m m
7	MSB	LSB	0							
p	p	d	b b b m m							
SM30.0 и SM30.1	SM130.0 и SM130.1	mm: Выбор протокола 00 = протокол интерфейса точка-точка (PPI/непривилегированный режим) 01 = протокол свободно программируемого обмена данными 10 = PPI/привилегированный режим 11 = резерв (по умолчанию PPI/непривилегированный режим) Примечание: Когда вы выбираете код mm = 10 (PPI/привилегированный режим), S7–200 становится master-устройством в сети и допускает выполнение команд NETR и NETW. В режимах PPI биты с 2-го по 7 игнорируются.								
SM30.2 – SM30.4	SM130.2 – SM130.4	bbb: скорость передачи в режиме свободно программируемого обмена данными 000 =38 400 Бод 100 =2 400 Бод 001 =19 200 Бод 101 =1 200 Бод 010 =9 600 Бод 110 =115 200 Бод 011 =4 800 Бод 111 =57 600 Бод								
SM30.5	SM130.5	d: Количество битов данных на символ 0 =8 битов на символ 1 =7 битов на символ								
SM30.6 и SM30.7	SM130.6 и SM130.7	pp: выбор контроля четности 00 = нет контроля четности 10 = контроль по нечетности 01 = контроль по четности 11 = контроль по нечетности								

SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ)

Вы можете сохранить значение, хранящееся в памяти переменных в постоянную память (ЭСППЗУ) под управлением вашей программы. Для этого загрузите адрес ячейки, подлежащей сохранению, в SMW32. Затем загрузите SMB31 с командой сохранить значение. После того как вы загрузили команду для сохранения значения, не меняйте значения в памяти переменных, пока CPU не сбросит бит SM31.7, показывая, что операция сохранения завершена.

В конце каждого цикла S7–200 проверяет, не отдана ли команда на сохранение значения в постоянной памяти. Если команда отдана, указанное значение сохраняется в постоянной памяти.

Как описано в таблице D–12, SMB31 определяет размер данных, подлежащих сохранению в постоянной памяти, а также предоставляет в распоряжение команду, которая инициирует выполнение операции сохранения. SMW32 хранит начальный адрес в памяти переменных для данных, подлежащих сохранению в постоянной памяти.

Таблица D–12. Байт специальной памяти SMB31 и слово специальной памяти SMW32

SM-байт	Описание								
Формат	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> MSB 7 <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">c</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">s</td> <td style="padding: 2px;">s</td> </tr> </table> </div> <div style="text-align: center;"> LSB 0 </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> MSB 15 Адрес в V-памяти </div> <div style="text-align: center;"> LSB 0 </div> </div>	c	0	0	0	0	0	s	s
c	0	0	0	0	0	s	s		
SM31.0 и SM31.1	ss: размер данных 00 = байт 10 = слово 01 = байт 11 = двойное слово								
SM31.7	c: Сохранить в ЭСПЗУ 0 = Нет запроса на сохранение 1 = Программа пользователя обращается с запросом на сохранение данных S7–200 сбрасывает этот бит после каждой операции сохранения.								
SMW32	Адрес в V-памяти для данных, подлежащих сохранению, хранится в SMW32. Это значение вводится как смещение от V0. Когда операция сохранения выполняется, значение из этого адреса в V-памяти сохраняется в соответствующей ячейке V-памяти постоянного запоминающего устройства (ЭСППЗУ).								

SMB34 и SMB35: регистры интервалов времени для прерываний, управляемых временем

Как описано в таблице D–13, SMB34 задает интервал времени для циклического прерывания 0, а SMB35 задает интервал времени для циклического прерывания 1. Вы можете задать интервал времени (шагами по 1 мс) от 1 до 255 мс. Значение интервала времени воспринимается S7–200, когда соответствующее событие, вызывающее прерывание, управляемое временем, назначается программе обработки прерывания. Для изменения этого интервала времени вы должны снова назначить событие, вызывающее циклическое прерывание, той же самой или другой программе обработки прерывания. Вы можете завершить событие, вызывающее циклическое прерывание, отсоединив это событие.

Таблица D–13. Байты специальной памяти SMB34 и SMB35

SM-байт	Описание
SMB34	Этот байт задает интервал времени (шагами по 1 мс от 1 мс до 255 мс) для прерывания, управляемого временем, 0.
SMB35	Этот байт задает интервал времени (шагами по 1 мс от 1 мс до 255 мс) для прерывания, управляемого временем, 1.

SMB36 – SMB65: регистры HSC0, HSC1 и HSC2

Как описано в таблице D–14, SMB36 – SM65 используются для контроля и управления скоростными счетчиками HSC0, HSC1 и HSC2.

Таблица D–14. Байты специальной памяти SMB36 – SMD62

SM-байт	Описание
SM36.0 – SM36.4	Резерв
SM36.5	Бит состояния текущего направления счета HSC0: 1 = прямой счет
SM36.6	Бит состояния «текущее значение равно предустановленному значению» для HSC0: 1 = равно
SM36.7	Бит состояния «текущее значение больше предустановленного значения» для HSC0: 1 = больше
SM37.0	Бит управления уровнем активности для сброса HSC0: 0= активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM37.1	Резерв
SM37.2	Выбор скорости счета для квадратурных счетчиков: 0 = 4-кратная скорость; 1 = 1-кратная скорость
SM37.3	Бит управления направлением счета HSC0: 1 = прямой счет
SM37.4	Актуализация направления HSC0: 1 = актуализировать направление
SM37.5	Актуализация предустановленного значения HSC0: 1 = записать новое предустановленное значение в HSC0
SM37.6	Актуализация текущего значения HSC0: 1 = записать новое текущее значение в HSC0
SM37.7	Бит разблокировки HSC0: 1 = разблокировать
SMD38	Новое текущее значение HSC0
SMD42	Новое предустановленное значение HSC0
SM46.0 – SM46.4	Резерв
SM46.5	Бит состояния текущего направления счета HSC1: 1 = прямой счет
SM46.6	Бит состояния «текущее значение равно предустановленному значению» для HSC1: 1 = равно
SM46.7	Бит состояния «текущее значение больше предустановленного значения» для HSC1: 1 = больше
SM47.0	Бит управления уровнем активности для сброса HSC1: 0= активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM47.1	Бит управления уровнем активности для пуска HSC1: 0= активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM47.2	Выбор скорости счета для квадратурного счетчика HSC1: 0 = 4-кратная скорость; 1 = 1-кратная скорость
SM47.3	Бит управления направлением счета HSC1: 1 = прямой счет
SM47.4	Актуализация направления HSC1: 1 = актуализировать направление
SM47.5	Актуализация предустановленного значения HSC1: 1 = записать новое предустановленное значение в HSC1
SM47.6	Актуализация текущего значения HSC1: 1 = записать новое текущее значение в HSC1
SM47.7	Бит разблокировки HSC1: 1 = разблокировать
SMD48	Новое текущее значение HSC1
SMD52	Новое предустановленное значение HSC1

Таблица D–14. Байты специальной памяти SMB36 – SMD62, продолжение

SM-байт	Описание
SM56.0 – SM56.4	Резерв
SM56.5	Бит состояния текущего направления счета HSC2: 1 = прямой счет
SM56.6	Бит состояния «текущее значение равно предустановленному значению» для HSC2: 1 = равно
SM56.7	Бит состояния «текущее значение больше предустановленного значения» для HSC2: 1 = больше
SM57.0	Бит управления уровнем активности для сброса HSC2: 0= активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM57.1	Бит управления уровнем активности для пуска HSC2: 0= активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM57.2	Выбор скорости счета для квадратурного счетчика HSC2: 0 = 4-кратная скорость; 1 = 1-кратная скорость
SM57.3	Бит управления направлением счета HSC2: 1 = прямой счет
SM57.4	Актуализация направления HSC2: 1 = актуализировать направление
SM57.5	Актуализация предустановленного значения HSC2: 1 = записать новое предустановленное значение в HSC2
SM57.6	Актуализация текущего значения HSC2: 1 = записать новое текущее значение в HSC2
SM57.7	Бит разблокировки HSC1: 1 = разблокировать
SMD58	Новое текущее значение HSC2
SMD62	Новое предустановленное значение HSC2

SMB66 – SMB85: регистры ПТО/PWM

Как описано в таблице D–15, байты SMB66 – SMB85 используются для контроля и управления функциями вывода импульсной последовательности и широтно-импульсной модуляции. За полным описанием этих битов обратитесь к информации о командах скоростного вывода последовательностей импульсов в главе 6.

Таблица D–15. Байты специальной памяти SMB66 – SMB85

SM-байт	Описание
SM66.0 – SM66.3	резерв
SM66.4	Профиль ПТО0 прерван: 0 = нет ошибки, 1 = прерван из-за ошибки в расчете приращения
SM66.5	Профиль ПТО0 прерван: 0 = прерван не по команде пользователя, 1 = прерван по команде пользователя
SM66.6	Переполнение конвейера ПТО0 (очищается системой при использовании внешних профилей, в противном случае должен быть сброшен пользователем): 0 = нет переполнения, 1 = конвейер переполнен
SM66.7	Бит холостого хода ПТО0: 0 = ПТО активен, 1 = ПТО не активен
SM67.0	Актуализация значения времени цикла ПТО0/PWM0: 1 = записать новое значение времени цикла
SM67.1	Актуализация значения ширины импульсов PWM0: 1 = записать новую ширину импульсов
SM67.2	Актуализация значения количества импульсов ПТО0: 1 = записать новое количество импульсов
SM67.3	База времени ПТО0/PWM0: 0 = 1 мкс/такт; 1 = 1 мс/такт
SM67.4	Обновлять PWM0 синхронно: 0 = асинхронное обновление, 1 = синхронное обновление
SM67.5	Режим работы ПТО0: 0 = односегментный режим (время цикла и количество импульсов хранятся в SM-памяти), 1 = многосегментный режим (таблица профиля хранится в памяти переменных)
SM67.6	Выбор ПТО0 или PWM0: 0 = ПТО, 1 = PWM
SM67.7	Бит разблокировки ПТО0/PWM0: 1 = разблокировать

Таблица D–15. Байты специальной памяти SMB66 – SMB85, продолжение

SM-байт	Описание
SMW68	Значение времени цикла PTO0/PWM0 (от 2 до 65 535 единиц базы времени)
SMW70	Значение ширины импульсов PWM0 (от 0 до 65 535 единиц базы времени);
SMD72	Количество импульсов PTO0 (от 1 до $2^{32}-1$);
SM76.0 – SM76.3	Резерв
SM76.4	Профиль PTO1 прерван: 0 = нет ошибки, 1 = прерван из-за ошибки в расчете приращения
SM76.5	Профиль PTO1 прерван: 0 = прерван не по команде пользователя, 1 = прерван по команде пользователя
SM76.6	Переполнение конвейера PTO1 (очищается системой при использовании внешних профилей, в противном случае должен быть сброшен пользователем): 0 = нет переполнения, 1 = конвейер переполнен
SM76.7	Бит холостого хода PTO1: 0 = PTO активен, 1 = PTO не активен
SM77.0	Актуализация значения времени цикла PTO1/PWM1: 1 = записать новое значение времени цикла
SM77.1	Актуализация значения ширины импульсов PWM1: 1 = записать новую ширину импульсов
SM77.2	Актуализация значения количества импульсов PTO1: 1 = записать новое количество импульсов
SM77.3	База времени PTO1/PWM1: 0 = 1 мкс/такт; 1 = 1 мс/такт
SM77.4	Обновлять PWM1 синхронно: 0 = асинхронное обновление, 1 = синхронное обновление
SM77.5	Режим работы PTO1: 0 = односегментный режим (время цикла и количество импульсов хранятся в SM-памяти), 1 = многосегментный режим (таблица профиля хранится в памяти переменных)
SM77.6	Выбор PTO1 или PWM1: 0 = PTO, 1 = PWM
SM77.7	Бит разблокировки PTO1/PWM1: 1 = разблокировано
SMW78	Значение времени цикла PTO1/PWM1 (от 2 до 65 535 единиц базы времени)
SMW80	Значение ширины импульсов PWM1 (от 0 до 65 535 единиц базы времени)
SMD82	Количество импульсов PTO1 (от 1 до $2^{32}-1$);

SMB86 – SMB94 и SMB186 – SMB194: управление приемом сообщений

Как описано в таблице D–16, SMB86 – SMB94 и SMB186 – SMB194 используются для управления и чтения состояния команды приема сообщения.

Таблица D–16. Байты специальной памяти SMB86 – SMB94 и SMB186 – SMB194

Порт 0	Порт 1	Описание								
SMB86	SMB186	<p>Байт состояния приема сообщения</p> <p style="text-align: right;">MSB 7</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>n</td> <td>r</td> <td>e</td> <td>0</td> <td>0</td> <td>t</td> <td>c</td> <td>p</td> </tr> </table> <p style="text-align: right;">LSB 0</p> <p>n: 1 = Прием сообщения завершен командой пользователя r: 1 = Прием сообщения завершен: ошибка во входных параметрах или отсутствие условия начала или конца e: 1 = Получен символ конца t: 1 = Прием сообщения завершен: истекло время c: 1 = Прием сообщения завершен: достигнуто максимальное количество символов p: 1 = Прием сообщения завершен из-за ошибки, обнаруженной при проверке четности</p>	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	<p>Байт управления приемом сообщений</p> <p style="text-align: right;">MSB 7</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>en</td> <td>sc</td> <td>ec</td> <td>il</td> <td>c/m</td> <td>tmr</td> <td>bk</td> <td>0</td> </tr> </table> <p style="text-align: right;">LSB 0</p> <p>en: 0 = Функция приема сообщений заблокирована. 1 = Функция приема сообщений разблокирована. Бит блокировки/деблокировки приема сообщений проверяется при каждом выполнении команды RCV. sc: 0 = Игнорировать SMB88 или SMB188. 1 = Использовать SMB88 или SMB188 для обнаружения начала сообщения. ec: 0 = Игнорировать SMB89 или SMB189. 1 = Использовать SMB89 или SMB189 для обнаружения конца сообщения. il: 0 = Игнорировать SMW90 или SMW190. 1 = Использовать значение SMW90 или SMW190 для обнаружения бездействия линии. c/m: 1 = Таймер измеряет время между символами. 0 = Таймер измеряет время сообщения. tmr: 1 = Игнорировать SMW92 или SMW192. 0 = Завершить прием, если превышено время, указанное в SMW92 или SMW192. bk: 1 = Игнорировать паузу. 0 = Использовать паузу для обнаружения начала сообщения.</p>	en	sc	ec	il	c/m	tmr	bk	0
en	sc	ec	il	c/m	tmr	bk	0			
SMB88	SMB188	Символ начала сообщения								
SMB89	SMB189	Символ конца сообщения								
SMW90	SMW190	Время бездействия линии в миллисекундах. Первый символ, принятый по истечении времени бездействия линии, является началом нового сообщения.								
SMW92	SMW192	Значение контрольного времени при измерении времени между символами и времени сообщения в миллисекундах. Если этот интервал времени истек, то прием сообщения завершается.								
SMB94	SMB194	<p>Максимальное количество символов, которое должно быть принято (от 1 до 255 байтов).</p> <p>Примечание: Этот диапазон должен быть установлен на ожидаемый максимальный размер буфера, даже если завершение приема с помощью подсчета символов не используется.</p>								

SMW98: ошибки в шине расширения

Как описано в таблице D–17, SMW98 дает информацию о количестве ошибок в шине расширения ввода/вывода.

Таблица D–17. Байты специальной памяти SMW98

SM-слово	Описание
SMW98	Эта ячейка увеличивается на 1 каждый раз, когда обнаруживается контрольная четности в шине расширения ввода/вывода. Она очищается после включения питания и может быть очищена пользователем.

SMB130: регистр управления свободно программируемым обменом данными (см. SMB30)

См. таблицу D–11.

SMB131 – SMB165: регистры HSC3, HSC4 и HSC5

Как описано в таблице D–18, байты от SMB131 до SMB165 используются для контроля и управления скоростными счетчиками HSC3, HSC4 и HSC5.

Таблица D–18. Байты специальной памяти SMB131 – SMB165

SM-байт	Описание
SMB131 – SMB135	Резерв
SM136.0 – SM136.4	Резерв
SM136.5	Бит состояния текущего направления счета HSC3: 1 = прямой счет
SM136.6	Бит состояния «текущее значение равно предустановленному значению» для HSC3: 1 = равно
SM136.7	Бит состояния «текущее значение больше предустановленного значения» для HSC3: 1 = больше
SM137.0 – SM137.2	Резерв
SM137.3	Бит управления направлением счета HSC3: 1 = прямой счет
SM137.4	Актуализация направления HSC3: 1 = актуализировать направление
SM137.5	Актуализация предустановленного значения HSC3: 1 = записать новое предустановленное значение в HSC3
SM137.6	Актуализация текущего значения HSC3: 1 = записать новое текущее значение в HSC3
SM137.7	Бит разблокировки HSC3: 1 = разблокировать
SMD138	Новое текущее значение HSC3
SMD142	Новое предустановленное значение HSC3
SM146.0 – SM146.4	Резерв
SM146.5	Бит состояния текущего направления счета HSC4: 1 = прямой счет
SM146.6	Бит состояния «текущее значение равно предустановленному значению» для HSC4: 1 = равно
SM146.7	Бит состояния «текущее значение больше предустановленного значения» для HSC4: 1 = больше
SM147.0	Бит управления уровнем активности для сброса HSC1: 0 = активен при высоком уровне сигнала, 1 = активен при низком уровне сигнала
SM147.1	Резерв
SM147.2	Выбор скорости счета для квадратурных счетчиков: 0 = 4-кратная скорость; 1 = 1-кратная скорость
SM147.3	Бит управления направлением счета HSC4: 1 = прямой счет
SM147.4	Актуализация направления HSC4: 1 = актуализировать направление

Таблица D–18. Байты специальной памяти SMB131 – SMB165, продолжение

SM-байт	Описание
SM147.5	Актуализация предустановленного значения HSC4: 1 = записать новое предустановленное значение в HSC4
SM147.6	Актуализация текущего значения HSC4: 1 = записать новое текущее значение в HSC4
SM147.7	Бит разблокировки HSC4: 1 = разблокировать
SMD148	Новое текущее значение HSC4
SMD152	Новое предустановленное значение HSC4
SM156.0 – SM156.4	Резерв
SM156.5	Бит состояния текущего направления счета HSC5: 1 = прямой счет
SM156.6	Бит состояния «текущее значение равно предустановленному значению» для HSC5: 1 = равно
SM156.7	Бит состояния «текущее значение больше предустановленного значения» для HSC5: 1 = больше
SM157.0 – SM157.2	Резерв
SM157.3	Бит управления направлением счета HSC5: 1 = прямой счет
SM157.4	Актуализация направления HSC5: 1 = актуализировать направление
SM157.5	Актуализация предустановленного значения HSC5: 1 = записать новое предустановленное значение в HSC5
SM157.6	Актуализация текущего значения HSC5: 1 = записать новое текущее значение в HSC5
SM157.7	Бит разблокировки HSC5: 1 = разблокировать
SMD158	Новое текущее значение HSC5
SMD162	Новое предустановленное значение HSC5

SMB166 – SMB185: таблица определения профилей PTO0, PTO1

Как описано в таблице D–19, байты SMB166 – SMB185 используются, чтобы показать количество шагов активного профиля и адрес таблицы профиля в памяти переменных.

Таблица D–19. Байты специальной памяти SMB166 – SMB185

SM-байт	Описание
SMB166	Текущее количество записей активного шага профиля для PTO0
SMB167	Резерв
SMD168	Адрес в памяти переменных таблицы профиля для PTO0, заданный как смещение от V0.
SMB170	Байт состояния линейного PTO0
SMB171	Байт результата линейного PTO0
SMD172	Задаёт частоту, которая должна генерироваться, когда генератор линейного PTO0 работает в ручном режиме. Частота задается в виде двойного целого числа в герцах. SMB172 – это старший байт, а SMB175 – младший байт
SMB176	Текущее количество записей активного шага профиля для PTO1
SMB177	Резерв
SMD178	Адрес в памяти переменных таблицы профиля для PTO1, заданный как смещение от V0.
SMB180	Байт состояния линейного PTO1
SMB181	Байт результата линейного PTO1
SMD182	Задаёт частоту, которая должна генерироваться, когда генератор линейного PTO1 работает в ручном режиме. Частота задается в виде двойного целого числа в герцах. SMB182 – это старший байт, а SMB178 – младший байт

SMB186 – SMB194: управление приемом сообщений (см. SMB86 – SMB94)

См. таблицу D–16.

SMB200 – SMB549: состояние интеллектуальных модулей

Как показано в таблице D–20, байты SMB200 – SMB549 зарезервированы для информации, предоставляемой интеллектуальными модулями расширения, например, модулем EM 277 PROFIBUS-DP. Информацию о том, как ваш модуль использует байты SMB200 – SMB549, вы найдете в данных вашего конкретного модуля в Приложении A.

Для CPU S7–200 с ПЗУ версии, меньшей 1.2, для обеспечения совместимости вы должны установить интеллектуальный модуль рядом с CPU.

Таблица D–20. Байты специальной памяти SMB200 – SMB549

Байты специальной памяти SMB200 – SMB549							
Интеллектуальный модуль в слоте 0	Интеллектуальный модуль в слоте 1	Интеллектуальный модуль в слоте 2	Интеллектуальный модуль в слоте 3	Интеллектуальный модуль в слоте 4	Интеллектуальный модуль в слоте 5	Интеллектуальный модуль в слоте 6	Описание
SMB200 – SMB215	SMB250 – SMB265	SMB300 – SMB315	SMB350 – SMB365	SMB400 – SMB415	SMB450 – SMB465	SMB500 – SMB515	Имя модуля (16 символов ASCII)
SMB216 – SMB219	SMB266 – SMB269	SMB316 – SMB319	SMB366 – SMB369	SMB416 – SMB419	SMB466 – SMB469	SMB516 – SMB519	Номер версии ПО (4 символа ASCII)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	Код ошибки
SMB222 – SMB249	SMB272 – SMB299	SMB322 – SMB349	SMB372 – SMB399	SMB422 – SMB449	SMB472 – SMB499	SMB522 – SMB549	Информация, зависящая от конкретного типа модуля

Номера для заказа S7-200



CPU	Номер для заказа
CPU 221 DC/DC/DC 6 входов/4 выхода	6ES7 211-0AA23-0XB0
CPU 221 AC/DC/Relay 6 входов/4 реле	6ES7 211-0BA23-0XB0
CPU 222 DC/DC/DC 8 входов/6 выходов	6ES7 212-1AB23-0XB0
CPU 222 AC/DC/Relay 8 входов/6 реле	6ES7 212-1BB23-0XB0
CPU 224 DC/DC/DC 14 входов/10 выходов	6ES7 214-1AD23-0XB0
CPU 224 AC/DC/Relay 14 входов/10 реле	6ES7 214-1BD23-0XB0
CPU 224XP DC/DC/DC 14 входов/10 выходов	6ES7 214-2AD23-0XB0
CPU 224XP AC/DC/Relay 14 входов/10 реле	6ES7 214-2BD23-0XB0
CPU 226 DC/DC/DC 24 входа/16 выходов	6ES7 216-2AD23-0XB0
CPU 226 AC/DC/Relay 24 входа/16 реле	6ES7 216-2BD23-0XB0
Модули расширения	Номер для заказа
EM 221, 24 В пост. тока, цифровой, 8 входов	6ES7 221-1BF22-0XA0
EM 221, цифровой, 8 входов перем. тока (8 x 120/230 В перем. тока)	6ES7 221-1EF22-0XA0
EM 221, цифровой ввод, 16 x 24 В пост. тока	6ES7 221-1BH22-0XA0
EM 222, 24 В пост. тока, цифровой, 8 выходов	6ES7 222-1BF22-0XA0
EM 222, цифровой вывод, 8 реле	6ES7 222-1HF22-0XA0
EM 222, цифровой, 8 выходов перем. тока (8 x 120/230 В перем. тока)	6ES7 222-1EF22-0XA0
EM 222, цифровой вывод, 4 x 24 В пост. тока – 5 А	6ES7 222-1BD22-0XA0
EM 222, цифровой вывод, 4 реле – 10 А	6ES7 222-1HD22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 4 входа/4 выхода	6ES7 223-1BF22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 4 входа/4 релейных выхода	6ES7 223-1HF22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 8 входов/8 выходов	6ES7 223-1BH22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 8 входов/8 релейных выходов	6ES7 223-1PH22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 16 входов/16 выходов	6ES7 223-1BL22-0XA0
EM 223, 24 В пост. тока, цифровой ввод-вывод, 16 входов/16 релейных выходов	6ES7 223-1PL22-0XA0
EM 231, аналоговый ввод, 4 входа	6ES7 231-0HC22-0XA0
EM 231, аналоговый ввод, RTD, 2 входа	6ES7 231-7PB22-0XA0
EM 231, аналоговый ввод, термopара, 4 входа	6ES7 231-7PD22-0XA0
EM 232, аналоговый вывод, 2 выхода	6ES7 232-0HB22-0XA0
EM 235, аналоговый ввод-вывод, 4 входа/1 выход	6ES7 235-0KD22-0XA0
EM 241, модемный модуль	6ES7 241-1AA22-0XA0
EM 253, модуль позиционирования	6ES7 253-1AA22-0XA0
EM 277 PROFIBUS-DP	6ES7 277-0AA22-0XA0
Модуль AS Interface (CP 243-2)	6GK7 243-2AX01-0XA0
Модуль Internet (CP 243-1 IT) (с электронной документацией на компакт-диске)	6GK7 243-1GX00-0XE0
Модуль Ethernet (CP 243-1) (с электронной документацией на компакт-диске)	6GK7 243-1EX00-0XE0

Сменные модули и кабели	Номер для заказа
Модуль памяти, 32К (программа пользователя)	6ES7 291-8GE20-0XA0
Модуль памяти, 64К (программа пользователя, рецепт и протокол данных)	6ES7 291-8GF23-0XA0
Модуль памяти, 256К (программа пользователя, рецепт и протокол данных)	6ES7 291-8GH23-0XA0
Часы реального времени с батарейным модулем	6ES7 297-1AA23-0XA0
Батарейный модуль BC 293, CPU 22х	6ES7 291-8BA20-0XA0
Кабель для модулей расширения, .8 метров, CPU 22х/EM	6ES7 290-6AA20-0XA0
Кабель, RS-232/PPI Multi-Master	6ES7 901-3CB30-0XA0
Кабель, USB/PPI Multi-Master	6ES7 901-3DB30-0XA0
Программное обеспечение	Номер для заказа
STEP 7-Micro/WIN 32 (V4.0), отдельная лицензия (компакт-диск)	6ES7 810-2CC03-0YX0
STEP 7-Micro/WIN 32 (V4.0), лицензия на обновление (компакт-диск)	6ES7 810-2CC03-0YX3
S7-200 Toolbox: TP-Designer для TP070, версия 1.0 (компакт-диск)	6ES7 850-2BC00-0YX0
STEP 7-Micro/WIN, дополнительный пакет: библиотека команд STEP 7-Micro/WIN 32, V1.1 (компакт-диск)	6ES7 830-2BC00-0YX0
S7-200 PC Access V1.0 (OPC-сервер), отдельная лицензия	6ES7 840-2CC01-0YX0
S7-200 PC Access V1.0 (Version), лицензия на многократное копирования	6ES7 840-2CC01-0YX1
Конфигурационное программное обеспечение WinCC Flexible Micro	6AV6 610-0AA01-0AA0

Коммуникационные платы	Номер для заказа
CP 5411: укороченная AT ISA	6GK 1 541-1AA00
CP 5511: PCMCIA, Type II	6GK 1 551-1AA00
CP 5611: плата PCI (версия 3.0 или выше)	6GK 1 561-1AA00
Руководства	Номер для заказа
Системное руководство «Программируемый контроллер S7-200» (немецкий язык)	6ES7 298-8FA24-8AH0
Системное руководство «Программируемый контроллер S7-200» (английский язык)	6ES7 298-8FA24-8BH0
Системное руководство «Программируемый контроллер S7-200» (французский язык)	6ES7 298-8FA24-8CH0
Системное руководство «Программируемый контроллер S7-200» (испанский язык)	6ES7 298-8FA24-8DH0
Системное руководство «Программируемый контроллер S7-200» (итальянский язык)	6ES7 298-8FA24-8EH0
Руководство S7-200 Point-to-Point Interface Communication [Обмен данными через двухточечный интерфейс S7-200] (английский/немецкий)	6ES7 298-8GA00-8XH0
TP070 Touch Panel [Сенсорная панель TP070], руководство пользователя (английский язык)	6AV6 591-1DC01-0AB0
TP170 micro, руководство по эксплуатации (английский язык)	6AV6 691-1DB01-0AB0
Руководство CP 243-2 SIMATIC NET AS-Interface Master (английский язык)	6GK7 243-2AX00-8BA0
WinCC Flexible Micro User Manual (английский язык)	6AV6 691-1AA01-0AB0

Кабели, сетевые соединители и повторители	Номер для заказа
Кабель MPI	6ES7 901-0BF00-0AA0
Сетевой кабель PROFIBUS	6XVI 830-0AH10
Штекер для подключения к сетевой шине с портом для устройства программирования, вертикальный кабельный вывод	6ES7 972-0BB11-0XA0
Штекер для подключения к сетевой шине (без порта для устройства программирования), вертикальный кабельный вывод	6ES7 972-0BA11-0XA0
Штекер RS–485 для подключения к шине с кабельным выводом под углом 35° (без порта для устройства программирования)	6ES7 972-0BA40-0XA0
Штекер RS–485 для подключения к шине с кабельным выводом под углом 35° (с портом для устройства программирования)	6ES7 972-0BB40-0XA0
Клеммный блок CPU 22х/EM, 7 клемм, съемный	6ES7 292-1AD20-0AA0
Клеммный блок CPU 22х/EM, 12 клемм съемный	6ES7 292-1AE20-0AA0
Клеммный блок CPU 22х/EM, 14 клемм съемный	6ES7 292-1AF20-0AA0
Клеммный блок CPU 22х/EM, 18 клемм съемный	6ES7 292-1AG20-0AA0
Повторитель RS–485, IP 20, с гальванической развязкой	6ES7 972-0AA00-0XA0
Интерфейсы оператора	Номер для заказа
Текстовый дисплей TD 200	6ES7 272-0AA30-0YA0
Текстовый дисплей TD 200C (заказная передняя панель) ¹	6ES7 272-1AA10-0YA0
Панель оператора OP3	6AV3 503-1DB10T
Панель оператора OP7	6AV3 607-1JC20-0AX1
Панель оператора OP17	6AV3 617-1JC20-0AX1
Сенсорная панель TP070	6AV6 545-0AA15-2AX0
Сенсорная панель TP170 micro	6AV6 640-0CA01-0AX0
Разное	Номер для заказа
Упоры для профильной шины	6ES5 728-8MAII
Клеммный блок с 12 зажимами для полевой проводки (CPU 221, CPU 222) 10 штук	6ES7 290-2AA00-0XA0
Комплект запасных крышек, содержит по 4 элемента каждого из следующих видов: крышки для клеммных блоков на 7, 12, 14, 18, 2х12, 2х14 клемм; дверца для доступа к CPU, дверца для доступа к EM	6ES7 291-3AX20-0XA0
Имитатор с 8 клеммами	6ES7 274 1XF00-0XA0
Имитатор с 14 клеммами	6ES7 274 1XH00-0XA0
Имитатор с 24 клеммами	6ES7 274 1XK00-0XA0
Шаблоны для передней панели TD 200C (пакет пустых шаблонов)	6ES7 272-1AF00-7AA0

- 1 Содержит пустой шаблон для заполнения пользователем. Если Вам нужны дополнительные пустые шаблоны, закажите шаблоны для передней панели TD 200C.

Времена выполнения команд STL

F

Времена выполнения команд очень важны, если ваше приложение содержит функции, критичные к времени. Времена выполнения команд показаны в таблице F–2.



Совет

При использовании времен исполнения из таблицы F–2 вам следует учесть влияние на эти времена потока сигнала, косвенной адресации и обращения к определенным областям памяти. Эти факторы могут непосредственно влиять на приведенные времена исполнения.

Влияние потока сигнала

В таблице F–2 показано время, необходимое для выполнения командой логической операции или функции при наличии для этой команды потока сигнала (когда значение вершины стека равно 1).

Если поток сигнала отсутствует, то время выполнения для этой команды равно 1 мкс.

Влияние косвенной адресации

В таблице F–2 показано время, необходимое для выполнения командой логической операции или функции, при использовании прямой адресации операндов или константы.

При использовании командой косвенной адресации операндов время выполнения для этой команды увеличивается на 14 мкс для каждого косвенно адресованного в этой команде операнда.

Влияние обращения к определенным областям памяти

Обращение к определенным областям памяти, например, AI, AQ, L и аккумуляторам, требует дополнительного времени на выполнение команды.

В таблице F–1 показано дополнительное время, которое необходимо добавить к времени выполнения команды, когда в операнде указаны эти области памяти.

Таблица F–1. Добавка к времени выполнения при обращении к указанным областям памяти

Область памяти	Добавка к времени выполнения
Локальный аналоговый вход (AI) фильтрация деактивизирована	9,4 мкс
фильтрация активизирована	8,4 мкс
Аналоговый вход расширения (AI) фильтрация деактивизирована	134 мкс
фильтрация активизирована	8,4 мкс
Локальный аналоговый выход (AQ)	92 мкс
Аналоговый выход расширения (AQ)	48 мкс
Локальная память (L)	2,8 мкс
Аккумуляторы (AC)	2,8 мкс

Таблица F-2. Времена выполнения команд

Команда	мкс	Команда	мкс
= Используется: I SM, T, C, V, S, Q, M L	0,24 1,3 10,5	BITIM	16
+D	29	BIR Используется: локальные входы выходы расширения	23 30
-D	29	BIW Используется: локальные выходы выходы расширения	24 32
*D	47	BMB Время = База + (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	10 28 5,7
/D	250	BMD Время= База + (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	11 29 10,6
+I	25	BMW Время= База + (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	10 28 8,6
-I	25	BTI	16
*I	37	CALL Без параметров: С параметрами: Время = База + Σ (время операнда) База Время операнда бит (вход, выход) байт (вход, выход) слово (вход, выход) двойн. слово (вход, выход)	9 14 10, 11 8, 7 10, 9 12, 10
/I	64	Указание: обработка выходных операндов происходит при возврате из подпрограммы	
=I Используется: локальные выходы выходы расширения	16 24	CEVNT	24
+R	71 тип, 99 макс.	CFND Максимальное время = База + N1 < ((LM1 < N2) + LM2) База Умножитель длины 1 (LM1) Умножитель длины 2 (LM2) N1 – длина исходной строки N2 - длина набора символов - строки символов	35 8,6 9,5
-R	72 тип, 100 макс.	CITIM	23
*R	56 тип. 166 макс.	COS	900 тип. 1070 макс.
/R	177 тип. 230 макс.	CRET Имеется поток сигнала Поток сигнала отсутствует	16 0,8
A Используется: I SM, T, C, V, S, Q, M L	0,22 0,72 6,1	CRETI Поток сигнала отсутствует	0,2
AB <=, =, >=, >, <, <>	18	CSCRE	3,1
AD <=, =, >=, >, <, <>	27	CTD При изменении счетного входа Иначе	27 19
AENO	0,4	CTU При изменении счетного входа Иначе	31 19
AI Используется: локальные входы выходы расширения	15 21	CTUD При изменении счетного входа Иначе	37 24
ALD	0,22	DECB	16
AN Используется: I SM, T, C, V, S, Q, M L	0,22 0,72 6,1	DECD	22
ANDB	19	DECO	19
ANDD	30	DECW	20
ANDW	25	DISI	9
ANI Используется: локальные входы выходы расширения	15 21	DIV	67
AR <=, =, >=, >, <, <>	29		
AS=, <> Время = База + (LM < N) База Умножитель длины (LM) N – число сравниваемых символов	33 6,3		
ATCH	12		
ATH Время = База + (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	23 31 10,2		
ATT	36		
AW <=, =, >=, >, <, <>	23		
BCDI	35		

Команда	МКС	Команда	МКС
DLED	14	LD Используется: I SM, T, C, V, S, Q, M L	0,22 0,8 6
DTA	302	LDB <=, =, >=, >, <, <>	18
DTI	21	LDD <=, =, >=, >, <, <>	27
DTCH	12	LDI Используется: локальные входы входы расширения	15 21
DTR	35 тип. 40 макс.	LDN Используется: I SM, T, C, V, S, Q, M L	0,3 0,9 6,1
DTS	305	LDNI Используется: локальные входы входы расширения	15 21
ED	8	LDR<=, =, >=, >, <, <>	29
ENCO	24 макс.	LDS	0,22
END Поток сигнала отсутствует	0,2	LDS=, <> Время = База + (LM < N) База Умножитель длины (LM) N - число сравниваемых символов	33 6,3
ENI	11	LDW <=, =, >=, >, <, <>	24
EU	8	LIFO	37
EXP	720 тип. 860 макс.	LN	680 тип. 820 макс.
FIFO Время = База + (длина<LM) База Умножитель длины (LM)	30 7	LPP	0,22
FILL Время= База + (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	15 29 3,2	LPS	0,24
FND <, =, >, <> Время = База + (длина<LM) База Умножитель длины (LM)	39 6,5	LRD	0,22
FOR Время = База + (число циклов<LM) База Множитель циклов (LM)	35 28	LSCR	7,3
GPA	16	MOVB	15
HDEF	18	MOVD	20
HSC	30	MOVR	20
HTA Время= База+ (длина<LM) База (постоянная длина) База (переменная длина) Умножитель длины (LM)	20 28 5,2	MOVW	18
IBCD	52	MUL	37
INCB	15	NETR	99
INCD	22	NETW Время = База + (LM < N) База Умножитель длины (LM) N – число байтов для передачи	95 4
INCW	20	NEXT	0
INT Обычно при 2 прерывании	24	NOP	0,22
INVB	16	NOT	0,22
INVD	22	O Используется: I SM, T, C, V, S, Q, M L	0,22 0,72 6,4
INVW	20	OB <=, =, >=, >, <, <>	18
ITA	136	OD <=, =, >=, >, <, <>	26
ITB	17	OI Используется: локальные входы входы расширения	15 21
ITD	20	OLD	0,22
ITS	139	ON Используется: I SM, T, C, V, S, Q, M L	0,22 0,72 6,4
JMP	1,8		
LBL	0,22		

Команда	МКС	Команда	МКС
ONI	Используется: локальные входы входы расширения 15 21	RRW	Время = База + (LM<N) База Умножитель длины (LM) N – это счетчик сдвигов 26 1,2
OR<=, =, >=, >, <, <>	29	RTA	Время = База + (LM < N) База (для первой цифры в результате) Умножитель длины (LM) N - число дополнительных цифр в результате 149 96
ORB	19	RTS	Время = База + (LM < N) База (для первой цифры в результате x) Умножитель длины (LM) N - число дополнительных цифр в результате 154 96
ORD	29	S	Длина = 1 и задана как константа Иначе: Время = База + (длина<LM) База Умножитель длины (LM) Если длина хранится как переменная, добавьте к базе 2,9 14 0,5 17
ORW	25	SCAT	Время = База + (LM < N) База Умножитель длины (LM) N – число присоединенных символов 30 5,3
OS=, < >	Время + База + (LM < N) База Умножитель длины (LM) N - число сравниваемых символов 33 6,3	SCPY	Время = База + (LM < N) База Умножитель длины (LM) N – число копируемых символов 27 4,6
OW <=, =, >=, >, <, <>	24	SCORE	0,24
PID	Обычно Переход из ручного режима в автоматический Коэффициент пересчет Автонастройка 400 800 макс. 770 макс. 650 макс.	SCRT	10
PLS:	Используется: PWM PTO, односегм. режим PTO, многосегм. режим 31 36 50	SEG	15
R	Длина=1 и определена как константа База для счетчиков (C) База для таймеров (T) База для всех остальных Иначе: Время = База + (длина < LM) База для счетчиков База для таймеров (T) База для всех остальных Умножитель длины (LM) для операнда C Умножитель длины (LM) для операнда T Умножитель длины (LM) для всех остальных Если длина хранится как переменная, добавьте к базе 9,3 16 2,9 8,6 8,3 14 5,1 9,9 0,5 17	SFND	Максимальное время = База + ((N1-N2) < LM2) + (N2<LM1) База Умножитель длины 1 (LM1) Умножитель длины 2 (LM2) N1 - длина исходной строки N2 – длина строки поиска 39 7,6 6,8
RCV	51	SHRB	Время = База + (длина<LM1) + ((длина /8) < LM2) База (постоянная длина) База (переменная длина) Умножитель длины 1 (LM1) Умножитель длины 2 (LM2) 48 52 1,0 1,5
RET	16	SI	Время = База + (длина<LM) База LM у локального выхода LM у выхода расширения Если длина хранится как переменная, добавьте к базе 8,9 13 21 17
RI	Время = База + (длина<LM) База Умножитель длины (LM) у локальных выходов Умножитель длины (LM) у выходов расширения Если длина хранится как переменная, добавьте к базе 8,9 13 21 17	SIN	900 тип. 1070 макс.
RLB	Время = База + (LM < N) База Умножитель длины (LM) N – это счетчик сдвигов 23 0,2	SLB	Время = База + (LM<N) База Умножитель длины (LM) N - это счетчик сдвигов 23 0,2
RLD	Время = База + (LM<N) База Умножитель длины (LM) N – это счетчик сдвигов 28 1,4	SLD	Время = База + (LM<N) База Умножитель длины (LM) N - это счетчик сдвигов 29 1,1
RLW	Время = База + (LM<N) База Умножитель длины (LM) N – это счетчик сдвигов 27 0,9		
ROUND	56 тип. 110 макс.		
RRB	Время = База + (LM<N) База Умножитель длины (LM) N – это счетчик сдвигов 22 0,5		
RRD	Время = База + (LM<N) База Умножитель длины (LM) N - это счетчик сдвигов 28 1,7		

Команда	МКС	Команда	МКС
SLEN	21	STI	Время = База + (LM < N) База (для 1-го символа источника) Умножитель длины (LM) N - число дополнительных символов источника
SLW	27 0,6		58 27
SPA	371	STOP	Поток сигнала отсутствует
SQRT	460 тип. 550 макс.	STR	Время = База + (LM < N) База (для 1-го символа источника) Умножитель длины (LM) N - число дополнительных символов источника
SRB	22 0,6	SWAP	17
SRD	28 1,5	TAN	1080 тип. 1300 макс.
SRW	27 1	TODR	331
SSCPY	42 5,3	TODRX	391 тип. 783 тип.
STD	69 27	TODW	436
		TODWX	554
		TOF	36
		TON	33
		TONR	32
		TRUNC	53 тип. 106 макс.
		WDR	7
		XMT	42
		XORB	19
		XORD	29
		XORW	25

Краткая справочная информация об S7-200



Чтобы облегчить вам поиск информации, в этом разделе собраны следующие данные:

- биты специальной памяти
- описания событий, вызывающих прерывания
- обзор областей памяти и характеристик CPU S7-200
- скоростные счетчики HSC0, HSC1, HSC2, HSC3, HSC4, HSC5
- команды S7-200

Таблица G-1. Биты специальной памяти

Биты специальной памяти			
SM0.0	Всегда включен	SM1.0	Результат операции = 0
SM0.1	Первый цикл	SM1.1	Переполнение или недопустимое значение
SM0.2	Потеряны сохраняемые данные	SM1.2	Отрицательный результат
SM0.3	Включение питания	SM1.3	Деление на 0
SM0.4	30 с выкл./ 30 с вкл.	SM1.4	Таблица заполнена
SM0.5	0,5 с выкл./ 0,5 с вкл.	SM1.5	Таблица пуста
SM0.6	1 цикл выкл./1 цикл вкл.	SM1.6	Ошибка преобразования BCD в двоичный код
SM0.7	Переключатель в положении RUN	SM1.7	Ошибка преобразования ASCII в 16-ричный код

Таблица G–2. Прерывающие события по приоритетам

Номер события	Описание прерывания	Группа приоритета	Приоритет в группе
8	Порт 0: прием символа	Коммуникационные прерывания (наивысшая)	0
9	Порт 0: передача завершена		0
23	Порт 0: прием сообщения завершен		0
24	Порт 1: прием сообщения завершен		1
25	Порт 1: прием символа		1
26	Порт 1: передача завершена		1
19	Прерывание при завершении РТО 0	Прерывания по вводу/выводу (средняя)	0
20	Прерывание при завершении РТО 1		1
0	IO.0, нарастающий фронт		2
2	IO.1, нарастающий фронт		3
4	IO.2, нарастающий фронт		4
6	IO.3, нарастающий фронт		5
1	IO.0, падающий фронт		6
3	IO.1, падающий фронт		7
5	IO.2, падающий фронт		8
7	IO.3, падающий фронт		9
12	HSC0: CV=PV (текущее значение = предустановленному)		10
27	HSC0: направление изменено		11
28	HSC0: внешний сброс		12
13	HSC1: CV=PV (текущее значение = предустановленному)		13
14	HSC1: направление изменено		14
15	HSC1: внешний сброс		15
16	HSC2: CV=PV		16
17	HSC2: направление изменено		17
18	HSC2: внешний сброс		18
32	HSC3: CV=PV (текущее значение = предустановленному)		19
29	HSC4: CV=PV (текущее значение = предустановленному)		20
30	HSC4: направление изменено		21
31	HSC4: внешний сброс		22
33	HSC5: CV=PV (текущее значение = предустановленному)	23	
10	Прерывание, управляемое временем, 0	Прерывания, управляемые временем (низшая)	0
11	Прерывание, управляемое временем, 1		1
21	Таймер T32, прерывание по СТ=РТ		2
22	Таймер T96, прерывание по СТ=РТ		3

Таблица G-3. Области памяти и функции CPU S7-200

Описание	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Размер программы пользователя с редактированием в режиме RUN без редактирования в режиме RUN	4096 байтов	4096 байтов	8192 байта	12288 байтов	16384 байта
	4096 байтов	4096 байтов	12288 байтов	16384 байта	24576 байтов
Размер данных пользователя	2048 байтов	2048 байтов	8192 байта	10240 байтов	10240 байтов
Регистр входов образа процесса	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7
Регистр выходов образа процесса	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7
Аналоговые входы (только чтение)	AIW0 – AIW30	AIW0 – AIW30	AIW0 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	AQW0 – AQW30	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB8191	VB0 – VB10239	VB0 – VB10239
Локальная память (L) ¹	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63
Битовая память (M)	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7
Специальная память (SM) Только чтение	SM0.0 – SM179.7 SM0.0 – SM29.7	SM0.0 – SM299.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7
Таймеры	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)
Задержка включения с запоминанием	1 мс	T0, T64	T0, T64	T0, T64	T0, T64
	10 мс	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68	T1 – T4 и T65 – T68
	100 мс	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95	T5 – T31 и T69 – T95
Задержка включения/ выключения	1 мс	T32, T96	T32, T96	T32, T96	T32, T96
	10 мс	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100	T33 – T36 и T97 – T100
	100 мс	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255	T37 – T63 и T101 – T255
Счетчики	C0 – C255	C0 – C255	C0 – C255	C0 – C255	C0 – C255
Скоростные счетчики	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5
Реле управления последовательностью (S)	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7
Аккумуляторные регистры	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3
Переходы/метки	0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
Вызов/подпрограмма	0 – 63	0 – 63	0 – 63	0 – 63	0 – 127
Программы обработки прерываний	0 – 127	0 – 127	0 – 127	0 – 127	0 – 127
Положительные/ отрицательные фронты	256	256	256	256	256
PID-регуляторы	0 – 7	0 – 7	0 – 7	0 – 7	0 – 7
Порты	Порт 0	Порт 0	Порт 0	Порт 0, Порт 1	Порт 0, Порт 1

¹ LB60 – LB63 зарезервированы STEP 7-Micro/WIN, версия 3.0 или выше.

Таблица G–4. Скоростные счетчики HSC0, HSC3, HSC4 и HSC5

Режим	HSC0			HSC3			HSC4			HSC5
	Датчик такт. имп.	Направление	Сброс	Датчик такт. имп.	Датчик такт. имп.	Направление	Сброс	Датчик такт. имп.	Датчик такт. имп.	
0	I0.0			I0.1	I0.3				I0.4	
1	I0.0		I0.2		I0.3		I0.5			
2										
3	I0.0	I0.1			I0.3	I0.4				
4	I0.0	I0.1	I0.2		I0.3	I0.4	I0.5			
5										
Режим	HSC0			HSC4			HSC5			
	Датчик такт. имп.: вперед	Датчик такт. имп.: назад	Сброс		Датчик такт. имп.: вперед	Датчик такт. имп.: назад	Сброс			
6	I0.0	I0.1			I0.3	I0.4				
7	I0.0	I0.1	I0.2		I0.3	I0.4	I0.5			
8										
Режим	HSC0			HSC4			HSC5			
	Фаза А	Фаза В	Сброс		Фаза А	Фаза В	Сброс			
9	I0.0	I0.1			I0.3	I0.4				
10	I0.0	I0.1	I0.2		I0.3	I0.4	I0.5			
11										
Режим	HSC0			HSC3			HSC5			
	Датчик такт. имп.			Датчик такт. имп.						
12	Q0.0			Q0.1						

Таблица G–5. Скоростные счетчики HSC1 и HSC2

Режим	HSC1				HSC2			
	Датчик такт. имп.	Датчик такт. имп.: назад	Сброс	Пуск	Датчик такт. имп.	Направление	Сброс	Пуск
0	I0.6				I1.2			
1	I0.6		I1.0		I1.2		I1.4	
2	I0.6		I1.0	I1.1	I1.2		I1.4	I1.5
3	I0.6	I0.7			I1.2	I1.3		
4	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
5	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
Режим	HSC1				HSC2			
	Датчик такт. имп.: вперед	Датчик такт. имп.: назад	Сброс	Пуск	Датчик такт. имп.: вперед	Датчик такт. имп.: назад	Сброс	Пуск
6	I0.6	I0.7	I1.0		I1.2	I1.3		
7	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
8	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
Режим	Фаза А	Фаза В	Сброс	Пуск	Фаза А	Фаза В	Сброс	Пуск
9	I0.6	I0.7			I1.2	I1.3		
10	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
11	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5

Булевы операции		
LD	Бит	Загрузить
LDI	Бит	Загрузить непосредственно
LDN	Бит	Загрузить инверсное значение
LDNI	Бит	Загрузить непосредственно инверсное значение
A	Бит	И
AI	Бит	Непосредственное И
AN	Бит	И-НЕ
ANI	Бит	Непосредственное И-НЕ
O	Бит	ИЛИ
OI	Бит	Непосредственное ИЛИ
ON	Бит	ИЛИ-НЕ
ONI	Бит	Непосредственное ИЛИ-НЕ
LDBx	IN1, IN2	Загрузить результат сравнения байтов IN1 (x:<, <=, =, >=, >, <>) IN2
ABx	IN1, IN2	Результат сравнения байтов IN1 (x:<, <=, =, >=, >, <>) IN2 И вершина стека
OBx	IN1, IN2	Результат сравнения байтов IN1 (x:<, <=, =, >=, >, <>) IN2 ИЛИ вершина стека
LDWx	IN1, IN2	Загрузить результат сравнения слов IN1 (x:<, <=, =, >=, >, <>) IN2
AWx	IN1, IN2	Результат сравнения слов IN1 (x:<, <=, =, >=, >, <>) IN2 И вершина стека
OWx	IN1, IN2	Результат сравнения слов IN1 (x:<, <=, =, >=, >, <>) IN2 ИЛИ вершина стека
LDDx	IN1, IN2	Загрузить результат сравнения двойных слов IN1 (x:<, <=, =, >=, >, <>) IN2
ADx	IN1, IN2	Результат сравнения двойных слов IN1 (x:<, <=, =, >=, >, <>) IN2 И вершина стека
ODx	IN1, IN2	Результат сравнения двойных слов IN1 (x:<, <=, =, >=, >, <>) IN2 ИЛИ вершина стека
LDRx	IN1, IN2	Загрузить результат сравнения вещественных чисел IN1 (x:<, <=, =, >=, >, <>) IN2
ARx	IN1, IN2	Результат сравнения вещественных чисел И вершина стека IN1 (x:<, <=, =, >=, >, <>) IN2
ORx	IN1, IN2	Результат сравнения вещественных чисел IN1 (x:<, <=, =, >=, >, <>) IN2 ИЛИ вершина стека
NOT		Инверсия стека
EU		Обнаружение нарастающ. фронта
ED		Обнаружение падающ. фронта
=	Бит	Присваивание значения
=I	Бит	Непосредственное присваивание значения
S	Бит, N	Установка N битов
R	Бит, N	Сброс N битов
SI	Бит, N	Непосредственная установка N битов
RI	Бит, N	Непосредственный сброс N битов
LDSx	IN1, IN2	Загрузить результат сравнения строк IN1 (x: =, <>) IN2
ASx	IN1, IN2	Результат сравнения строк IN1 (x: =, <>) IN2 И вершина стека
OSx	IN1, IN2	Результат сравнения строк IN1 (x: =, <>) IN2 ИЛИ вершина стека
ALD		И для 1-го и 2-го уровней стека
OLD		ИЛИ для 1-го и 2-го уровней стека
LPS		Дублирование вершины стека
LRD		Копирование 2-го уровня стека
LPP		Извлечение вершины стека
LDS	N	Дублирование N-го бита стека
AENO		И ENO

Арифметические операции, увеличение и уменьшение на 1		
+I	IN1, OUT	Сложить целые, двойные целые или вещественные числа
+D	IN1, OUT	IN1+OUT=OUT
+R	IN1, OUT	IN1+OUT=OUT
-I	IN1, OUT	Вычесть целые, двойные целые или вещественные числа
-D	IN1, OUT	OUT-IN1=OUT
-R	IN1, OUT	OUT-IN1=OUT
MUL	IN1, OUT	Перемножить целые числа (16*16->32)
*I	IN1, OUT	Перемножить целые, двойные целые или вещественные числа
*D	IN1, OUT	IN1 * OUT = OUT
*R	IN1, IN2	IN1 * OUT = OUT
DIV	IN1, OUT	Разделить целые числа (16/16->32)
/I	IN1, OUT	Разделить целые, двойные целые или вещественные числа
/D	IN1, OUT	OUT / IN1 = OUT
/R	IN1, OUT	OUT / IN1 = OUT
SQRT	IN, OUT	Квадратный корень
LN	IN, OUT	Натуральный логарифм
EXP	IN, OUT	Натуральная экспонента
SIN	IN, OUT	Синус
COS	IN, OUT	Косинус
TAN	IN, OUT	Тангенс
INCB	OUT	Увеличить на 1 байт, слово или двойное слово
INCW	OUT	Увеличить на 1 байт, слово или двойное слово
INCD	OUT	Увеличить на 1 байт, слово или двойное слово
DECB	OUT	Уменьшить на 1 байт, слово или двойное слово
DECW	OUT	Уменьшить на 1 байт, слово или двойное слово
DECD	OUT	Уменьшить на 1 байт, слово или двойное слово
PID	TBL, LOOP	PID-регулятор
PID-регулятор		
TON	Txxx, PT	Таймер с задержкой включения
TOF	Txxx, PT	Таймер с задержкой выключения
TONR	Txxx, PT	Таймер с задержкой включения с запоминанием
BITIM	OUT	Запуск интервального таймера
CITIM	IN, OUT	Расчет интервального таймера
CTU	Cxxx, PV	Прямой счет
CTD	Cxxx, PV	Обратный счет
CTUD	Cxxx, PV	Реверсивный счетчик
Часы реального времени		
TODR	T	Прочитать часы реального времени
TODW	T	Записать в часы реального времени
TODRX	T	Прочитать расширенные часы реального времени
TODWX	T	Установить расширенные часы реального времени
Команды управления программой		
END		Условный конец программы
STOP		Переход в состояние STOP
WDR		Сброс контроля времени (300 мс)
JMP	N	Перейти к указанной метке
LBL	N	Определить метку для перехода
CALL	N [N1...]	Вызвать подпрограмму [N1, ... до 16 возможных параметров]
CRET		Условный возврат из подпрограммы
FOR	INDX,INIT,FINAL	Цикл For/Next
NEXT		
LSCR	N	Загрузка, переход, условный конец
SCRT	N	и конец сегмента реле управления
CSCRE		последовательностью
SCORE		
DLED	IN	Диагностический светодиод

Команды передачи, сдвига и циклического сдвига		
MOVB IN, OUT MOVW IN, OUT MOVD IN, OUT MOVVR IN, OUT	Передать байт, слово, двойное слово, вещественное число	
BIR IN, OUT BIW IN, OUT	Передача непосредственно считанного байта Передача байта для непосредственной записи	
BMB IN, OUT, N BMW IN, OUT, N BMD IN, OUT, N	Групповая передача байтов, слов, двойных слов	
SWAP IN	Переставить местами байты	
SHRB DATA, S_BIT, N	Вдвинуть бит в регистр сдвига	
SRB OUT, N SRW OUT, N SRD OUT, N	Сдвинуть вправо байт, слово, двойное слово	
SLB OUT, N SLW OUT, N SLD OUT, N	Сдвинуть влево байт, слово, двойное слово	
RRB OUT, N RRW OUT, N RRD OUT, N	Циклически сдвинуть вправо байт, слово, двойное слово	
RLB OUT, N RLW OUT, N RLD OUT, N	Циклически сдвинуть влево байт, слово, двойное слово	
Логические операции		
ANDB IN1, OUT ANDW IN1, OUT ANDD IN1, OUT	Логическое И с байтами, словами и двойными словами	
ORB IN1, OUT ORW IN1, OUT ORD IN1, OUT	Логическое ИЛИ с байтами, словами и двойными словами	
XORB IN1, OUT XORW IN1, OUT XORD IN1, OUT	Логическое исключающее ИЛИ с байтами, словами и двойными словами	
INVB OUT INWV OUT INVD OUT	Инвертирование байта, слова или двойного слова (дополнение до 1)	
Строковые команды		
SLEN IN, OUT SCAT IN, OUT SCPY IN, OUT SSCPY IN, INDX, N, OUT CFND IN1, IN2, OUT SFND IN1, IN2, OUT	Длина строки Слияние строк Копирование строки Копирование подстроки из строки Поиск первого вхождения символа в строку Поиск строки в строке	
Табличные команды, поиск и преобразование		
ATT DATA, TBL	Добавить данные к таблице	
LIFO TBL, DATA FIFO TBL, DATA	Получить данные из таблицы	
FND= TBL, PTN, INDX FND<> TBL, PTN, INDX FND< TBL, PTN, INDX FND> TBL, PTN, INDX	Найти в таблице значение данных, удовлетворяющее заданному критерию	
FILL IN, OUT, N	Заполнить диапазон памяти по образцу	
BCDI OUT IBCD OUT	Преобразовать BCD в целое Преобразовать целое в BCD	
BTI IN, OUT ITB IN, OUT ITD IN, OUT	Преобразовать байт в целое Преобразовать целое в байт Преобразовать целое в двойное целое	
DTI IN, OUT	Преобразовать целое в двойное целое	
DTR IN, OUT	Преобразовать двойное целое в вещественное	
TRUNC IN, OUT	Преобразовать вещественное в двойное слово	
ROUND IN, OUT	Преобразовать вещественное в двойное целое	
ATH IN, OUT, LEN HTA IN, OUT, LEN ITA IN, OUT, FMT DTA IN, OUT, FM	Преобразовать ASCII в 16-ричн. Преобразовать 16-ричн. в ASCII Преобразовать целое в ASCII Преобразовать двойное целое в ASCII	
RTA IN, OUT, FM	Преобразовать вещественное в ASCII	
DECO IN, OUT ENCO IN, OUT	Декодировать Закодировать	
SEG IN, OUT	Генерировать 7-сегментное изображение	
ITS IN, FMT, OUT DTS IN, FMT, OUT	Преобразовать целое в строку Преобразовать двойное целое в строку	
RTS IN, FMT, OUT	Преобразовать вещественное в строку	
STI STR, INDX, OUT STD STR, INDX, OUT	Преобразовать подстроку в целое Преобразовать подстроку в двойное целое	
STR STR, INDX, OUT	Преобразовать подстроку в вещественное	
Команды прерывания		
CRETI	Условный возврат из прерывания	
ENI DISI	Разблокировать прерывания Заблокировать прерывания	
ATCH INT, EVNT DTCH EVNT	Назначить программу обработки прерывания событию Отсоединить событие	
Команды обмена данными		
XMT TBL, PORT RCV TBL, PORT	Передача в режиме свободно программируемого обмена данными Прием сообщения в режиме свободно программируемого обмена данными	
NETR TBL, PORT NETW TBL, PORT	Читать из сети Записать через сеть	
GPA ADDR, PORT SPA ADDR, PORT	Получить адрес порта Установить адрес порта	
Скоростные команды		
HDEF HSC, MODE	Определить режим скоростного счетчика	
HSC N	Активизировать скоростной счетчик	
PLS Q	Импульсный выход	

Предметный указатель

Символы

&, 33

*, 33

А

Аварийный контроль, PID-регулятор, 153

Автоматическая настройка, PID, 372

Автоматическая настройка PID

автоматическое определение гистерезиса, 377

исключительные ситуации, 379

отклонение, 377

ошибочные состояния, 379

последовательность, 378

предпосылки, 377

регулируемая переменная за пределами допустимого диапазона, 379

Автоматический режим, PID-регулятор, 153

Автоматическое определение гистерезиса, 377

Адрес

назначение, 62

символический, 63

скоростные счетчики, 128

сеть, 213

установка удаленного, 213

Modbus, 351

Адресация

аккумуляторы, 29

аналоговые входы, 30

аналоговые выходы, 31

байт.бит, 26

биты памяти, 27

биты специальной памяти (SM), 30

встроенные входы/выходы, 32

входы-выходы модулей расширения, 32

косвенная (указатели), 33

локальная память, 30

области памяти, 27–30

память S7-200, 26

память переменных, 27

память реле управления очередностью (SCR), 31

прямой доступ, 26

регистр входов образа процесса, 27

регистр выходов образа процесса, 27

скоростные счетчики, 29

счетчики, 28

таймеры, 28

Адресация входов и выходов, 32

Аккумуляторы, 29

Аналоговые выходы, настройка значений, 44

Аналоговые модули, 3

аналоговый вход EM 231, 406

EM 231, RTD, 413

EM 231, термopара, 413

EM 232, аналоговый вывод, 410

EM 235, аналоговый ввод/вывод, 407

Аналоговый вход (AI)

адресация, 30

фильтрация, 45

Аналоговый выход (AQ), адресация, 31

Аналоговый потенциометр

потенциометры, 50

SMB28 и SMB29, 472

Аппаратура, устранение неисправностей, 251

Арифметические операции

вычитание, 142

декрементирование, 146

деление, 142

деление целых чисел с остатком (DIV), 144

инкрементирование, 146

сложение, 142

умножение, 142

умножение целого числа на двойное целое (MUL), 144

Арифметические операции с вещественными числами, пример, 143

Арифметические операции с целыми числами, пример, 143

Асинхронные обновления, команда PWM, 138

Б

Байт состояния, скоростной счетчик, 128

Байты специальной памяти, EM 277 PROFIBUS-DP, 431

Батарейный модуль, 452

номер для заказа, 484

Безмодемный адаптер, 234

Безусловные входы, 59

Библиотека протокола USS, управление приводами MicroMaster, 333

Библиотека протоколов Modbus, 349

Библиотеки команд, 65

Библиотеки команд STEP 7-Micro/WIN 32, 482

Битовая память (M), 27

Битовые логические операции

блок с двумя устойчивыми состояниями, 79

блок с двумя устойчивыми состояниями и преимуществом сброса, 79

катушки, 75

контакты, 72

логический стек, 77

Биты специальной памяти, 468–476

краткая справка, 491

Биты специальной памяти, модуль позиционирования EM 253, 300

Блок данных, 54

Блокирование, скоростные счетчики, 133

Блок с двумя устойчивыми состояниями и преимуществом сброса, 79

- Блок с двумя устойчивыми состояниями и преимуществом установки, 77
- Булевы операции
 - блок с двумя устойчивыми состояниями, 79
 - катушки, 75
 - контакты, 72
 - логический стек, 77

В

- Варианты режимов поиска опорной точки, 305–309
 - модуль позиционирования EM 253, 305–309
- Ввод значений в таблицу, 191
 - пример, 191
- Ввод команд, 12
- Вложенность, подпрограммы, 206
- Восстановление, потеря пароля, 49
- Времена исполнения, команды STL, 485
- Времена цикла: SMW22 – SMW26, 472
- Время оборота маркера, 235
 - сравнение, 237
- Время, установка, 80
- Время цикла (функция PTO), 135
- Встроенные входы/выходы, 32
- Встроенные переменные, в текстовых и SMS-сообщениях, 313
- Входы, 24, 25
 - аналоговые, фильтрация, 45
 - калибровка, 405
 - модуля позиционирования EM 253, 267
 - пуска и сброса (HSC), 125
 - скоростных счетчиков, 122
 - условные/безусловные, 59
 - цифрового модуля расширения, 397
 - цифровые, фильтрация, 45
 - CPU, 392
- Входы и выходы модулей расширения, 32
 - ошибки шины (SMW98), 477
- Входы и выходы, непосредственное чтение и запись, 42
- Входы контура регулирования
 - преобразование, 151
 - нормализация, 151
- Выбор
 - кабеля PPI Multi-Master, 226
 - наборы команд, 57
 - платы CP, 226
 - протокола обмена данными, 216
 - редактора программ, 55
 - режима работы S7-200, 41
 - термопары с помощью двухпозиционных переключателей, 416
 - RTD с помощью двухпозиционных переключателей, 420–421
- Вывод импульсов
 - скоростной, 50
 - команда вывода импульсов (PLS), 135
 - команда вывода последовательности импульсов (PTO), 135
 - широотно-импульсная модуляция (PWM), 135

- Вывод импульсов с использованием широтно-импульсной модуляции, PWMx_RUN, 256
- Вывод последовательности импульсов
 - PTOx_ADV, 266
 - PTOx_CTRL, 262
 - PTOx_LDPOS, 265
 - PTOx_MAN, 264
 - PTOx_RUN, 263
- Выделение памяти, Мастер протоколирования данных, 369
- Выполнение
 - команд, 25
 - логики управления, 24
- Высвечивание сегментов семисегментного индикатора, 104
- Высоковольтное испытание изоляции, 386
- Выход ШИМ, настройка, 255
- Выходы, 24
 - команды без выходов, 60
 - модуль позиционирования EM 255, 267
 - цифровой модуль расширения, 399
 - CPU, 392
- Выходы, аналоговые, настройка значений, 44
- Выходы и реле, 22
- Выходы переменного тока и реле, 22
- Выходы постоянного тока и реле, 22
- Выходы, цифровые, настройка состояний, 43
- Выход PTO, 260
- Вычитающий счетчик, 118

Г

- Групповая пересылка байтов, 168
- Групповая пересылка двойных слов, 168
- Групповая пересылка слов, 168

Д

- Данные
 - передача, 89
 - прием, 89, 94
 - сохранение и восстановление, 36
- Данные о мощности CPU, 389
- Дата, установка, 80
- Двухпозиционные переключатели
 - кабель RS-232/PPI Multi-Master, 456
 - термопара, 416
 - RTD, 420–421
- Демонтаж
 - клеммный блок, 19
 - модуль расширения, 19
 - CPU, 19
- Дерево команд, 11, 55
- Диагностика, самотестирование CPU, 25
- Диагностическая информация, модуль позиционирования EM 253, 293
- Диагностический светодиод, 47
- Диалоговое окно с информацией о ПЛК, 60
- Диапазон байтов и целых чисел, 26
- Диапазоны операндов, 71
- Диапазоны, PID-регулятор, 152

Дифференциальная составляющая, PID-алгоритм, 150
 Длина строки, 186
 Доступ
 данные S7-200, 26, 33
 прямая адресация, 26
 таблица состояний, 64

З

Завершение по инициативе пользователя, 94
 Загрузка
 нового предустановленного значения в HSC, 133
 нового текущего значения в HSC, 133
 Загрузка в CPU
 конфигураций протоколов данных, 36
 программы, 14, 36
 рецептов, 36
 Загрузка из CPU, программы, 36
 Заземление, 20, 21
 Зазор, выбор рабочей зоны, 308
 Закладки, 246
 Запросы на обмен данными, обработка, 25
 Запуск, STEP 7-Micro/WIN, 9
 Защита паролем, модемный модуль EM 241, 313
 Защита, пароли, 48
 Защита с помощью диода, 22
 Защитные схемы, 22
 Защитный контур, 385
 Значения вещественных чисел, 26, 31
 Значения таблицы профилей, генераторы PTO/PWM, 140

И

Извлечение данных после запуска, 39
 Извлечение программы из модуля памяти, 38
 Изменение
 направления счета скоростного счетчика, 132
 предустановленного значения в HSC, 133
 текущего значения в HSC, 133
 Изменение указателей, 34
 Имитаторы входов, 458
 Имитаторы, входы, 458
 Имитаторы, номера для заказа, 485
 Импульсный таймер команда (TP), 203
 Индикаторные панели
 сенсорная панель TP070, 5
 текстовый дисплей TD 200, 5
 Индуктивные нагрузки, 22
 Инициализация
 протокол Modbus, 350
 скоростные счетчики, 129
 Инкрементирование указателей, 34
 Интегральная составляющая, PID-алгоритм, 149
 Интеллектуальные модули, 3
 поддерживающие их CPU, 325
 состояние (SMB200 – SMB549), 481
 Интерфейс, коммуникационный, выбор, 212
 Интерфейс с международной телефонной линией, модемный модуль EM 241, 310

Интерфейсы оператора, номер для заказа, 485
 Исключительные ситуации, автоматическая настройка PID, 379
 Исполнение, программа, 14
 Источник питания, 8, 17

К

Кабели
 выбор, 212
 настройка
 свободно программируемый обмен данными, 233
 удаленный модем, 232
 номера для заказа, 484
 оконечная нагрузка, 225
 расширение ввода-вывода, 453
 сеть, 223, 224
 смещение, 225
 RS-232/PPI Multi-Master, 454
 USB/PPI Multi-Master, 454
 Кабель с разъемами для модулей расширения, монтаж, 453
 Кабель Multi-Master, 9
 Кабель PPI Multi-Master, 4
 выбор, 210, 226
 настройка, 232
 настройка для свободно программируемого обмена данными, 233
 переключатели для выбора скорости передачи, 233
 режим свободно программируемого обмена данными, 229
 с модемом, 233
 с радиомодемом, 234
 стандарт RS-232, 229
 Кабель RS-232/ PPI Multi-Master
 настройка для гипертерминала, 241, 485
 настройка порта, 222
 настройка с помощью двухпозиционных переключателей, 456
 технические данные, 454
 Кабель USB/ PPI Multi-Master
 настройка порта, 222
 выбор, 212
 номер для заказа, 454
 Калибровка
 EM 231, 405
 EM 235, 405
 входы, 405
 Катушки
 непосредственная установка, 75
 непосредственное присваивание, 75
 непосредственный сброс, 75
 присваивание, 75
 сброс, 75
 установка, 75
 Квадратный корень, 145
 Клеммный блок
 демонтаж, 19
 обратная установка, 19

- Коды ошибок, 464
 - команда PWMx_RUN, 256
 - команды для модуля позиционирования EM 253, 294
 - команды протокола USS, 344
 - команды PTO, 266
 - модуль позиционирования EM 253, 294
 - нарушения правил компиляции, 466
 - ошибки модуля позиционирования EM 253, 295
 - ошибки этапа выполнения, 465
 - фатальные ошибки, 464
- Коды ошибок исполнения, команды протокола USS, 344
- Коды ошибок команд, модуль позиционирования EM 253, 294
- Коды ошибок модуля, модуль позиционирования EM 253, 295
- Коды стран, поддерживаемые EM 241, 310
- Команда блокирования прерываний, 155
- Команда, ввод, 12
- Команда возврата из подпрограммы, 206
 - пример, 209
- Команда возврата из программы обработки прерываний, 155
- Команда вывода импульсов (PLS), 135
- Команда вывода последовательности импульсов (PTO), 50, 135
 - время цикла, 136
 - значения таблицы профилей, 140
 - Мастер управления позиционированием, 135
 - много сегментная конвейерная обработка, 137
 - настройка с помощью SM-памяти, 138
 - одно сегментная конвейерная обработка, 136
 - управление шаговым двигателем, 140
- Команда вызова подпрограммы, 206
 - пример, 207
- Команда вычитания, 142
- Команда декодирования, 114
 - пример, 114
- Команда деления, 142
- Команда деления целых чисел с остатком (DIV), 144
 - пример, 144
- Команда дублирования вершины логического стека, 77
- Команда загрузки реле управления очередностью, 174
- Команда загрузки стека, 77
- Команда записи через сеть, 83, 84
 - коды ошибок, 84
 - пример, 84
- Команда заполнения памяти комбинацией двоичных разрядов, 194
 - пример, 194
- Команда извлечения вершины стека, 77
- Команда инвертирования байта, 163
- Команда инвертирования двойного слова, 163
- Команда инвертирования слова, 163
- Команда кодирования, 114
 - пример, 114
- Команда копирования второго уровня стека, 77
- Команда копирования подстроки из строки, 188
- Команда копирования строки, 186
- Команда логического сопряжения по И, 164
 - пример, 165
- Команда непосредственного присваивания, 75
- Команда округления, 103
- Команда округления отбрасыванием, 103
- Команда отсоединения прерывания, 155
- Команда непосредственного сброса, 75
- Команда непосредственной установки, 75
- Команда очистки прерывающих событий, 155
- Команда передачи, 88
 - передача данных, 89
 - пример, 95
 - режим свободно программируемого обмена данными, 88
- Команда перемещения значения бита в регистр сдвига, 183
 - пример, 184
- Команда перестановки байтов, пример, 185
 - пример, 185
- Команда пересылки байта, 166
- Команда пересылки байта для непосредственной записи, 167
- Команда пересылки вещественного числа, 166
- Команда пересылки двойного слова, 166
- Команда пересылки непосредственно считанного байта, 167
- Команда пересылки слова, 166
- Команда перехода к следующему реле управления очередностью, 174
- Команда-подпрограмма RCPx_Read, рецепт, 363
- Команда поиска значения в таблице, 195
 - пример, 197
- Команда поиска первого появления символа в строке, 189
- Команда поиска строки в строке, 189
- Команда получения адреса порта, 97
- Команда преобразования байта в целое число, 102
- Команда преобразования вещественного числа в строку, 109, 112
- Команда преобразования вещественного числа в строку символов ASCII, 107
 - пример, 108
- Команда преобразования двойного целого числа в вещественное, 102
- Команда преобразования двойного целого числа в строку, 109, 112
- Команда преобразования двойного целого числа в целое, 102
- Команда преобразования двойного целого числа в ASCII, 106
- Команда преобразования подстроки в вещественное число, 109, 112
- Команда преобразования подстроки в двойное целое число, 109, 112
- Команда преобразования подстроки в целое число, 109, 112
- Команда преобразования строки символов ASCII в 16-ричное число, 105
 - пример, 108

- Команда преобразования целого числа в байт, 102
- Команда преобразования целого числа в двойное целое, 102
- Команда преобразования целого числа в строку, 109, 112
- Команда преобразования целого числа в строку символов ASCII, 105
пример, 108
- Команда преобразования целого числа в BCD, 102
- Команда преобразования шестнадцатеричного числа в строку символов ASCII, 105
- Команда преобразования BCD в целое число, 102
- Команда приема, 88
завершение по инициативе пользователя, 94
максимальное число символов, 94
межсимвольный таймер, 93
обнаружение конечного символа, 93
обнаружение обрыва провода, 92
ошибки четности, 94
прием данных, 89
пример, 95
распознавание бездействия линии, 91
распознавание начального символа, 92
режим свободно программируемого обмена данными, 88
таймер сообщения, 93
условия завершения сообщения, 91
условия начала сообщения, 91
SMB86 – SMB94, SMB186 – SMB194, 478
- Команда присваивания, 75
- Команда разблокирования прерываний, 155
- Команда сброса, 75
пример, 75
- Команда сброса контроля времени, 169
пример, 170
- Команда сдвига байта влево, 181
- Команда сдвига байта вправо, 181
- Команда сдвига двойного слова влево, 181
- Команда сдвига двойного слова вправо, 181
- Команда сдвига слова вправо, 181
- Команда соединения строк, 186
- Команда сравнения байтов, 98
- Команда сравнения вещественных чисел, 98
- Команда сравнения двойных слов, 98
- Команда сравнения строк, 100
- Команда сравнения целых чисел, 98
- Команда увеличения на 1 (инкрементирование), 144
пример, 144
- Команда удаления первой записи, 192
пример, 192
- Команда удаления последней записи, 192
пример, 193
- Команда уменьшения на 1, 146
пример, 146
- Команда умножения, 142
- Команда умножения целого числа на двойное целое (MUL), 144
пример, 144
- Команда условного возврата из подпрограммы, 206
- Команда условного возврата из программы обработки прерываний, 155
- Команда условного завершения, 169
пример, 170
- Команда условного завершения реле управления очередностью, 175
- Команда установки, 75
пример, 75
- Команда установки адреса порта, 97
- Команда установки расширенных часов реального времени, 81
- Команда установки часов реального времени, 80
- Команда циклического сдвига байта влево, 181
- Команда циклического сдвига байта вправо, 181
- Команда циклического сдвига двойного слова влево, 181
- Команда циклического сдвига двойного слова вправо, 181
- Команда циклического сдвига слова влево, 181
- Команда циклического сдвига слова вправо, 181
- Команда чтения из сети, 85, 84
коды ошибок, 84
пример, 86
- Команда чтения расширенных часов реального времени, 81
- Команда MBUS_INIT, 353
- Команда MBUS_SLAVE, 355
- Команда MODx_CTRL, модемный модуль EM 241, 321
- Команда MODx_XFR, модемный модуль EM 241, 321
- Команда MODx_MSG, модемный модуль EM 241, 322
- Команда Next, 171
пример, 172
- Команда NOT, 72
- Команда PID-регулятор, 147
аварийный контроль, 1531
автоматический режим, 153
виды регуляторов, 150
диапазоны, 152
дифференциальная составляющая, 150
интегральная составляющая, 149
Мастер, 147
нормализация входов контура регулирования, 150
ошибочные состояния, 153
переменные, 152
преобразование входов контура регулирования, 150
преобразование регулирующего воздействия в масштабированную целую величину, 151
пропорциональная составляющая, 149
регулятор, 150
режимы, 153
ручной режим, 153
с отрицательной обратной связью, 152
с положительной обратной связью, 152
таблица контура регулирования, 154

- Команда PWMx_RUN, коды ошибок, 256
- Команда STOP, 169
 - пример, 170
- Командный байт, модуль позиционирования EM 255, 301
- Команды
 - байт – целое, 102
 - битовая логика, 72
 - блокирование прерывания, 155
 - блок с двумя устойчивыми состояниями и преимуществом сброса, 79
 - блок с двумя устойчивыми состояниями и преимуществом установки, 79
 - ввод значения в таблицу, 191
 - вещественное – ASCII, 107
 - возврат из подпрограммы, 206
 - возврат из программы обработки прерываний, 155
 - вывод импульсов (PLS), 135
 - вывод последовательности импульсов (PTO), 135
 - вызов подпрограммы, 206
 - вычитание, 142
 - вычитающий счетчик, 118
 - групповая пересылка байтов, 168
 - групповая пересылка двойных слов, 168
 - групповая пересылка слов, 168
 - двойное целое – вещественное, 102
 - двойное целое – целое, 102
 - двойное целое – ASCII, 106
 - декодирование, 114
 - декрементирование, 146
 - деление, 142
 - деление целых чисел с остатком (DIV), 144
 - длина строки, 186
 - дублирование вершины логического стека, 77
 - загрузка реле управления очередностью, 174
 - загрузка стека, 77
 - запись через сеть, 83
 - заполнение памяти, 194
 - извлечение вершины стека, 77
 - импульсный таймер (TP), 203
 - инвертирование байта, 163
 - инвертирование двойного слова, 163
 - инвертирование слова, 163
 - инкрементирование, 146
 - исполнение, 25
 - квадратный корень, 145
 - кодирование, 114
 - копирование второго уровня стека, 77
 - копирование подстроки из строки, 188
 - копирование строки, 186
 - косинус, 145
 - логическое сопряжение 1-го и 2-го уровня стека по И, 77
 - логического сопряжение первого и второго уровня стека по ИЛИ, 77
 - метка, 173
 - модемный модуль EM 243, 320
 - модуль позиционирования EM 255, 275
 - назначение прерывания, 155
 - натуральная экспонента, 145
 - натуральный логарифм, 145
 - непосредственная установка, 75
 - непосредственное присваивание, 75
 - непосредственно управляемый контакт, 72
 - непосредственный сброс, 75
 - обратный счет, 115
 - округление, 103
 - округление отбрасыванием, 103
 - определение скоростного счетчика (HDEF), 120
 - останов, 169
 - отрицательный фронт, 72
 - отсоединение прерывания, 155
 - очистка прерывающих событий, 155
 - передача, 88
 - перемещение значения бита в регистр сдвига, 183
 - перестановка байтов, 185
 - пересылка байта, 166
 - пересылка байта для непосредственной записи, 167
 - пересылка вещественного числа, 166
 - пересылка двойного слова, 166
 - пересылка непосредственно считанного байта, 167
 - пересылка слова, 166
 - переход к следующему реле управления очередностью, 175
 - переход – метка, 173
 - поиск значения в таблице, 195
 - поиск первого появления символа в строке, 189
 - поиск строки в строке, 189
 - положительный фронт, 72
 - получение адреса порта, 97
 - поразрядное исключаящее ИЛИ, 164
 - поразрядное логическое И, 164
 - преобразование вещественного числа в строку, 109, 112
 - преобразование двойного целого числа в строку, 109, 112
 - преобразование подстроки в вещественное число, 109, 112
 - преобразование подстроки в двойное целое число, 109, 112
 - преобразование подстроки в целое число, 109, 112
 - преобразование целого числа в строку, 109, 112
 - прерывание, 155–161
 - прием, 88
 - присваивание, 75
 - протокол Modbus Slave, 352
 - протокол USS, 335
 - прямой счет, 115
 - разблокирование прерываний, 155
 - реверсивный счетчик, 116, 118
 - регулятор (PID), 147
 - сброс, 75
 - сброс контроля времени, 169

- сдвиг байта влево, 181
сдвиг байта вправо, 181
сдвиг двойного слова влево, 181
сдвиг двойного слова вправо, 181
сдвиг слова влево, 181
сдвиг слова вправо, 181
семисегментный индикатор, 104
синус, 145
скоростной вывод импульсов (PLS), 135
скоростной счетчик (HSC), 120
сложение, 142
соединение строк, 186
создание команд позиционирования, 304
сравнение, 61
стандартный контакт, 72
суммирующий счетчик, 118
таблица, 192–197
таймер с задержкой включения с
запоминанием (TONR), 198
таймер с задержкой включения (TON), 198,
203
таймер с задержкой выключения (TOF), 198,
203
тангенс, 145
удаление первой записи, 192
удаление последней записи, 192
умножение, 142
умножение целого числа на двойное целое
(MUL), 144
условный возврат из подпрограммы, 206
условный возврат из программы обработки
прерываний, 155
условное завершение, 169
условное завершение реле управления
очередностью, 174
установка, 75
установка расширенных часов реального
времени, 81
установка адреса порта, 97
установка часов реального времени, 80
целое – байт, 102
целое – двойное целое, 102
целое – ASCII, 105
целое – BCD, 102
циклический сдвиг байта влево, 181
циклический сдвиг байта вправо, 181
циклический сдвиг двойного слова влево, 181
циклический сдвиг двойного слова вправо,
181
циклический сдвиг слова влево, 181
циклический сдвиг слова вправо, 181
часы реального времени, 80
чтение из сети, 83
чтение расширенных часов реального
времени, 81
чтение часов реального времени, 80
шестнадцатеричное число – ASCII, 105
широотно-импульсная модуляция (PWM), 135
AENO, 77
ASCII – шестнадцатеричное число, 105
BCD – целое, 102
for, 171
MBUS_INIT, 353
MBUS_SLAVE, 355
MODx_CTRL, 321
MODx_MSG, 322
MODx_XFR, 321
next, 171
NOT, 72
OR, 164
PID-регулятор, 147
POSx_CFG, 286
POSx_CLR, 285
POSx_CTRL, 276
POSx_DIS, 284
POSx_GOTO, 278
POSx_LD OFF, 281
POSx_LD POS, 282
POSx_MAN, 277
POSx_RSEEK, 280
POSx_RUN, 279
POSx_SRATE, 283
PTOx_ADV, 266
PTOx_CTRL, 262
PTOx_LD POS, 265
PTOx_MAN, 264
PTOx_RUN, 263
PWMx_RUN, 256
Команды групповой пересылки, пример, 168
Команды для реле управления очередностью
загрузка реле управления очередностью, 174,
177
ограничения, 174
пример, 174
переход к следующему реле управления
очередностью, 174
разделение потоков управления, 177
слияние потоков управления, 177
условное завершение реле управления
очередностью, 174
Команды для часов реального времени, 80
Команды инвертирования, пример, 163
Команды позиционирования, модуль
позиционирования EM 253, 302
Кабель для модулей расширения, 453
Команды, краткий справочник, 497
Команды обмена данными
запись через сеть, 83
передача, 88
получение адреса порта, 97
прием, 88
установка адреса порта, 97
чтение из сети, 83
Команды обнаружения фронтов сигнала, 73
Команды перехода
метка, 173
переход на метку, 173
пример, 173

- Команды преобразования
 - декодирование, 114
 - кодирование, 114
 - округление, 103
 - округление отбрасыванием, 103
 - пример, 103
 - семисегментный индикатор, 104
 - стандартные, 101
 - строка, 109, 112
 - шестнадцатеричных чисел в ASCII, 105
 - ASCII, 105
 - ASCII в шестнадцатеричные числа, 105
 - Команды преобразования ASCII, 105
 - Команды прерывания
 - блокирование прерывания, 155
 - назначение прерывания, 155
 - отсоединение прерывания, 155
 - очистка прерывающих событий, 155
 - пример, 161
 - разблокирование прерываний, 155
 - условный возврат из программы обработки прерываний, 155
 - Команды протокола USS
 - коды ошибок исполнения, 344
 - пример программы, 343
 - указания по использованию, 335
 - USS4_DRV_CTRL, 337
 - USS4_INIT, 336
 - USS4_RPM_x и USS4_WPM_x, 340, 341
 - Команды сдвига
 - виды, 181
 - пример, 182
 - Команды сравнения, 59
 - пример, 98
 - сравнение байтов, 98
 - сравнение вещественных чисел, 96
 - сравнение двойных слов, 96
 - сравнение строк, 98
 - сравнение целых чисел, 98
 - Команды счетчиков IEC
 - вычитающий счетчик, 118
 - пример, 118
 - реверсивный счетчик, 118
 - суммирующий счетчик, 118
 - Команды счетчиков SIMATIC
 - обратный счет, 115
 - примеры, 117
 - прямой счет, 115
 - реверсивный счетчик, 116
 - Команды управления программой
 - команды перехода, 173
 - основы управления программой, 169
 - останов, 169
 - реле управления очередностью (SCR), 174
 - сброс контроля времени, 169
 - условное завершение, 169
 - цикл for-next, 171
 - Команды цикла For-next
 - пример, 172
 - for, 171
 - next, 171
 - Команды циклического сдвига, 181
 - виды, 181
 - пример, 182
 - Команды PTO, коды ошибок, 266
 - Команды STL
 - времена исполнения, 485
 - краткая справка, 495
 - Коммуникационные платы, номера для заказа, 484
 - Коммуникационные процессоры. См. Платы CP
 - модуль CP 243-1 Ethernet, 446
 - модуль CP 243-1 IT Internet, 448
 - CP 243-2 AS-Interface, 450
 - Коммуникационный порт
 - назначение контактов разъема, 225
 - прерывания, 158
 - режим свободно программируемого обмена данными, 227
 - Коммуникационный процессор CP 243-2
 - Мастер, 450
 - присоединительные элементы, 451
 - режим, 451
 - функции, 451
 - Конвейерная обработка, импульсы PTO, 136
 - Контактный план. См. Редактор LAD
 - Контакты, 72
 - пример, 74
 - Контроль
 - переменных, 64
 - Контурные регулирования с отрицательной обратной связью, 152
 - Контурные регулирования с положительной обратной связью, 152
 - Конфигурационная таблица, модемный модуль EM 241, 315, 327
 - Косвенная адресация, 33
 - & и *, 33
 - изменение указателей, 34
 - Косинус, 145
 - Коэффициент актуализации пропусков (GUF), 235
 - Краткая справочная информация, 491
- ## Л
- Ламповые нагрузки, указания, 22
 - Логика управления, 24
 - Логические соединения
 - MPI, 217
 - PPI, 216
 - Логическое сопряжение 1-го и 2-го уровня стека по И, 77
 - Логическое сопряжение 1-го и 2-го уровня стека по ИЛИ команда, 77
 - Локальная память, 30
- ## М
- Максимальная и стартстопная скорости, Мастер управления позиционированием, 257
 - Максимальное число символов, 94

- Мастера
 - команд, HSC, 121
 - команд, NETR, NETW, 83
 - расширения функций модема, 232, 316
 - управления позиционированием, 269
 - AS-i, 450
 - Ethernet, 446
 - Internet, 449
 - PID, 147
 - TD 200, 5
- Мастер команд
 - HSC, 121
 - NETR, NETW, 83
- Мастер протоколирования данных
 - выделение памяти, 369
 - загрузка проекта, 370
 - использование, 367
 - обзор, 366
 - подпрограмма DATx_WRITE, 371
 - пример применения, 366
 - проводник S7-200, 370
 - терминология, 366
- Мастер расширения функций модема, 232
- Мастер расширения функций модема, модемный модуль EM 243, 316
- Мастер рецептов, 358
 - выделение памяти, 61
 - загрузка в CPU, 362
 - команды, 363
 - терминология, 359
- Мастер управления позиционированием, 135, 269
 - PTOx_ADV, 266
 - PTOx_CTRL, 262
 - PTOx_LDPOS, 265
 - PTOx_MAN, 264
 - PTOx_RUN, 263
- Мастер AS-i, 450
- Масштабируемое целое, преобразование регулирующих воздействий, 151
- Международные стандарты, 384
- Межсимвольный таймер, 93
- Метка, 173
- Метка времени, протокол с историей, 48
- Модем
 - удаленный, кабель PPI Multi-Master, 232
 - с кабелем PPI Multi-Master, 233
- Модемный модуль, 437
 - защита паролем, 313
 - интерфейс с международными телефонными линиями, 310
 - команда MODx_CTRL, 321
 - команда MODx_MSG, 322
 - команда MODx_XFR, 321
 - команды, 320
 - конфигурационная таблица, 315
 - Мастер расширения функций модема, 316
 - обратный вызов для защиты от несанкционированного доступа, 314
 - ошибки, возвращаемые командами, 323
 - пейджинговая связь, 312
 - передача данных, 313
 - пример, 325
 - светодиоды состояний, 316
 - служба коротких сообщений, 312
 - текстовая пейджинговая связь, 312
 - телефонный номер для передачи сообщений, 329
 - формат текстовых сообщений, 331
 - характеристики, 310
 - цифровая пейджинговая связь, 312
 - штекер RJ11, 310
 - SMS-сообщения, 312
- Модемный модуль EM 241, 437
 - адреса Modbus, 312
 - биты специальной памяти, 325
 - защита паролем, 313
 - интерфейс с международными телефонными линиями, 310
 - интерфейс со STEP 7-Micro/WIN, 311
 - команда MOD_XFR, 321
 - команда MODx_CTRL, 321
 - команда MODx_MSG, 322
 - команды, 320
 - конфигурационная таблица, 315, 327
 - Мастер расширения функций модема, 316
 - монтаж, 438
 - обратный вызов для защиты от несанкционированного доступа, 314
 - ошибки, возвращаемые командами, 323
 - пейджинговая связь, 312
 - передача данных, 313
 - поддерживаемые страны, 310
 - пример, 325
 - протокол Modbus RTU, 311
 - светодиоды состояний, 316
 - Служба коротких сообщений (SMS), 312
 - совместимость с CPU, 438
 - текстовая пейджинговая связь, 312
 - телефонный номер для передачи сообщений, 329
 - технические данные, 437
 - формат текстовых сообщений, 331
 - характеристики, 310
 - цифровая пейджинговая связь, 312
 - штекер RJ11, 310
- Модули
 - памяти, 452
 - номера для заказа, 484
- Модули памяти, номер для заказа, 482
- Модули расширения, 3
 - адресация входов-выходов, 32
 - аналоговые
 - номера для заказа, 402, 413
 - схемы соединений, 404
 - технические данные, 402
 - технические данные входов, 402
 - технические данные выходов, 403
 - демонтаж, 19
 - монтаж, 18
 - номера для заказа, 481
 - потребности в мощности, 17, 459
 - размеры, 18
 - регистр идентификатора и ошибок, 471

- цифровые
 - общие технические данные, 396
 - схемы соединений, 399–401
 - технические данные входов, 397
 - технические данные выходов, 398
- цифровые, номера для заказа, 396
- Модули AS-interface, номера для заказа руководств, 483
- Модули CPU
 - данные о мощности, 389
 - демонтаж, 19
 - монтаж, 18
 - номера для заказа, 387, 481
 - области памяти, 70, 493
 - самодиагностика, 25
 - схемы соединений, 392–394
 - технические данные, 387, 388
 - технические данные входов, 389
 - технические данные выходов, 390
 - характеристики, 70
- Модуль аналогового ввода EM 231
 - калибровка, 405
 - монтаж, 410
 - схема соединений для входов, 408
 - точность и повторяемость, 411
 - технические данные, 412
 - формат слова данных для входов, 408
- Модуль аналогового ввода/вывода EM 235
 - калибровка, 405
 - монтаж, 410
 - настройка, 407
 - схема соединений для входов, 409
 - схема соединений для выходов, 410
 - технические данные, 412
 - точность и повторяемость, 411
 - формат слова данных для входов, 408
 - формат слова данных для выходов, 409
- Модуль аналогового вывода EM 232
 - схема соединений для выходов, 410
 - формат слова данных для выходов, 409
- Модуль памяти, 36, 452
 - коды ошибок, 464
 - сохранение программы, 37
 - электростатический разряд, 36
- Модуль памяти, извлечение программы, 38
- Модуль позиционирования
 - входы и выходы, 267
 - диагностическая информация, 293
 - коды ошибок, 294
 - коды ошибок команд, 294
 - коды ошибок модуля, 295
 - команды, 275
 - команды позиционирования, 302
 - Мастер управления позиционированием, 269
 - настройка, 269, 293
 - настройка профилей перемещения, 259
 - отображение и управление работой, 292
 - панель управления EM 253, 292–294
 - пример программы управления позиционированием, 304
 - программирование, 268
 - режимы поиска опорной точки, 305–309
- создание команд, 304
- специальная память, 300
- таблица настроек и профилей, 296
- указания по использованию команд, 275
- устранение зазора, 308
- характеристики, 267
- ACCEL_TIME, 258
- DECEL_TIME, 258
- POSx_CFG, 286
- POSx_CLR, 285
- POSx_CTRL, 276
- POSx_DIS, 284
- POSx_GOTO, 278
- POSx_LDOFF, 281
- POSx_LDPOS, 282
- POSx_MAN, 277
- POSx_RSEEK, 280
- POSx_RUN, 279
- POSx_SRATE, 283
- Модуль позиционирования EM 253
 - входы и выходы, 267
 - диагностическая информация, 293
 - коды ошибок, 294
 - коды ошибок команд, 294
 - коды ошибок модуля, 295
 - командный байт, 301
 - команды, 275
 - команды позиционирования, 302
 - Мастер управления позиционированием, 269
 - настройка, 269, 293
 - отображение и управление работой, 292
 - панель управления EM 255, 292–294
 - пример программы для управления модулем, 304
 - примеры, 287–291
 - программирование, 268
 - режимы поиска опорной точки, 305–309
 - совместимость с CPU, 441
 - создание команд, 304
 - специальная память, 300
 - схема подключения, 443–444
 - таблица настроек и профилей, 296
 - устранение зазора, 308
 - указания по использованию команд, 275
 - характеристики, 267
 - ACCEL_TIME, 258
 - DECEL_TIME, 258
 - POSx_CFG, 286
 - POSx_CLR, 285
 - POSx_CTRL, 276
 - POSx_DIS, 284
 - POSx_GOTO, 278
 - POSx_LDOFF, 281
 - POSx_LDPOS, 282
 - POSx_MAN, 277
 - POSx_RSEEK, 280
 - POSx_RUN, 279
 - POSx_SRATE, 283
- Модуль с термопарой (EM 231)
 - индикаторы состояния, 417
 - настройка, 415
 - основы, 415

- температурные диапазоны, 418–419
- Модуль CP 243-1 Ethernet
 количество соединений, 217
 коммуникационные процессоры, 446
 Мастер, 446
 настройка, 446
 номер для заказа, 443, 481
 присоединительные элементы, 446
 технические данные, 445
 функции, 446
- Модуль CP 243-1 IT Internet
 количество соединений, 217
 коммуникационные процессоры, 448
 Мастер Internet, 449
 настройка, 449
 номер для заказа, 448, 481
 присоединительные элементы, 449
 технические данные, 448
 функции, 448
- Модуль CP 243-2 AS-Interface
 номер для заказа, 450
 характеристики, 450
- Модуль CP 243-2 AS-Interface, номер для заказа, 483
- Модуль EM 231c термопарой
 индикаторы состояния, 417
 клеммы, 414
 настройка, 415
 основы, 415
 совместимость с CPU, 414
 температурные диапазоны и точность, 418–419
 технические данные, 413
 установка двухпозиционных переключателей, 416
- Модуль EM 231 RTD
 индикаторы состояния, 422
 клеммы, 414
 настройка, 420
 совместимость с CPU, 414
 температурные диапазоны и точность, 423–424
 технические данные, 413
 установка двухпозиционных переключателей, 420–421
- Модуль EM 277 PROFIBUS-DP
 байты специальной памяти, 431
 в сети PROFIBUS, 428
 дополнительные характеристики, 432
 как slave-устройство DP, 427
 количество соединений, 216
 конфигурационные возможности, 429
 настройка, 428–429
 переключатели адресов, 426
 протокол DP, 427
 режим обмена данными, 430
 светодиодные индикаторы состояния, 432
 светодиоды состояния, 426
 совместимость с CPU, 426
 технические данные, 425
 файл базы данных устройства, 433–434
- Модуль Ethernet, Мастер, 444
- Модуль RTD (EM 231), 420
- Монтаж
 источник питания, 17
 кабель с разъемами для модулей расширения, 451
 модуль расширения, 18
 монтажные размеры, 18
 профильная шина, 18
 размеры, 18
 тепловыделяющие устройства, 16
 требования к зазорам, 16
 указания, 16
 устройства высокого напряжения, 16
 щит управления, 18
 электрические помехи, 16
 CPU, 18
 EM 231, 408
 EM 235, 408
 S7-200, 17
 STEP 7-Micro/WIN, 4
- Монтаж в щите управления, 18
- Монтаж на профильной шине, 18
- Морское агентство Американское судовое бюро, 384
- Морское агентство Bureau Veritas (BV), 384
- Морское агентство Det Norske Veritas (DNV), 384
- Морское агентство Germanisher Lloyd (GL), 384
- Морское агентство Lloyds Register of Shipping (LRS), 384
- Морское агентство Nippon Kaiji Kyokai (NK), 384
- ## Н
- Наблюдение, 14
 переменных с помощью таблицы состояний, 249
 состояния программы, 248
- Набор команд IEC 1131-3, 57
- Набор команд SIMATIC, 57
- Наборы команд
 выбор, 57
 IEC 1131-3, 57
 SIMATIC, 57
- Навигационная панель, 55
- Назначение
 адресов, 62
 начальных значений, 62
 прерываний скоростным счетчикам (HSC), 128
 сетевых адресов, 213
- Назначение контактов коммуникационного порта, 225
- Назначение контактов, коммуникационный порт, 225
- Назначение прерывания, 155
- Наибольший адрес станции (HSA), 233
- Направление, изменение в HSC, 132
- Нарушения правил компиляции, 466
- Настройка
 значений аналогового выхода, 44
 модуля позиционирования EM 255, 293
 модуля CP 243-1 Ethernet, 446

- модуля CP 243-1 IT Internet, 449
 - операции ПТО/PWM, 138
 - отображения состояния FBD и LAD, 248
 - отображения состояния STL, 249
 - пароля, 49
 - сети, 222
 - состояний цифровых выходов, 43
 - сохраняемых областей памяти, 44
 - таблицы символов для Modbus, 349
 - EM 231, 406
 - EM 231 RTD, 420
 - EM 231 thermocouple, 415
 - EM 235, 407
 - EM 277 PROFIBUS-DP, 426–427
 - HyperTerminal, 241–244
 - Настройка
 - привода MM3, 345
 - привода MM4, 348
 - Настройки параметров для обмена данными, STEP 7-Micro/WIN, 10
 - Натуральная экспонента, 145
 - Натуральный логарифм, 145
 - Национальные стандарты, 384
 - Неинтеллектуальный терминал, настройка кабеля RS-232/PPI Multi-Master, 241, 487
 - Непосредственно управляемый контакт, 72
 - Нефатальные ошибки, 60
 - Номера для заказа, 483–485
 - аналоговый модуль расширения, 402, 413
 - имитаторы входов, 458
 - кабели PPI Multi-Master, 454
 - модуль CP 243-2 AS-Interface, 450
 - цифровой модуль расширения, 396
 - CPU, 387
 - Нормализация, входы контура регулирования, 150
- О**
- Обзор, управление без обратной связи, 254
 - Области памяти
 - диапазоны операндов, 71
 - краткая справка, 493
 - очистка, 49
 - CPU, 70
 - Область битовой памяти (M), 27
 - сохранения при потере питания, 38
 - Область локальной памяти (L), 30
 - Область памяти для входов (I), 27
 - Область памяти для реле управления очередностью (S), 31
 - Область памяти для счетчиков (C), 28
 - Область памяти для таймеров (T), 28
 - Область памяти переменных (V), 27
 - Область специальной памяти (SM), 30
 - аналоговые потенциометры, 50
 - Обмен данными, 212
 - интерфейс, выбор, 212
 - конфликты, 238
 - модем, 233–234
 - поддерживаемые протоколы, 216
 - протокол Modbus Slave, 350
 - радиомодем, 234–235
 - скорость передачи, 233, 234
 - с приводами MicroMaster, 334
 - фоновое время, 43
 - S7-200, 10
 - Обмен данными между равноправными узлами, 219–220
 - Обмен данными с приводами, расчет потребностей во времени, 334
 - Обмен данными через PPI, переход в режим свободно программируемого обмена данными, 89
 - Обнаружение конечного символа, 93
 - Обнаружение обрыва провода, 92
 - Обновление, характеристик PWM, 138
 - Обработка, запросы на обмен данными, 25
 - Обработка ошибок, 60
 - Обратная установка, клеммный блок, 19
 - Обратный вызов для защиты от несанкционированного доступа, модемный модуль EM 241, 314
 - Обратный вызов, модемный модуль EM 241, 314
 - Обратный счет, 115
 - пример, 117
 - Обращение к слову, 26
 - Ограничение доступа, 48
 - Ограничение пикового напряжения, 22
 - Оконечная нагрузка, сетевой кабель, 225
 - Операции с подпрограммами
 - вызов подпрограммы, 206
 - пример, 209
 - условный возврат из подпрограммы, 206
 - Определение скоростного счетчика (HDEF), 120
 - Определение таблицы профилей ПТО0, ПТО1 (SMB166 – SMB185), 480
 - Оптимизация, производительность сети, 235
 - Отклонение, 377
 - Открытие, редактор программ, 11
 - Отладка
 - редактирование в режиме RUN, 246
 - характеристики, 246
 - принудительное присваивание значений, 250
 - исполнение программы в течение определенного количества циклов, 250
 - Отладка программного обеспечения, 245
 - Отображение
 - состояния программы, 248
 - элементов программы, 55
 - Отображение адресов в S7-200, Modbus, 351
 - Отрицательный фронт, 72, 73
 - Очереди, программы обработки прерываний, 159
 - Очередь коммуникационных прерываний, 159
 - Очередь прерываний по вводу/выводу, 159
 - Очередь прерываний, управляемых временем, 159
 - Ошибки
 - автоматическая настройка PID, 379
 - ввода-вывода, 60
 - исполнения программы, 61
 - исполнения протокола Modbus Slave, 355
 - команды чтения из сети и записи через сеть, 84

компиляции программы, 60
 модемный модуль EM 241 команды, 323
 нефатальные, 60
 настройка модемного модуля EM 241, 315
 обработка, 60
 просмотр, 60
 фатальные, 61
 этапа выполнения, 60
 PID-регулятор, 153
 SMB1, ошибки исполнения, 468
 Ошибки ввода-вывода, 60
 Ошибки, выявляемые контролем четности
 SMB3, 469
 SMB30 и SMB130, 94
 Ошибки исполнения, 61
 Ошибки компиляции, 60
 Ошибки этапа исполнения, 60
 Ошибки этапа исполнения, коды ошибок, 465

П

Пакет для программирования, 4
 Память
 доступ, 26
 сохраняемая, 44
 CPU, 2
 Память выходов (Q), 27
 Память для программ, увеличение, 48
 Память SM, 30
 протокол Modbus Slave, 350
 управление функциями РТО/PWM, 138
 Панели инструментов, 55
 Панель управления настройкой PID, 380
 Панель управления EM 253, 290–292
 Параметры
 в подпрограмму, 207
 типы для подпрограммы, 207
 Пароль
 восстановление потерянного, 49
 настройка, 49
 ограничение доступа, 49
 очистка, 49
 функции CPU, 48
 Пейджинговая связь, модемный модуль, 312
 Передача данных, модемный модуль EM 241,
 313
 Перезапуск, после фатальных ошибок, 62
 Переключатели адресов, EM 277 PROFIBUS-DP,
 426
 Переключатель режимов, 41
 Переменные
 контроль, 64
 контроль с помощью таблицы состояний, 249
 символический адресация, 62
 PID-регулятор, 152
 Переполнение очереди (SMB4), 469
 Платы CP
 выбор, 226
 номера для заказа, 484
 Платы CP, поддерживаемые STEP 7 Micro/WIN,
 227

Повторители
 номера для заказа, 485
 сеть, 224
 Подавление помех, входной фильтр, 45
 Подключение, 20, 21
 Подключение
 кабеля Multi-Master, 9
 модема к S7-200, 233
 привода MM3, 344
 привода MM4, 347
 радиомодема к S7-200, 234
 сетевых устройств, 237
 Подпрограммы
 вложенность, 206
 вызов из программы обработки прерываний,
 158
 параметр поток сигнала, 207
 пример, 53
 с параметрами, 207
 типы данных, 207
 типы параметров, 207
 указания, 54
 Положение рабочей зоны, модуль
 позиционирования EM 253, 308
 Положительный фронт, 72, 73
 Пользовательские библиотеки, 65
 Поразрядное исключающее ИЛИ, 164
 пример, 165
 Поразрядное логическое ИЛИ, 164
 пример, 165
 Поразрядные логические операции
 И, ИЛИ, исключающее ИЛИ, 164
 инвертирование, 163
 Порт, настройка, кабели PPI Multi-Master, 222
 Порядок, прерывающих событий, 160
 Постоянная память, 36, 39
 Постоянная память, копирование памяти
 переменных, 40
 Постоянные значения, 32
 Построение сети, 223
 Потенциометры, аналоговый потенциометр, 50
 Потеря пароля, 49
 Потеря питания, сохраняемая память, 38, 44
 Поток сигнала, параметр подпрограммы, 207
 Потребности в мощности, 17
 расчет, 459, 461
 CPU, 459
 модуль расширения, 459
 пример, 460
 таблица для расчета, 461
 Предотвращение сетевых конфликтов, 238
 Предустановленное значение
 изменение в HSC, 133
 установка для HSC, 128
 Преобразование
 входов контура регулирования, 150
 регулирующего воздействия в
 масштабированную целую величину, 151
 Прерывание, управляемое временем, пример,
 161

- Прерывания
 - назначение – HSC, 128
 - скоростные счетчики, 122
- Прерывания по вводу/выводу, 158
- Прерывания, управляемые временем, 158
- Прерывания, управляемые временем, регистры интервалов времени (SMB34, SMB35), 474
- Прерывающие события
 - виды, 156
 - краткая справка, 492
 - приоритет, 160
- Приводы. См. Приводы MicroMaster
- Привод MicroMaster
 - обмен данными, 334
 - подключение, 344
 - управление, 333
 - чтение и запись, 340, 341
- Привод MM3
 - подключение, 344
 - наладка, 345
- Привод MM4
 - подключение, 345
 - наладка, 346
- Прием данных, 94
- Прием данных, управляемый прерываниями, 94
- Пример программы, 10
- Примеры
 - арифметические операции с вещественными числами, 143
 - арифметические операции с целыми числами, 143
 - ввод значений в таблицу, 191
 - вызов подпрограммы, 208
 - команда генерирования комбинации двоичных разрядов для семисегментного индикатора, 104
 - команда групповой пересылки, 168
 - команда заполнения памяти комбинацией двоичных разрядов, 194
 - команда обратного счета, 117
 - команда останова, 170
 - команда перемещения значения бита в регистр сдвига, 184
 - команда перестановки байтов, 185
 - команда поиска в таблице, 197
 - команда преобразования вещественного числа в строку символов ASCII, 108
 - команда преобразования строки символов ASCII в 16-ричное число, 108
 - команда преобразования целого числа в строку символов ASCII, 108
 - команда сброса, 75
 - программа с командами USS, 343
 - команда увеличения на 1, 146
 - команда удаления первой записи, 192
 - команда удаления последней записи, 193
 - команда уменьшения на 1, 146
 - команда условного завершения, 170
 - команда установки, 75
 - команда DIV, 144
 - команда MUL, 144
 - команды возврата из подпрограммы, 209
 - команды вызова подпрограммы, 209
 - команды декодирования, 114
 - команды для реле управления очередностью, 174
 - команды инвертирования, 163
 - команды кодирования, 114
 - команды передачи, 95
 - команды перехода на метку, 173
 - команды прерывания, 161
 - команды сдвига, 182
 - команды сравнения, 98
 - команды счетчиков IEC, 118
 - команды циклического сдвига, 182
 - команды чтения из сети и записи через сеть, 86
 - контакт, 74
 - модемный модуль EM 241, 325
 - модуль позиционирования, 287–291
 - подпрограмма, 53
 - позарядные логические операции И, ИЛИ и исключающее ИЛИ, 165
 - прерывание, управляемое временем, 161
 - прием команды, 95
 - программы обработки прерываний, 53
 - протокол Modbus Slave, программирование, 356
 - разделение потоков управления, 177
 - расчет потребности в мощности, 459
 - реверсивный счетчик, 117
 - режим свободно программируемого обмена данными, 228
 - режимы скоростного счетчика, 123
 - сеть с передачей маркера, 236
 - скоростной счетчик, 134
 - слияние потоков управления, 178
 - стандартные команды преобразования, 102
 - стековые операции, 78
 - счетчики SIMATIC, 117
 - таймер с задержкой включения, 200
 - таймер с задержкой включения с запоминанием, 202
 - таймер с задержкой выключения, 201
 - таймеры IEC, 203
 - таймеры SIMATIC, 200, 201, 202
 - условные переходы, 179
 - цикл for-next, 172
- Принудительное присваивание значений, 249, 250
- Приоритет
 - прерывающие события, 160
 - программы обработки прерываний, 159
- Присваивание начальных значений, 62
- Присоединительные элементы
 - модуля CP 243-1 Ethernet, 446
 - модуля CP 243-1 IT Internet, 449
 - CP 243-2, 451
- Проводник S7-200, 36, 41
 - Мастер протоколирования данных, 370
- Программа
 - аналоговые входы, 25
 - основные элементы, 53
 - ошибки компиляции, 60

- создание, 10
 - создание с помощью STEP 7-Micro/WIN, 55
 - загрузка в CPU, 14, 36
 - редактирование в режиме RUN, 246
 - ошибки исполнения, 63
 - наблюдение, 14
 - контроль состояния, 248
 - исполнение, 14
 - сохранение, 13
 - совместное использование данных с программой обработки прерываний, 157
 - задание числа циклов, 250
 - таблица состояний, 64
 - сохранение, 36–40
 - структурирование, 53
 - подпрограммы, 54
 - функции для отладки, 246
 - Программа, пример, управление модуль позиционирования, 304
 - Программирование
 - модуль позиционирования EM 253, 268
 - скоростные счетчики, 121
 - Программное обеспечение, номера для заказа, 484
 - Программы обработки прерываний, 25, 42
 - ввод-вывод, 158
 - виды, поддерживаемые S7-200, 158
 - вызов подпрограмм, 158
 - коммуникационный порт, 158
 - нарастающий/падающий фронт, 158
 - очереди, 159
 - пример, 53
 - приоритет, 159
 - системная поддержка, 157
 - совместное использование данных с главной программой, 157
 - указания, 54, 157
 - управляемые временем, 158
 - Проектирование
 - системы с микро ПЛК, 52
 - схем защиты, 52
 - Проектирование системы, микро ПЛК, 52
 - Производительность, оптимизация сеть, 235
 - Пропорциональная составляющая, PID-алгоритм, 149
 - Просмотр, ошибок, 60
 - Протокол обмена данными
 - выбор, 216
 - двухточечный интерфейс (PPI), 216, 237
 - многоточечный интерфейс (MPI), 217, 237
 - определяемый пользователем, 227
 - PROFIBUS, 217, 237
 - TCP/IP, 217
 - Протокол, обмен данными
 - выбор, 216
 - определяемый пользователем, 227
 - Протокол, определяемый пользователем, режим свободно программируемого обмена данными, 227
 - Протокол с историей, события CPU, 48
 - Протоколы
 - поддерживаемые STEP 7-Micro/WIN, 227
 - PROFIBUS-DP, 427
 - Протокол Modbus RTU, 352
 - модемный модуль EM 241, 311
 - отображение адресов, 312
 - функции, поддерживаемые модемным модулем, 311
 - Протокол Modbus Slave
 - адреса, 351
 - время исполнения, 350
 - инициализация, 350
 - используемые ресурсы, 350
 - коды ошибок исполнения, 355
 - команды, 352
 - настройка таблицы символов, 351
 - отображение адресов – S7-200, 351
 - поддерживаемые функции, 352
 - пример программирования, 356
 - специальная память, 350
 - таблица CRC, 350
 - MBUS_INIT, 353
 - MBUS_SLAVE, 355
 - Протокол MPI, 217, 237
 - Протокол PPI, 216, 237
 - сеть с несколькими master-устройствами, 218
 - сеть с одним master-устройством, 218
 - сложная сеть, 219
 - Протокол PROFIBUS, 217, 237
 - Протокол USS, требования, 334
 - Профиль перемещения
 - модуль позиционирования, 259
 - определение, 259
 - формирование шагов, 260
 - Прямой счет, 115
 - Пуск, скоростной счетчик, 125
- ## Р
- Работа со сложными сетями, 238
 - Радиомодем
 - режим PPI, 234
 - с кабелем PPI Multi-Master, 234
 - PPI/режим свободно программируемого обмена данными, 234
 - Развязка
 - сеть, 223
 - указания по подключению, 20
 - Разделение потоков управления
 - пример, 177
 - Размеры , CPU, 2
 - Разрешающая способность таймера, 199, 200
 - Разрешающий вход (EN), 59
 - Разрешающий выход (ENO), 59
 - Разрешение, таймер, 199, 200
 - Распознавание бездействия линии, 91
 - Распознавание начального символа, 91
 - Расчет
 - время оборота маркера, 235
 - потребности в мощности, 459–461

- Реверсивный счетчик, 116, 118
 - пример, 117
 - Регистратор импульсов, 45
 - Регистр входов образа процесса (I), 25, 27
 - Регистр выходов образа процесса (Q), 25, 27
 - Регистр образа процесса, 42
 - Регистры идентификации и ошибок модулей ввода/вывода, 472
 - Регистры управления свободно программируемым обменом данными (SMB30 и SMB130), 473
 - Регистры HSC0, HSC1, HSC2 (SMB36 – SMB65), 475
 - Регистры HSC3, HSC4, HSC5 (SMB131 – SMB165), 479
 - Регулируемая переменная за пределами допустимого диапазона, автоматическая настройка PID, 379
 - Регулирующие воздействия, преобразование в масштабированную целую величину, 151
 - Регулятор
 - выбор типа, 150
 - диапазоны/переменные, 152
 - команды, 147–158
 - настройка смещения, 152
 - ошибочные состояния, 153
 - положительная/отрицательная обратная связь, 152
 - преобразование входов, 150
 - преобразование выходов, 151
 - режимы, 153
 - Редактирование в режиме RUN, 43, 246
 - блокирование, 48
 - Редактор блоков данных
 - назначение адресов, 62
 - назначение начальных значений, 62
 - Редакторы
 - контактный план (LAD), 56
 - список команд (STL), 55
 - функциональный план (FBD), 56
 - Редакторы программ, 55
 - выбор, 55
 - контактный план (LAD), 55
 - открытие, 11
 - соглашения, 58
 - список команд (STL), 55
 - функциональный план (FBD), 55
 - Редактор FBD
 - описание, 56
 - соглашения, 58
 - функции, 56
 - Редактор LAD
 - описание, 56
 - соглашения, 58
 - характеристики, 56
 - Редактор STL
 - описание, 55
 - характеристики, 55
 - Режим обмена данными, EM 277 PROFIBUS-DP, 430
 - Режим останова, 14, 41
 - Режим работы, профиль перемещения, 259
 - Режим свободно программируемого обмена данными
 - определение, 158
 - ошибка, выявленная контролем четности (SMB3), 469
 - передача и прием команды, 88
 - переход от PPI, 89
 - прием данных, управляемый прерываниями, 94
 - пример, 228
 - принимаемые символы (SMB2), 469
 - протоколы, определяемые пользователем, 227
 - разблокирование, 88
 - стандарт RS-232, 229
 - Режимы
 - скоростные счетчики, 122
 - PID-регулятор, 153
 - Режимы поиска опорной точки, 305
 - Режимы работы, CPU
 - биты состояния, 468
 - изменение, 41
 - протокол свободно программируемого обмена данными, 228
 - Режим PPI, радиомодем, 234
 - Режим RUN, 14, 41
 - редактирование программа, 246
 - Рекомендации
 - к подпрограммам, 54
 - по вертикальному монтажу, 18
 - по изменению указателей для косвенной адресации, 34
 - по монтажу при сильных вибрациях, 18
 - по программам обработки прерываний, 54, 157
 - по проектированию системы автоматизации с микроконтроллером, 52
 - Реле, 22
 - срок службы, 385
 - Рецепты, загрузка из CPU и в CPU, 36
 - Руководства, номера для заказа, 482
 - Ручной режим, PID-регулятор, 153
- ## С
- Сброс, скоростной счетчик, 125
 - Светодиоды
 - диагностические, 47
 - модемный модуль EM 241, 316
 - Светодиоды состояний
 - модуль позиционирования EM 253, 441
 - EM 231 RTD, 422
 - EM 231 thermocouple, 417
 - EM 277 PROFIBUS-DP, 426, 432
 - Свободно программируемый обмен данными, 469
 - SMB30 и SMB130, 472
 - Свободно программируемый обмен данными, настройка, кабель PPI Multi-Master, 233
 - Свойства регистратора импульсов, 46
 - Семисегментный индикатор, 104
 - пример, 104

- Сенсорная панель TP070, 5
 номера для заказа, 485
 номер для заказа руководства, 484
- Серводвигатели, управление перемещением без обратной связи, 257
- Сертификаты агентств, 384
- Сертификаты морских агентств, 384
- Сетевые соединители, номера для заказа, 485
- Сеть
 адреса, 213
 адреса устройств, 216
 время оборота маркера, 235
 кабель, 223, 224
 кабель PPI Multi-Master, 226
 коммуникационный порт, 225
 компоненты, 223–226
 коэффициент актуализации пропусков (GUF), 235
 модем, 233
 наибольший адрес станции (HSA), 235
 обмен данными, 212, 435
 оконечная нагрузка кабеля, 225
 оптимизация производительности, 235
 плата CP, 226
 повторители, 224
 поиск CPU, 215
 построение, 223
 примеры конфигураций, 218, 219, 220
 радиомодем, 234
 развязка, 223
 расчет расстояний, 223
 скорость передачи, 223
 сложная, 238
 смещение, кабель, 225
 сложная PPI, 219
 указания по построению, 223
 устройства человеко-машинного интерфейса, 227
 master-устройства, 213
 MPI, больше 187,5 кБод, 220
 MPI, меньше 187,5 кБод, 219
 PPI с несколькими master-устройствами, 218
 PROFIBUS, 213
 PROFIBUS-DP, 220
 PPI с одним master-устройством, 218
 slave-устройства, 213
- Сеть с передачей маркера, пример, 236
- Сеть Ethernet, CP 243-1, 221
- Сеть MPI
 больше 187,5 кБод, 220
 меньше 187,5 кБод, 219
- Сеть PPI с несколькими master-устройствами, 218
- Сеть PPI с одним master-устройством, 218
- Сеть PROFIBUS-DP
 назначение контактов, 225
 повторители, 224
 технические данные кабеля, 223
 S7-315-2 и EM 277, 220
 STEP 7-Micro/WIN и человеко-машинный интерфейс, 220
- Символическая адресация, 63
- Символические имена, создание списка, 53
- Синус, 145
- Синхронные обновления, PWM команда, 138
- Система микро ПЛК, проектирование, 52
- Системная поддержка, для программ обработки прерываний, 157
- Системное руководство по S7-200, номера для заказа, 484
- Системный блок данных, 54
- Скорости передачи
 оптимальная, 235
 сеть, 223
 установка, 213
 установка переключателей
 кабель Multi-Master, 9
 кабель PPI Multi-Master, 215, 227, 229, 233, 234
- Скорость передачи, сеть, 223
- Скоростной вывод импульсов, 50, 135
 SMB66-SMB85, 476
- Скоростной счетчик (HSC), 120
 пример, 134
- Скоростные счетчики, 50
 адресация, 128
 байт состояния, 128
 блокирование, 133
 временные диаграммы, 123–125
 выбор активного состояния, 126
 изменение направления, 132
 изменение предустановленного значения, 133
 изменение текущего значения, 133
 инициализирующая последовательность, 129
 команды сброса и запуска, 125
 назначение прерывания, 128
 настройка управляющего байта, 126
 область памяти, адресация, 29
 определение режимов и входов, 122
 прерывания, 122
 программирование, 121
 режимы, 123, 494
 управляющий байт, 121
 установка текущего и предустановленного значений, 127
 SMB36-SMB65, 473
- Слияние потоков управления
 пример, 178
- Сложение, 142
- Сложная сеть PPI, 219
- Служба коротких сообщений, модемный модуль EM 241, 312
- Смещение
 сетевой кабель, 223
 PID-регулятор, 145, 147
- События CPU, протокол с историей, 48
- Совместимость
 EM 231 RTD, 412
 EM 231 thermocouple, 412
 EM 241 modem, 436
 EM 253 Position, 439
 EM 277 PROFIBUS-DP, 424

- Соглашения
 - программирование S7-200, 59
 - редакторы программ, 58
- Соединение
 - с S7-200, 10
- Создание
 - программы, 10
 - программы с помощью Micro/WIN, 55
 - протоколов, определяемых пользователем, 227
 - списка символических имен, 53
 - чертежей конфигурации, 53
- Сообщения, сеть с передачей маркера, 236
- Состояние
 - исполнение, 248
 - контроль программы, 248
 - отображение в LAD и FBD, 248
 - отображение в STL, 249
 - условное завершение цикла, 248
- Состояние входов/выходов(SMB5), 470
- Состояние исполнения, 43
- Сохранение
 - битов памяти (M) в ЭСППЗУ, 38
 - данных программы S7-200, 36
 - значений в ЭСППЗУ, 473
 - памяти переменных (V) в ЭСППЗУ, 39
 - программы, 13
 - программы в модуль памяти, 37
- Сохранение в постоянной памяти с помощью программы, 39
- Сохраняемая память, 36–39, 44
 - диапазоны, 44
- Сохраняемая память данных, 36
- Сохраняемые биты памяти, 38
- Сохраняемый таймер с задержкой включения (TONR), 198
 - пример, 202
- Специальная память
 - модемный модуль EM 241, 325
 - протокол Modbus Slave, 350
- Список команд. См. Редактор STL
- Сравнение времен оборота маркера, 237
- Срок службы электрической части, 385
- Стандартная профильная шина, 17
- Стандартные команды преобразования, 101
- Стандартный контакт, 72
- Стандарты, национальные и международные, 384
- Стандарт RS-232
 - кабель PPI Multi-Master, 229
 - режим свободно программируемого обмена данными, 229
- Стандарт RS-485, 224
- Станции оператора, определение, 52
- Стековые операции
 - дублирование вершины логического стека, 77
 - загрузка стека, 77
 - извлечение вершины стека, 77
 - копирование второго уровня стека, 77
 - логическое сопряжение первого и второго уровня стека по И, 77
 - логическое сопряжение первого и второго уровня стека по ИЛИ, 77
 - пример, 78
 - AENO, 77
- Строка, представление, 32
- Строки, формат, 31
- Строковая константа, 31
- Строковые команды
 - длина строки, 186
 - копирование подстроки из строки, 188
 - копирование строки, 186
 - поиск первого появления символа в строке, 189
 - поиск строки в строке, 189
 - соединение строк, 186
- Структурирование, программа, 53
- Суммирующий счетчик, 118
- Схема соединений для входов
 - EM 231, 408
 - EM 235, 409
- Схема соединений для выходов
 - EM 232, 410
 - EM 235, 410
- Схемы защиты, проектирование, 52
- Схемы соединений
 - аналоговые модули расширения, 404
 - входы и выходы CPU, 392
 - модули CPU, 392–394
 - модуль позиционирования EM 253, 443–444
 - цифровые модули расширения, 399–401
 - CPU 221 AC/DC/Relay, 392
 - CPU 221 DC/DC/DC, 392
 - CPU 222 AC/DC/Relay, 393, 394
 - CPU 222 DC/DC/DC, 393, 394
 - CPU 224 AC/DC/Relay, 393, 394
 - CPU 224 DC/DC/DC, 393, 394
 - CPU 226 AC/DC/Relay, 395
 - CPU 226 DC/DC/DC, 395
 - EM 221 DI 8 x AC, 400
 - EM 221 DI 8x24 VDC, 400
 - EM 222 DO 8 x Relay, 400
 - EM 222 DO 8x24 VDC, 400
 - EM 223 4x24 VDC In/4x24 VDC Out, 399
 - EM 223 DI 16/DO 16 x DC 24V, 401
 - EM 223 DI 16/DO 16 x 24 VDC Relay, 401
 - EM 223 DI 4/DO 4 x DC 24V/Relay, 401, 402
 - EM 223 DI 8/ DO 8 x 24 VDC/Relay, 401
 - EM 223 DI 8/DO 8 x 24 VDC, 401
 - EM 231 AI 4 x 12 Bit, 404
 - EM 231 RTD, 414
 - EM 231 thermocouple, 414
 - EM 232 AQ 2 x 12 Bit, 404
 - EM 235 AI 4/AQ 1 x 12 Bit, 404
- Счетчики, 28
 - определение скоростного счетчика (HDEF), 120
 - скоростной счетчик (HSC), 120
 - IEC
 - вычитающий счетчик, 118
 - реверсивный счетчик, 118
 - суммирующий счетчик, 118

- SIMATIC
 обратный счет, 115
 прямой счет, 115
 реверсивный счетчик, 116
- Счетчики, скоростные, 50
- Т**
- Таблица глобальных переменных, 63
 Таблица контура регулирования, 154
 PID, 374
 Таблица локальных переменных, 55, 64
 Таблица настроек и профилей, модуль позиционирования EM 255, 296
 Таблица перекрестных ссылок, 246
 Таблица профилей, модуль позиционирования EM 253, 296
 Таблица символов, 63
 адресация, 63
 конфигурирование для Modbus, 351
 Таблица состояний, 64
 контроль значений, 249
 принудительное присваивание значений, 250
 Таблица CRC, протокол Modbus Slave, 350
 Табличные команды
 ввод значения в таблицу, 191
 заполнение памяти, 194
 поиск значения в таблице, 195
 удаление первой записи, 192
 удаление последней записи, 192
 Таймерные команды
 прерывания, 159
 IEC
 импульсный таймер (TP), 203
 таймер с задержкой включения (TON), 203
 таймер с задержкой выключения (TOF), 203
 SIMATIC
 таймер с задержкой включения (TON), 198
 таймер с задержкой выключения (TOF), 198
 таймер с задержкой включения с запоминанием (TONR), 198
 Таймерные команды IEC, 203
 пример, 203
 Таймерные команды SIMATIC, 198
 пример, 200, 201, 202
 Таймер с задержкой включения (TON), 198, 203
 пример, 200
 Таймер с задержкой выключения (TOF), 198, 203
 пример, 201
 Таймер сообщения, 94
 Тангенс, 145
 Текстовая пейджинговая связь, модемный модуль EM 241, 312
 Текстовые дисплеи
 номер для заказа, 485
 шаблоны для передней панели (TD 200C), 485
 Текстовый дисплей TD 200, 5
 Текстовый дисплей TD 200
 номер для заказа, 485
- Текстовый дисплей TD 200C, номер для заказа, 485
 Текущее значение
 изменение в HSC, 133
 установка для HSC, 127
 Температурные диапазоны
 EM 231 RTD, 423–424
 EM 231 thermocouple, 418–419
 Технические данные
 аналоговый модуль расширения, 402
 кабель RS-232/PPI Multi-Master, 454
 модуль CP 243-1 Ethernet, 445
 модуль CP 243-1 IT Internet, 448
 модуль CP 243-2 AS-Interface, 450
 система микро ПЛК, 52
 цифровой модуль расширения, 396
 CPU, 387, 388
 EM 231 RTD, 413
 EM 231 thermocouple, 413
 EM 241 modem, 437
 EM 277 PROFIBUS-DP, 425
 Технические данные входов
 аналоговый модуль расширения, 402
 цифровой модуль расширения, 397
 CPU, 389
 Технические данные выходов
 аналоговый модуль расширения, 403
 цифровой модуль расширения, 398
 CPU, 390
 Типы данных, параметры подпрограммы, 207
 Требования к зазорам, 16
 Требования к оборудованию, 4
 Требования, протокол Modbus Slave, 350
- У**
- Увеличение памяти для программ, блокировка редактирования в режиме RUN, 48
 Удаленный адрес, установка для S7-200, 215
 Указания
 для индуктивных нагрузок, 22
 для ламповых нагрузок, 22
 по заземлению и подключению, 20
 по монтажу, 16
 по построению сети, 223
 по применению подавляющих цепочек, 22
 по разделению потенциалов, 20
 Указания по подключению линий переменного тока, 21
 Указания по подключению линий постоянного тока, 21
 Указатели, косвенная адресация, 33
 Управление без обратной связи, 254
 Управление перемещением без обратной связи, использование шаговых двигателей и серводвигателей, 257
 Управляющий байт, установка для HSC, 126
 Условия окружающей среды, технические данные
 эксплуатация, 385
 транспортировка и хранение, 385
 Условные входы, 59

Условные переходы, пример, 179
 Установка
 времени, 80
 даты, 80
 скорости передачи, 213
 текущего и предустановленного значений для HSC, 127
 удаленного адреса для S7-200, 215
 управляющего байта (HSC), 126
 Установка DIP-переключателей, кабель PPI Multi-Master, 9, 229
 Устранение неисправностей
 аппаратура S7-200, 251
 коды ошибок, 464
 нефатальные ошибки, 60
 указания, 251
 фатальные ошибки, 61
 Устройства человеко-машинного интерфейса, 227

Ф

Файл базы данных устройства (GSD), EM 277 PROFIBUS-DP, 433–434
 Фатальные ошибки, 61
 просмотр, 464
 Фильтрация
 аналоговые входы, 45
 цифровые входы, 45
 Формат адреса байта, 26
 Формат слова данных для входов
 EM 231, 408
 EM 235, 408
 Формат слова данных для выходов
 EM 232, 409
 EM 235, 409
 Формат сообщений для передачи данных CPU, модемный модуль EM 241, 332
 Формат текстовых сообщений, модемный модуль EM 241, 330
 Формат телефонного номера для передачи сообщений, модемный модуль EM 241, 329
 Функции
 CPU, 70
 модемный модуль EM 241, 310
 модуль позиционирования EM 253, 267
 Функции памяти
 команда групповой пересылки байтов, 168
 команда перестановки байтов, 185
 команда пересылки байтов, 166
 команда перемещения значения бита в регистр сдвига, 183
 команды циклического сдвига, 181
 команды сдвига, 181
 Функции, Modbus, 352
 Функции PTO/PWM, регистры (WMB66 – SMB85), 475
 Функциональный план. См. Редактор FBD

Ц

Целостность байта, PROFIBUS, 430

Целостность буфера, PROFIBUS, 430
 Целостность данных, PROFIBUS, 430
 Целостность слова, PROFIBUS, 430
 Цепь цифрового входа, 47
 Цикл, 24
 задание количества, 250
 таймеры, 200
 Цифровая пейджинговая связь, модемный модуль EM 241, 312
 Цифровой модуль расширения, адресация, 32
 Цифровые входы
 запись в, 25
 регистратор импульсов, 45
 фильтрация, 45
 чтение, 42
 Цифровые выходы, установка состояний, 43
 Цифровые модули, 4

Ч

Часы
 биты состояния, 468
 модуль, 452
 Часы реального времени
 установка расширенных часов реального времени, 81
 установка часов реального времени, 80
 чтение расширенных часов реального времени, 81
 чтение часов реального времени, 80
 Часы реального времени (TOD), 80
 Часы TOD, 80
 Чертежи, создание конфигурации, 53
 Чертежи конфигурации, 53
 Числа, представление, 26, 32
 Числа с плавающей точкой, 31, 151
 Числовые команды
 квадратный корень, 143
 косинус, 143
 натуральная экспонента, 143
 натуральный логарифм, 143
 синус, 143
 тангенс, 143
 Число, представление, 31
 Чтение данных из выходов, 24, 25
 Чтение часов реального времени команда, 80

Ш

Шаблоны для передней панели, TD 200 C, номер для заказа, 485
 Шаги, формирование для профиля перемещения, 260
 Шаговые двигатели, управление перемещением без обратной связи, 257
 Шина
 монтаж, 18
 размеры, 18

Широтно-импульсная модуляция (PWM), 50, 135, 255
 время цикла, 137
 значения таблицы профилей, 140
 конфигурирование с помощью битов SM-памяти, 138
 Мастер управления позиционированием, 135
 настройка выхода PWM, 255
 способы изменения, 138
 управление шаговым двигателем, 140
 Штекер RJ11, модемный модуль EM 241, 310, 438

Э

Электромагнитная совместимость, S7-200, 385
 стандарты излучения, 385
 стандарты помехоустойчивости, 385
 ЭСППЗУ
 коды ошибок, 464
 сохранение битов памяти (M), 38
 сохранение памяти переменных (V), 39, 473

А

ACCEL_TIME (время ускорения), модуль позиционирования EM 255, 258
 AENO, 77
 AI. См. Аналоговый вход (AI)
 AQ. См. Аналоговый выход (AQ)

С

CP 243-2 AS-I, технические данные, 450
 CP5411, 482
 CP5511, 482
 CP5611, 482
 CPU
 буферизация памяти, 3
 защита паролем, 48
 идентификационный регистр (SMB6), 471
 имитаторы входов, 458
 количество соединений, 216
 коммуникационные порты, 3
 модули расширения, 3
 монтажные размеры, 18
 обработка ошибок, 60
 память, 3
 поиск в сети, 215
 потребности в мощности, 17, 459
 размеры, 3
 скорость исполнения, 3
 цикл, 24
 цифровые входы/выходы, 3
 часы реального времени, 3
 CPU 224, пример программы для обмена данными DP, 435
 CPU, поддерживающие интеллектуальные модули, 325

D

DATx_WRITE, подпрограмма протоколирования данных, 371
 DECEL_TIME (время замедления), модуль позиционирования EM 255, 258

E

EN, 59, 67
 ENO, 59, 67
 Ethernet, 216
 TCP/IP, 217

G

GSD-файл, EM 277 PROFIBUS-DP, 433–434

H

HSC, 50
 Мастер команд, 121
 HyperTerminal, настройка кабеля RS-232/PPI Multi-Master, 239–483

I

Internet-устройства, CP 243-1 IT, 221
 Internet-модуль, Мастер, 449

M

Master-устройства, 213
 Micro/WIN. См. STEP 7-Micro/WIN

N

NETR, NETW, Мастер команд, 83

O

OP3, OP7, OP17, номера для заказа, 485

P

PID, таблица контура регулирования, расширенная, 374
 POSx_CFG, 286
 POSx_CLR, 285
 POSx_CTRL, 276
 POSx_DIS, 284
 POSx_GOTO, 278
 POSx_LD OFF, 281
 POSx_LD POS, 282
 POSx_MAN, 277
 POSx_RSEEK, 280
 POSx_RUN, 279
 POSx_SRATE, 2831

PPI/режим свободно программируемого обмена данными, радиомодем, 234
 PROFIBUS, master- и slave-устройства, 213
 PROFIBUS-DP
 модуль (EM 277), 428
 пример программы, 435
 стандартный обмен данными, 427
 целостность данных, 430
 PTO_CTRL, 262
 PTOx_CTRL, 262
 PTOx_LDPOS, 265
 PTOx_MAN, 264
 PTOx_RUN, 263
 PWMx_RUN, 254
 широотно-импульсная модуляция, 256

S

S-память, 31
 S7-200

 адресация, 26
 аккумуляторы, 29
 аналоговые входы (AI), 30
 аналоговые выходы (AQ), 31
 доступ к данным, 26
 загрузка в CPU, 36
 загрузка из CPU, 36
 защита паролем, 48
 исполнение логика управления, 24
 источник питания, 8
 как slave-устройство, 213, 427
 коды ошибок, 464
 компоненты системы, 2
 локальная память, 30
 модем, 233
 модули расширения, 3
 модули CPU, 2
 монтаж, 17
 области памяти, 70
 область битовой памяти (M), 27
 область локальной памяти (L), 30
 область памяти реле управления очередностью (S), 31
 область памяти счетчиков (C), 28
 область памяти таймеров (T), 28
 область памяти для входов (I), 27
 область памяти переменных (V), 27
 область специальной памяти (SM), 30
 обработка ошибок, 60
 память, 26
 поддерживаемые программы обработки прерываний, 158
 подключение с помощью STEP 7-Micro/WIN, 10
 постоянные значения, 32
 примеры конфигурации сетей, 218, 219, 220, 221
 программы обработки прерываний, 157
 радиомодем, 234
 размеры, 2
 реакция на фатальную ошибку, 464

 регистр входов образа процесса (I), 27
 регистр выходов образа процесса (Q), 27
 регистр образа процесса, 42
 режим RUN, 14, 41
 режим STOP, 14, 41
 свойства регистратора импульсов, 46
 сетевой адрес, 214–216
 скорости передачи, 213, 214–216
 скоростные счетчики, 29
 соглашения по программированию, 59
 сохранение данных, 36
 сохраняемая память, 44
 специальные функции, 42
 стандарт RS-232, 229
 схемы соединений, 392–394
 технические данные, 385
 указания по заземлению, 21
 указания по монтажу, 16
 указания по подключению, 21
 условия окружающей среды, 385
 устранение неисправностей, 251
 устранение неисправностей аппаратуры, 251
 цикл, 24, 42
 чтение и запись данных, 24
 электромагнитная совместимость, 385
 таблица состояний, 249
 С-память, 28
 М-память, 27
 Q-память, 27
 S-память, 31
 SM-память, 30
 Т-память, 28
 V-память, 27
 S7-300, примеры конфигураций сетей, 219
 S7-400, примеры конфигураций сетей, 219
 Slave-устройство, 213
 EM 277 PROFIBUS-DP, 427
 S7-200, 427
 SMB0: биты состояния, 468
 SMB1: биты состояния, 468
 SMB130: регистры управления свободно программируемым обменом данными, 472
 SMB131 – SMB165: регистры HSC3, HSC4, HSC5, 477
 SMB166 – SMB185: определение таблицы профилей PTO0, , 478
 SMB186 – SMB194: управление приемом сообщений, 476
 SMB2: принимаемые символы при свободно программируемом обмене данными, 469
 SMB200 – SMB549: состояние интеллектуальных модулей, 479
 SMB28, SMB29: аналоговый потенциометр, 50, 472
 SMB3: свободно программируемый обмен данными, ошибка, выявленная контролем четности, 469
 SMB30 и SMB130: свободно программируемый обмен данными, управляющие регистры, 472
 SMB31 и SMW32: управление записью в ЭСППЗУ, 475

SMB34 и SMB35: регистры прерываний, управляемых временем, 473
 SMB36 – SMB65: регистры HSC0, HSC1, HSC2, 473
 SMB4: переполнение очереди, 469
 SMB5: состояние входов-выходов, 470
 SMB6: регистр идентификатора CPU, 470
 SMB66 – SMB85: регистры PTO/PWM, 475
 SMB7: резерв, 470
 SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода, 471
 SMB86 – SMB94, SMB186 – SMB194: управление приемом сообщений, 476
 SMS, модемный модуль, 312
 SMW22 – SMW26: времена цикла, 472
 SMW98: ошибки шины ввода-вывода модулей расширения, 477
 STEP 7-Micro/WIN
 запуск, 9
 инструментальные средства отладки, 245
 интерфейс для EM 241, 311
 кабель PPI Multi-Master, 227
 как master-устройство, 213
 монтаж, 4
 наборы команд
 выбор, 57
 IEC 1131-3, 57
 SIMATIC, 57
 настройки параметров для обмена данными, 10
 номера для заказа, 482
 открытие, 55
 пакет программирования, 4
 плата CP, 227
 примеры конфигураций сетей, 218–221

редакторы программ, 55
 сетевой адрес, 214–216
 скорость передачи, 214–216
 создание программ, 55
 требования к компьютеру, 4
 требования к оборудованию, 4
 установка связи с S7-200, 10

T

T-память, 28
 TC/IP, протокол обмена данными, 217
 TCP/IP, протокол, 217
 TD 200, Мастер, 5
 Teleservice, 311
 TP-Designer для TP070, версия 1.0, 482

U

Команда USS4_DRV_CTRL, 337
 Команда USS4_INIT, 336
 Команда USS4_RPM_x, 340, 341
 Команда USS4_WPM_x, 340, 341

V

V-память, 27
 копирование в постоянную память, 40
 назначение адресов, 62
 сохранение в ЭСППЗУ, 39

Области памяти и функции S7-200

Описание	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Размер программы пользователя с редактированием в режиме RUN без редактирования в режиме RUN	4096 байт	4096 байт	8192 байта	12288 байт	16384 байта
	4096 байт	4096 байт	12288 байт	16384 байта	24576 байт
Размер данных пользователя	2048 байт	2048 байт	8192 байта	10240 байт	10240 байт
Регистр входов образа процесса	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7	I0.0 – I15.7
Регистр выходов образа процесса	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7	Q0.0 – Q15.7
Аналоговые входы (только чтение)	AIW0 – AIW30	AIW0 – AIW30	AIW0 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	AQW0 – AQW30	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB8191	VB0 – VB10239	VB0 – VB10239
Локальная память (L) ¹	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63	LB0 – LB63
Битовая память (M)	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7	M0.0 – M31.7
Специальная память (SM) Только чтение	SM0.0 – SM179.7 SM0.0 – SM29.7	SM0.0 – SM299.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7	SM0.0 – SM549.7 SM0.0 – SM29.7
Таймеры	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)	256 (T0 – T255)
Задержка включения с запоминанием	1 мс	T0, T64	T0, T64	T0, T64	T0, T64
	10 мс	T1 – T4, and T65 – T68	T1 – T4, and T65 – T68	T1 – T4, and T65 – T68	T1 – T4, and T65 – T68
	100 мс	T5 – T31, and T69 – T95	T5 – T31, and T69 – T95	T5 – T31, and T69 – T95	T5 – T31, and T69 – T95
		T32, T96	T32, T96	T32, T96	T32, T96
Задержка включения/ выключения	1 мс	T33 – T36, and T97 – T100	T33 – T36, and T97 – T100	T33 – T36, and T97 – T100	T33 – T36, and T97 – T100
	10 мс	T37 – T63, and T101 – T255	T37 – T63, and T101 – T255	T37 – T63, and T101 – T255	T37 – T63, and T101 – T255
	100 мс				
Счетчики	C0 – C255	C0 – C255	C0 – C255	C0 – C255	C0 – C255
Скоростные счетчики	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5	HC0 – HC5
Реле управления очередностью (S)	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7	S0.0 – S31.7
Аккумуляторные регистры	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3	AC0 – AC3
Переходы/метки	0 – 255	0 – 255	0 – 255	0 – 255	0 – 255
Вызов/Подпрограмма	0 – 63	0 – 63	0 – 63	0 – 63	0 – 127
Программы обработки прерываний	0 – 127	0 – 127	0 – 127	0 – 127	0 – 127
Положительные/ отрицательные фронты	256	256	256	256	256
PID-регуляторы	0 – 7	0 – 7	0 – 7	0 – 7	0 – 7
Порты	Порт 0	Порт 0	Порт 0	Порт 0, Порт 1	Порт 0, Порт 1

¹ LB60 – LB63 зарезервированы пакетом STEP 7-Micro/WIN, версия 3.0 или выше.

STL	Страница	STL	Страница	STL	Страница	STL	Страница	STL	Страница
=	73	AW > =	96	IBCD	99	MOVB	164	RLW	179
+D	140	AW <>	96	INCB	144	MOVD	164	ROUND	99
-D	140	BCDI	99	INCD	144	MOVR	164	RRB	179
* D	140	BIR	165	INCW	144	MOVW	164	RRD	179
/ D	140	BITIM	196	INVB	161	MUL	142	RRW	179
+I	140	BIW	165	INVD	161	NEXT	169	RTA	103
-I	140	BMB	166	INWV	161	NETR	81	RTS	107
=I	73	BMD	166	ITA	103	NETW	81	S	73
* I	140	BMW	166	ITB	99	NOT	70	SCAT	184
/ I	140	BTI	99	ITD	99	O	70	SCPY	184
+R	140	CALL	204	ITS	107	OB =	96	SCRE	172
-R	140	CEVNT	153	JMP	171	OB > =	96	SCRT	172
*R	140	CFND	187	LBL	171	OB >	96	SEG	99
/R	140	CITIM	196	LD	70	OB <	96	SFND	187
A	70	COS	143	LDB <=	96	OB < =	96	SHRB	181
AB < =	96	CRET	204	LDB =	96	OB <>	96	SI	73
AB =	96	CRETI	153	LDB >=	96	OD <	96	SIN	143
AB >	96	CSCRE	172	LDB >	96	OD < =	96	SLB	179
AB<	96	CTD	113	LDB <	96	OD =	96	SLD	179
AB > =	96	CTU	113	LDB <>	96	OD >	96	SLEN	184
AB <>	96	CTUD	113	LDD >=	96	OD > =	96	SLW	179
AD <	96	DECB	144	LDD <	96	OD <>	96	SPA	95
AD < =	96	DECD	144	LDD <=	96	OI	70	SQRT	143
AD =	96	DECO	112	LDD =	96	OLD	75	SRB	179
AD >	96	DECW	144	LDD >	96	ON	70	SRD	179
AD > =	96	DISI	153	LDD <>	96	ONI	70	SRW	179
AD <>	96	DIV	142	LDI	70	OR=	96	SSCPY	186
AENO	75	DLED	178	LDN	70	OR <	96	STD	110
AI	70	DTA	103	LDNI	70	OR<=	96	STI	110
ALD	75	DTCH	153	LDR=	96	OR >	96	STOP	167
AN	70	DTI	99	LDR <	96	OR >=	96	STR	110
ANDB	162	DTR	99	LDR<=	96	OR <>	96	SWAP	183
ANDD	162	DTS	107	LDR >	96	ORB	162	TAN	143
ANDW	162	ED	70	LDR>=	96	ORD	162	TODR	78
ANI	70	ENCO	112	LDR <>	96	ORW	162	TODRX	78
AR=	96	END	167	LDS	75	OS=	98	TODW	78
AR <	96	ENI	153	LDS=	98	OS<>	98	TODWX	78
AR<=	96	EU	70	LDS<>	98	OW <	96	TOF	196
AR >	96	EXP	143	LDW <=	96	OW < =	96	TON	196
AR>=	96	FIFO	190	LDW <	96	OW =	96	TONR	196
AR <>	96	FILL	192	LDW =	96	OW >	96	TRUNC	99
AS=	98	FND <	193	LDW >	96	OW > =	96	WDR	167
AS<>	98	FND <>	193	LDW >=	96	OW <>	96	XMT	86
ATCH	153	FND =	193	LDW <>	96	PID	145	XORB	162
ATH	103	FND >	193	LIFO	190	PLS	133	XORD	162
ATT	189	FOR	169	LN	143	R	73	XORW	162
AW <	96	GPA	95	LPP	75	RCV	86		
AW < =	96	HDEF	118	LPS	75	RI	73		
AW=	96	HSC	118	LRD	75	RLB	179		
AW >	96	HTA	103	LSCR	172	RLD	179		