SIEMENS

SIMATIC

Программируемый контроллер S7-200

Системное руководство

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 в сети	7
Устранение неисправностей аппаратуры и инструментальные средства для тестирования программ	8
Создание программы для модуля позиционирования	9
Создание программы для модема	10
Управление приводом MicroMaster с помощью библиотеки протокола USS	11
Использование библиотеки протокола Modbus	12
Технические данные	A
Расчет баланса мощностей	B
Сообщения об ошибках	C
Специальные биты памяти (SM)	D
Номера для заказа S7–200	E
Времена выполнения команд STL	F
Краткая информация об S7–200	G
Предметный указатель	

6ES7 298-8FA22-8BH0

Издание 3

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, тяжким телесным повреждениям или существенному имущественному ущербу, если не будут предприняты соответствующие меры предосторожности.

Предупреждение
Указывает на потенциально опасную ситуацию, которая **может** привести к смерти, тяжким телесным повреждениям или существенному имущественному ущербу, 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 2002 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 2002
Технические данные могут быть изменены.

Предисловие

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.

Целевая группа

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.

Область применения руководства

Информация, содержащаяся в данном руководстве, относится, в частности, к следующим продуктам:

- Модели CPU S7–200: CPU 221, CPU 222, CPU 224, CPU 226 и CPU 226XM
- Модули расширения S7–200 EM 22x
- STEP 7-Micro/WIN, версия 3.2, пакет 32–битного программного обеспечения для S7–200
- Библиотека команд и дизайнер TP STEP 7-Micro/WIN для TP070, версия 1.0, набор программных инструментальных средств для заказчиков, которые используют S7–200 с другими компонентами, например, сенсорная панель TP070, Modbus или привод MicroMaster

Соответствие стандартам

Серия SIMATIC S7–200 удовлетворяет следующим стандартам:

- Директива Европейского сообщества (CE) по устройствам низкого напряжения 73/23/EEC
  EN 61131-2: Программируемые контроллеры – Требования к оборудованию
- Директива Европейского сообщества (CE) по электромагнитной совместимости 89/336/EEC

  Стандарты электромагнитных излучений
  EN 50081-1: жилищное строительство, торговля и легкая промышленность
  EN 50081-2: промышленная среда

  Стандарты устойчивости к электромагнитным помехам
  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 взаимные исследования]: FM класс I, раздел 2, группы опасных помещений A, B, C и D, T4A и класс I, зона 2, IIC, T4

За полной информацией обратитесь к Приложению.

Допуски к эксплуатации в морском деле

К моменту сдачи этого руководства в печать серия продуктов SIMATIC S7–200 соответствовала стандартам нижеприведенных морских агентств. За информацией о текущих допусках к эксплуатации обращайтесь в местные представительства 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-8071
Bureau Veritas [Бюро Веритас] (BV)	09051 / A2 BV
Nippon Kaiji Kyokai [Ниппон Кайдзи Кёкай - Япония] (NK)	A-534

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 об эксплуатации 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 (Создание программы для модуля позиционирования) дает информацию о командах и мастере, с помощью которых можно создать программу для модуля позиционирования 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. 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.
- 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.

Дополнительная информация и помощь

Информация об S7–200 и STEP 7-Micro/WIN

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 получает доступ к обширной оперативной помощи в режиме online для STEP 7-Micro/WIN. Онлайновая помощь включает в себя полезную информацию, необходимую для начального знакомства с программированием S7–200, а также много других тем.

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 и приемы

Компакт-диск с документацией содержит советы и приемы (Tips and Tricks), 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 и приемов можно найти также в Интернете на сайте S7–200.

Интернет: www.siemens.com/S7-200

Для получения дополнительной информации о продуктах и услугах фирмы Siemens, технической поддержки, получения ответов на часто задаваемые вопросы (FAQ – frequently asked questions), обновления продуктов и советов относительно применений используйте следующие адреса в Интернете:

- www.ad.siemens.de *для общей информации о фирме Siemens*

  Этот сайт департамента Автоматизации и приводов фирмы Siemens содержит информацию о линии продуктов SIMATIC и других продуктах, которые могут быть получены от фирмы Siemens.

- www.siemens.com/S7-200 *для информации о продуктах 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.

Техническая помощь и приобретение продуктов S7–200

Местные представительства или дистрибьюторы фирмы Siemens

Для получения ответов на любые технические вопросы, для обучения работе с продуктами S7–200 или для заказа продуктов S7–200 обращайтесь в представительства или к дистрибьюторам фирмы Siemens. Так как персонал, занимающийся продажами, технически обучен и имеет конкретные знания о возможностях использования и процессах, а также об отдельных продуктах фирмы Siemens, которые вы используете, он может дать вам наискорейшие и наиболее эффективные ответы на любые проблемы, которые могут вам встретиться.

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:

- *Для вызовов, исходящих с территории Соединенных Штатов Америки*
  Местное время: с понедельника по пятницу с 8.00 до 19.00 по времени Восточного побережья
  Тел.: +1 800 241-4453
  Факс: +1 (0) 770 740-3699
  E–Mail: drives.support@sea.siemens.com
- *Для вызовов, исходящих из Америки вне США*
  Местное время: с понедельника по пятницу с 8.00 до 19.00 по времени Восточного побережья
  Тел.: +1 (0) 770 740-3505
  Факс: +1 (0) 770 740-3699
  E–Mail: drives.support@sea.siemens.com
- *Для вызовов, исходящих из Европы и Африки*
  Местное время (Нюрнберг): с понедельника по пятницу с 07.00 до 17.00
  Тел.: +49 (0) 180 5050-222
  Факс: +49 (0) 180 5050-223
  E–Mail: techsupport@ad.siemens.de
- *Для вызовов, исходящих из Азии и Австралии*
  Местное время (Singapore): с понедельника по пятницу с 08.30 до 17.30
  Тел.: +65 (0) 740-7000
  Факс: +65 (0) 740-7001
  E–Mail: drives.support@sae.siemens.com.sg

Содержание

1 Обзор продукта 1

CPU S7–200 2
Модули расширения S7–200 3
Пакет для программирования STEP 7-Micro/WIN 3
Возможности обмена данными 4
Индикаторные панели 4

2 Первые шаги 5

Подключение CPU S7–200 6
Создание программы-примера 8
Загрузка программы-примера 11
Перевод S7–200 в режим RUN 11

3 Монтаж S7–200 13

Указания по монтажу устройств S7–200 14
Монтаж и демонтаж модулей S7–200 15
Указания по заземлению и подключению 18

4 Основы ПЛК 21

Выполнение логики управления с помощью S7–200 22
Доступ к данным S7–200 24
Как в S7–200 производится сохранение и извлечение данных 34
Сохранение вашей программы в модуле памяти 36
Установка режима работы CPU S7–200 37
Использование вашей программы для сохранения памяти переменных в ЭСППЗУ 38
Функции S7–200 39

5 Основы программирования, соглашения и функции 47

Рекомендации по проектированию системы с микроконтроллером 48
Основные элементы программы 49
Использование STEP 7-Micro/WIN для создания вашей программы 51
Наборы команд SIMATIC и IEC 1131-3 53
Соглашения, используемые в редакторах программ 54
Создание программы управления с помощью мастеров 56
Устранение ошибок в S7–200 56
Назначение адресов и начальных значений в редакторе блоков данных 58
Использование таблицы символов для символической адресации переменных 58
Использование локальных переменных 59
Контроль над программой с помощью таблицы состояний 59
Создание библиотеки команд 60
Функции тестирования программы 60

6 Набор команд S7–200 61

Соглашения, используемые для описания команд 63
Области памяти и функции S7–200 64
Битовые логические операции 66
Контакты 66
Катушки 68
Стековые операции 70
Функциональный блок с двумя устойчивыми состояниями: преимущество установки и преимущество сброса 72
Команды, связанные с временем 73
Команды для обмена данными 74
Команды чтения из сети и записи в сеть 74
Команды передачи и приема (свободно программируемый обмен данными) 79
Команды получения и установки адреса порта 88
Команды сравнения 89
Сравнение числовых величин 89
Сравнение строк 91
Команды преобразования 92
Стандартные команды преобразования 92
Команды преобразования ASCII 96
Команды преобразования строк 100
Команды кодирования и декодирования 105
Команды счета 106
Счетчик SIMATIC 106
Счетчик IEC 109
Скоростные счетчики 111
Команда вывода импульсов 125
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 155
Логические операции 162
Операции инвертирования 162
Операции И, ИЛИ и исключающее ИЛИ 163
Команды пересылки 165
Пересылка байта, слова, двойного слова или вещественного числа 165
Непосредственное чтение или запись и пересылка байта 166
Команды групповой пересылки 167
Команды управления программой 168
Условное завершение 168
Останов 168
Сброс времени контроля 168
Команды формирования программного цикла For-Next 170
Команды перехода 172
Команды для реле управления очередностью (SCR) 173
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
Операции с подпрограммами 203

7 Обмен данными в сети 207

Основы обмена данными в сети для S7–200 208
Выбор протокола связи для вашей сети 211
Инсталляция и деинсталляция интерфейсов для обмена данными 216
Построение сети 218
Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными 222
Использование в сети модемов и STEP 7-Micro/WIN 224
Для опытных пользователей 228

8 Устранение неисправностей аппаратуры и инструментальные средства для тестирования программ 235

Функции для отладки вашей программы 236
Отображение состояния программы 238
Использование таблицы состояний для контроля и изменения данных в S7–200 239
Принудительное присваивание значений 240
Исполнение программы в течение определенного количества циклов 240
Указания по устранению неисправностей аппаратуры 241

9 Создание программы для модуля позиционирования 243

Функции модуля позиционирования 244
Конфигурирование модуля позиционирования 246
Команды позиционирования, созданные мастером управления позиционированием 257
Примеры программ для модуля позиционирования 269
Наблюдение за модулем позиционирования с помощью панели управления EM 253 274
Коды ошибок для модуля и команд позиционирования 276
Для опытных пользователей 278

10 Создание программы для модемного модуля 287

Функции модемного модуля 288
Использование мастера конфигурирования модемного модуля 294
Обзор команд модемного модуля и ограничений 298
Команды для модемного модуля 299
Пример программы для модемного модуля 303
CPU S7–200, поддерживающие интеллектуальные модули 303
Биты специальной памяти для модемного модуля 304
Для опытных пользователей 306
Формат телефонных номеров для передачи сообщений 308
Формат текстовых сообщений 309
Формат сообщений для передачи данных CPU 310

11 Управление приводом MicroMaster с помощью библиотеки протокола USS 311

Требования для использования протокола USS 312
Расчет времени, необходимого для обмена данными с приводом 313
Использование команд USS 314
Команды для протокола USS 315
Примеры программ для протокола USS 322
Коды ошибок выполнения команд USS 323
Подключение и наладка привода MicroMaster серии 3 324
Подключение и наладка привода MicroMaster серии 4 327

12 Использование библиотеки протокола Modbus 329

Требования для использования протокола Modbus 330
Инициализация и время исполнения протокола Modbus 330
Адресация Modbus 331
Использование команд протокола slave-устройств Modbus 332
Команды протокола slave-устройств Modbus 333

A Технические данные 337

Общие технические данные 338
Технические данные CPU 340
Технические данные цифровых модулей расширения 346
Технические данные аналоговых модулей расширения 351
Технические данные модулей расширения для термопар и RTD 361
Технические данные модуля EM 277 PROFIBUS-DP 373
Технические данные модемного модуля EM 241 385
Технические данные модуля позиционирования EM 253 387
Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2) 393
Дополнительные съемные модули 395
Кабель с разъемами для модулей расширения 395
Кабель PC/PPI 396
Имитаторы входов 398

B Расчет баланса мощностей 399

C Сообщения об ошибках 403

Коды фатальных ошибок и сообщения 404
Ошибки этапа выполнения 405
Нарушение правил компиляции 406

D Биты специальной памяти (SM) 407

SMB0: биты состояния 408
SMB1: биты состояния 408
SMB2: принимаемые символы при свободно программируемом обмене данными 409
SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными 409
SMB4: переполнение очереди ожидания 409
SMB5: состояние входов/выходов 410
SMB6: идентификационный регистр CPU 410
SMB7: резерв 410
SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода 411

SMW22 – SMW26: времена цикла 412
SMB28 и SMB29: аналоговый потенциометр 412
SMB30 и SMB130: регистры управления свободно программируемым обменом данными 412
SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ) 413
SMB34 и SMB35: регистры интервалов времени для прерываний, управляемых временем 413
SMB36 – SMB65: регистры HSC0, HSC1 и HSC2 413
SMB66 – SMB85: регистры PTO/PWM 415
SMB86 – SMB94 и SMB186 – SMB194: управление приемом сообщений 416
SMW98: ошибки в шине расширения 417
SMB130: регистр управления свободно программируемым обменом данными (см. SMB30) 417
SMB131 – SMB165: регистры HSC3, HSC4 и HSC5 417
SMB166 – SMB185: таблица определения профилей PTO0, PTO1 418
SMB186 – SMB194: управление приемом сообщений (см. SMB86 – SMB94) 418
SMB200 – SMB549: состояние интеллектуальных модулей 419

E Номера для заказа S7–200 421

F Времена выполнения команд STL 425

G Краткая информация об S7–200 431

Предметный указатель 437

Обзор продукта

1

Семейство S7–200 включает в себя различные малые устройства управления (микро-ПЛК), с помощью которых можно управлять широким спектром устройств для решения ваших задач автоматизации.

S7–200 контролирует входы и изменяет выходы под управлением программы пользователя. Программа пользователя 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

CPU S7–200 2

Модули расширения S7–200 3

Пакет для программирования STEP 7-Micro/WIN 3

Возможности обмена данными 4

Индикаторные панели 4

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.

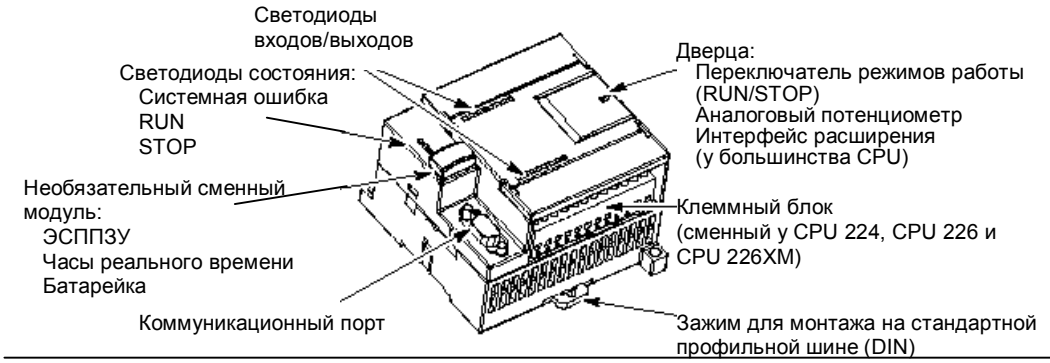


Рис. 1-1. Микро-ПЛК S7-200

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 1–1 приведены некоторые из функций различных CPU. Подробную информацию для конкретных CPU вы найдете в Приложении А.

Таблица 1–1. Сравнение различных исполнений CPU S7–200

Характеристика	**CPU 221**	**CPU 222**	**CPU 224**	**CPU 226**	**CPU 226XM**
Размеры (мм)	90 x 80 x 62	90 x 80 x 62	120,5 x 80 x 62	190 x 80 x 62	190 x 80 x 62
Память программ	2048 слов	2048 слов	4096 слов	4096 слов	8192 слов
Память данных	1024 слов	1024 слов	2560 слов	2560 слов	5120 слов
Буферизация памяти	тип. 50 часов	тип. 50 часов	тип. 190 часов	тип. 190 часов	тип. 190 часов
Встроенные входы/ выходы	6 входов/ 4 выхода	8 входов / 6 выходов	14 входов / 10 выходов	24 входа/ 16 выходов	24 входа / 16 выходов
Модули расширения	0	2	7	7	7
Скоростные счетчики 1-фазные 2-фазные	 4 при 30 кГц 2 при 20 кГц	 4 при 30 кГц 2 при 20 кГц	 6 при 30 кГц 4 при 20 кГц	 6 при 30 кГц 4 при 20 кГц	 6 при 30 кГц 4 при 20 кГц
Импульсные выходы (DC)	2 при 20 кГц	2 при 20 кГц	2 при 20 кГц	2 при 20 кГц	2 при 20 кГц
Аналоговый потенциометр	1	1	2	2	2
Часы реального времени	Сменный модуль	Сменный модуль	встроены	встроены	встроены
Порты связи	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	0,37 микросекунды/операцию				

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 вы можете расширить функциональные возможности своего CPU S7–200. В таблице 1–2 приведен список имеющихся в настоящее время модулей расширения. Подробную информацию для конкретных модулей вы найдете в Приложении А.

Таблица 1–2. Модули расширения S7–200

Модули расширения	Виды		
Цифровые модули			
Ввод	8 вх. пост. тока	8 вх. перем. тока	
Вывод	8 вых. пост. тока	8 вых. перем. тока	8 реле
Комбинация	4 вх. пост. тока/ 4 вых. пост. тока 4 вх. пост. тока/4 реле	8 вх. пост. тока/ 8 вых. пост. тока 8 вх. пост. тока/ 8 реле	16 вх. пост. тока/ 16 вых. пост. тока 16 вх. пост. тока/16 реле
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	Модем	PROFIBUS–DP
Другие модули	AS-интерфейс		

Пакет для программирования STEP 7-Micro/WIN

Пакет для программирования STEP 7-Micro/WIN предоставляет дружественную 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 онлайновую 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 95, Windows 98,
  Windows 2000, Windows Me (Millennium Edition) или Windows NT 4.0 (или выше)
- не менее 50 Мбайт свободного пространства на жестком диске
- мышь (рекомендуется)

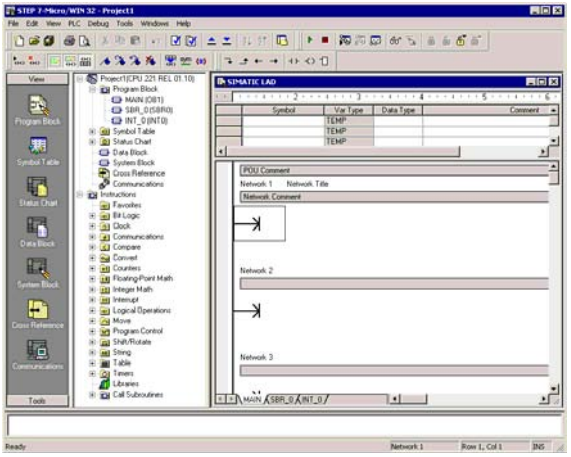


Рис. 1–2. STEP 7-Micro/WIN

Установка STEP 7-Micro/WIN

Вставьте компакт-диск со STEP 7-Micro/WIN в дисковод для компакт-дисков своего компьютера. Программа 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 NT или Windows 2000 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 PC/PPI или с использованием платы коммуникационного процессора (CP) с кабелем MPI для сетей MPI и PROFIBUS-DP.

Использование кабеля PC/PPI 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 PC/PPI может использоваться также для присоединения к S7–200 других устройств для обмена данными.

Для использования кабеля MPI вы должны также установить в своем компьютере плату CP. Плата CP предоставляет дополнительные аппаратные средства, необходимые для построения соединений с более высокими скоростями передачи и обработки обмена данными в скоростных сетях.

Индикаторные панели

Текстовый дисплей TD 200

TD 200 – это 2-строчный текстовый дисплей с 20 символами в каждой строке, который может быть присоединен 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 представляет собой недорогой интерфейс с вашим приложением, позволяющий вам просматривать, контролировать и изменять процессные переменные вашего приложения.

Функции и технические данные TD 200.

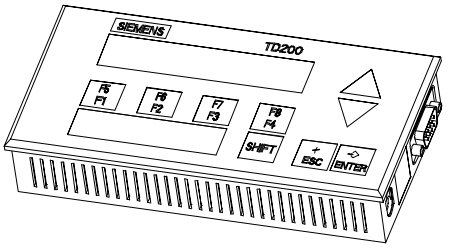


Рис. 1–3. Текстовый дисплей TD 200

Сенсорная панель TP070

Сенсорная панель TP070 – это индикаторное устройство, которое может быть присоединено к S7–200. С помощью этой сенсорной панели вы можете настроить свой интерфейс оператора в соответствии с требованиями пользователя.

TP070 может отображать пользовательские графики, ползунковые индикаторы, переменные приложения, экранные кнопки и т.д. на удобной для пользователя сенсорной панели.

Дополнительный компакт-диск TP-Designer for TP070, версии 1.0 предоставляет программное обеспечение дизайнера TP для программирования вашего TP070.

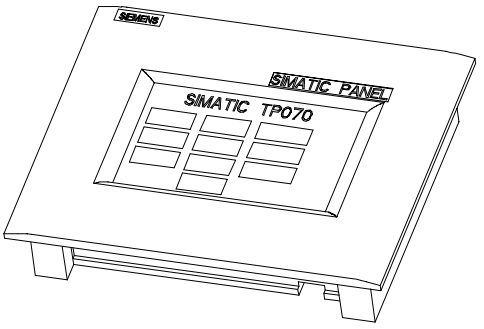


Рис. 1–4. Сенсорная панель TP070

Первые шаги 2

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 PC/PPI, CPU S7–200 и устройство программирования, на котором установлено программное обеспечение STEP 7-Micro/WIN.

В этой главе

Подключение CPU S7–200 6
Создание программы-примера 8
Загрузка программы-примера 11
Перевод S7–200 в режим RUN 11

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 исполнений CPU S7–200 постоянного и переменного тока.

Перед монтажом или демонтажем любого электрического устройства вы должны обеспечить, чтобы питание этого устройства было выключено. Выполните 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.

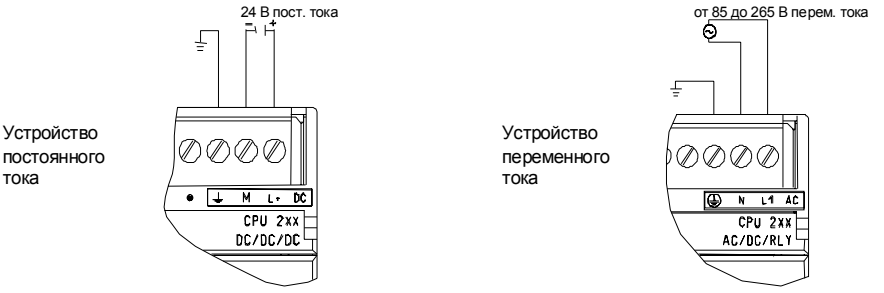


Рис. 2–1. Подключение питания к CPU S7–200

Присоединение кабеля PC/PPI

На рис. 2–2 показан кабель PC/PP, соединяющий S7–200 с устройством программирования. Для подключения кабеля PC/PPI:

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

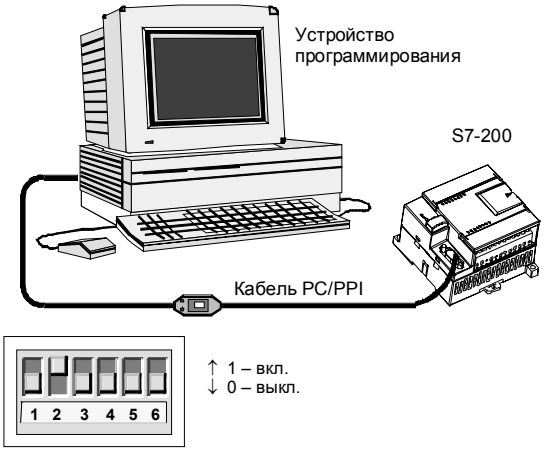


Рис. 2–2. Присоединение кабеля PC/PPI

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.

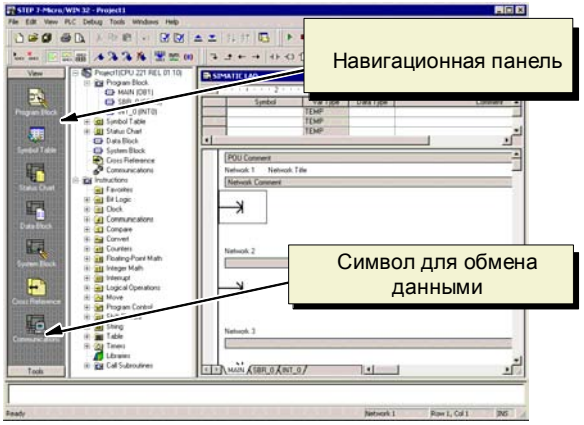


Рис. 2–3. Новый проект STEP 7-Micro/WIN

Проверка параметров обмена данными для STEP 7-Micro/WIN

Проект-пример использует для STEP 7-Micro/WIN и кабеля PC/PPI настройки по умолчанию. Проверьте эти настройки:

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.

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

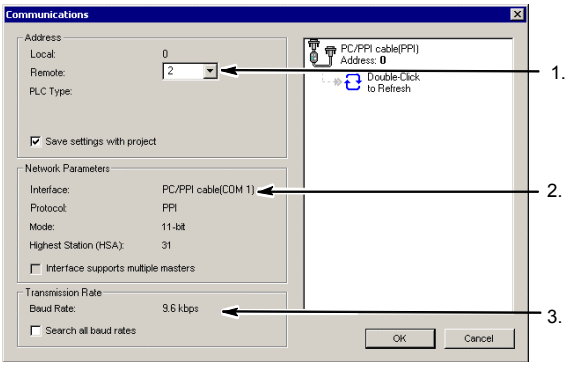


Рис. 2–4. Проверка параметров обмена данными

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 OK.

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.

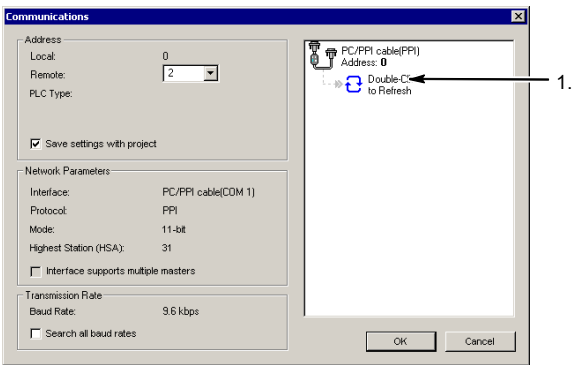


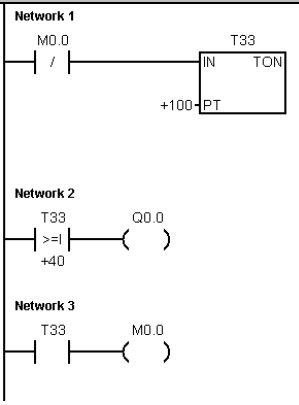
Рис. 2–5. Установка связи с S7–200

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.

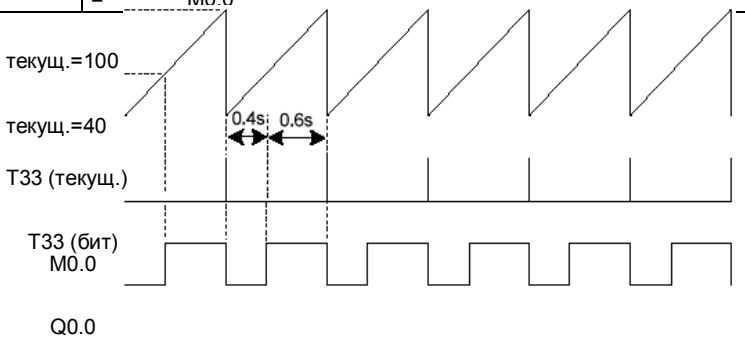
Команды для этого примера введите в редакторе LAD (KOP). Следующий пример показывает всю программу в виде контактного плана (LAD) и в виде списка команд (STL, AWL). Комментарии к сегменту в STL-программе объясняют логику для каждого сегмента. Импульсная диаграмма показывает, как программа работает.

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 быстр, 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 быстр, 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.

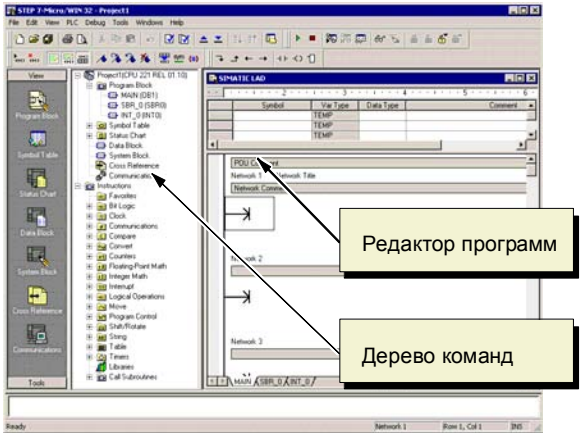


Рис. 2–6. Окно STEP 7-Micro/WIN

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 нормально замкнутый 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.

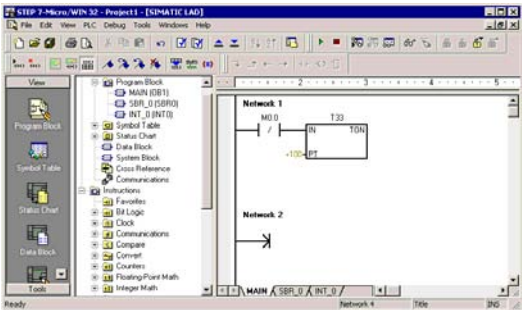


Рис. 2–7. Сегмент 1

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 S7–200. Для ввода команды сравнения:

1. Дважды щелкните на символе компаратора (Compare), чтобы отобразить команды сравнения. Выберите команду >=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.

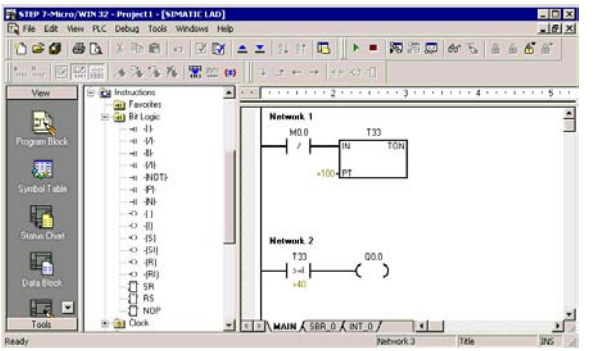


Рис. 2–8. Сегмент 2

5. Введите следующую величину для сравнения со значением таймера: 40
6. Нажмите клавишу Return, чтобы ввести это значение.

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 нормально открытый 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.

Чтобы ввести катушку для включения M0.0:

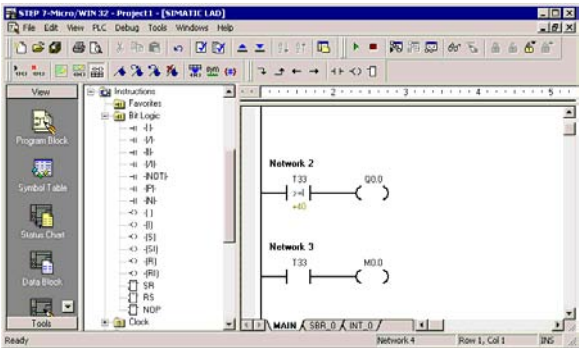


Рис. 2–9. Сегмент 3

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 OK.

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

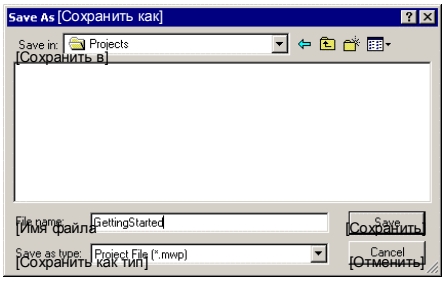


Рис. 2–10. Сохранение примера проекта

Загрузка программы-примера

	Совет Каждый проект STEP 7-Micro/WIN ставится в соответствие модели CPU (CPU 221, CPU 222, CPU 224, CPU 226 или CPU 226XM). 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]».

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 OK.

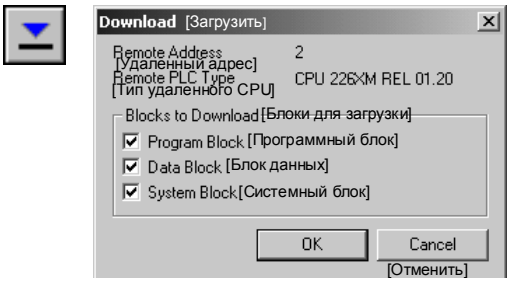


Рис. 2–11. Загрузка программы

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].

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 OK, 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.

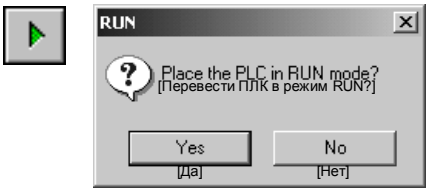


Рис. 2–12. перевод S7–200 в режим RUN

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 установка была простой. 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 14

Монтаж и демонтаж модулей S7–200 15

Указания по заземлению и подключению 18

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 обращайте 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° C. 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.

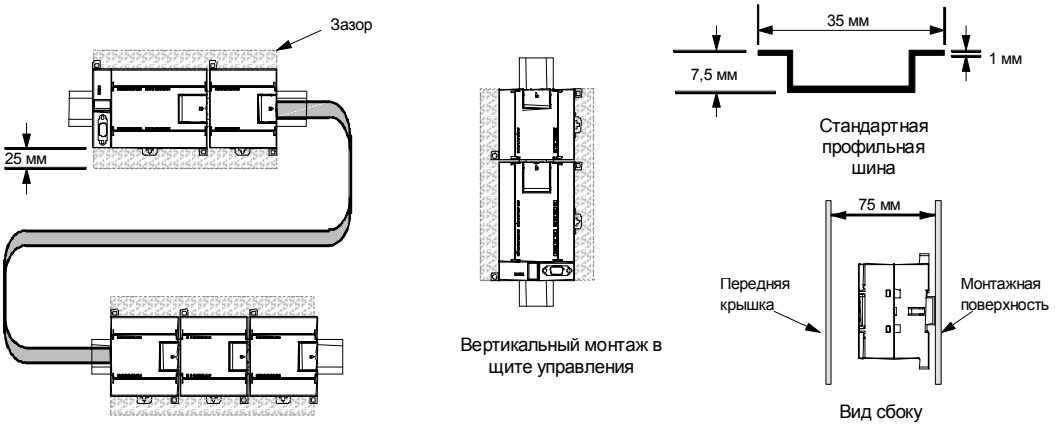


Рис. 3–1. Возможности монтажа, ориентация и зазор

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 поставлять 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 A 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 B, вы можете рассчитать, какую мощность (или ток) может поставлять 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 A 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 (M) 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 удостоверяйтесь, что перед монтажом или демонтажем устройства питающее напряжение S7–200 отключено.

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

Предупреждение

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

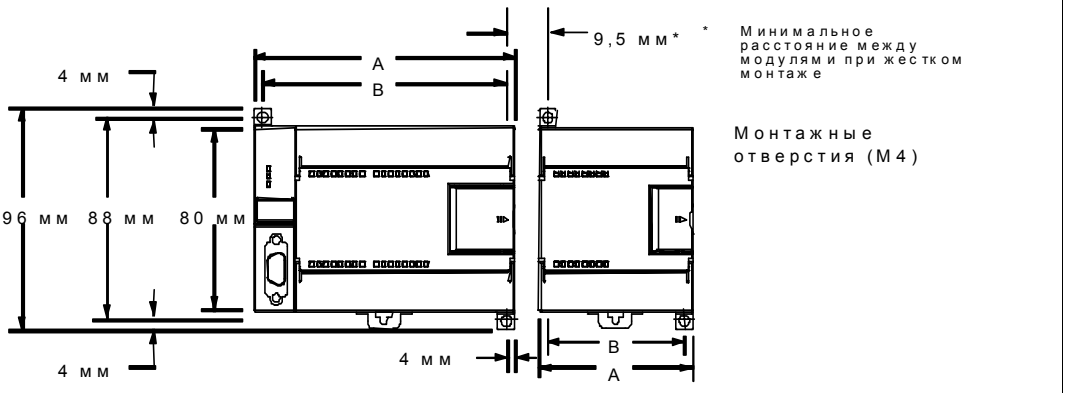
Если устройство S7–200 заменяется другим вариантом, неправильно ориентировано или смонтировано в неправильном порядке, то это может привести к гибели или серьезным телесным повреждениям персонала и/или к повреждению оборудования.

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 A	Ширина B
CPU 221 и CPU 222	90 мм	82 мм
CPU 224	120,5 мм	112,5 мм
CPU 226 и CPU 226XM	196 мм	188 мм
Модули расширения: 8 входов/выходов пост. тока и реле (8I, 8Q, 4I/4Q, 2 AQ)	46 мм	38 мм
Модули расширения: 16 цифровых входов/выходов (8I/8Q), аналоговые входы/выходы (4AI, 4AI/1AQ), RTD, термопара, PROFIBUS, 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. Проделайте отверстия 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.

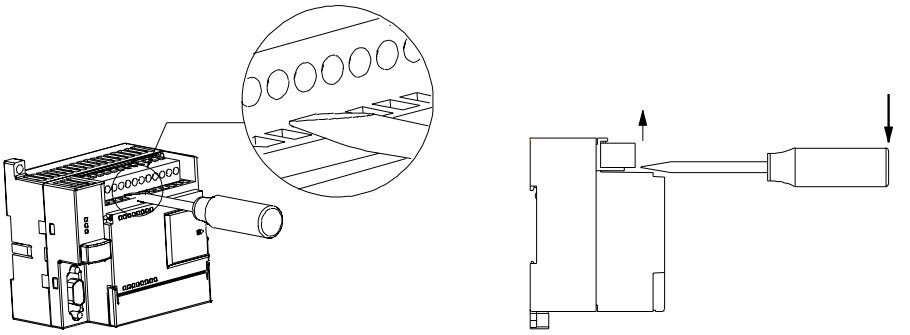
2. Отсоедините от устройства все провода и кабели. Это облегчается тем, что большинство CPU S7–200 и модулей расширения имеют съемные клеммные блоки.
3. Если к демонтируемому устройству присоединены модули расширения, то откройте переднюю дверцу и отсоедините плоский кабель соседнего модуля расширения.
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 A, 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 E.

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.



Ри

с. 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 удостоверяйтесь, 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 Voltage [Ограниченное напряжение].

Предупреждение
Использование источников питания без потенциальной развязки или с однократной изоляцией для цепей низкого напряжения из 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 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 мм^2 (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 нейтральным проводом.

Прокладывайте провода по возможно более короткому пути и обращайте внимание на то, чтобы поперечное сечение провода соответствовало требуемому току. Клеммный блок можно присоединять проводами с поперечным сечением от 2 мм^2 до 0,3 мм^2 (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 A 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 A. 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.

Защитные схемы

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 (A), включенного параллельно индуктивной нагрузке, но если ваше приложение требует быстрого отключения, то рекомендуется, кроме того, использовать стабилитрон (B). Рассчитывайте стабилитрон на ток в выходной цепи.

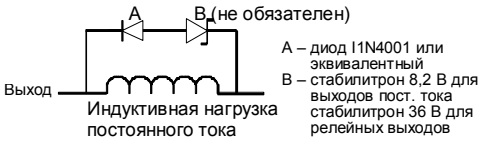


Рис. 3–3. Защитная схема для нагрузки постоянного тока

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. 3–3 показан пример защитной схемы для нагрузки переменного тока. В большинстве приложений металлооксидный варистор (MOV) может ограничить пиковое напряжение и обеспечить защиту для внутренних цепей S7–200. Обратите внимание на то, чтобы рабочее напряжение варистора было, по крайней мере, на 20 % больше, чем номинальное напряжение сети.

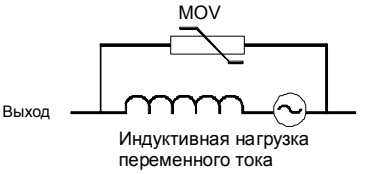


Рис. 3–4. Защитная схема для нагрузки переменного тока

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 22
Доступ к данным S7–200 24
Как в S7–200 производится сохранение и извлечение данных 34
Сохранение программы в модуле памяти 36
Установка режима работы CPU S7–200 37
Использование вашей программы для сохранения памяти переменных в ЭСППЗУ 38
Функции S7–200 39

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.

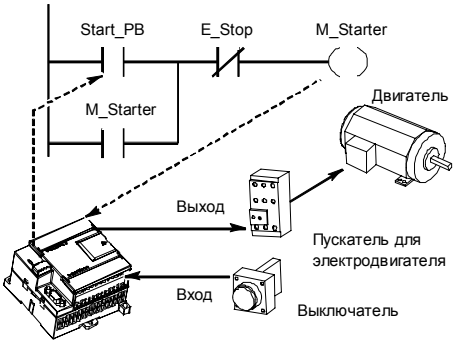


Рис. 4–1. Управление входами и выходами

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.

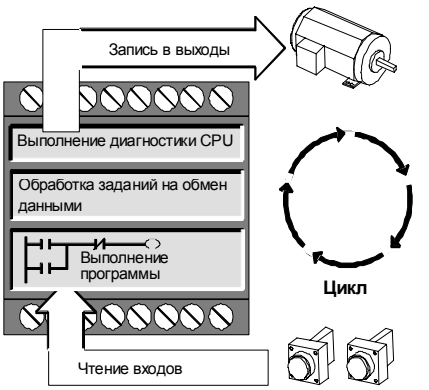


Рис. 4–2. Цикл S7–200

Выполнение цикла 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 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 физического модуля всякий раз, когда ваша программа обращается к аналоговому входу.

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, термопарами и AS–интерфейсом.

Обработка 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	*Неприменимо*	*Неприменимо*	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 областью памяти и адресом байта (I = input [вход], 3 = байт 3) следует точка («.»), чтобы отделить адрес бита (бит 4).

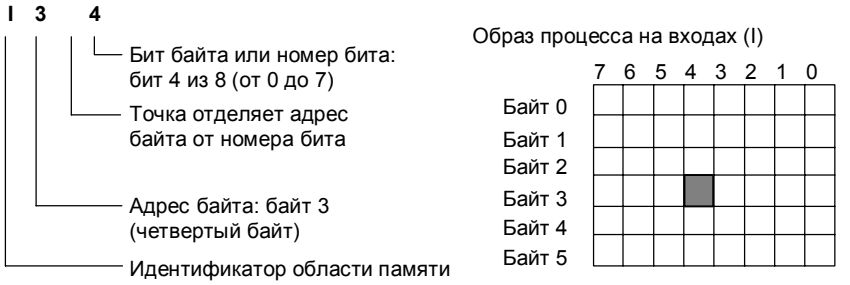


Рис. 4–3. Адресация байт.бит

Когда вы применяете 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.

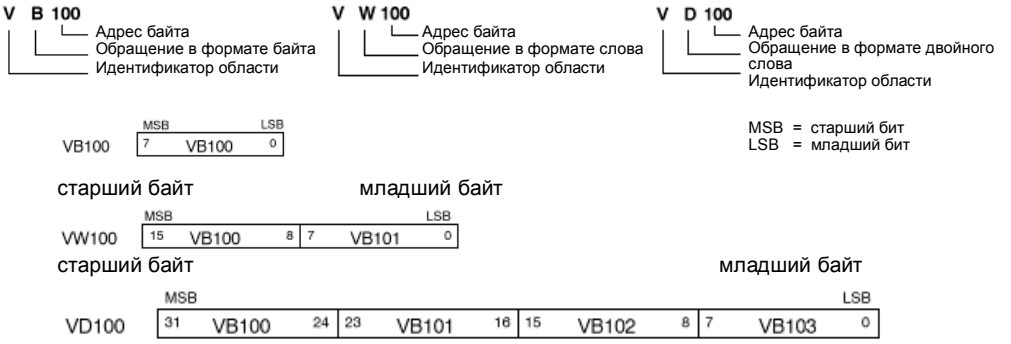


Рис. 4–4. Обращение к одному и тому же адресу в формате байта, слова и двойного слова

Обращение к данным в областях памяти

Регистр входов образа процесса: I

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

Бит:	I*[адрес байта].[адрес бита]*	I0.1
Байт, слово или двойное слово:	I*[длина][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*[адрес байта].[адрес бита]*	Q1.1
Байт, слово или двойное слово:	Q*[длина][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*[адрес байта].[адрес бита]*	V10.2
Байт, слово или двойное слово:	V*[длина][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*[адрес байта].[адрес бита]*	M26.7
Байт, слово или двойное слово:	M*[длина][начальный адрес байта]*	MD20

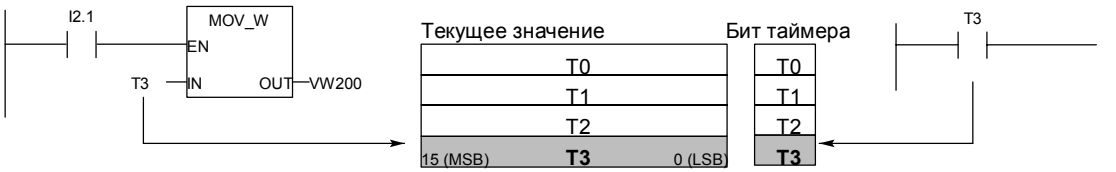
Таймеры: T

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 (значение таймера).
- 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.

Формат: T*[номер таймера]* T24



Обращается к текущему значению Обращается к биту 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.

Формат: C*[номер счетчика]* C24

Обращается к текущему значению Обращается к биту 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 главе 6.

Формат: AC*[номер аккумулятора]* AC0

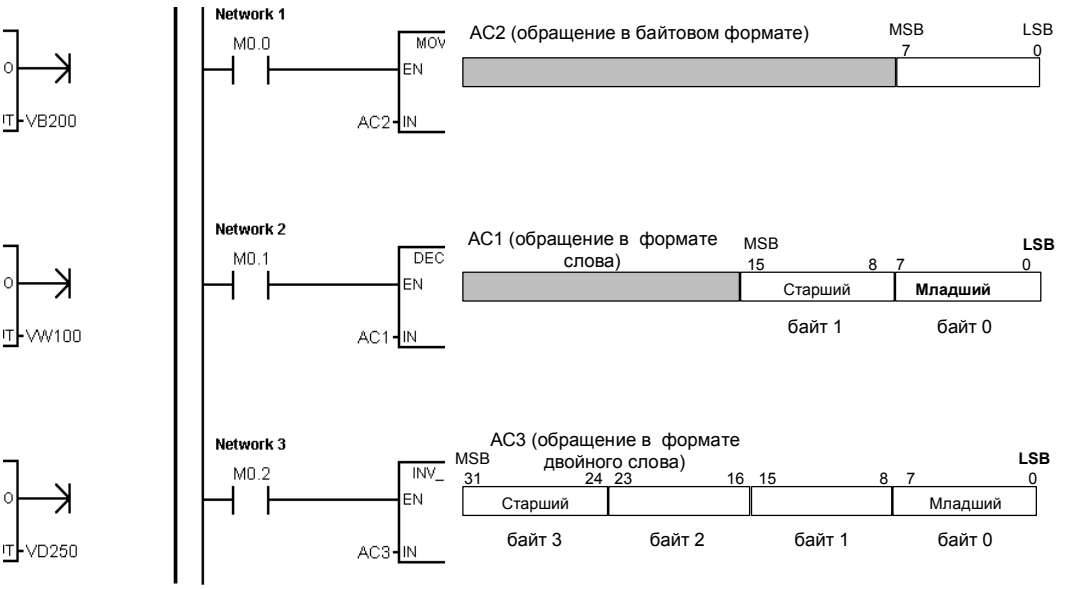


Рис. 4–7. Обращение к аккумуляторам

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*[адрес байта].[адрес бита]*	SM0.1
Байт, слово или двойное слово:	SM*[длина][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.

Совет
При программировании в LAD или FBD последние четыре байта зарезервированы для STEP 7-Micro/WIN. Если вы программируете на STL, то все 64 байта локальной памяти имеются в вашем распоряжении, но мы рекомендуем вам и в этом случае не использовать эти последние четыре байта.

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 подпрограмм 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*[адрес байта].[адрес бита]*	L0.0
Байт, слово или двойное слово:	L*[длина] [начальный адрес байта]*	LB33

Аналоговые входы: AI

S7–200 преобразует аналоговые величины (например, температуру или напряжение) в цифровые величины, имеющие длину слова (16 битов). Вы обращаетесь к этим значениям 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 вы обращаетесь к этим значениям с помощью адресов четных байтов (например, AIW0, AIW2, AIW4). Аналоговые входы можно только считывать.

Формат:	AIW*[начальный адрес байта]*	AIW4

Аналоговые выходы: AQ

S7–200 преобразует цифровые величины, имеющие длину слова (16 битов), в ток или напряжение пропорционально цифровой величине. Вы обращаетесь к этим значениям 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 вы записываете эти значения с адресами четных байтов (например, AQW0, AQW2, AQW4). Аналоговые выходы можно только записывать.

Формат:	AQW*[начальный адрес байта]*	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*[адрес байта].[адрес бита]*	S3.1
Байт, слово или двойное слово:	S*[длина][начальный адрес байта]*	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.

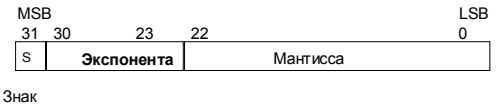


Рис. 4–8. Формат вещественного числа

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.

Длина	Символ 1	Символ 2	Символ 3	Символ 4	...	Символ 254
Байт 0	Байт 1	Байт 2	Байт 3	Байт 4		Байт 254

Рис. 4–9. Формат строк

Задание констант 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]’	Текст в одинарных кавычках.’
Вещественное	ANSI/IEEE 754-1985	+1.175495E-38 (положительное) –1.175495E-38 (отрицательное)

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 модулей расширения 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.

Входы и выходы аналоговых модулей расширения всегда назначаются парами. 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 224)

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, T (только текущее значение) и C (только текущее значение). Косвенную адресацию нельзя использовать для обращения к отдельному биту или к областям памяти AI, AQ, HC, SM 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.

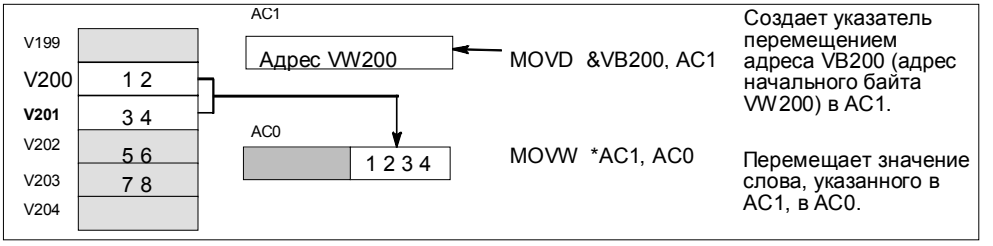


Рис. 4–11. Создание и использование указателя

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.

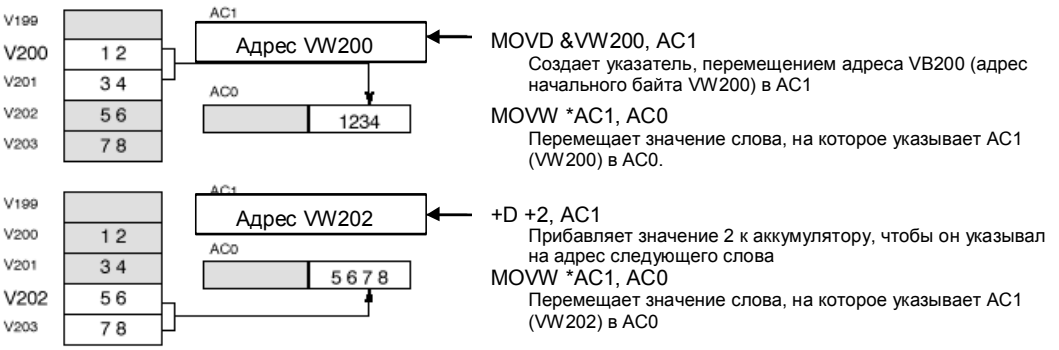
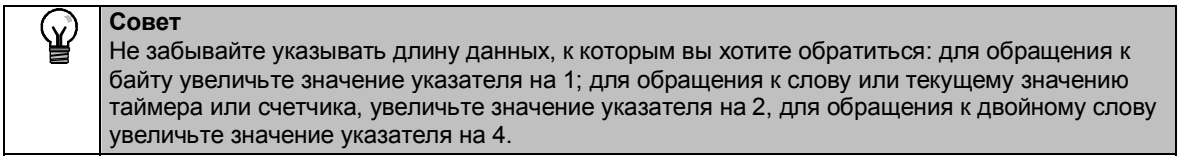


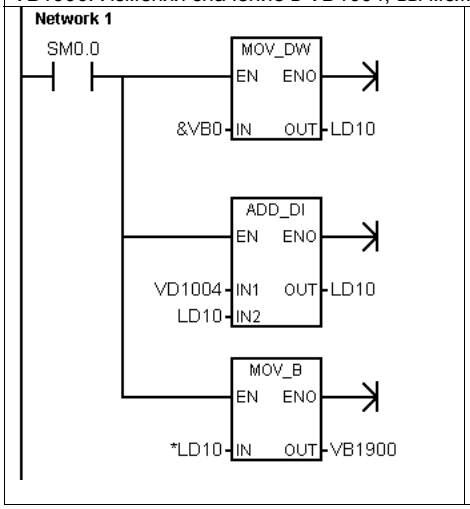
Рис. 4–12. Изменение указателя



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

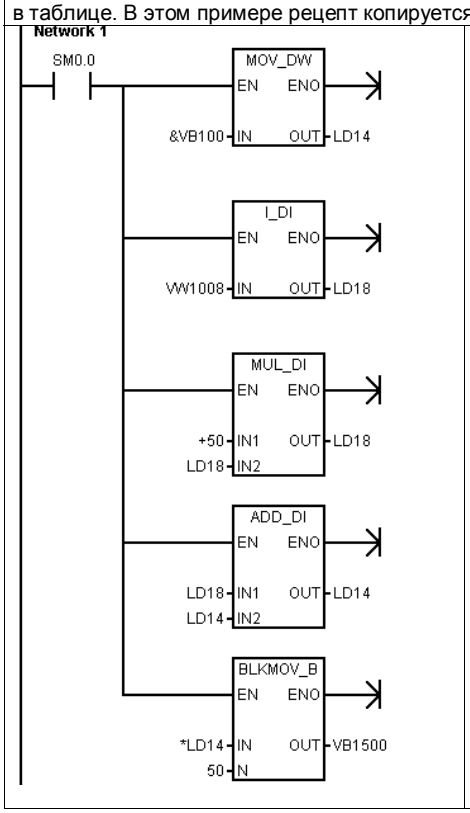
В этом примере используется LD10 как указатель на адрес VB0. Затем вы увеличиваете указатель на величину смещения, хранящуюся в VD1004. Теперь LD10 указывает на другой адрес в памяти переменных (VB0 + смещение). Значение, хранящееся в памяти переменных по адресу, на который указывает LD10, копируется 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
MOVD   &VB0, LD10
+D     VD1004, LD10
MOVB   *LD10, VB1900
```

Пример программы для обращения к данным в таблице с использованием смещения

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
MOVD   &VB100, LD14
ITD    VW1008, LD18
*D     +50, LD18
+D     LD18, LD14
BMB    *LD14, VB1500, 50
```

Как в S7–200 производится сохранение и извлечение данных

S7–200 предоставляет несколько методов, гарантирующих, что ваша программа, данные программы и конфигурационные данные вашего S7–200 будут сохраняться надлежащим образом.

S7–200 предоставляет в распоряжение конденсатор большой емкости, который поддерживает целостность ОЗУ после отключения питания. В зависимости от исполнения S7–200, этот конденсатор может поддерживать ОЗУ в течение нескольких дней.

S7–200 предоставляет в распоряжение ЭСППЗУ для постоянного, устойчивого к исчезновению напряжения, хранения всей вашей программы, выбранных пользователем областей данных и конфигурационных данных.

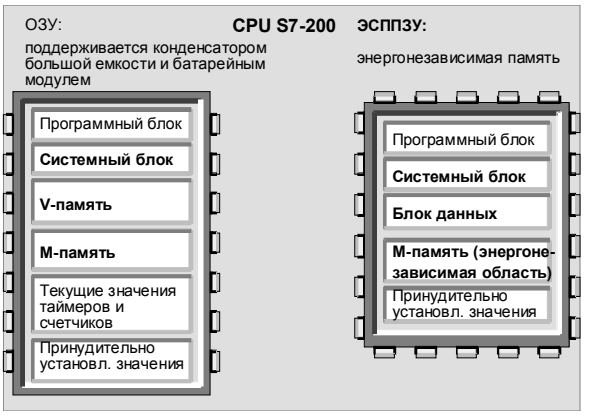


Рис. 4–13. Области памяти CPU S7-200

S7–200 поддерживает необязательный батарейный модуль, который расширяет промежуток времени, в течение которого ОЗУ может сохранять свое содержимое после отключения питания CPU. Батарейный модуль берет на себя буферизацию только после того, как конденсатор разрядился.

Загрузка компонентов вашего проекта в CPU и из CPU

Ваш проект состоит из трех компонентов: программного блока, блока данных (не обязателен) и системного блока (не обязателен).

На рис. 4–14 показано, как проект загружается в S7–200.

При загрузке проекта его компоненты сохраняются в ОЗУ. S7–200 также автоматически копирует программу пользователя, блок данных и системный блок в ЭСППЗУ для энергонезависимого хранения.

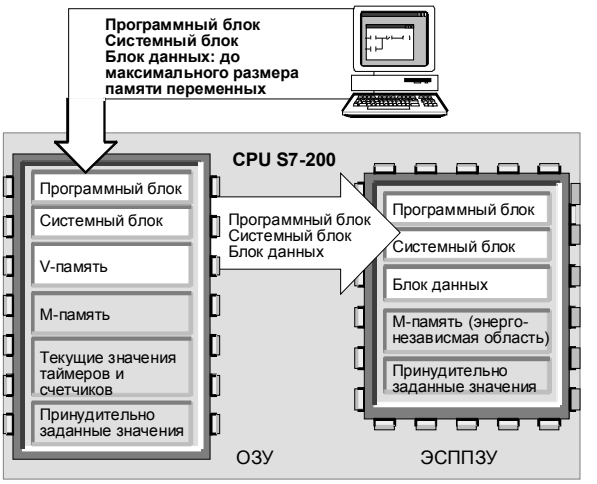


Рис. 4–14. Загрузка проекта в S7–200

На рис. 4–15 показано, как проект загружается из S7–200.

Когда вы загружаете проект в свой компьютер, S7–200 загружает системный блок из ОЗУ, а программный блок и блок данных из ЭСППЗУ.

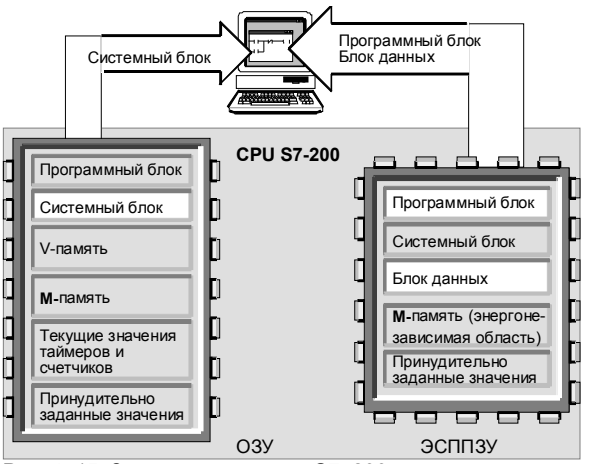


Рис. 4–15. Загрузка проекта из S7–200

Сохранение данных из битовой памяти (M) при потере питания

Если первые 14 байтов битовой памяти (от MB0 до MB13) были определены при конфигурировании как реманентные (сохраняемые), то они сохраняются в ЭСППЗУ, когда S7–200 теряет питание.

Как показано на рис. 4–16, S7–200 передает эти сохраняемые области из M-памяти в ЭСППЗУ.

По умолчанию первые 14 байтов битовой памяти устанавливаются как не сохраняемые. Эта установка деактивизирует процесс сохранения, который обычно происходит при выключении S7–200.

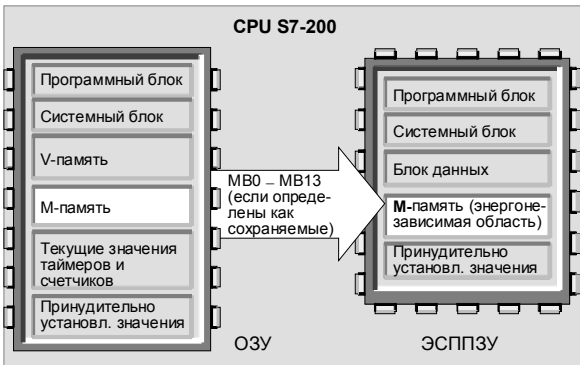


Рис. 4–16. Сохранение битовой памяти при потере питания

Восстановление данных после запуска

При запуске S7–200 восстанавливает программный блок и системный блок из ЭСППЗУ, как показано на рис. 4–17. При запуске S7–200 проверяет ОЗУ, чтобы убедиться, что конденсатор большой емкости успешно выполнил свою задачу по поддержанию данных, хранящихся в ОЗУ. Если содержимое ОЗУ было успешно сохранено, то сохраняемые области ОЗУ остаются неизменными.

Сохраняемые и несохраняемые области памяти переменных восстанавливаются из соответствующего блока данных в ЭСППЗУ. Если содержимое ОЗУ не было сохранено (например, после длительного перерыва в питании), S7–200 очищает ОЗУ (включая сохраняемую и несохраняемую области) и устанавливает бит потери сохраняемых данных (SM0.2) для первого цикла обработки программы, следующего за включением питания, а затем копирует в ОЗУ данные, сохраненные в ЭСППЗУ.

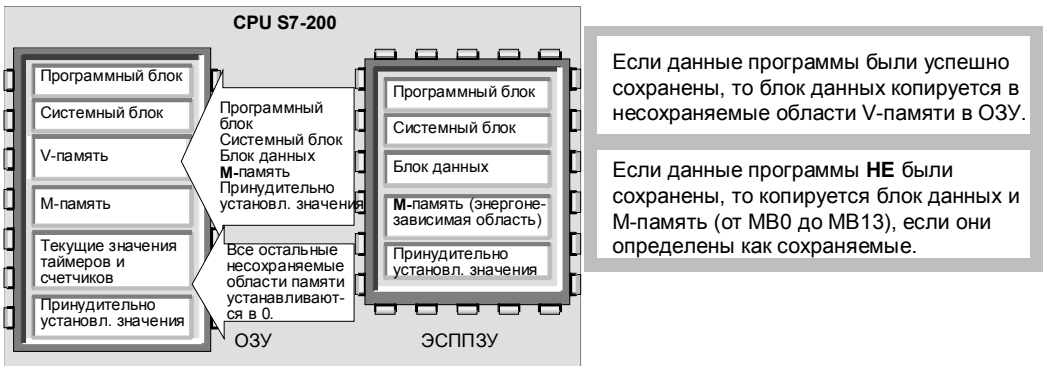


Рис. 4–17. Восстановление данных после запуска

Сохранение программы в модуле памяти

S7–200 поддерживает дополнительный модуль памяти, который предоставляет в распоряжение вашей программе сменное ЭСППЗУ. S7–200 хранит в модуле памяти следующие элементы: программный блок, блок данных, системный блок и принудительно устанавливаемые значения.

Вы можете скопировать свою программу в модуль памяти из ОЗУ только тогда, когда S7–200 получает питание и находится в состоянии STOP, а модуль памяти вставлен. Модуль памяти можно вставлять и вынимать, когда S7–200 включен.

Осторожно
Электростатический разряд может повредить модуль памяти или гнездо на CPU S7–200.

При работе с модулем вы должны находиться на заземленном проводящем полу и/или носить на руке заземленный браслет. Храните модуль в токопроводящем контейнере.

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

Копирование вашей программы в модуль памяти

После того как модуль памяти вставлен, используйте для копирования программы следующую процедуру:

1. Переведите CPU S7–200 в состояние STOP.
2. Загрузите программу в S7–200, если она еще не была загружена.
3. Выберите команду меню **PLC → Program Memory Cartridge [ПЛК → Запрограммировать модуль памяти]**, чтобы скопировать программу в модуль памяти. На рис. 4–18 показаны элементы памяти CPU, которые сохраняются в модуле памяти.

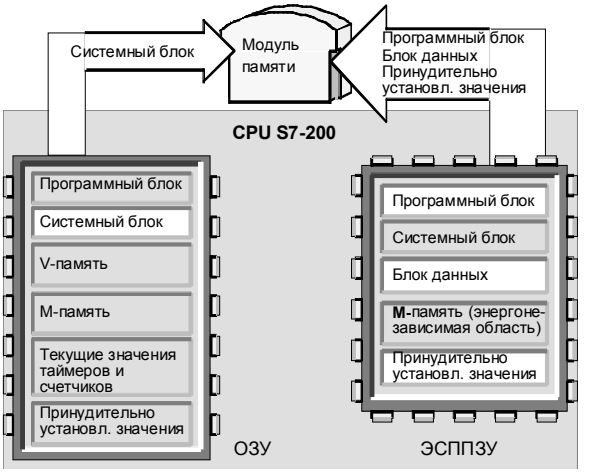


Рис. 4–18. Копирование в модуль памяти

4. Не обязательно: Вытащите модуль памяти и поставьте на место крышку S7–200.

Восстановление программы из модуля памяти

Для передачи программы из модуля памяти в S7–200 вы должны выключить, а затем снова включить питание S7–200 при вставленном модуле памяти.

Примечание
Включение CPU S7–200 с пустым модулем памяти или с 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.

Вытащите модуль памяти и снова включите питание S7–200. После этого модуль памяти может быть снова вставлен и перепрограммирован, если необходимо.

Как показано на рис. 4–19, S7–200 выполняет следующие задачи, после того как вы выключили и снова включили питание при вставленном модуле памяти:

1. Если содержимое модуля памяти отличается от содержимого ЭСППЗУ, то S7–200 очищает ОЗУ.
2. S7–200 копирует содержимое модуля памяти в ОЗУ.
3. S7–200 копирует программный блок, системный блок и блок данных в ЭСППЗУ.

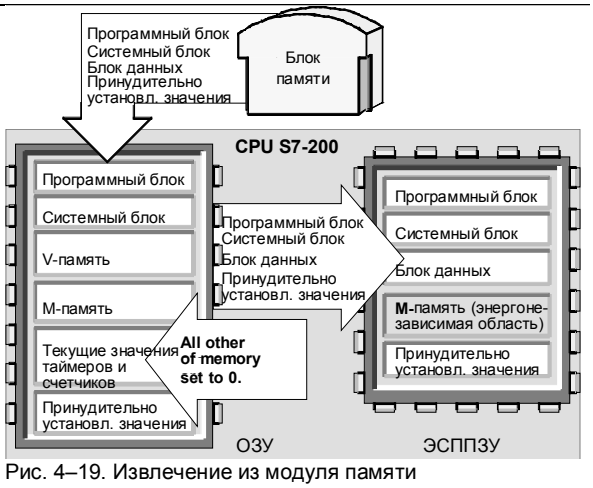


Рис. 4–19. Извлечение из модуля памяти

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.

Использование вашей программы для сохранения памяти переменных в ЭСППЗУ

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

Операция сохранения в ЭСППЗУ не обновляет данные в модуле памяти.

Совет
Так как число операций сохранения в ЭСППЗУ ограничено (минимум 100 000, обычно 1 000 000), вы должны обеспечить, чтобы сохранялись только необходимые значения. В противном случае ЭСППЗУ может износиться, а 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 ЭСППЗУ

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. 4–20 показан формат SMB31 и SMW32.

Выполните следующие шаги, чтобы запрограммировать S7–200 на сохранение или запись конкретного значения в V-памяти:

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. 4–20.
3. Установите SM31.7 в 1.

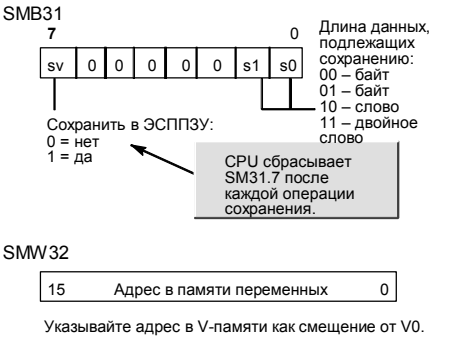


Рис. 4–20. SMB31 и SMW32

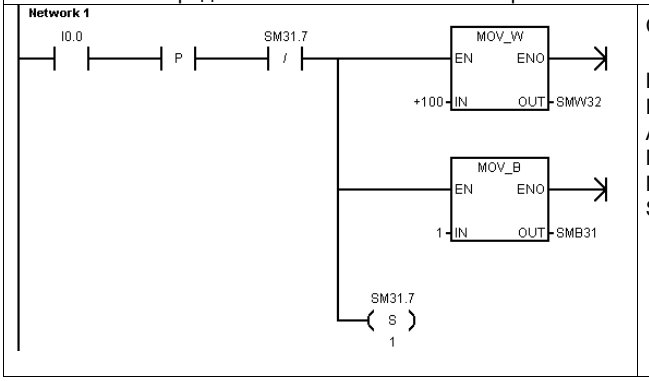
В конце каждого цикла выполнения программы S7–200 проверяет SM31.7; если SM31.7 равен 1, то указанное значение сохраняется в ЭСППЗУ. Операция завершается, когда S7–200 сбрасывает SM31.7 в 0.

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

Пример программы: Копирование V-памяти в ЭСППЗУ

Этот пример передает VB100 в ЭСППЗУ. 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.




```
Сегмент 1     //Передать ячейку памяти
              // переменных (VB100) в ЭСППЗУ
LD      I0.0
EU
AN      SM31.7
MOVW    +100, SMW32
MOVB    1, SMB31
S       SM31.7, 1
```

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 во входном регистре 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, сохраняются 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 обработки коммуникационных задач

При проектировании вы можете установить долю времени цикла (в процентах), которая планируется для обработки коммуникационных заданий, которые связаны с процессами компиляции в режиме RUN или с состоянием выполнения. (Редактирование в режиме RUN и состояние исполнения – это возможности, предоставляемые пакетом STEP 7-Micro/WIN для облегчения отладки вашей программы.) Увеличение доли времени, отводимой на коммуникационные задачи, увеличивает время цикла, вследствие чего ваш процесс управления осуществляется медленнее.

По умолчанию доля времени цикла, отводимая на коммуникационные задания, 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 щелкните на закладке Background Time [Фоновое время].
2. Измените свойства для фонового времени обмена данными и щелкните на OK.
3. Загрузите измененный системный блок в S7–200.

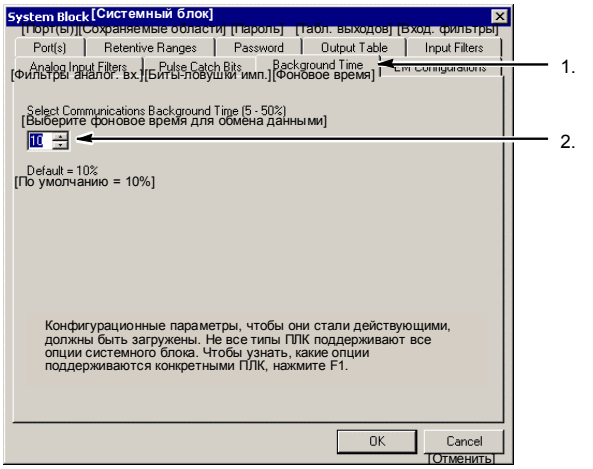


Рис. 4–21. Фоновое время для обмена данными

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, которая загружается и сохраняется в S7–200 и действительна только для цифровых выходов.

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и щелкните на закладке Output Table [Таблица выходов].
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 OK.
5. Загрузите измененный системный блок в S7–200.

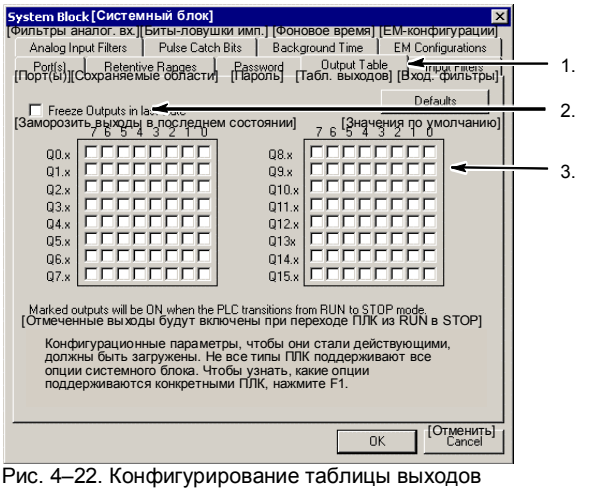


Рис. 4–22. Конфигурирование таблицы выходов

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, C 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 MB0 до MB13 в качестве сохраняемого, то активизируется специальная функция, которая при потере питания автоматически сохраняет эти ячейки памяти в ЭСППЗУ.

Для определения сохраняемой памяти:

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

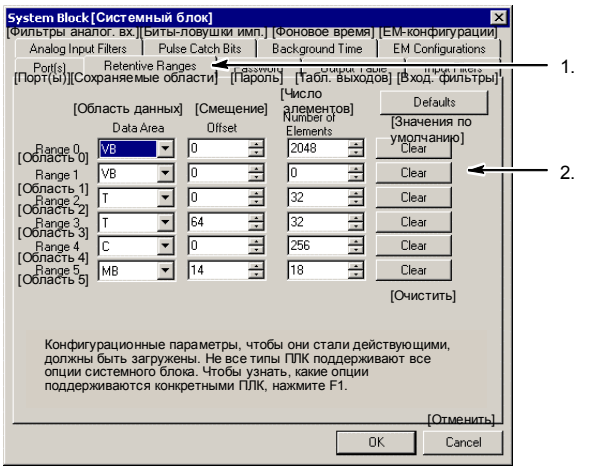


Рис. 4–23. Сохраняемая память

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, которая загружается и хранится в S7–200. По умолчанию время фильтра равно 6,4 мс. Как показано на рис. 4–24, каждая заданная 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 щелкните на закладке Input Filters [Входные фильтры].
2. Введите величину задержки для каждой группы входов и щелкните на OK .
3. Загрузите измененный системный блок в S7–200.

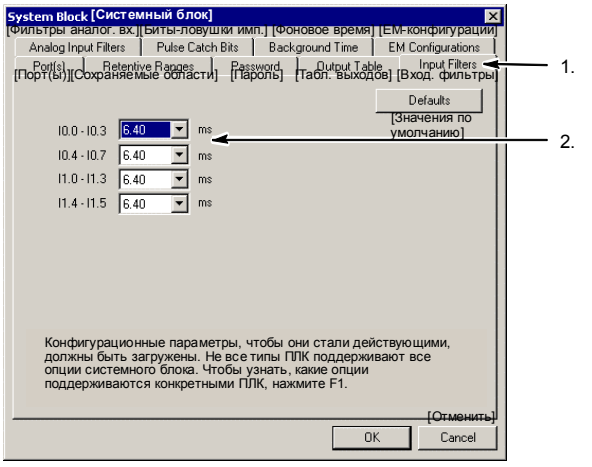


Рис. 4–24. Конфигурирование входных фильтров

Совет
Фильтр цифровых входов оказывает также влияние на входную величину в командах чтения, входных прерываний и ловушек 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 превышает определенную величину. Это изменение, называемое зоной нечувствительности, задается в отсчетах цифрового значения аналогового входа.

По умолчанию фильтр активизирован для всех аналоговых входов.

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** и щелкните на закладке Analog Input Filters [Фильтры аналоговых входов].
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.

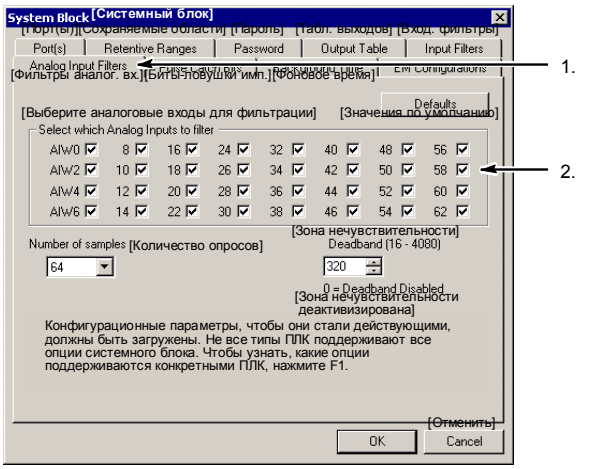
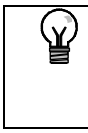


Рис. 4–25. Фильтр аналоговых входов



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 главного модуля AS–интерфейса.

S7–200 дает возможность регистрировать короткие импульсы

S7–200 имеет в своем распоряжении функцию "Захват импульса", которая может быть использована для всех или некоторых встроенных цифровых входов. Функция " Захват импульса" 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 " Захват импульса" 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 " Захват импульса" можно активизировать для любого встроенного цифрового входа.

Чтобы вызвать диалоговое окно для конфигурирования захвата импульса:

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

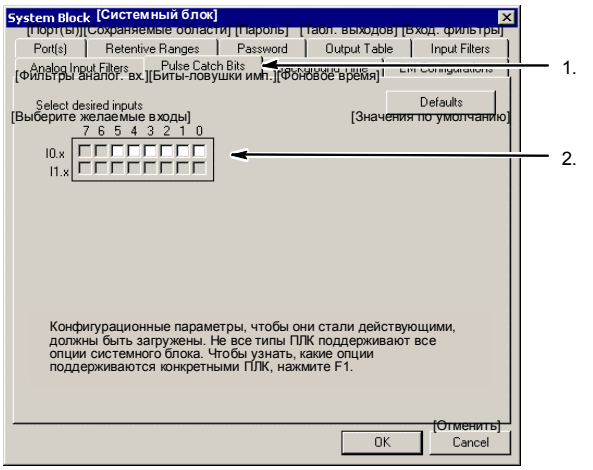


Рис. 4–26. Захват импульсов

На рис. 4–27 показано, как действует S7–200 с активизированным и деактивизированным захватом импульсов.

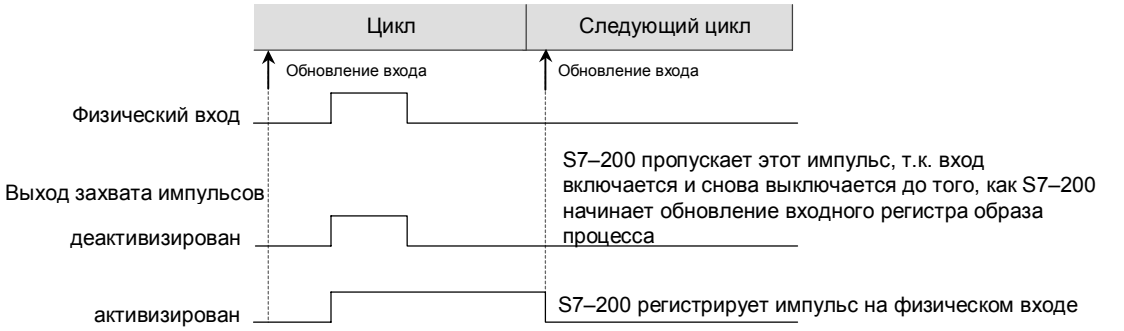


Рис. 4–27. Функционирование S7–200 с активизированным и деактивизированным захватом импульсов

Так как функция захвата 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. 4–28 дано схематическое представление цепи цифрового входа.

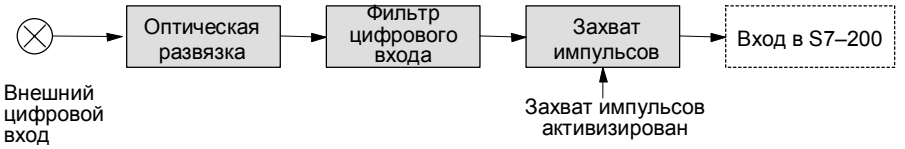


Рис. 4–28. Цепь цифрового входа

На рис. 4–29 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 6–44.)

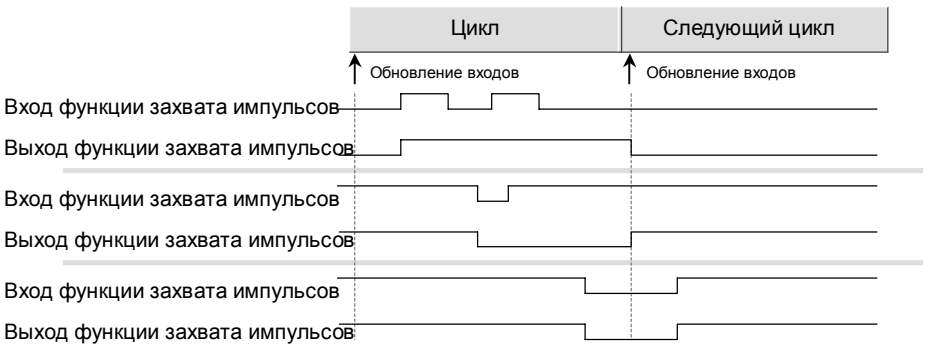


Рис. 4–29. Реакции функции захвата импульсов на различные входные условия

S7–200 предоставляет защиту с помощью пароля

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

Благодаря установке пароля только уполномоченные лица имеют доступ к определенным функциям и памяти: 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. По умолчанию для S7–200 установлен уровень 1 (без ограничений).

Ввод пароля через сеть не оказывает влияния на парольную защиту S7–200.

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			

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. 4–30) позволяет установить пароль для S7–200:

1. Выберите команду меню **View > Component > System Block [Вид > Компонент > Системный блок]** для отображения диалогового окна System Block [Системный блок] и щелкните на закладке Password [Пароль].
2. Выберите желаемый уровень защиты для S7–200.
3. Введите пароль и подтвердите его.
4. Щелкните на OK .
5. Загрузите измененный системный блок в S7–200.

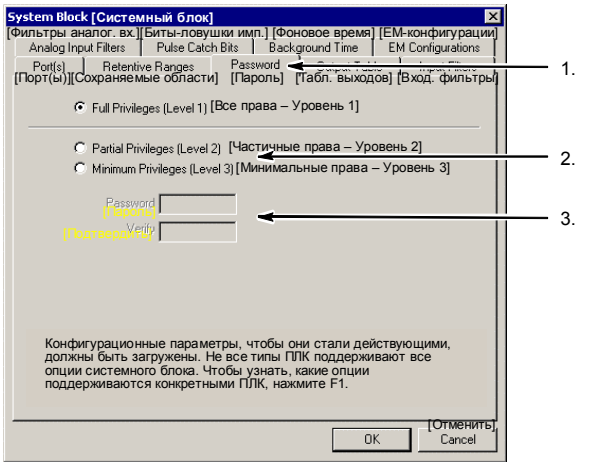


Рис. 4–30. Создание пароля

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 адреса, скорости передачи и часов реального времени. Для стирания программы в 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 OK.
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.

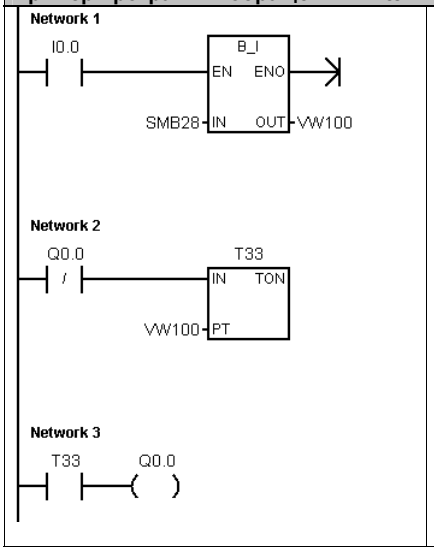
Всегда соблюдайте соответствующие меры безопасности и перед очисткой памяти S7–200 обеспечьте, что ваш процесс находится в безопасном состоянии.

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




```
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
```

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 A/B-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 (PTO) 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 и заданного времени цикла. Количество импульсов может быть задано в диапазоне от 1 до 4 294 967 295. Время цикла может быть задано в микро- или миллисекундах в пределах от 50 мкс до 65 535 мкс или от 2 мс до 65 535 мс. Функция «Последовательность скоростных импульсов» (PTO) 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. Конфигурация импульсов может включать в себя до 255 сегментов с участками, соответствующими разгону, работе или замедлению.

Функция «Широтно-импульсная модуляция» обеспечивает фиксированное время цикла с переменной относительной длительностью импульсов. Время цикла и ширина импульсов могут быть заданы в микро- или миллисекундах. Время цикла имеет диапазон от 50 мкс до 65 535 мкс или от 2 мс до 65 535 мс. Ширина импульсов имеет диапазон от 0 мкс до 65 535 мкс или от 0 мс до 65 535 мс. 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.

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 48
Основные элементы программы 49
Использование STEP 7-Micro/WIN для создания вашей программы 51
Наборы команд SIMATIC и IEC 1131-3 53
Соглашения, используемые в редакторах программ 54
Создание программы управления с помощью мастеров 56
Устранение ошибок в S7–200 56
Назначение адресов и начальных значений в редакторе блоков данных 58
Использование таблицы символов для символической адресации переменных 58
Использование локальных переменных 59
Контроль над программой с помощью таблицы состояний 59
Создание библиотеки команд 60
Функции тестирования программы 60

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 ваш 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/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)

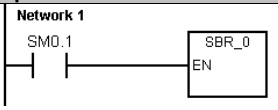
Создание списка символических имен (не обязателен)

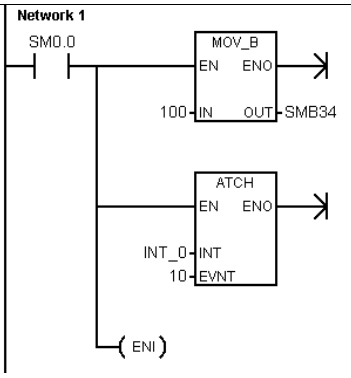
Если вы выбираете для адресации использование символических имен, то составьте список символических имен для абсолютных адресов. Включите не только сигналы физических входов-выходов, но также и другие элементы, которые нужно использовать в вашей программе.

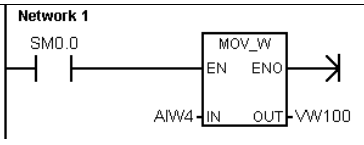
Основные элементы программы

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		Network 1 //В 1-ом цикле вызвать подпрограмму 0. LD SM0.1 CALL SBR_0

S B R 0		Network 1 //Установить для прерывания,
		//управляемого временем,
		// интервал 100 мс.
		//Разблокировать прерывание 0.
		LD SM0.0
		MOVB 100, SMB34
		ATCH INT_0, 10
		ENI

INT 0		Network 1 //Опросить аналоговый вход 4. LD SM0.0 MOVW AIW4,VW100

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

Системный блок

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

Блок данных

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 3.2 [Пуск > SIMATIC > STEP 7 MicroWIN 3.2]**. 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 перетаскивать 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.

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.

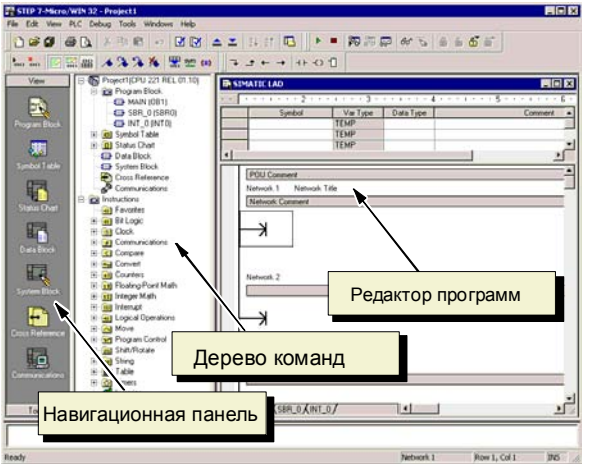


Рис. 5–1. STEP 7-Micro/WIN

Редактор программ

STEP 7-Micro/WIN имеет в своем распоряжении три редактора для создания программ: цепная логическая схема (LAD), , называемая также контактным планом (KOP), 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.

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

```
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

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 исполняется 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.

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

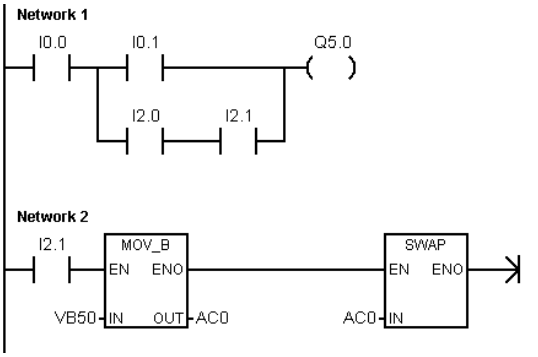


Рис. 5–3. Пример программы в виде цепной логической схемы

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.

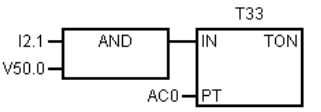


Рис. 5–4. Пример программы в виде функционального плана

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 конфигурация 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: при отсутствии 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 собой 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.

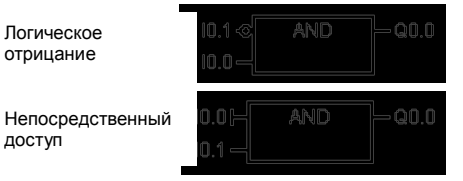


Рис. 5–5. Соглашения для FBD

- 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 действительны только для физических входов.
- Блок без входов или выходов: Блок без входа обозначает команду, не зависимую 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 EN/ENO и их типы данных 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 таблица 5–2.

Таблица 5–2. Представление условного и безусловного входов

Поток сигнала	LAD	FBD
Команда, зависящая от потока сигнала (условная)	1 —(JMP)	JMP
Команда, не зависящая от потока сигнала (безусловная)	├—(NEXT)	NEXT

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 также 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.

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:

Мастер команд

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, сгенерированный 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.

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

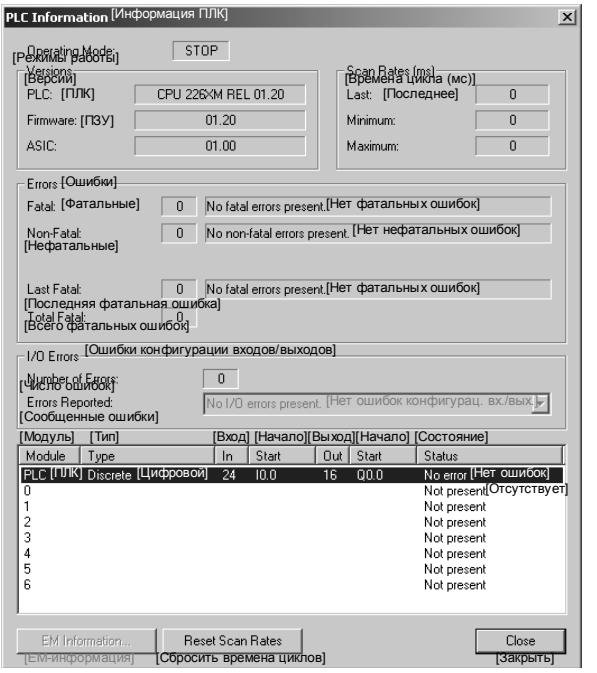


Рис. 5–6. Диалоговое окно с информацией ПЛК

Нефатальные ошибки

В случае нефатальных ошибок речь идет об ошибках в построении программы пользователя, об ошибке при исполнении 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 ЭСППЗУ и не потеряется.) После исправления своей программы вы можете загрузить ее снова. Список нарушений правил компиляции вы найдете в Приложении C.

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.

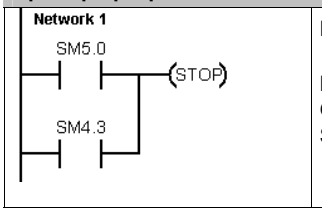
Информация о состоянии модуля хранится в битах специальной памяти (SM). Ваша программа может контролировать и анализировать эти биты. Подробную информацию о битах специальной памяти, используемых для индикации ошибок конфигурации входов/выходов вы найдете в Приложении D. Бит SM5.0 является глобальным 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 C). 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




```
Network 1   //При возникновении ошибки конфигурации входов/выходов
            //или ошибки выполнения программы перейти в STOP
LD      SM5.0
O       SM4.3
STOP
```

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 светодиод System Fault [Ошибка системы] и светодиод 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:

- Выключение и затем включение питания.
- Перевод переключателя 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 неспособным к обмену данными. 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

Блок 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. Редактор блоков данных выделяет необходимое место в памяти переменных в соответствии со сделанными вами ранее назначениями адресов и размером (байт, слово или двойное слово) данных.

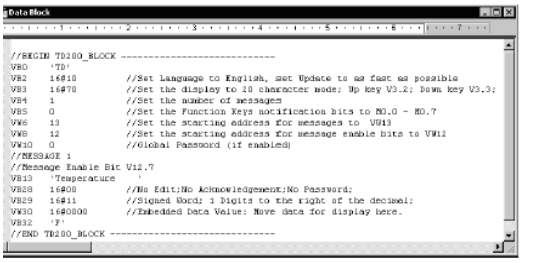


Рис. 5–7. Редактор блоков данных

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

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.

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.

		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

Для присвоения адресу символического имени:

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.

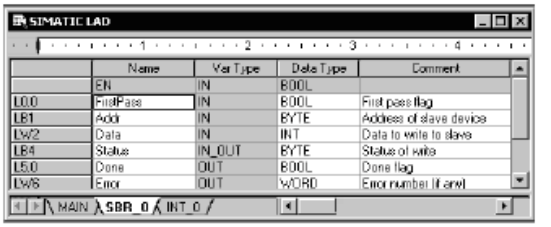


Рис. 5–9. Таблица локальных переменных

Контроль над программой с помощью таблицы состояний

Таблица 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.

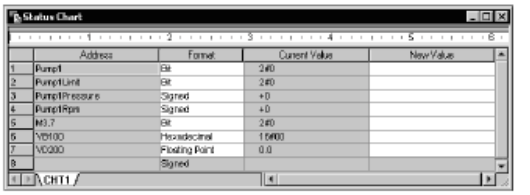


Рис. 5–10. Таблица состояний

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 перемененные на логические группы, чтобы каждую группу можно было наблюдать в собственно 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 и программы обработки прерываний.
2. Обеспечьте, чтобы все адреса в памяти переменных в подпрограммах и 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.

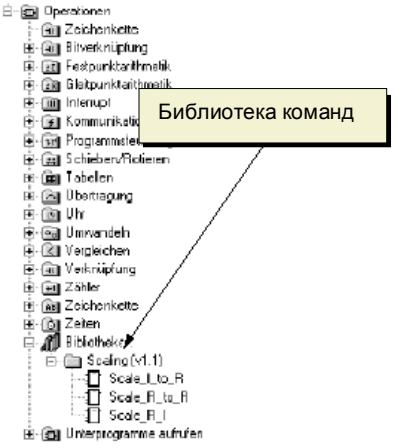


Рис. 5–11. Дерево команд с библиотеками

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 63
Области памяти и функции S7–200 64
Битовые логические операции 66
- Контакты 66
- Катушки 68
- Стековые операции 70
- Функциональный блок с двумя устойчивыми состояниями: преимущество установки и преимущество сброса 72

Команды, связанные со временем 73
Команды для обмена данными 74
- Команды чтения из сети и записи через сеть 74
- Команды передачи и приема (свободно программируемый обмен данными) 79
- Команды получения и установки адреса порта 88

Команды сравнения 89
- Сравнение числовых величин 89
- Сравнение строк 91

Команды преобразования 92
- Стандартные команды преобразования 92
- Команды преобразования ASCII 96
- Команды преобразования строк 100
- Команды кодирования и декодирования 105

Команды счета 106
- Счетчик SIMATIC 106
- Счетчик IEC 109

Скоростные счетчики 111
Команда вывода импульсов 125
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 155
Логические операции 162
- Операции инвертирования 162
- Поразрядные логические операции И, ИЛИ и исключающее ИЛИ 163

Команды пересылки 165
- Пересылка байта, слова, двойного слова или вещественного числа 165
- Непосредственное чтение или запись и пересылка байта 166
- Команды групповой пересылки 167

Команды управления программой 168
- Условное завершение 168
- Останов 168

Сброс времени контроля 168
Команды формирования программного цикла For-Next 170
Команды перехода 172
Команды для реле управления очередностью (SCR) 173
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
Операции с подпрограммами 203

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.

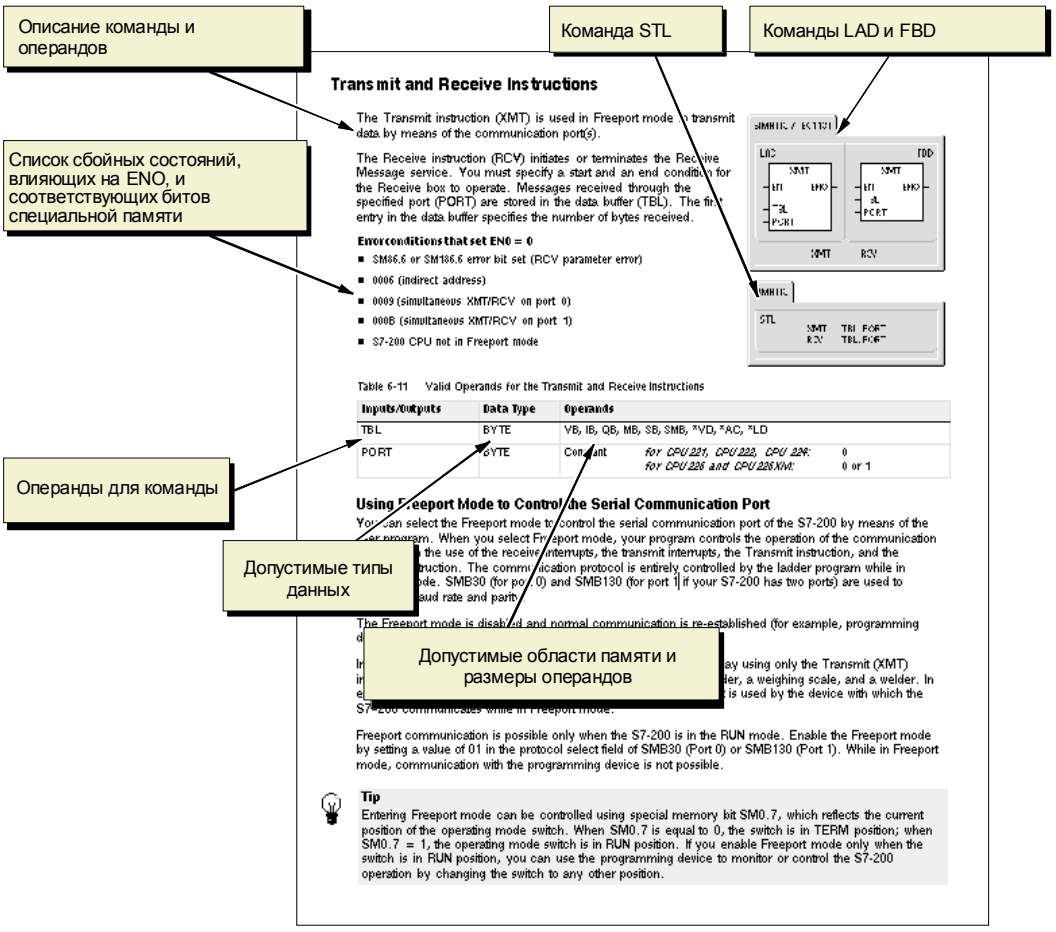


Рис. 6–1. Описания команд

Области памяти и функции S7–200

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

Описание	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Размер программы пользователя	2 К слов	2 К слов	4 К слов	4 К слов	8 К слов
Размер данных пользователя	1 К слов	1 К слов	2.5 К слов	2.5 К слов	5 К слов
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 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	--	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB5119	VB0 – VB5119	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)	SM0.0 – SM179.7	SM0.0 – SM299.7	SM0.0 – SM549.7	SM0.0 – SM549.7	SM0.0 – SM549.7
(только чтение)	SM0.0 – SM29.7	SM0.0 – SM29.7	SM0.0 – SM29.7	SM0.0 – SM29.7	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	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, HC3, HC4 и HC5	HC0, HC3, HC4 и 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	От 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
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 226	CPU 226XM
Доступ к биту (байт.бит)	I	от 0.0 до 15.7	от 0.0 до 15.7	от 0.0 до 15.7	от 0.0 до 15.7
	Q	от 0.0 до 15.7	от 0.0 до 15.7	от 0.0 до 15.7	от 0.0 до 15.7
	V	от 0.0 до 2047.7	от 0.0 до 2047.7	от 0.0 до 5119.7	от 0.0 до 10239.7
	M	от 0.0 до 31.7	от 0.0 до 31.7	от 0.0 до 31.7	от 0.0 до 31.7
	SM	от 0.0 до 179.7	от 0.0 до 299.7	от 0.0 до 549.7	от 0.0 до 549.7
	S	от 0.0 до 31.7	от 0.0 до 31.7	от 0.0 до 31.7	от 0.0 до 31.7
	T	от 0 до 255	от 0 до 255	от 0 до 255	от 0 до 255
	C	от 0 до 255	от 0 до 255	от 0 до 255	от 0 до 255
	L	от 0.0 до 59.7	от 0.0 до 59.7	от 0.0 до 59.7	от 0.0 до 59.7
Доступ к байту	IB	от 0 до 15	от 0 до 15	от 0 до 15	от 0 до 15
	QB	от 0 до 15	от 0 до 15	от 0 до 15	от 0 до 15
	VB	от 0 до 2047	от 0 до 2047	от 0 до 5119	от 0 до 10239
	MB	от 0 до 31	от 0 до 31	от 0 до 31	от 0 до 31
	SMB	от 0 до 179	от 0 до 299	от 0 до 549	от 0 до 549
	SB	от 0 до 31	от 0 до 31	от 0 до 31	от 0 до 31
	L	от 0 до 63	от 0 до 63	от 0 до 63	от 0 до 255
	AC	от 0 до 3	от 0 до 3	от 0 до 3	от 0 до 255
Доступ к слову	IW	от 0 до 14	от 0 до 14	от 0 до 14	от 0 до 14
	QW	от 0 до 14	от 0 до 14	от 0 до 14	от 0 до 14
	VW	от 0 до 2046	от 0 до 2046	от 0 до 5118	от 0 до 10238
	MW	от 0 до 30	от 0 до 30	от 0 до 30	от 0 до 30
	SMW	от 0 до 178	от 0 до 298	от 0 до 548	от 0 до 548
	SW	от 0 до 30	от 0 до 30	от 0 до 30	от 0 до 30
	T	от 0 до 255	от 0 до 255	от 0 до 255	от 0 до 255
	C	от 0 до 255	от 0 до 255	от 0 до 255	от 0 до 255
	LW	от 0 до 58	от 0 до 58	от 0 до 58	от 0 до 58
	AC	от 0 до 3	от 0 до 3	от 0 до 3	от 0 до 3
	AIW	*отсутствует*	от 0 до 30	от 0 до 62	от 0 до 62
	AQW	*отсутствует*	от 0 до 30	от 0 до 62	от 0 до 62
Доступ к двойному слову	ID	от 0 до 12	от 0 до 12	от 0 до 12	от 0 до 12
	QD	от 0 до 12	от 0 до 12	от 0 до 12	от 0 до 12
	VD	от 0 до 2044	от 0 до 2044	от 0 до 5116	от 0 до 10236
	MD	от 0 до 28	от 0 до 28	от 0 до 28	от 0 до 28
	SMD	от 0 до 176	от 0 до 296	от 0 до 546	от 0 до 546
	SD	от 0 до 28	от 0 до 28	от 0 до 28	от 0 до 28
	LD	от 0 до 56	от 0 до 56	от 0 до 56	от 0 до 56
	AC	от 0 до 3	от 0 до 3	от 0 до 3	от 0 до 3
	HC	0, 3, 4, 5	0, 3, 4, 5	от 0 до 5	от 0 до 5

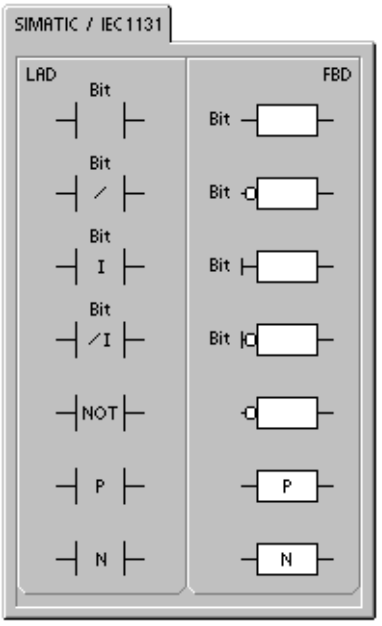
Битовые логические операции

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).

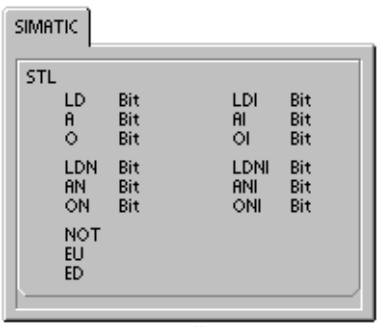
Нормально открытый контакт замкнут (включен), когда бит равен 1, а нормально замкнутый контакт замкнут (включен), когда бит равен 0. В FBD к блокам И и ИЛИ может быть подключено не более 32 входов. В STL команды, представляющие нормально открытый контакт, загружают значение адресного бита в вершину стека 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 [НЕ]

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.

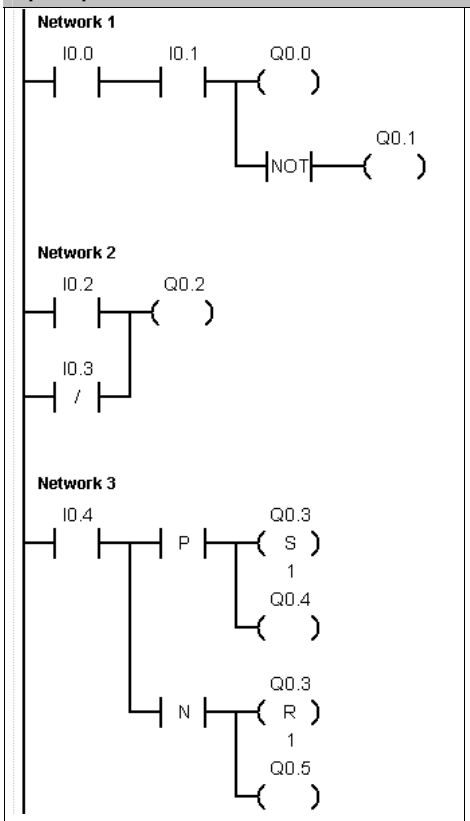
При редактировании программы в режиме RUN необходимо ввести параметр для команд обнаружения положительного и отрицательного фронта. Подробную информацию о редактировании в режиме RUN вы найдете в главе 5.

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

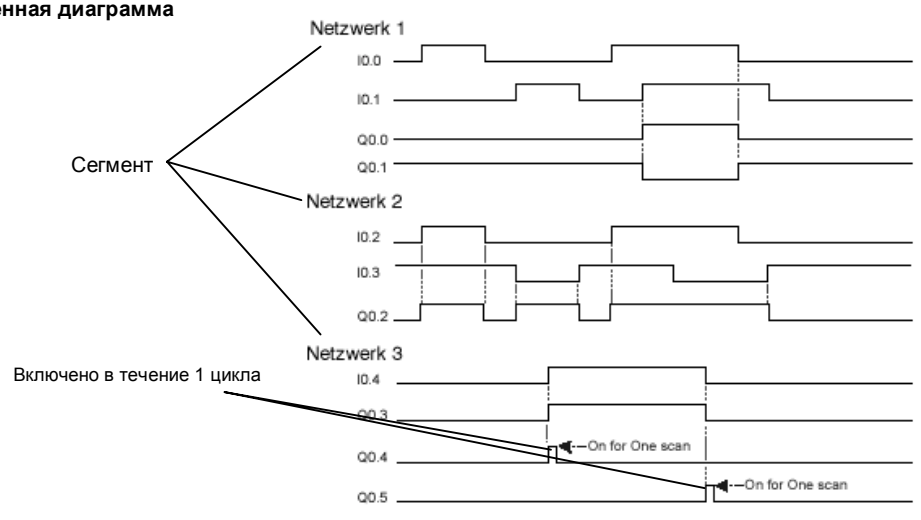
Совет
Так как команды обнаружения положительного и отрицательного фронта требуют переключения, соответственно, с 0 на 1 или с 1 на 0, то нет возможности обнаружить нарастающий или падающий фронт в первом цикле. В первом цикле состояние указанного в этих командах бита устанавливает S7–200. В следующих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

Пример: Контакты




```
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.
LD    I0.0
A     I0.1
=     Q0.0
NOT
=     Q0.1
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".
LD    I0.2
ON    I0.3
=     Q0.2
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.
LD    I0.4
LPS
EU
S     Q0.3, 1
=     Q0.4
LPP
ED
R     Q0.3, 1
=     Q0.5
```

Временная диаграмма



Катушки

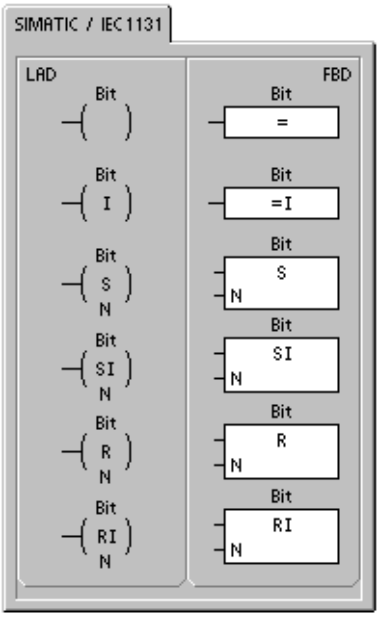
Присваивание

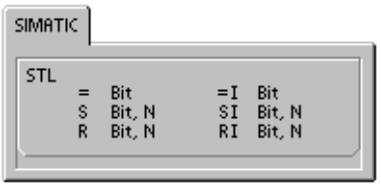
Команда присваивания (=) записывает новое значение для выходного бита в регистр образа процесса. При выполнении команды присваивания S7–200 устанавливает или сбрасывает выходной бит в регистре образа процесса. В LAD и FBD указанный бит устанавливается равным потоку сигнала. В STL значение, находящееся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

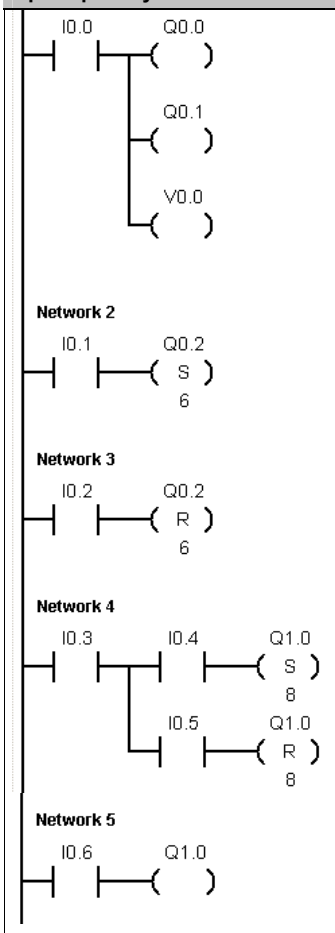
Сбойные состояния, устанавливающие ENO = 0

- 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

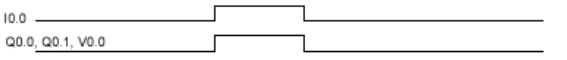
Пример: Катушки




```
Network 1   //Команды присваивания присваивают битовые значения
            // внешним входам и выхода (I, Q) и внутренней памяти
            //(M, SM, T, C, V, S, L).
LD      I0.0
=       Q0.0
=       Q0.1
=       V0.0
Network 2   //Установка в 1 группы из 6 последовательных битов.
            //Указание адреса начального бита и количества битов,
            //подлежащих установке.
            //Индикатор состояния программы для команды установки (S)
            //включен, когда значение первого бита (Q0.2) равно 1.
LD      I0.1
S       Q0.2, 6
Network 3   //Сброс в 0 группы из 6 последовательных битов.
            //Указание адреса начального бита и количества битов,
            //подлежащих сбросу.
            //Индикатор состояния программы для команды сброса (R)
            //включен, когда значение первого бита (Q0.2) равно 0.
LD      I0.2
R       Q0.2, 6
Network 4   //Установка и сброс группы из 8 выходных битов (Q1.0 – Q1.7).
LD      I0.3
LPS
A       I0.4
S       Q1.0, 8
LPP
A       I0.5
R       Q1.0, 8
Network 5   //Команды установки и сброса выполняют функцию реле с
            //блокировкой.
            //Обеспечьте, чтобы биты установки и сброса не были
            //переписаны другой командой присваивания. В этом примере
            //сегмент (Network) 4 устанавливает и сбрасывает 8 выходных
            //битов (Q1.0 – Q1.7) группой.
            //В режиме RUN сегмент 5 может заменять значение бита Q1.0
            //и управлять индикаторами состояния программы для
            //установки и сброса в сегменте 4.
LD      I0.6
=       Q1.0
```

Временная диаграмма

Сегмент 1



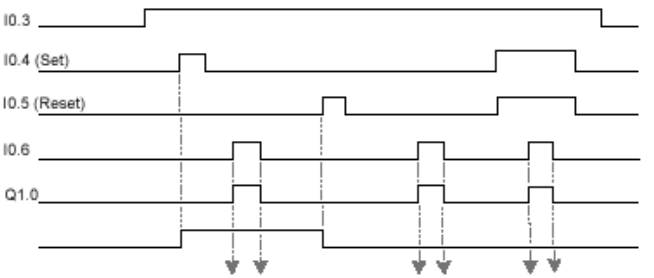
Сегменты 2 и 3

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.

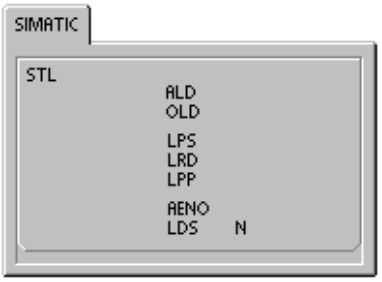
Стековые операции

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 И. 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.

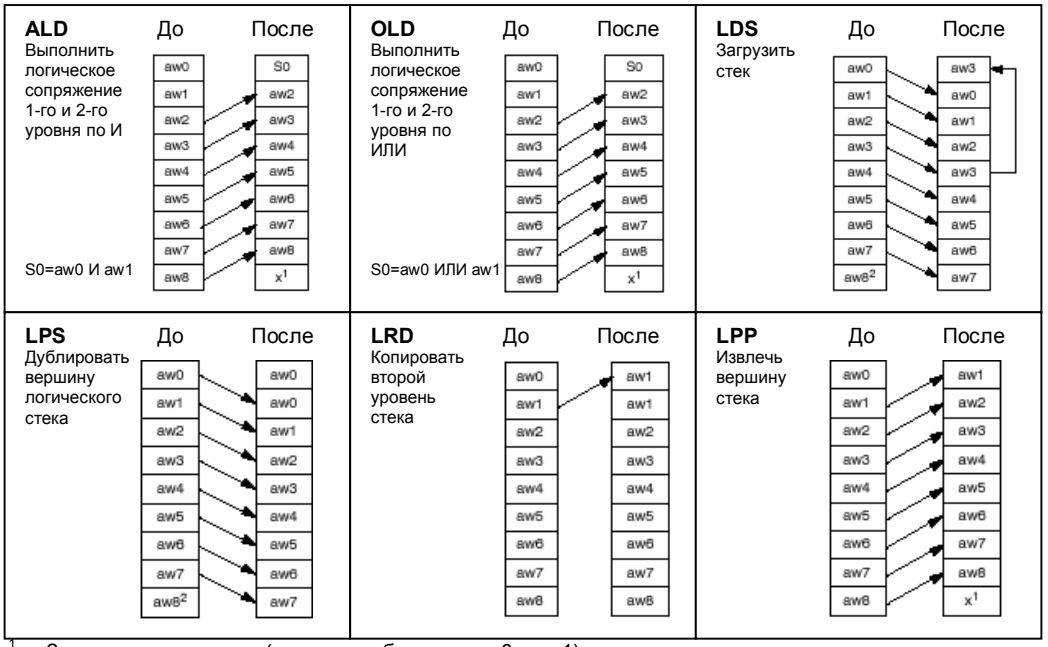
Загрузка стека

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)

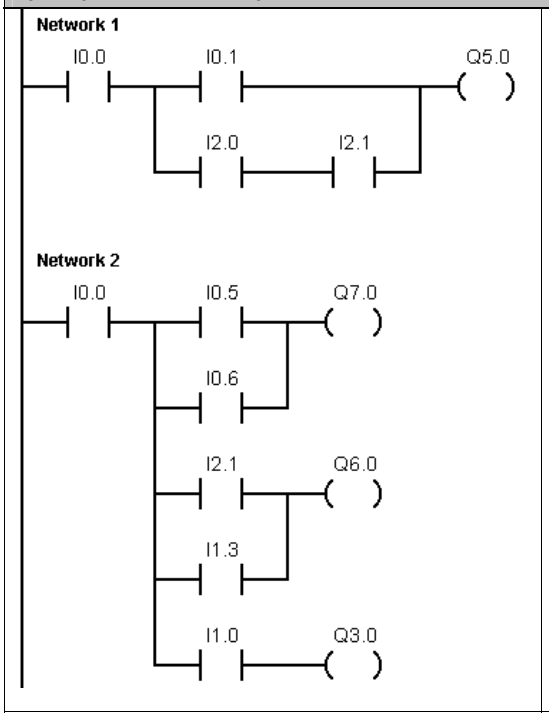
Как показано на рис. 6–2, S7–200 использует логический стек для решения задач логики управления. В этих примерах «aw0» –«aw7» означают начальные значения логического стека, «nw» – новое значение, подготовленное командой, а «S0» означает расчетное значение, которое сохраняется в логическом стеке.



[1] Значение неизвестно (оно может быть равно 0 или 1).
[2] После выполнения команды "Дублировать вершину логического стека" или "Загрузить стек" значение iv8 теряется.

Рис. 6–2. Принцип действия стековых операций

Пример: Стековые операции



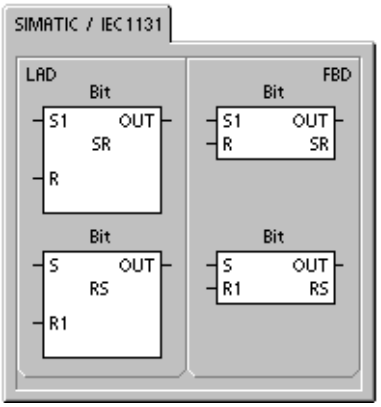

```
Network 1
LD      I0.0
LD      I0.1
LD      I2.0
A       I2.1
OLD
ALD
=       Q5.0
Network 2
LD      I0.0
LPS
LD      I0.5
O       I0.6
ALD
=       Q7.0
LRD
LD      I2.1
O       I1.3
ALD
=       Q6.0
LPP
A       I1.0
=       Q3.0
```

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

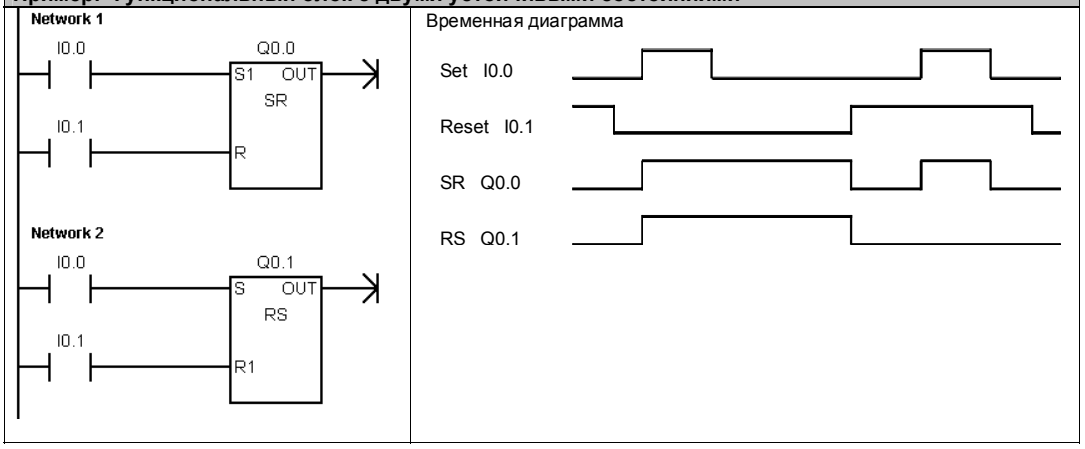
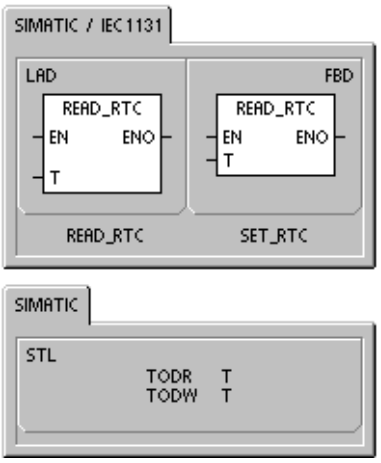


Таблица 6–7. Таблица состояний для функциональных блоков с двумя устойчивыми состояниями

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. 6–3 показан формат 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 деактивизирует день недели.

Рис. 6–3. 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).

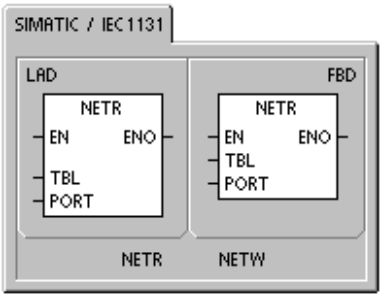
Часы реального времени в S7–200 используют только две младшие значащие цифры для года, так что 2000-й год представлен как 00. ПЛК S7-200 никак не использует информацию о годе. Однако пользовательские программы, которые используют арифметические операции или сравнения со значением года, должны учитывать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

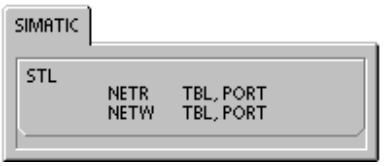
Високосные годы обрабатываются правильно до 2096 года.

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 байта состояния таблицы (см. рис. 6–4)

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.

Мастер команд

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 6–9. 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 226 и CPU 226XM:	0 или 1

На рис. 6–4 описана таблица, к которой обращается параметр TBL, а в таблице 6–10 приведены коды ошибок.

Байтовое смещение	7				0
0	D	A	E	0	Код ошибки
1					
2	Указатель				
3	на область данных				
4	в удаленной станции				
5	(I, Q, M или V)				
6	Длина данных				
7	Байт данных 0				
8	Байт данных 1				
	⋮				
22	Байт данных 15				

D Завершена (операция выполнена): 0 = не выполн. 1 = выполнена
A Активна (операция поставлена в очередь): 0 = не активна 1 = активна
E Ошибка (операция содержит ошибку): 0 = нет ошибки 1 = ошибка

Адрес удаленной станции: адрес ПЛК, к данным которого нужно получить доступ.

Указатель на область данных в удаленной станции: косвенный указатель на данные, к которым нужно получить доступ.

Длина данных: количество байтов данных, к которым нужно получить доступ в удаленной станции (от 1 до 16 байтов).

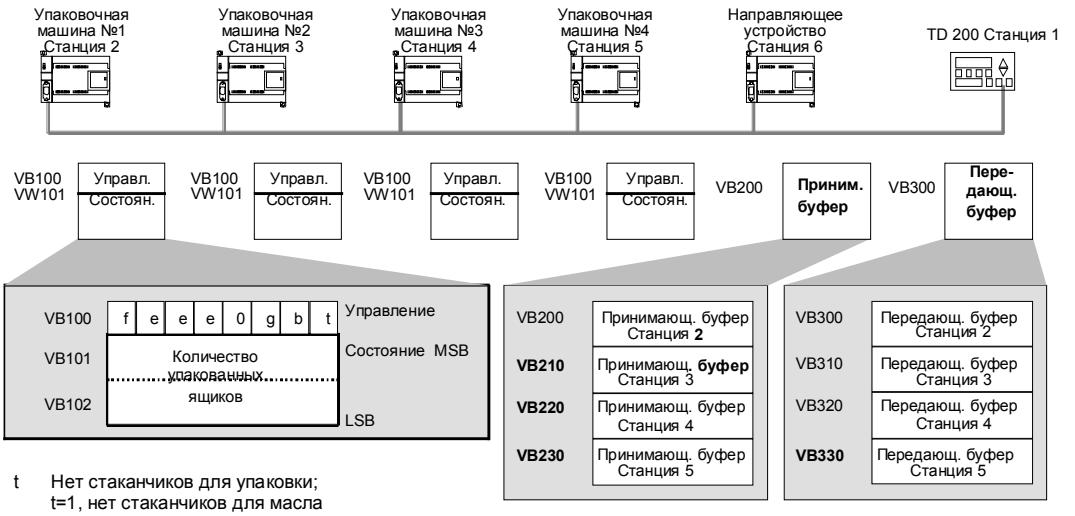
Область данных для приема или передачи: от 1 до 16 байтов, зарезервированных для данных.
Для команды чтения из сети хранит значения, которые были считаны из удаленной станции при выполнении команды.
Для команды записи через сеть хранит значения, подлежащие передаче в удаленную станцию при выполнении команды.

Рис. 6–4. Параметр TBL для команд чтения из сети и записи через сеть

Таблица 6–10. 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, сопряженная с переходом в автономный режим: коллизии, вызванные двойным адресом станции или неисправность 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.)
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. 6–5 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.



t Нет стаканчиков для упаковки;
t=1, нет стаканчиков для масла
b В наличии мало ящиков: b=1, в течение следующих 30 мин. ящики будут заполнены
g Мало клея: g=1, в течение следующих 30 мин. клей должен быть долит
eee Код ошибки, указывающий на вид неисправности
f Индикатор неисправности; f=1, в упаковочной машине возникла неисправность

Рис. 6–5. Пример команд чтения из сети и записи через сеть

На рис. 6–6 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*

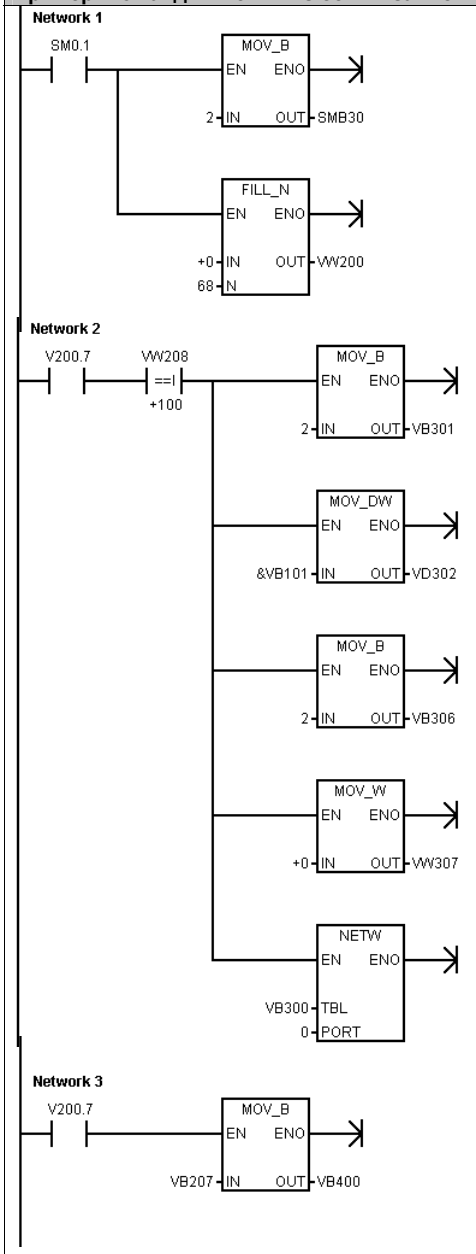
	7				0
VB200	D	A	E	0	Код ошибки
VB201	Адрес удаленной станции = 2				
VB202	Указатель на				
VB203	область данных				
VB204	в				
VB205	удаленной станции = (&VB100)				
VB206	Длина данных = 3 байта				
VB207	Управление				
VB208	Состояние (MSB)				
VB209	Состояние (LSB)				

Передающий буфер *для сброса счетчика упаковочной машины №1*

	7				0
VB300	D	A	E	0	Код ошибки
VB301	Адрес удаленной станции = 2				
VB302	Указатель на				
VB303	область данных				
VB304	в				
VB305	удаленной станции = (&VB101)				
VB306	Длина данных = 2 байта				
VB307	0				
VB308	0				

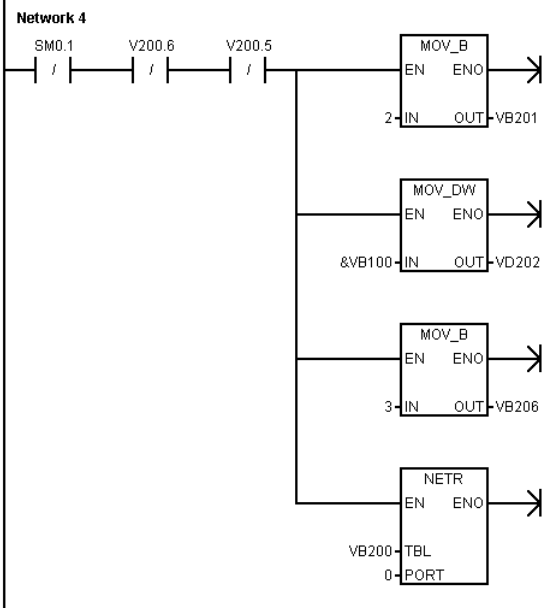
Рис. 6–6. Образец данных TBL для примера команд чтения из сети и записи через сеть

Пример: Команды чтения из сети и записи через сеть




```
Network 1      //В первом цикле разблокировать
               //привилегированный режим PPI
               //и очистить все принимающие и
               //передающие буфера.
LD      SM0.1
MOVB    2, SMB30
FILL    +0, VW200, 68

Network 2      //Если бит завершения команды NETR
               //(V200.7) установлен и 100 коробок
               //упаковано:
               //1. Загрузить адрес станции упаковочной
               //   машины №1.
               //2. Загрузить указатель на данные в
               //   удаленной станции.
               //3. Загрузить длину данных, подлежащих
               //   передаче.
               //4. Загрузить данные для передачи.
               //5. Сбросить количество коробок,
               //   упакованных упаковочной машиной №1
LD      V200.7
AW=     VW208, +100
MOVB    2, VB301
MOVD    &VB101, VD302
MOVB    2, VB306
MOVW    +0, VW307
NETW    VB300, 0

Network 3      //Если бит завершения команды NETR
               //установлен, загрузить управляющие
               //данные упаковочной машины №1.
LD      V200.7
MOVB    VB207, VB400
```

Пример: Команды чтения из сети и записи через сеть, продолжение

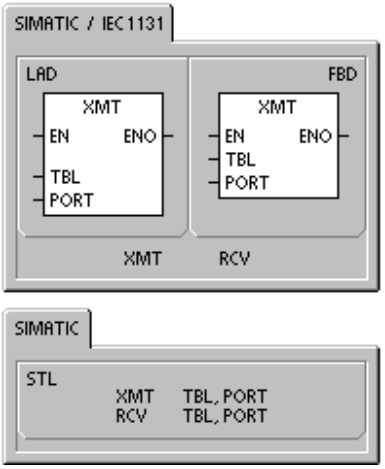



```
Network 4      //Если цикл не первый и нет ошибок:
               //1. Загрузить адрес станции упаковочной
               //   машины №1.
               //2. Загрузить указатель на данные в
               //   удаленной станции.
               //3. Загрузить длину данных, подлежащих
               //   приему.
               //4. Прочитать управляющую информацию
               //   и данные о состоянии упаковочной
               //   машины №1.
LDN     SM0.1
AN      V200.6
AN      V200.5
MOVB    2, VB201
MOVD    &VB100, VD202
MOVB    3, VB206
NETR    VB200, 0
```

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 обслуживание 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
- CPU S7–200 не находится в режиме свободно программируемого обмена данными

Таблица 6–11. 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 226 и CPU 226XM:	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. 222 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. 6–7 описан управляющий байт для свободно программируемого обмена данными. Для всех конфигураций генерируется один стоповый бит.

MSB 7							LSB 0
p	p	d	b	b	b	m	m

SMB30 = порт 0
SMB130 = порт 1

pp: Выбор способа проверки четности
- 00 = нет проверки четности
- 01 = проверка на четность
- 10 = нет проверки четности
- 11 = проверка на нечетность

d: Количество битов данных на символ
- 0 = 8 битов на символ
- 1 = 7 битов на символ

bbb: Скорость передачи для свободно программируемого обмена данными
- 000 = 38 400 Бод
- 001 = 19 200 Бод
- 010 = 9 600 Бод
- 011 = 4 800 Бод
- 100 = 2 400 Бод
- 101 = 1 200 Бод
- 110 = 115,2 КБод[1]
- 111 = 57,6 КБод[1]

[1] CPU S7–200 версии 1.2 или выше поддерживают скорости передачи 57,6 Кбод и 115,2 Кбод.

mm: Выбор протокола
- 00 = PPI/подчиненный режим
- 01 = Протокол свободно программируемого обмена данными
- 10 = PPI/привилегированный режим
- 11 = Резерв (по умолчанию PPI/подчиненный режим)

Рис. 6–7. 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. 6–8 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.

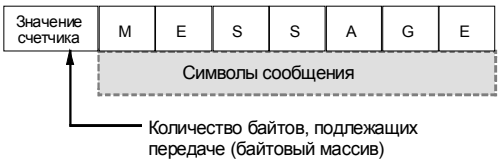


Рис. 6–8. Формат буфера передачи

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. 6–9 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.

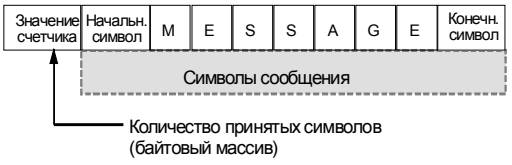


Рис. 6–9. Формат буфера приема

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.

Совет
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.

Таблица 6–12. Байты буфера приема (от SMB86 до SMB94 и от SM1B86 до SMB194)

Порт 0	Порт 1	Описание
SMB86	SMB186	Байт состояния приема сообщения MSB 7 … LSB 0 n r e 0 0 t c 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
SMB87	SMB187	Байт управления приемом сообщения MSB 7 … LSB 0 en sc ec il c/m tmr bk 0 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.
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).

Совет

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.

Команда приема поддерживает несколько условий начала сообщения. Если указанное вами условие начала приема содержит паузу или распознавание бездействия линия, то описанной выше проблемы удается избежать, принуждая функцию приема сообщений синхронизировать начало сообщения с началом символа, прежде чем символ будет помещен в буфер сообщений.

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 инициирует 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 линия с временем, полученным из SMW90 или SMW190. См. рис. 6–10. 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 = простой линии в мс

Рис. 6–10. 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 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 = 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 = не имеют значения, c/m = 1, tmr = 1, SMW92 = выдержка времени таймера в мс

Команда приема поддерживает несколько способов завершения сообщения. Сообщение может быть завершено одним или несколькими 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, чтобы проверить, не совпадает ли он 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 = 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. 6–11.

   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 = выдержка времени таймера в мс

Рис. 6–11. 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. 6–12.

   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 таймера в мс

Рис. 6–12. 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.

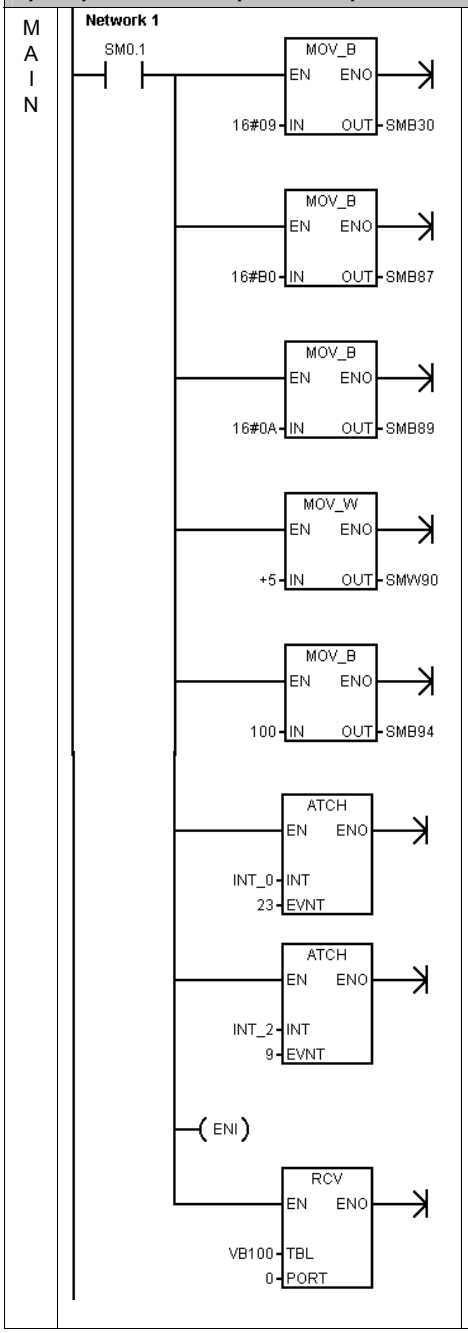
Прием данных, управляемый прерываниями

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.

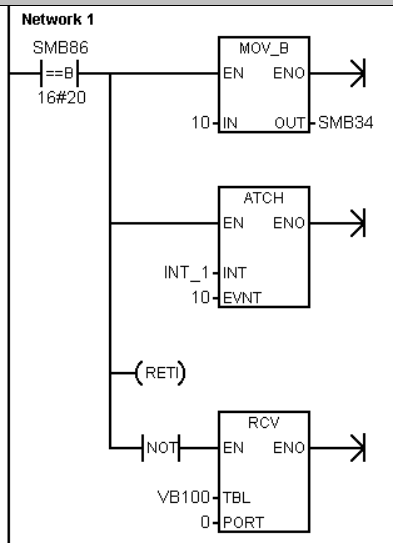
Совет
SMB2 и SMB3 совместно используются портами 0 и 1. Когда прием символа в порт 0 приводит к исполнению программы обработки прерывания, назначенной этому событию (прерывающее событие 8), SMB2 содержит символ, принятый портом 0, а SMB3 содержит результат контроля четности этого символа. Когда прием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

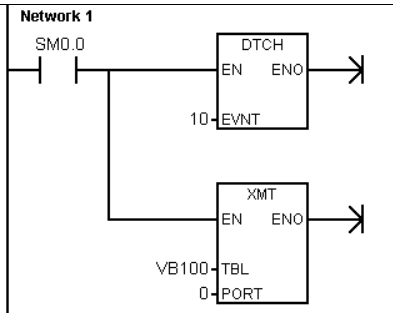
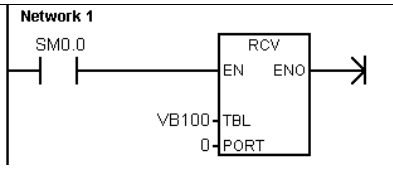
Пример: Команды передачи и приема




```
Network 1       //Эта программа принимает строку символов, пока не
                //будет получен символ перевода строки.
                //Затем сообщение возвращается отправителю.
LD      SM0.1           //В первом цикле:
MOVB    16#09, SMB30    //1.  Инициализировать свободно
                        //    программируемый обмен данными:
                        //    - выбрать 9600 Бод.
                        //    - выбрать 8 битов данных.
                        //    - выбрать отсутствие контроля
                        //      четности.
MOVB    16#B0, SMB87    //2.  Инициализировать управляющий байт
                        //    для приема сообщений:
                        //    - Прием сообщений разблокирован.
                        //    - Обнаруживать символ конца
                        //      сообщения.
                        //    - Обнаруживать условие бездействия
                        //      линии, как условие начала сообщения.
MOVB    16#0A, SMB89    //3.  В качестве символа конца сообщения
                        //    установить OA_H (перевод строки).
MOVW    +5, SMW90       //4.  Установить время ожидания
                        //    бездействия линии равным 5 мс.
MOVB    100, SMB94      //5.  Установить максимальное количество
                        //    символов равным 100.
ATCH    INT_0, 23       //6.  Назначить прерывание 0
                        //    событию "Прием завершен".
ATCH    INT_2, 9        //7.  Назначить прерывание 2
                        //    событию "Передача завершена".
ENI                     //8.  Разблокировать прерывания со
                        //    стороны пользователя.
RCV     VB100, 0        //9.  Разблокировать блок приема с
                        //    буфером в VB100.
```

Пример: Команды передачи и приема, продолжение		

INT 0		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.
		LDB= SMB86, 16#20
		MOVB 10, SMB34
		ATCH INT_1, 10
		CRETI
		NOT
		RCV VB100, 0

INT 1		Network 1 //Прерывание, управляемое 10-миллисекундным //таймером: //1. Отсоединить прерывание, управляемое таймером. //2. Передать сообщение обратно пользователю. LD SM0.0 DTCH 10 XMT VB100, 0
INT 2		Network 1 //Прерывание при завершении передачи: //Разблокировать новый прием. LD SM0.0 RCV VB100, 0

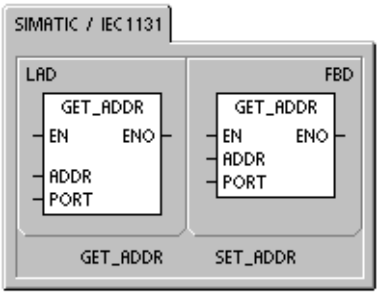
Команды получения и установки адреса порта

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)



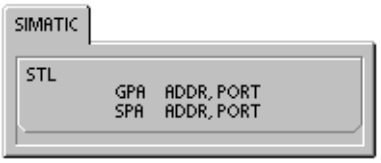


Таблица 6–13. Допустимые операнды для команд получения и установки адреса порта

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 226 и CPU 226XM:* 0 или 1

Команды сравнения

Сравнение числовых величин

Команды сравнения используются для сравнения двух величин:

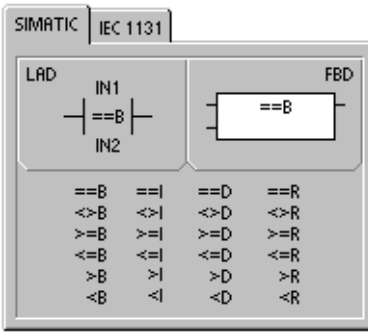
IN1 = IN2 IN1 >= IN2 IN1 <= IN2
IN1 > IN2 IN1 < IN2 IN1 <> IN2

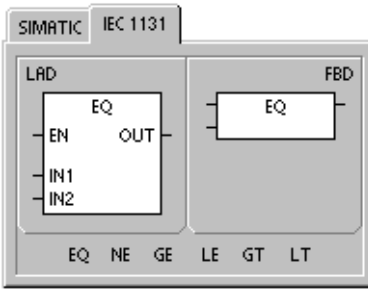
Операции сравнения байтов не учитывают знака.
Операции сравнения целых учитывают знак.
Операции сравнения двойных слов учитывают знак.
Операции сравнения вещественных чисел учитывают знак.

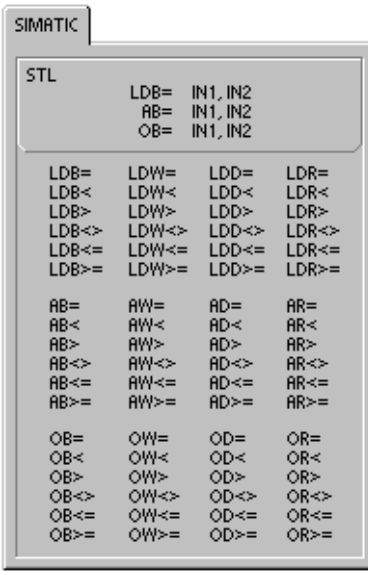
Для LAD и FBD: Если сравнение истинно, команда сравнения включает контакт (LAD) или выход (FBD).

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)

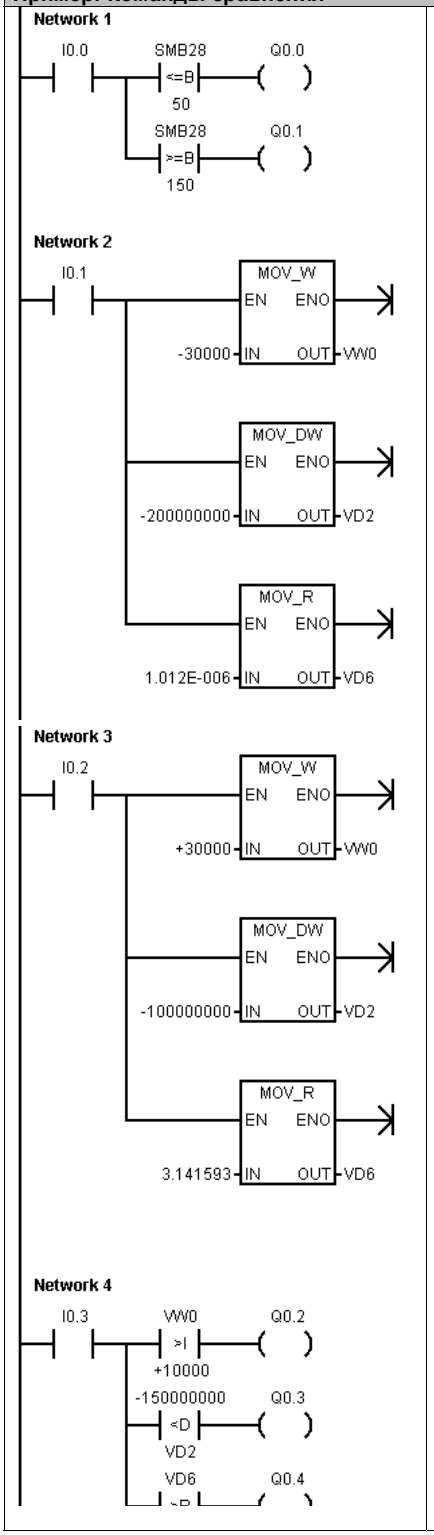
Во избежание появления таких состояний обратите внимание на правильную инициализацию всех указателей и величин, содержащих вещественные числа, перед выполнением команды сравнения.

Команды сравнения выполняются независимо от состояния потока сигнала.

Таблица 6–14. Допустимые операнды для команд сравнения

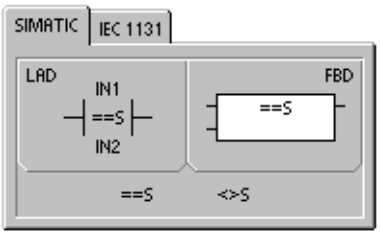
Входы/выходы	Тип	Операнды
IN1, IN2	BYTE INT DINT REAL	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, константа ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, константа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

Пример: Команды сравнения




```
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.
LD      I0.0
LPS
AB<=    SMB28, 50
=       Q0.0
LPP
AB>=    SMB28, 150
=       Q0.1

Network 2      //Загрузить адреса из памяти переменных с
               //небольшими значениями, чтобы сравнения
               //были ложными и индикаторы состояния
               //выключились.
LD      I0.1
MOVW    -30000, VW0
MOVD    -200000000, VD2
MOVR    1.012E-006, VD6

Network 3      //Загрузить адреса из памяти переменных с
               //большими значениями, чтобы сравнения были
               //истинными и индикаторы состояния
               //включились.
LD      I0.2
MOVW    +30000, VW0
MOVD    -100000000, VD2
MOVR    3.141593, VD6

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 > VW100
LD      I0.3
LPS
AW>     VW0, +10000
=       Q0.2
LRD
AD<     -150000000, VD2
=       Q0.3
LPP
AR>     VD6, 5.001E-006
=       Q0.4
```

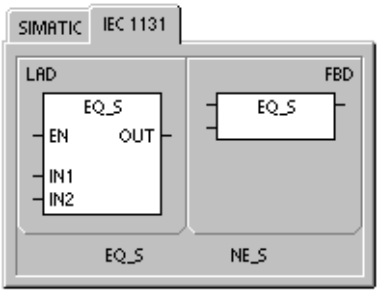
Сравнение строк

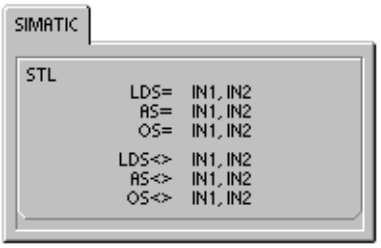


Команда сравнения строк сравнивает между собой две строки символов ASCII:

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 6–15. Допустимые операнды для команд сравнения строк

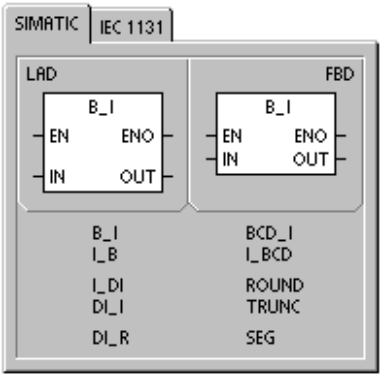
Входы/выходы	Тип	Операнды
IN1, IN2	BYTE (строка)	VB, LB, *VD, *LD, *AC
Выход (OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала

Команды преобразования

Стандартные команды преобразования

Команды преобразования чисел

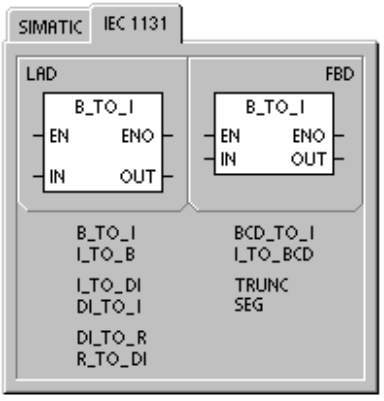
Команды преобразования байта в целое (BTI), целого в байт (ITB), целого в двойное целое (ITD), двойного целого в целое (DTI), двойного целого в вещественное (DTR), двоично-десятичного в целое (BCDI) и целого в двоично-десятичное (IBCD) преобразуют входную величину IN в указанный формат и сохраняют выходное 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 отбрасывание

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.

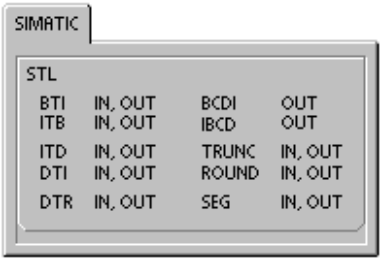


Таблица 6–16. Допустимые операнды для стандартных команд преобразования

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

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). Команда 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).	**Сбойные состояния, устанавливающие ENO = 0** ■ SM1.6 (недопустимое значение BCD) ■ 0006 (косвенный адрес) **Биты специальной памяти, на которые действует команда:** ■ SM1.6 (недопустимое значение BCD)

Принцип действия команды преобразования двойного целого в вещественное

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.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес)

Принцип действия команды преобразования двойного целого числа в целое

Команда преобразования двойного целого числа в целое (DTI) преобразует двойное целое число IN в целое число и помещает результат в переменную, указанную в OUT. Если преобразованное вами значение слишком велико, чтобы быть представленным на выходе, то устанавливается бит переполнения, а выход не изменяется.	**Сбойные состояния, устанавливающие ENO = 0** ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) **Биты специальной памяти, на которые действует команда:** ■ SM1.1 (переполнение)

Принцип действия команды преобразования целого числа в двойное целое

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.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес)

Принцип действия команды преобразования байта в целое число

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.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес)

Принцип действия команды преобразования целого числа в байт

Команда 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.	**Сбойные состояния, устанавливающие ENO = 0** ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) **Биты специальной памяти, на которые действует команда:** ■ SM1.1 (переполнение)

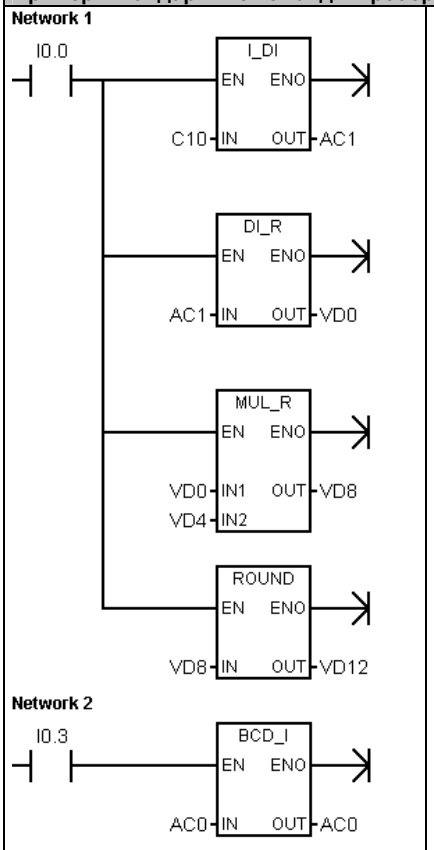
Совет
Для преобразования целого числа в вещественное используйте команду преобразования целого числа в двойное целое, а затем команду преобразования двойного целого числа в вещественное.

Принцип действия команд округления и округления отбрасыванием

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. 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.	**Сбойные состояния, устанавливающие ENO = 0** ■ SM1.1 (переполнение) ■ 0006 (косвенный адрес) **Биты специальной памяти, на которые действует команда:** ■ 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

C10	101	Значение счетчика = 101 дюйм
VD0	101.0	Значение счетчика (как вещественное число)
VD4	2.54	Константа 2,54 (дюймы в сантиметры)
VD8	256.54	256,54 сантиметра как вещественное число
VD12	**257**	257 сантиметров как двойное целое

BCD в целое

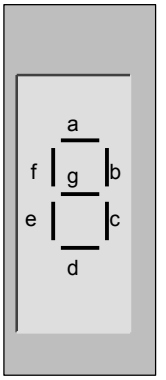
AC0	1234
	BCDI
AC0	04D2

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. 6–13 показано кодирование семисегментного дисплея, используемое командой SEG.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес)

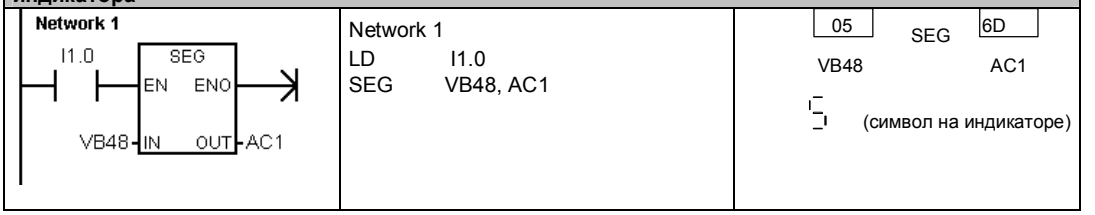
(IN) LSD	Отображение сегментов	(OUT) - g f e d c b a
0		0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2		0 1 0 1 1 0 1 1
3		0 1 0 0 1 1 1 1
4		0 1 1 0 0 1 1 0
5		0 1 1 0 1 1 0 1
6		0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1



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

Рис. 6–13. Кодирование семисегментного индикатора

Пример: Команда генерирования комбинации двоичных разрядов для 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.

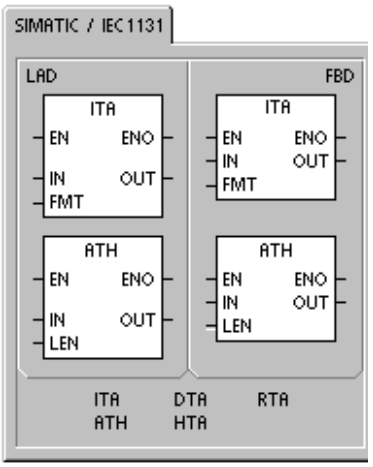
Максимальное количество шестнадцатеричных цифр, которое может быть преобразовано, равно 255.

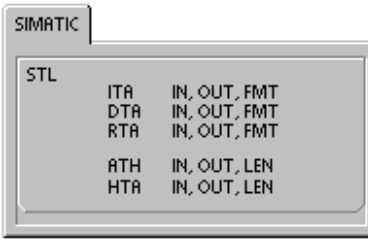
Сбойные состояния, устанавливающие ENO = 0

- 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 число в строку символов ASCII (RTA) преобразуют целое, двойное целое или вещественное число в символы ASCII.

Таблица 6–17. Допустимые операнды для команд преобразования ASCII

Входы/выходы	Тип данных	Операнды
IN	BYTE INT DINT REAL	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC 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 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.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес) ■ Недопустимый формат ■ *nnn* > 5

Массив символов ASCII всегда содержит 8 элементов.

На рис. 6–14 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. 6–14 показаны примеры чисел, форматированных с помощью десятичной точки (c = 0), с тремя цифрами справа от десятичной точки (nnn=011). Выходной буфер 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 выходном буфере выравниваются вправо.

FMT

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

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

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7
IN=12				0	.	0	1	2
IN=–123			–	0	.	1	2	3
IN=1234				1	.	2	3	4
IN = –12345		–	1	2		3	4	5

Рис. 6–14. 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, а также форму представления разделителя целой и дробной части – в виде запятой или точки. Результат преобразования помещается в 12 последовательных байтов, начиная с адреса OUT.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес) ■ Недопустимый формат ■ *nnn* > 5

Размер выходного буфера всегда равен 12 байтам.

На рис. 6–15 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. 6–15 также показаны примеры чисел, форматированных с помощью десятичной точки (c = 0), с четырьмя цифрами справа от десятичной точки (nnn=100). Выходной буфер 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 выходном буфере выравниваются вправо.

FMT

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

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

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8	Out +9	Out +10	Out +11
IN=–12						–	0	.	0	0	1	2
IN=1234567					1	2	3	.	4	5	6	7

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

Принцип действия команды преобразования вещественного числа в строку символов ASCII

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. Результат преобразования помещается в выходном буфере, начиная с адреса OUT.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес) ■ nnn > 5 ■ ssss < 3 ■ ssss< число символов в OUT

Длина 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. 6–16 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 *c* определяет использование запятой (c=1) или десятичной точки (c=0) в качестве разделителя целой и дробной части.

На рис. 6–16 также приведены примеры чисел, форматированных с помощью десятичной точки (c = 0), с одной цифрой справа от десятичной точки (nnn=001) и размером буфера, равным 6 байтам (ssss=0110). Выходной буфер 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 выходного буфера должен, по крайней мере, на три байта превышать количество цифр справа от десятичной точки.
- Числа в выходном буфере выравниваются вправо.

FMT

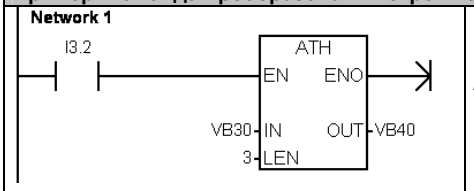
MSB 7	6	5	4	3	2	1	LSB 0
s	s	s	s	c	n	n	n

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

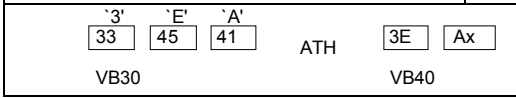
	Out	Out +1	Out +2	Out +3	Out +4	Out +5
IN = 1234.5	1	2	3	4	.	5
IN = -0.0004				0	.	0
IN = -3.67526			-	3	.	7
IN = 1.95				2	.	0

Рис. 6–16. Операнд FMT для команды преобразования вещественного числа в строку символов ASCII (RTA)

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

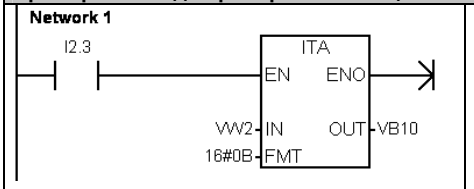



```
Network 1
LD      I3.2
ATH     VB30, VB40, 3
```

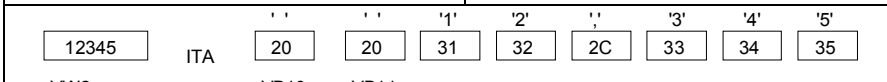


Указание: X показывает, что половина байта не меняется

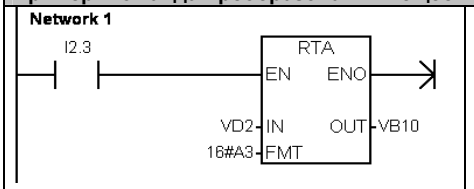
Пример: Команда преобразования целого числа в строку символов ASCII



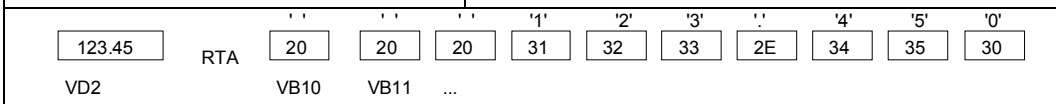

```
Network 1       //Преобразовать целое число из VW2
                //в 8 символов ASCII, начиная с VB10,
                //с помощью формата 16#0B
                //(запятая вместо десятичной точки,
                //за которой следуют 3 цифры).
LD      I2.3
ITA     VW2, VB10, 16#0B
```



Пример: Команда преобразования вещественного числа в строку символов ASCII



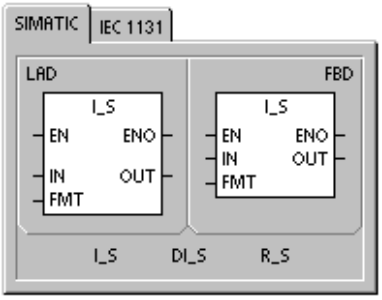

```
Network 1       //Преобразовать вещественное число из VD2
                //в 10 символов ASCII, начиная с адреса
                // VB10, с помощью формата 16#A3
                //(запятая вместо десятичной точки,
                // за которой следуют 3 цифры).
LD      I2.3
RTA     VD2, VB10, 16#A3
```



Команды преобразования строк

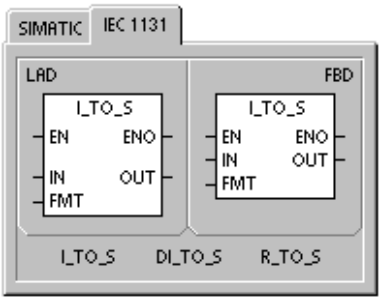
Преобразование числовых значений в строки

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).



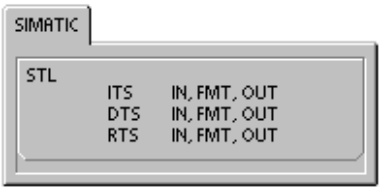
Принцип действия команды преобразования целого числа в строку символов

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. 6–17 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 *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. 6–17 показаны примеры чисел, форматированных с помощью десятичной точки (c = 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 6–18. Допустимые операнды для команд преобразования числовых значений в строки

Входы/выходы	Тип данных	Операнды
IN	BYTE (строка) INT DINT REAL	VB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC, 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
INDX, FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа
OUT	BYTE (строка) INT DINT, REAL	VB, LB, *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

FMT

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

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

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8
IN=12	8				0	.	0	1	2
IN=-123	8			–	0	.	1	2	3
IN=1234	8				1	.	2	3	4
IN = -12345	8		–	1	2	.	3	4	5

Рис. 6–17. Операнд FMT для команды преобразования целого числа в строку

Принцип действия команды преобразования двойного целого числа в строку

Команда преобразования двойного целого числа в строку (DTS) преобразует двойное целое число IN в строку ASCII длиной 12 символов. Операнд 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.

Сбойные состояния, устанавливающие ENO = 0

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

На рис. 6–18 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. 6–18 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 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.

FMT

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

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

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8	Out +9	Out +10	Out +11	Out +12
IN=12	12					.		0	.	0	0	1	2
IN=–1234567	12				–	1	2	3	.	4	5	6	7

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

Принцип действия команды преобразования вещественного числа в строку

Команда преобразования вещественного числа в строку (RTS) преобразует вещественное число IN в строку символов ASCII. Формат (FMT) определяет точность преобразования справа от десятичной точки, а также форму представления разделителя целой и дробной части – в виде запятой или точки, а также длину выходной строки. 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.	**Сбойные состояния, устанавливающие ENO = 0** ■ 0006 (косвенный адрес) ■ 0091 (операнд вышел за пределы допустимого диапазона) ■ Недопустимый формат: 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. 6–19 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. 6–19 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

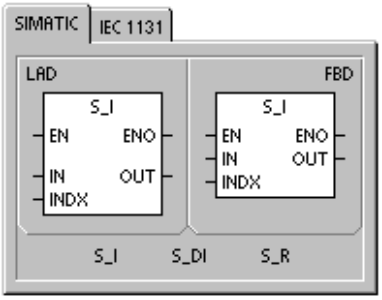
MSB 7	6	5	4	3	2	1	LSB 0
s	s	s	s	c	n	n	n

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

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6
IN=1234.5	6	1	2	3	4	.	5
IN= -0.0004	6				0	.	0
IN= –3.67526	6			–	3	.	7
IN = 1.95	6				2	.	0

Рис. 6–19. Операнд FMT для команды преобразования вещественного числа в строку

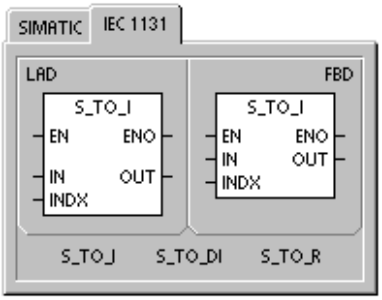
Преобразование подстрок в числовые значения



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.

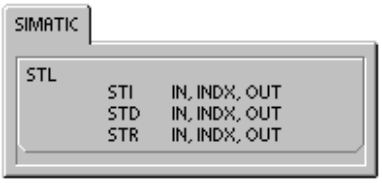
Сбойные состояния, устанавливающие ENO = 0

- 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 6–19. Допустимые операнды для команд преобразования подстрок в числовые значения

Входы/выходы	Тип данных	Операнды
IN	BYTE (строка)	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	BYTE (строка) INT DINT, REAL	VB, IB, QB, MB, SMB, SB, LB, *VD, *LD, *AC, константа 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

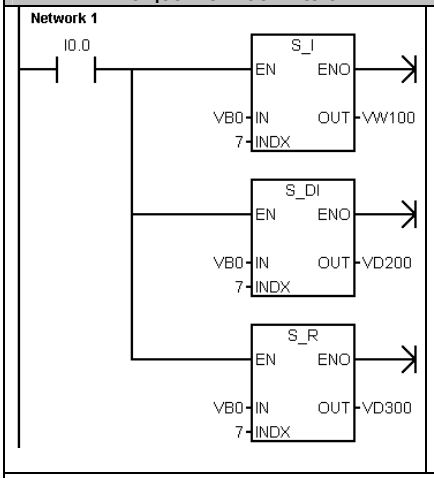
Входная строка	Выходное веществ. число
‘123’	123.0
‘-00456’	-456.0
‘123.45’	123.45
‘+2345’	2345.0
‘00.000000123’	0.000000123

Недопустимые входные строки

Входная строка
‘A123’
‘ ’
‘++123’
‘+–123’
‘+ 123’

Рис. 6–20. Примеры допустимых и недопустимых входных строк

Пример: Преобразование строк: Преобразование подстроки в целое, двойное целое и вещественное число

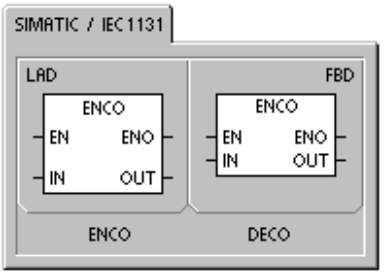



```
Network 1   //Преобразует числовую строку в целое.
            //Преобразует числовую строку в двойное целое.
            //Преобразует числовую строку в вещественное
            //число.
LD    I0.0
STI   VB0,7,VW100
STD   VB0,7,VD200
STR   VB0,7,VD300
```

VB0											VB11
11	'T'	'e'	'm'	'p'	' '	' '	'9'	'8'	'.'	'6'	'F'

После выполнения сегмента:
VW100 (целое) = 98
VD200 (двойное целое) = 98
VD300 (вещественное) = 98,6

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.

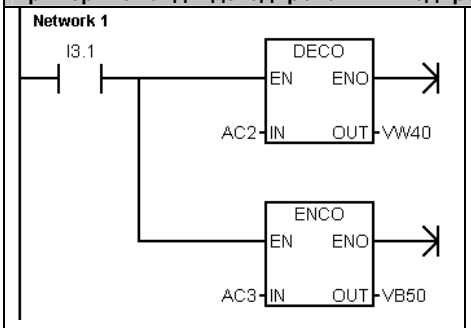
Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Таблица 6–20. 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

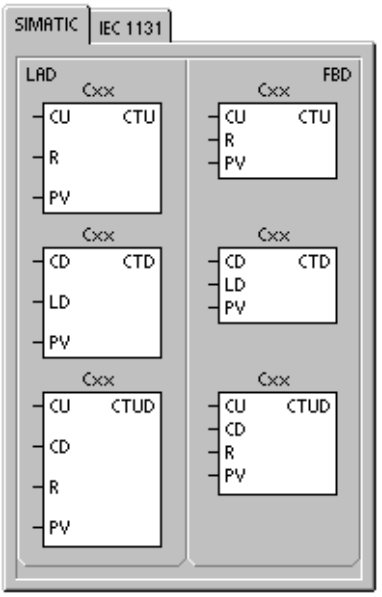
Пример: Команды декодирования и кодирования

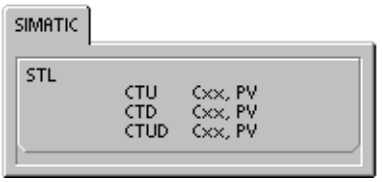



```
Network 1     //Аккумулятор 2 содержит биты ошибок.
              //1. Команда DECO устанавливает в VW40
              //   бит, соответствующий этому коду
              //   ошибки.
              //2. Команда ENCO преобразует младший
              //   установленный бит в код ошибки,
              //   который сохраняется в VB50.
LD      I3.1
DECO    AC2, VW40
ENCO    AC3, VB50
```

Команды счета

Счетчик SIMATIC





Прямой счет

Команда прямого счета (CTU) увеличивает текущее значение счетчика при появлении нарастающего фронта на входе (CU). Когда текущее значение Cxx больше или равно предустановленному значению PV, бит счетчика Cxx 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 Cxx равно нулю, бит счетчика Cxx включается. Счетчик сбрасывает свой бит Cxx и загружает текущее значение предустановленным значением PV, когда включается вход загрузки LD. Счетчик останавливается, когда он достигает нуля, и бит счетчика Cxx 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). Cxx 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 Cxx больше или равно предустановленному значению PV, то бит счетчика Cxx 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 6–21. Допустимые операнды для счетчиков SIMATIC

Входы/выходы	Типы данных	Операнды
Cxx	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).

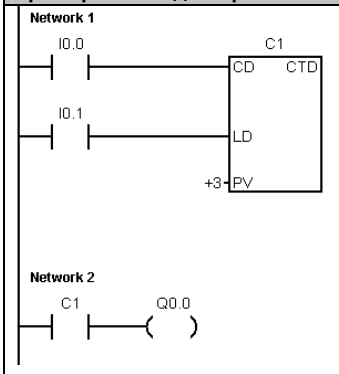
Если вы сбрасываете счетчик с помощью команды Сброс, то бит счетчика сбрасывается, а текущее значение устанавливается в ноль. Номер счетчика используется для обращения как к текущему значению, так и к биту счетчика.

Таблица 6–22. Принцип действия счетчиков

Тип	Режим	Бит счетчика	Выключение-включение питания/Первый цикл
CTU	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]

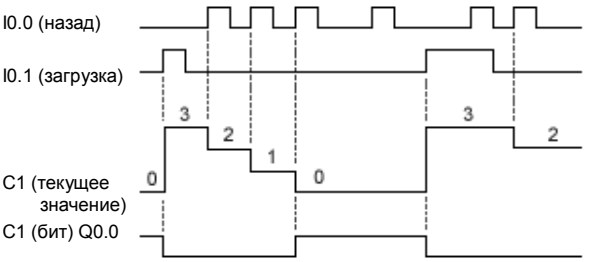
[1] Вы можете определить текущее значение счетчика как сохраняемую величину. Информацию о сохраняемой памяти CPU S7–200 вы найдете в главе 4.

Пример: Команда обратного счета SIMATIC

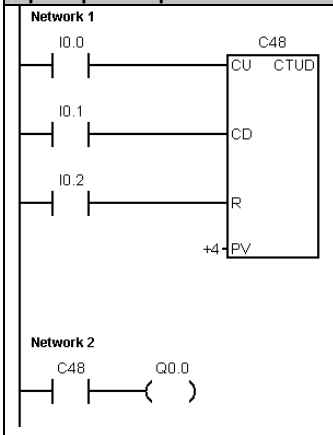



```
Network 1      //Текущее значение счетчика обратного счета C1
               //уменьшается с 3 до 0, когда I0.1 выключен.
               //Положительный фронт на I0.0 уменьшает текущее
               //значение C1
               //Включение I0.1 загружает предустановленное
               //значение 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

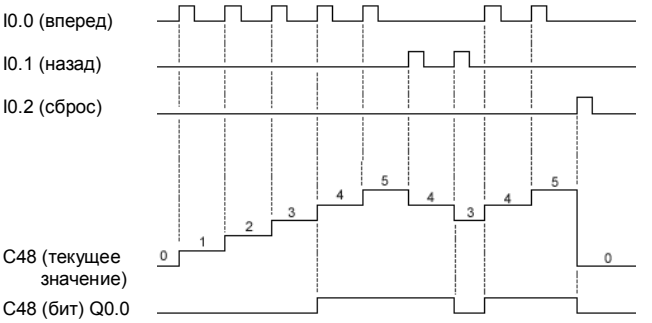


Пример: Реверсивный счетчик SIMATIC

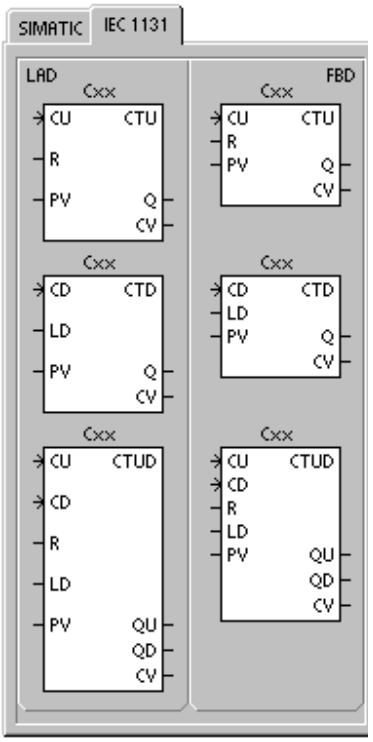



```
Network 1      //I0.0 считает вперед
               //I0.1 считает назад
               //I0.2 сбрасывает текущее значение в 0
LD     I0.0
LD      I0.1
LD     I0.2
CTUD   C48, +4

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

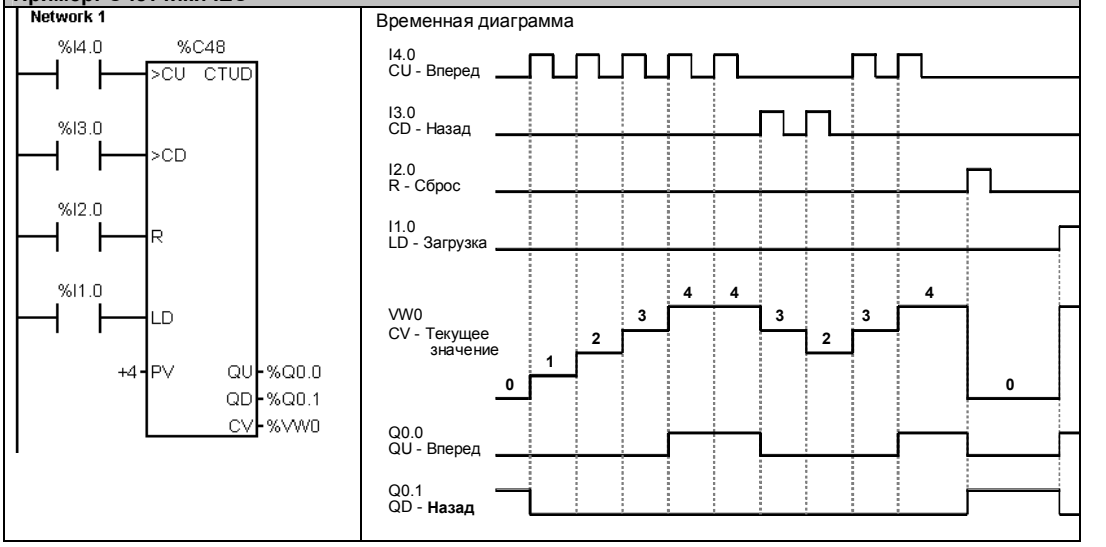
Реверсивный счетчик (CTUD) ведет счет вверх или вниз от предустановленного значения (CV) при положительном фронте на входе прямого счета (CU) или обратного счета (CD). Когда текущее значение равно предварительно установленному значению, включается выход прямого счета (QU). Когда текущее значение равно 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 6–23. 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

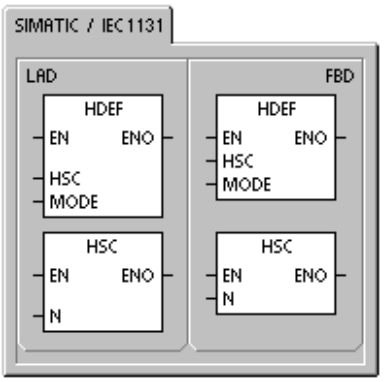
Совет
Так как для каждого счетчика имеется одно текущее значение, не назначайте один и тот же номер более чем одному счетчику. (Суммирующие, вычитающие и реверсивные счетчики обращаются к одному и тому же текущему значению.)

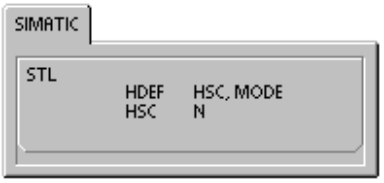
Пример: Счетчики IEC



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 6–25.

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 6–24. Допустимые операнды для скоростных счетчиков

Входы/выходы	Типы данных	Операнды
HSC, MODE	BYTE	Константа
N	WORD	Константа

Советы и уловки

За программами, использующими скоростные счетчики, обратитесь к советам и уловкам (Tips and Tricks) 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 A.

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 226 и CPU 226XM 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 A и B. 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 (**H**igh–Speed Counter **Def**inition = 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 6–25 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.

Совет
Примите во внимание, что все режимы HSC0 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.

Таблица 6–25. Входы для скоростных счетчиков

Режим	Описание	Входы			
		I0.0	I0.1	I0.2	
		I0.6	I0.7	I0.2	I1.1
		I1.2	I1.3	I1.1	I1.2
		I0.1			
		I0.3	I0.4	I0.5	
		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 A и B	Датчик тактовых импульсов A	Датчик тактовых импульсов B		
10		Датчик тактовых импульсов A	Датчик тактовых импульсов B	Сброс	
11		Датчик тактовых импульсов A	Датчик тактовых импульсов B	Сброс	Пуск

Примеры режимов HSC

Временные диаграммы на рисунках 6–21 - 6–25 показывают, как работает каждый счетчик в соответствии с режимом.

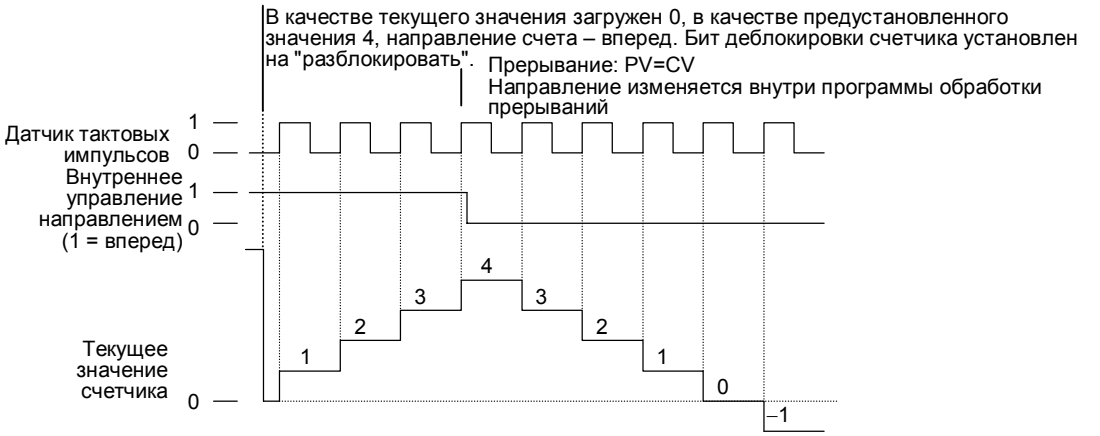


Рис. 6–21. Пример работы в режимах 0, 1 или 2

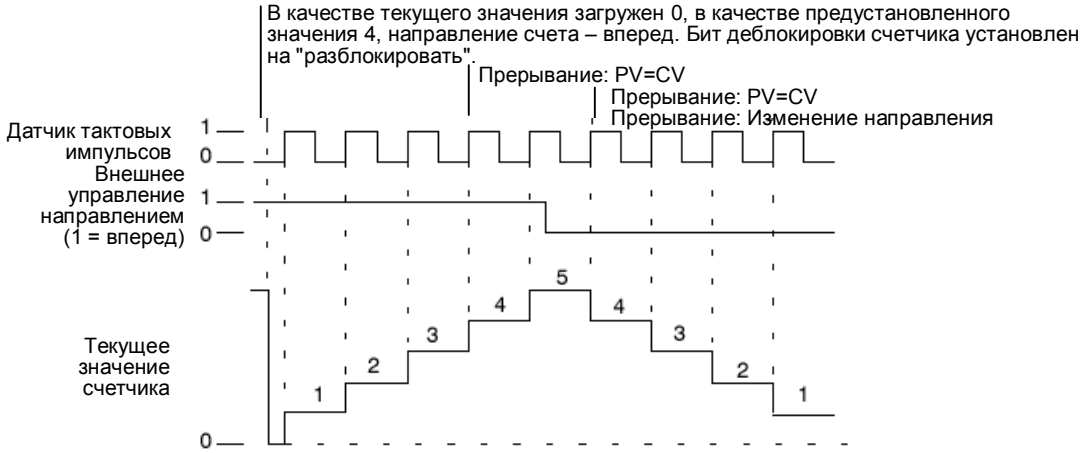


Рис. 6–22. Пример работы в режимах 3, 4 или 5

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.

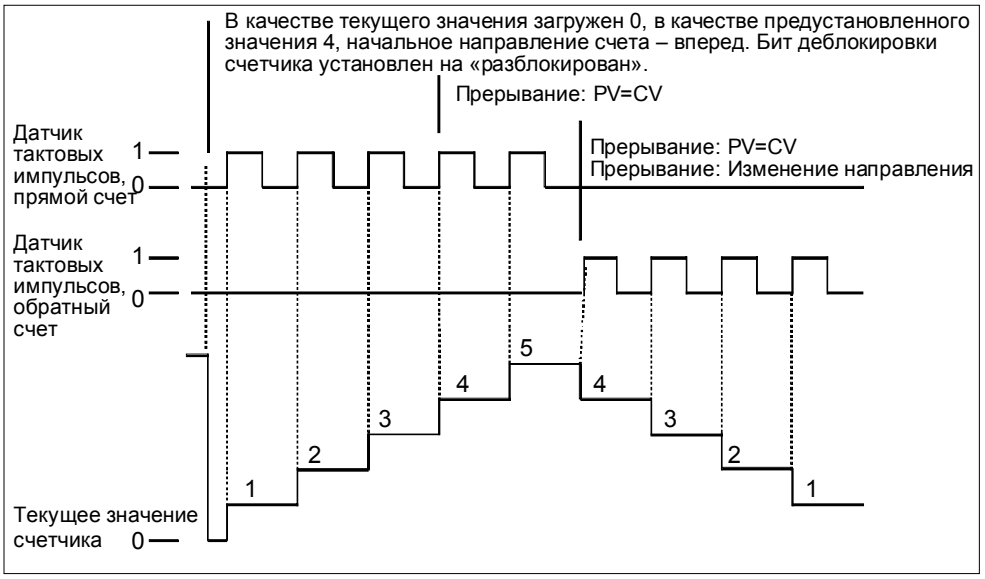


Рис. 6–23. Пример работы в режимах 6, 7 или 8

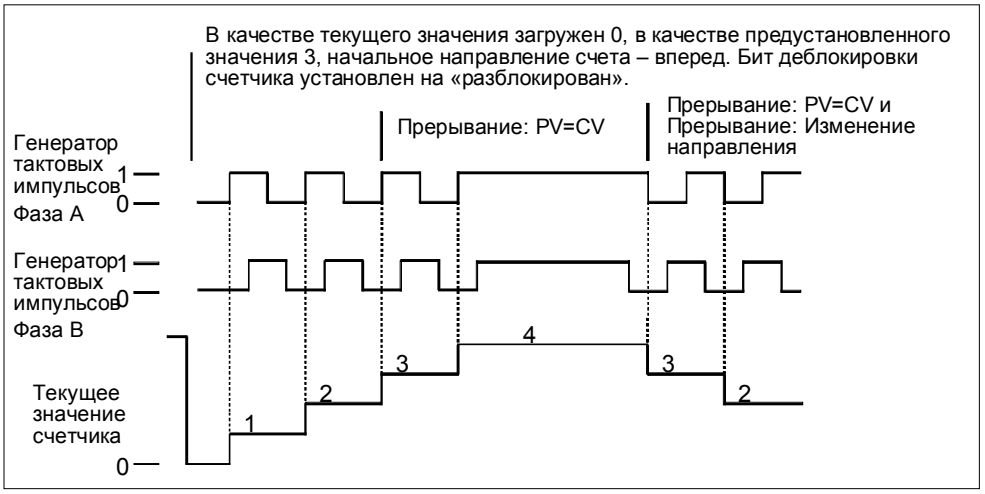


Рис. 6–24. Пример работы в режимах 9, 10 или 11 (квадратурный режим, однократная скорость)

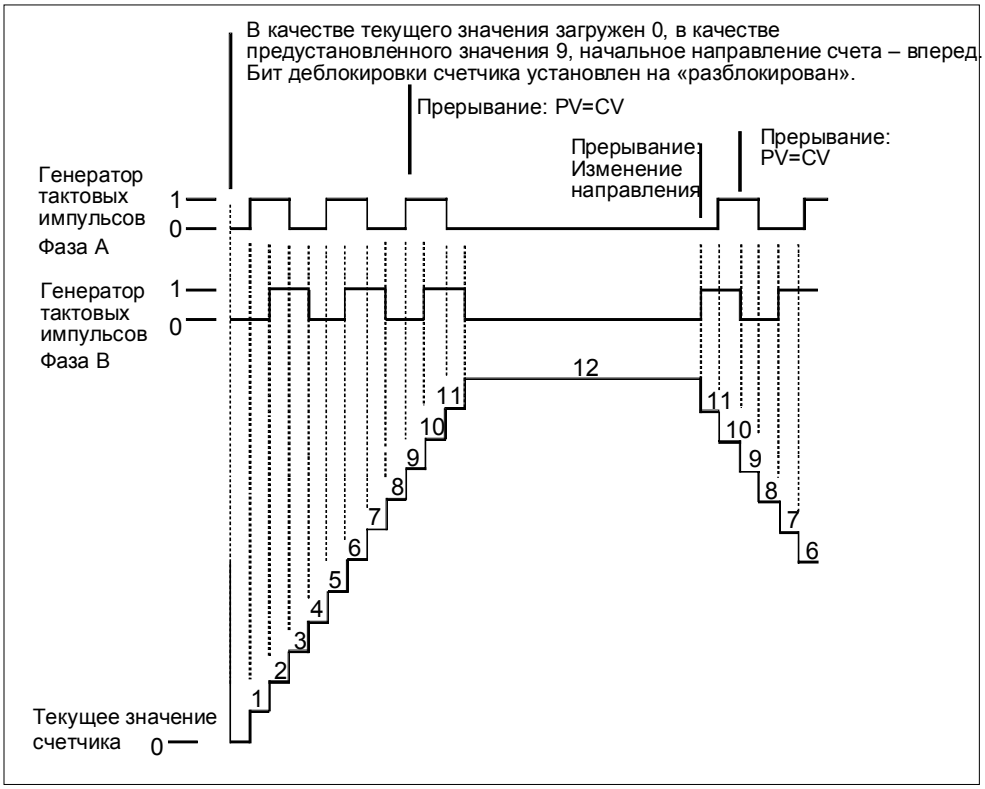


Рис. 6–25. Пример работы в режимах 9, 10 или 11 (квадратурный режим, четырехкратная скорость)

Принцип действия входов сброса и пуска

Работа входов сброса и запуска показана на рис. 6–26 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.

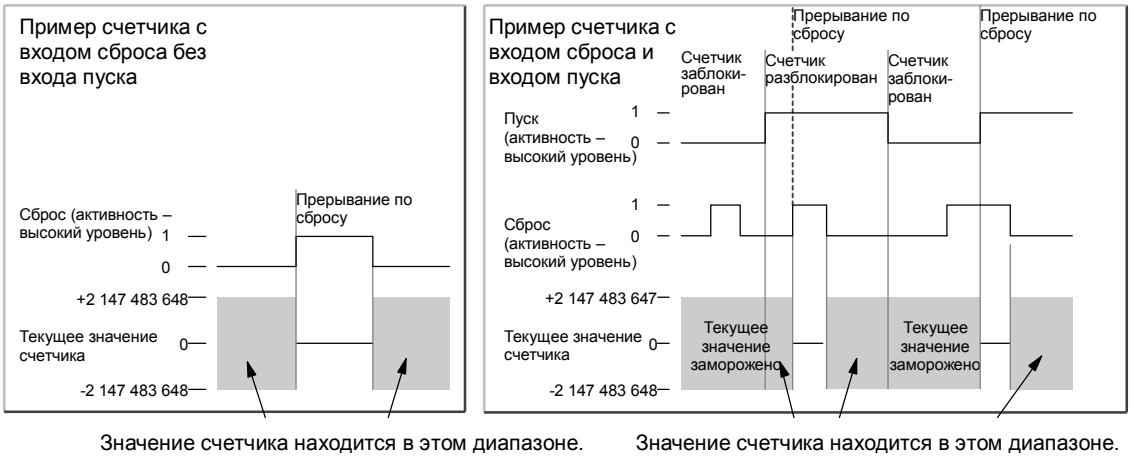


Рис. 6–26. Примеры работы счетчика, использующего вход сброса с входом и без входа пуска

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 6–26.

Совет
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.

Если команда HDEF была выполнена, вы не можете изменить настройку счетчика, не переведя сначала S7–200 в состояние STOP.

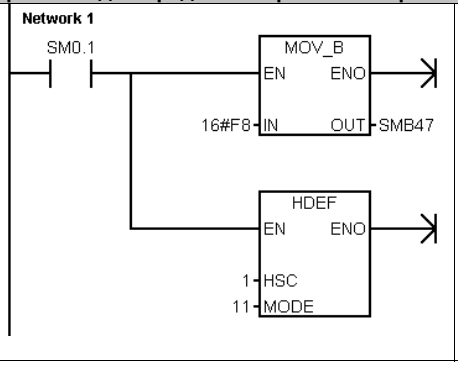
Таблица 6–26. Активный уровень для управляющих битов сброса, пуска и выбора однократной или четырехкратной скорости

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

M A I N




```
Network 1   //В первом цикле:
            //1. Выбрать для входов пуска и сброса
            //   в качестве активного высокий
            //   уровень и 4-кратную скорость счета.
            //2. Настроить HSC1 на квадратурный
            //   режим (A/B-счетчик) с входами
            //   сброса и пуска.
LD      SM0.1
MOVB    16#F8, SMB47
HDEF    1, 11
```

Настройка управляющего байта

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 6–27 описан каждый из этих управляющих битов.

Таблица 6–27. 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 6–28 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. Текущее значение непосредственно доступно для операций чтения, но оно может быть записано только с помощью команды HSC.

Таблица 6–28. Текущее и предустановленное значения HSC0, HSC1, HSC2, HSC3, HSC4 и HSC5

Загружаемое значение	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Новое текущее	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Новое предустановленное	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Адресация скоростных счетчиков (HC)

Для доступа к счетному значению скоростного счетчика указывается адрес этого счетчика с помощью типа памяти (HC) и номера счетчика (например, HC0). Текущее значение скоростного счетчика доступно только для чтения и может быть адресовано только как двойное слово (32 бита), как показано на рис. 6–27.

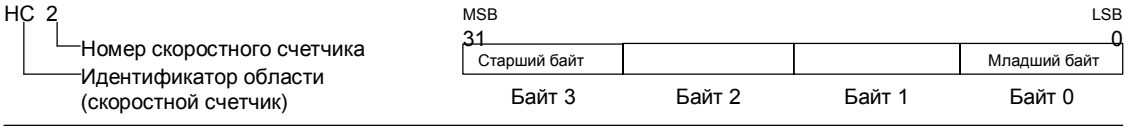


Рис. 6–27. Доступ к текущему значению скоростного счетчика

Назначение прерываний

Все режимы счетчиков поддерживают прерывание по равенству текущего значения предустановленному. 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 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 6–29 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 6–29. 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 вышеприведенные 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).

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 прямое направление счета
   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 0 при отсутствии внешнего сброса и пуска, 1 для внешнего сброса без пуска или 2 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 распознавать равенство текущего и предустановленного значений, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие CV = PV (событие 13). Подробную информацию об обработке прерываний вы найдете в разделе, посвященном командам прерывания.
7. Чтобы распознавать внешний сброс, запрограммируйте прерывание, поставив в соответствие программе обработки прерывания прерывающее событие «внешний сброс» (external reset) (событие 15).
8. Для разблокировки прерываний выполните команду разрешения всех прерываний (ENI).
9. Выполните команду HSC, чтобы S7–200 запрограммировал HSC1.
10. 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. 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 соответствие программе 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 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 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. 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 A и B (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 *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

   Пример (четырехкратная скорость счета):
   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: 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.

Изменение направления в режиме 0, 1 или 2

Следующие шаги описывают, как сконфигурировать изменение направления для HSC1 как однофазного счетчика с внутренним управлением направлением счета (режим 0, 1 или 2):

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.

Загрузка нового текущего значения (любой режим)

Изменение текущего значения приводит к принудительной блокировке счетчика на время выполнения изменения. Пока счетчик заблокирован, он не считает и не генерирует прерываний.

Следующие шаги описывают, как изменить текущее значение счетчика HSC1 (любой режим):

1. Загрузите SMB47, чтобы записать желаемое текущее значение:

   SMB47 = 16#C0 — Разблокирует счетчик
   Записывает новое текущее значение

2. Загрузите SMD48 (двойное слово) желаемым текущим значением (загрузите 0, чтобы его очистить).
3. Выполните команду HSC, чтобы S7–200 запрограммировал HSC1.

Загрузка нового предустановленного значения (любой режим)

Следующие шаги описывают, как изменить предустановленное значение HSC1 (любой режим):

1. Загрузите SMB47, чтобы записать желаемое предустановленное значение:

   SMB47 = 16#A0 — Разблокирует счетчик
   Записывает новое предустановленное значение

2. Загрузите SMD52 (двойное слово) желаемым предустановленным значением.
3. Выполните команду HSC, чтобы S7–200 запрограммировал HSC1.

Блокировка скоростного счетчика (любой режим)

Следующие шаги описывают, как заблокировать скоростной счетчик HSC1 (любой режим):

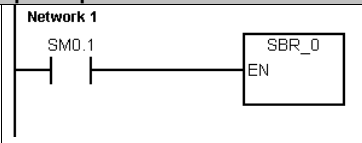
1. Загрузите SMB47, чтобы заблокировать счетчик:

   SMB47 = 16#00 — Блокирует счетчик

2. Выполните команду HSC, чтобы заблокировать счетчик.

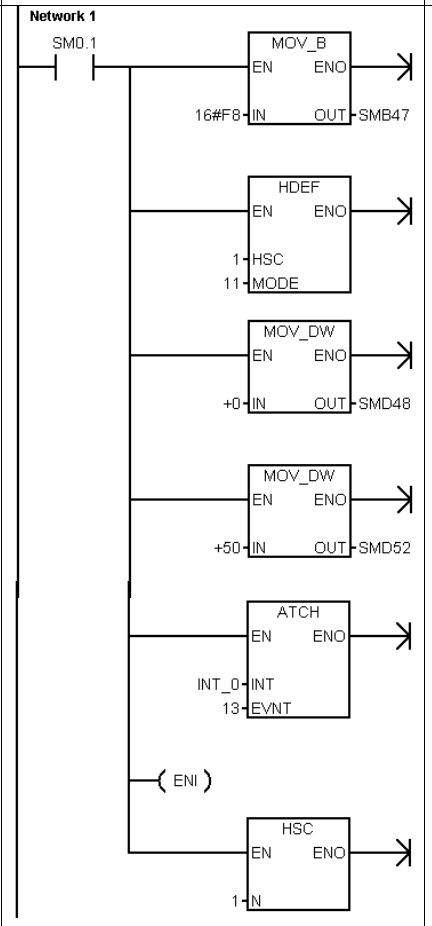
Пример: Скоростной счетчик

MAIN



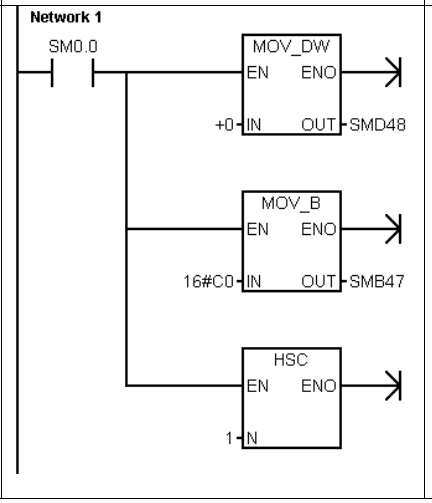

```
Network 1     //В первом цикле вызвать SBR_0.
LD      SM0.1
CALL    SBR_0
```

SBR 0




```
Network 1     // В первом цикле настроить HSC1:
              //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.
              //2.  Настроить HSC1 на квадратурный режим
              //    с входами сброса и пуска.
              //3.  Сбросить текущее значение HSC1.
              //4.  Ввести предустановленное значение HSC1,
              //    равное 50.
              //5.  Программе обработки прерываний INT_0
              //    поставить в соответствие событие 13 (текущее
              //    значение HSC1 = предустановленному)
              //6.  Разблокировать все прерывания.
              //7.  Запрограммировать HSC1.
LD      SM0.1
MOVB    16#F8, SMB47
HDEF    1, 11
MOVD    +0, SMD48
MOVD    +50, SMD52
ATCH    INT_0, 13
ENI
HSC     1
```

INT 0

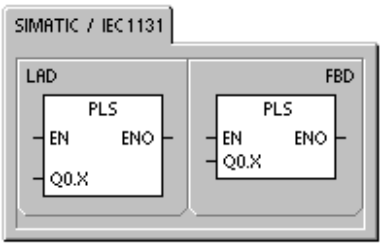


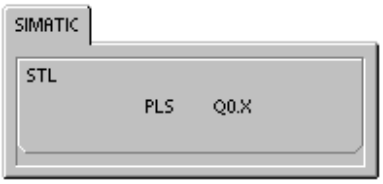

```
Network 1     // Запрограммировать HSC1:
              //1.  Сбросить текущее значение HSC1.
              //2.  Только записать новое текущее значение и
              //    оставить HSC1 активным.
LD      SM0.0
MOVD    +0, SMD48
MOVB    16#C0, SMB47
HSC     1
```

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). Для настройки импульсных выходов можно использовать мастер 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 6–30. 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 0.

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 и уловки

Программы, использующие команду PLS для функций PTO/PWM, содержатся в советах и уловках (Tips and Tricks) на компакт-диске с документацией. См. советы 7, 22, 23, 30 и 50.

Последовательность импульсов (PTO)

Функция PTO обеспечивает генерирование последовательности импульсов прямоугольной формы (с относительной длительностью 50%) с заданным количеством импульсов и заданным периодом следования импульсов. (См. рис. 6–28.) PTO 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 50 мкс до 65 535 мкс
  или
  от 2 мс до 65 535 мс.

Если при задании периода следования импульсов указывается нечетное число микросекунд или миллисекунд (напр., 75 мс), то это приводит к некоторому искажению относительной длительности импульсов.

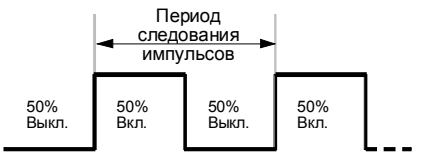


Рис. 6–28. Вывод последовательности импульсов (PTO)

Ограничения по количеству импульсов и периода следования импульсов приведены в таблице 6–31.

Таблица 6–31. Количество импульсов и периода следования импульсов в функции PTO

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 PTO 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.

Односегментная конвейерная обработка импульсов PTO

При односегментной конвейерной обработке вы несете ответственность за обновление ячеек области SM для следующей последовательности импульсов. Как только был запущен первый сегмент PTO, 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 PTO

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 16-битового значения периода следования импульсов, 16-битового значения приращения периода и 32-битового значения количества импульсов. Формат таблицы профиля показан в 6–32. 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 PTO номер активного в данный момент сегмента находится в SMB166 (или SMB176).

Таблица 6–32. 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 PTO не производится.

Широтно-импульсная модуляция (PWM)

Функция PWM обеспечивает фиксированный период следования импульсов при переменной относительной длительности импульсов. (См. рис. 6–29.) Период следования и ширину импульсов можно задавать в микро- или миллисекундах:

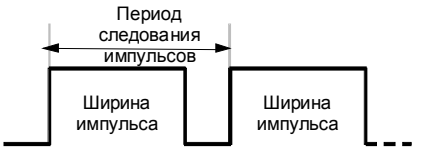


Рис. 6–29. Широтно-импульсная модуляция (PWM)

- Период следования импульсов: от 50 мкс до 65 535 мкс или от 2 мс до 65 535 мс
- Ширина импульса: от 0 мкс до 65 535 мкс или от 2 мс до 65 535 мс

Как показано в таблице 6–33, при установке ширины импульса равной периоду следования импульсов (что делает относительную длительность импульсов равной 100 процентам) выход включен постоянно. При установке ширины импульса равной 0 (что делает относительную длительность импульсов равной 0 процентов) выход выключен.

Таблица 6–33. Ширина и период следования импульсов и реакции функции PWM

Ширина импульса/период следования импульсов	Реакция
Ширина импульса >= 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 синхронно 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.

Совет
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.

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 6–34 описывает регистры, используемые для управления функционированием PTO/PWM. Вы можете использовать таблицу 6–35 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 байта (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.

Совет
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.

Таблица 6–34. Биты специальной памяти для управляющих регистров PTO / PWM

Q0.0	Q0.1	Биты состояния
SM66.4	SM76.4	Профиль PTO завершен (ошибка расчета приращения): 0 = нет ошибки 1 = завершен
SM66.5	SM76.5	Профиль PTO завершен по команде пользователя: 0 = нет завершения 1 = завершен
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 только в многосегментном режиме PTO
SMW168	SMW178	Начальный адрес таблицы профиля только в многосегментном режиме PTO (байтовое смещение от V0)

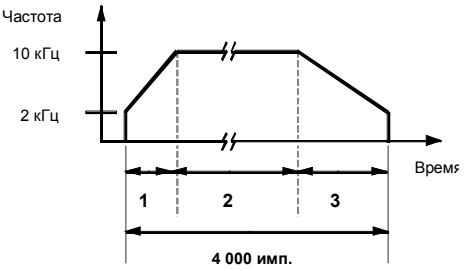
Таблица 6–35. Справочные данные об управляющем байте PTO/PWM

Управляющий регистр (16-ричное значение)		Результат выполнения операции PLS						
	Деблокировка	Функция	Сегментирование PTO	Метод обновления PWM	База времени	Кол-во импульсов	Ширина импульса	Период следования импульсов
16#81	Да	PTO	Односегм.		1 мкс/цикл			Загруж.
16#84	Да	PTO	Односегм.		1 мкс/цикл	Загружено		
16#85	Да	PTO	Односегм.		1 мкс/цикл	Загружено		Загруж.
16#89	Да	PTO	Односегм.		1 мс/цикл			Загруж.
16#8C	Да	PTO	Односегм.		1 мс/цикл	Загружено		
16#8D	Да	PTO	Односегм.		1 мс/цикл	Загружено		Загруж.
16#A0	Да	PTO	Многосегм.		1 мкс/цикл			
16#A8	Да	PTO	Многосегм.		1 мс/цикл			
16#D1	Да	PWM		Синхронный	1 мкс/цикл			Загруж.
16#D2	Да	PWM		Синхронный	1 мкс/цикл		Загружена	
16#D3	Да	PWM		Синхронный	1 мкс/цикл		Загружена	Загруж.
16#D9	Да	PWM		Синхронный	1 мс/цикл			Загруж.
16#DA	Да	PWM		Синхронный	1 мс/цикл		Загружена	
16#DB	Да	PWM		Синхронный	1 мс/цикл		Загружена	Загруж.

Расчет значений таблицы профиля

Возможность многосегментного режима конвейерной обработки генераторов PTO/PWM может быть полезной во многих приложениях, в частности, при управлении шаговыми двигателями.

Например, вы можете использовать PTO 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.



1 Сегмент 1 200 имп. **2** Сегмент 2 3400 имп. **3** Сегмент **3** 400 имп.

Рис. 6–30. Частотно-временная диаграмма

На рис. 6–30 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 PTO/PWM использует для настройки периода каждого импульса:

Приращение периода следования импульсов для сегмента =
$| End\_CT_{seg} - Init\_CT_{seg} | / Quantity_{seg}$

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 6–36 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 6–36. Значения таблицы профиля

Адрес	Значе-ние	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:

Период последнего импульса сегмента = $Init\_CT_{seg}$ + ($Delta_{seg}$ * ($Quantity_{seg}$ – 1))

где: $Init\_CT_{seg}$ = начальный период следования импульсов для этого сегмента

$Delta_{seg}$ = приращение периода следования импульсов для этого сегмента

$Quantity_{seg}$ = 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:

Длительность сегмента = $Quantity_{seg}$ * (Init_CT + (($Delta_{seg}$/2) * ($Quantity_{seg}$ – 1)))

где: $Quantity_{seg}$ = количество импульсов в данном сегменте

$Init\_CT_{seg}$ = начальный период следования импульсов для данного сегмента

$Delta_{seg}$ = приращение периода следования импульсов для данного сегмента

Пример вывода PWM

Совет
Следующее описание инициализации и последовательности функционирования PWM рекомендует использовать для инициализации вывода импульсов бит первого цикла (SM0.1). Использование бита первого цикла для вызова подпрограммы инициализации сокращает время цикла, так как в последующих циклах эта программа не вызывается. (Бит первого цикла устанавливается только в первом цикле при переходе в режим RUN.) Однако ваше приложение может иметь другие ограничения, требующие инициализации (или повторной инициализации) вывода импульсов. В этом случае вы можете использовать другое условие для вызова программы инициализации.

Инициализация вывода PWM

Обычно, чтобы инициализировать PWM для вывода импульсов, используется подпрограмма. Подпрограмма инициализации вызывается из основной программы. С помощью бита первого цикла (SM0.1) установите выход, используемый функцией PWM, в 0, и вызовите подпрограмму, в которой выполняется инициализация. Если вы используете вызов подпрограммы, то следующие циклы эту подпрограмму больше не вызывают, что сокращает время цикла и делает структуру подпрограммы более обозримой.

Сформировав в основной программе вызов подпрограммы для инициализации, действуйте следующим образом, чтобы создать в этой подпрограмме логику управления для формирования вывода импульсов на Q0.0:

1. Сформируйте управляющий байт, загрузив одно из следующих значений в SMB67: 16#D3 (для выбора микросекунд) или 16#DB (для выбора миллисекунд).

   Оба эти значения разблокируют функцию PTO/PWM, выбирают функцию PWM, устанавливают, что ширина и период следования импульсов должны обновляться, и выбирают базу времени (микросекунды или миллисекунды).
2. Загрузите значение периода следования импульсов в формате слова в SMW68.
3. Загрузите значение ширины импульса в формате слова в SMW70.
4. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM).
5. Если вы хотите предварительно загрузить новое значение управляющего байта для последующих изменений ширины импульса (не обязательно), загрузите одно из следующих значений в SMB67: 16#D2 (микросекунды) или 16#DA (миллисекунды).
6. Выйдите из подпрограммы.

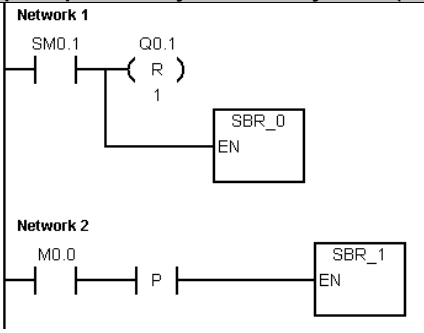
Изменение ширины импульсов для вывода PWM

Если вы предварительно загрузили SMB67 значением 16#D2 или 16#DA (см. шаг 5 выше), то вы можете изменять ширину выводимых импульсов (Q0.0) в подпрограмме. Создав вызов для этой подпрограммы, действуйте следующим образом, чтобы сформировать логику управления для изменения ширины импульсов:

1. Загрузите значение для новой ширины импульсов в формате слова в SMW70.
2. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM).
3. Выйдите из подпрограммы.

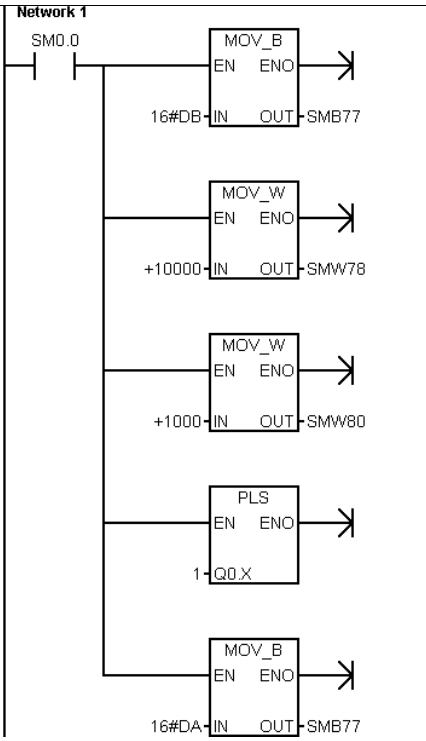
Пример: Широтно-импульсная модуляция (PWM)

MAIN



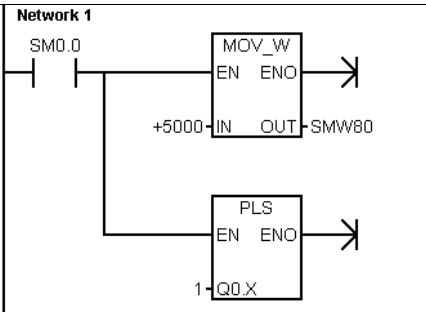

```
Network 1     //В первом цикле сбросить бит регистра
              //образа процесса и вызвать SBR_0.
LD      SM0.1
R       Q0.1, 1
CALL    SBR_0
Network 2     //Установить где-нибудь в программе M0.0,
              //чтобы перейти к относительной длительности
              //импульсов 50%.
LD      M0.0
EU
CALL    SBR_1
```

SBR 0



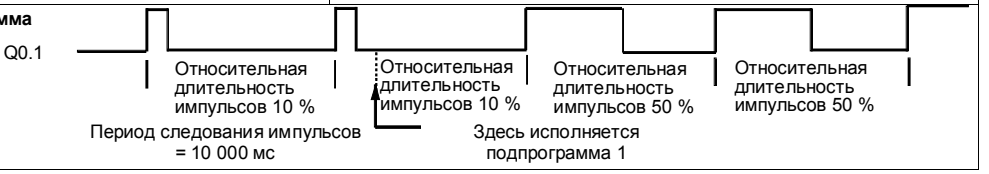

```
Network 1     //Начало подпрограммы 0:
              //1. Сформировать управляющий байт.
              //   - Выбрать функцию PWM.
              //   - Выбрать миллисекундные приращения
              //     и синхронные обновления.
              //   - Разблокировать загрузку значений
              //     ширины и периода следования импульсов.
              //   - Разблокировать функцию PWM.
              //2. Установить период следования импульсов
              //   10 000 мс.
              //3. Установить ширину импульсов 1 000 мс.
              //4. Вызвать функцию PWM: PLS1=>Q0.1.
              //5. Предварительно загрузить управляющий
              //   байт для последующих изменений
              //   ширины импульсов
LD      SM0.0
MOVB    16#DB, SMB77
MOVW    +10000, SMW78
MOVW    +1000, SMW80
PLS     1
MOVB    16#DA, SMB77
```

SBR 1




```
Network 1     //Начало подпрограммы 1:
              //Установить ширину импульсов 5 000 мс.
              //Подтвердить изменение ширины импульсов.
LD      SM0.0
MOVW    +5000, SMW80
PLS     1
```

Временная диаграмма



Пример вывода PTO

Совет
Следующее описание инициализации и последовательности функционирования PTO рекомендует использовать для инициализации вывода импульсов бит первого цикла (SM0.1). Использование бита первого цикла для вызова подпрограммы инициализации сокращает время цикла, так как в последующих циклах эта программа не вызывается. (Бит первого цикла устанавливается только в первом цикле при переходе в режим RUN.) Однако ваше приложение может иметь другие ограничения, требующие инициализации (или повторной инициализации) вывода импульсов. В этом случае вы можете использовать другое условие для вызова программы инициализации.

Инициализация вывода PTO для односегментного режима

Обычно, чтобы сконфигурировать и инициализировать PTO для вывода импульсов, используется подпрограмма. Подпрограмма инициализации вызывается из основной программы. С помощью бита первого цикла (SM0.1) установите выход, используемый функцией PTO, в 0, и вызовите подпрограмму, в которой выполняется инициализация. Если вы используете вызов подпрограммы, то следующие циклы эту подпрограмму больше не вызывают, что сокращает время цикла и делает структуру подпрограммы более обозримой.

Сформировав в основной программе вызов подпрограммы для инициализации, действуйте следующим образом, чтобы создать в этой подпрограмме логику управления для формирования вывода импульсов на Q0.0:

1. Сформируйте управляющий байт, загрузив одно из следующих значений в SMB67: 16#85 (для выбора микросекунд) или 16#8D (для выбора миллисекунд).

   Оба эти значения разблокируют функцию PTO/PWM, выбирают функцию PTO, устанавливают, что количество и период следования импульсов должны обновляться, и выбирают базу времени (микросекунды или миллисекунды).

2. Загрузите значение периода следования импульсов в формате слова в SMW68.
3. Загрузите значение для количества импульсов в формате двойного слова в SMD72.
4. (Не обязательно) Если по окончании последовательности импульсов вы хотите выполнить соответствующую операцию, вы можете запрограммировать прерывание, поставив программу обработки прерываний в соответствие событию "Последовательность импульсов завершена" (прерывающее событие 19). Используйте для этого команду ATCH и выполните затем команду ENI для разблокировки всех прерываний.
5. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM).
6. Выйдите из подпрограммы.

Изменение периода следования импульсов PTO (односегментный режим)

При односегментном режиме PTO вы можете изменять период следования импульсов в программе обработки прерываний или в подпрограмме. Для изменения периода следования импульсов в программе обработки прерываний или в подпрограмме, если вы работаете в односегментном режиме, действуйте следующим образом:

1. Сформируйте управляющий байт (чтобы разблокировать функцию PTO/PWM, выбрать функцию PTO, выбрать базу времени и установить, что период следования импульсов должен обновляться), загрузив одно из следующих значений в SMB67: 16#81 (для микросекунд) или 16#89 (для миллисекунд).
2. Загрузите новое значение периода следования импульсов в формате слова в SMW68.
3. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM). S7–200 завершает обрабатываемую в данный момент функцию PTO, прежде чем сгенерировать импульсы PTO с новым периодом следования.
4. Выйдите из программы обработки прерываний или подпрограммы.

Изменение количества импульсов PTO (односегментный режим)

При односегментном режиме PTO вы можете изменять количество импульсов в программе обработки прерываний или в подпрограмме. Для изменения количества импульсов в программе обработки прерываний или в подпрограмме, если вы работаете в односегментном режиме, действуйте следующим образом:

1. Сформируйте управляющий байт (чтобы разблокировать функцию PTO/PWM, выбрать функцию PTO, выбрать базу времени и установить, что количество импульсов должно обновляться), загрузив одно из следующих значений в SMB67: 16#84 (для микросекунд) или 16#8C (для миллисекунд).
2. Загрузите новое значение количества импульсов в формате двойного слова в SMD72.
3. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM). S7–200 завершает обрабатываемую в данный момент функцию PTO, прежде чем сгенерировать импульсы PTO с новым количеством импульсов.
4. Выйдите из программы обработки прерываний или подпрограммы.

Изменение периода следования и количества импульсов PTO (односегментный режим)

При односегментном режиме PTO вы можете изменять период следования и количество импульсов в программе обработки прерываний или в подпрограмме. Для изменения периода следования и количества импульсов в программе обработки прерываний или в подпрограмме, если вы работаете в односегментном режиме, действуйте следующим образом:

1. Сформируйте управляющий байт (чтобы разблокировать функцию PTO/PWM, выбрать функцию PTO, выбрать базу времени и установить, что период следования и количество импульсов должны обновляться), загрузив одно из следующих значений в SMB67: 16#85 (для микросекунд) или 16#8D (для миллисекунд).
2. Загрузите новое значение периода следования импульсов в формате слова в SMW68.
3. Загрузите новое значение количества импульсов в формате двойного слова в SMD72.
4. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM). S7–200 завершает обрабатываемую в данный момент функцию PTO, прежде чем сгенерировать импульсы PTO с новым количеством и периодом следования импульсов.
5. Выйдите из программы обработки прерываний или подпрограммы.

Инициализация вывода PTO для многосегментного режима

Обычно, чтобы сконфигурировать и инициализировать PTO для вывода импульсов в многосегментном режиме, используется подпрограмма. Подпрограмма инициализации вызывается из основной программы. С помощью бита первого цикла (SM0.1) установите выход, используемый функцией PTO, в 0, и вызовите подпрограмму, в которой выполняется инициализация. Если вы используете вызов подпрограммы, то следующие циклы эту подпрограмму больше не вызывают, что сокращает время цикла и делает структуру подпрограммы более обозримой.

Сформировав в основной программе вызов подпрограммы для инициализации, действуйте следующим образом, чтобы создать в этой подпрограмме логику управления для формирования вывода импульсов на Q0.0:

1. Сформируйте управляющий байт, загрузив одно из следующих значений в SMB67: 16#A0 (для выбора микросекунд) или 16#A8 (для выбора миллисекунд).

   Оба эти значения разблокируют функцию PTO/PWM, выбирают функцию PTO, выбирают многосегментный режим и выбирают базу времени (микросекунды или миллисекунды).

2. Загрузите значение начального смещения в памяти переменных для таблицы профиля в формате слова в SMW168.

3. Сформируйте значения сегментов в таблице профиля в памяти переменных. Обратите внимание на то, чтобы количество сегментов (первый байт таблицы) было указано правильно.

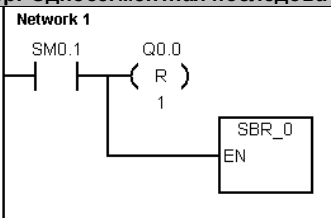
4. (Не обязательно) Если по окончании последовательности импульсов вы хотите выполнить соответствующую операцию, вы можете запрограммировать прерывание, поставив программу обработки прерываний в соответствие событию "Последовательность импульсов завершена" (прерывающее событие 19). Используйте для этого команду ATCH и выполните затем команду ENI для разблокировки всех прерываний.

5. Выполните команду PLS (чтобы S7–200 запрограммировал генератор PTO/PWM).

6. Выйдите из подпрограммы.

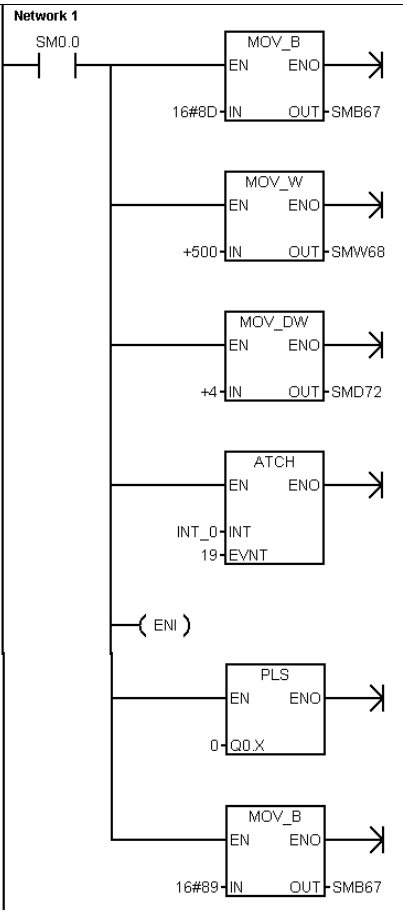
Пример: Односегментная последовательность импульсов (PTO)

MAIN



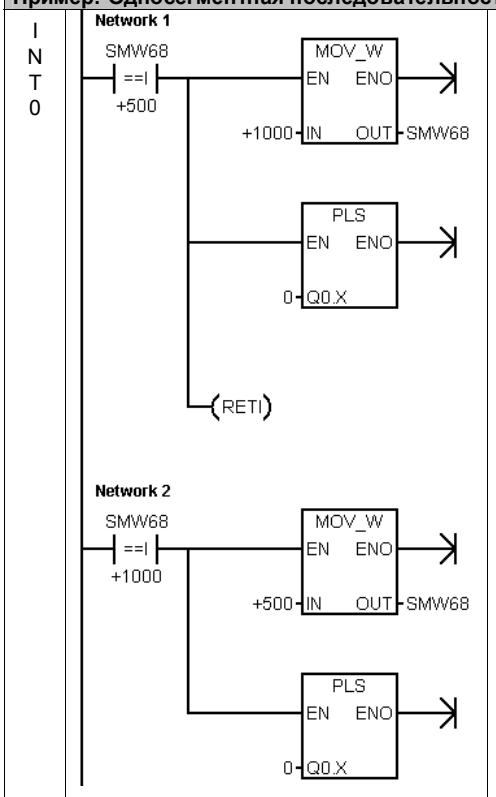

```
Network 1      //В первом цикле сбросить бит регистра
               // образа процесса и вызвать подпрограмму 0.
LD     SM0.1
R      Q0.0, 1
CALL   SBR_0
```

SBR0



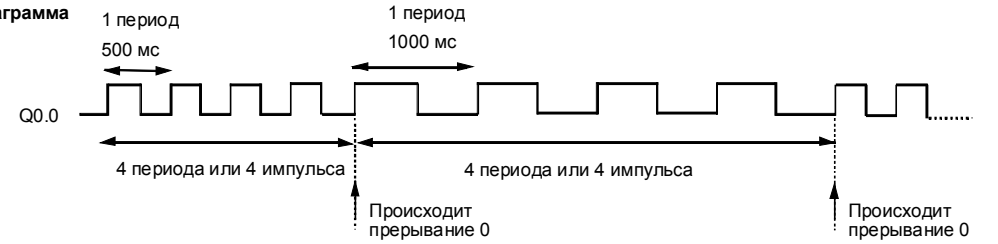

```
Network 1      //Начало подпрограммы 0: Сконфигурировать PTO
               //1. Сформировать управляющий байт:
               //   - Выбрать функцию PTO.
               //   - Выбрать односегментный режим.
               //   - Выбрать миллисекундные приращения.
               //   - Разблокировать загрузку количества
               //     и периода следования импульсов.
               //   - Разблокировать функцию PTO.
               //2. Установить период следования импульсов
               //   500 мс.
               //3. Установить количество импульсов 4.
               //4. Определить программу обработки прерываний 0
               //   как прерывание по завершению обработки PTO.
               //5. Разблокировать все прерывающие события.
               //6. Вызвать функцию PTO, PLS0 => Q0.0.
               //7. Предварительно загрузить управляющий байт
               //   для последующих изменений периода
               //   следования импульсов.
LD     SM0.0
MOVB   16#8D, SMB67
MOVW   +500, SMW68
MOVD   +4, SMD72
ATCH   INT_0, 19
ENI
PLS    0
MOVB   16#89, SMB67
```

Пример: Односегментная последовательность импульсов (PTO), продолжение



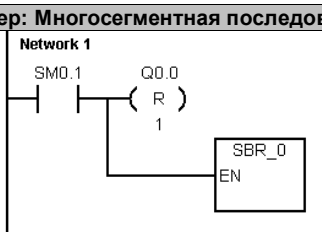

```
Network 1      //Если текущий период следования импульсов
               //равен 500 мс: установить период следования
               //импульсов равным 1000 мс и сгенерировать 4
               //импульса.
LDW=    SMW68, +500
MOVW    +1000, SMW68
PLS     0
CRETI

Network 2      //Если текущий период следования импульсов
               //равен 1000 мс: установить период следования
               //импульсов равным 500 мс и сгенерировать 4
               //импульса.
LDW=    SMW68, +1000
MOVW    +500, SMW68
PLS     0
```

Временная диаграмма



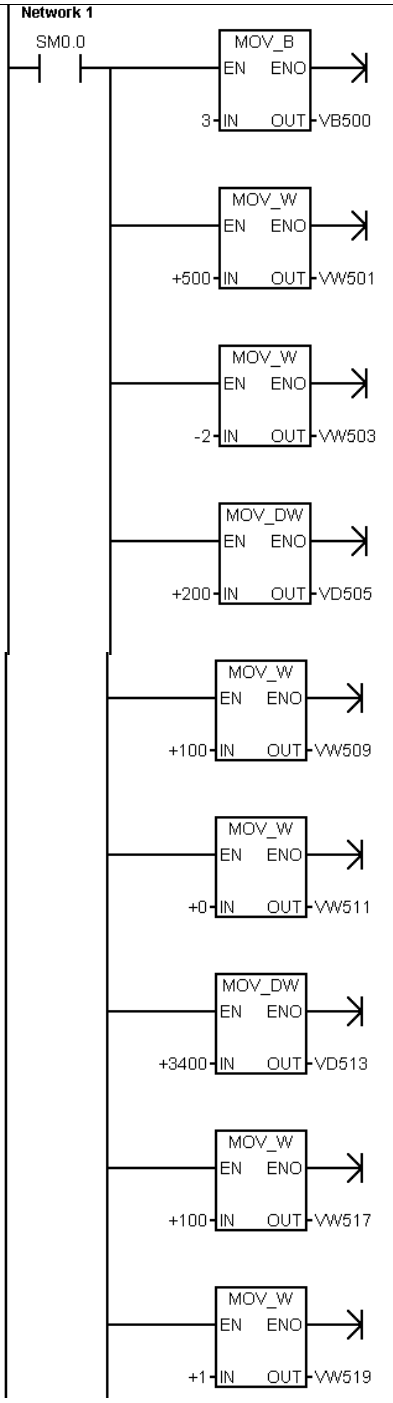
Пример: Многосегментная последовательность импульсов (PTO)

MAIN




```
Network 1      //В первом цикле сбросить бит регистра
               //образа процесса и вызвать подпрограмму 0
LD     SM0.1
R      Q0.0, 1
CALL   SBR_0
```

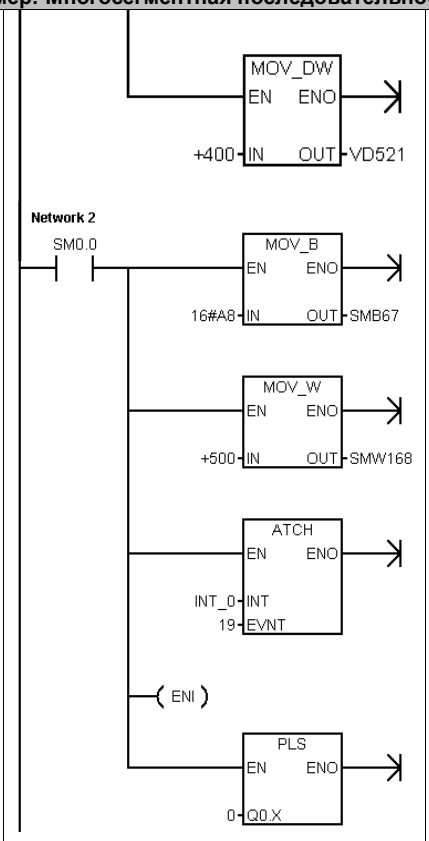
SBR0




```
Network 1      //Заранее загрузить таблицу профиля PTO:
               //Установить количество сегментов в таблице
               //профилей на 3.
               //Сконфигурировать каждый из 3 сегментов.
               //
               //1. Сконфигурировать сегмент 1:
               //   - Установить начальный период следования
               //     импульсов = 500 мс.
               //   - Установить приращение периода следования
               //     импульсов = –2 мс.
               //   - Установить количество импульсов = 200.
               //2. Сконфигурировать сегмент 2:
               //   - Установить начальный период следования
               //     импульсов = 100 мс.
               //   - Установить приращение периода следования
               //     импульсов = 0 мс.
               //   - Установить количество импульсов = 3400.
               //3. Сконфигурировать сегмент 3:
               //   - Установить начальный период следования
               //     импульсов = 100 мс.
               //   - Установить приращение периода следования
               //     импульсов = 1 мс.
               //   - Установить количество импульсов = 400.

LD     SM0.0
MOVB   3, VB500
MOVW   +500, VW501  //Сегмент 1
MOVW   -2, VW503
MOVD   +200, VD505
MOVW   +100, VW509  //Сегмент 2
MOVW   +0, VW511
MOVD   +3400, VD513
MOVW   +100, VW517  //Сегмент 3
MOVW   +1, VW519
MOVD   +400, VD521
```

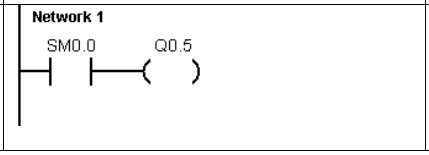
Пример: Многосегментная последовательность импульсов (PTO), продолжение

Продолжение SBR0




```
Network 2      //1. Сформировать управляющий байт:
               //   - Выбрать функцию PTO
               //   - Выбрать многосегментный режим
               //   - Выбрать миллисекундные приращения
               //   - Разблокировать функцию PTO
               //2. Установить начальный адрес таблицы профиля
               //   на V500.
               //3. Определить программу обработки прерываний 0
               //   как прерывание по завершению обработки PTO.
               //4. Разблокировать все прерывающие события
               //5. Вызвать функцию PTO, PLS0 => Q0.0.
LD     SM0.0
MOVB   16#A8, SMB67
MOVW   +500, SMW168
ATCH   INT_0, 19
ENI
PLS    0
```

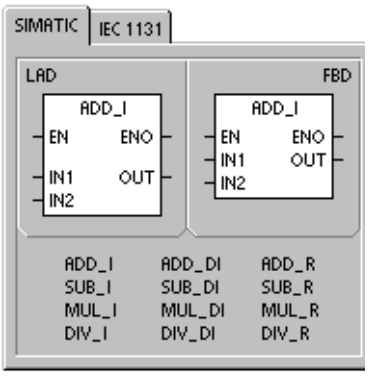
INT0

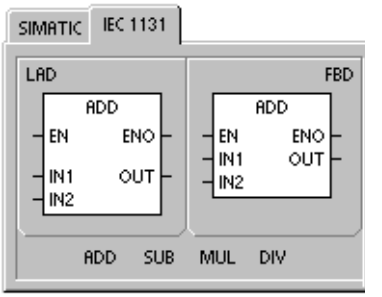


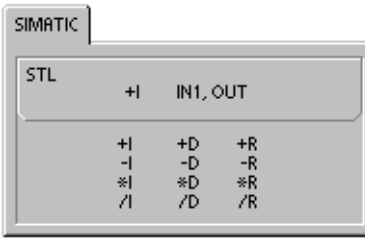

```
Network 1      //По окончании вывода профиля PTO
               //включить выход Q0.5
LD     SM0.0
=      Q0.5
```

Арифметические операции

Операции сложения, вычитания, умножения и деления







Сложение	Вычитание	
IN1 + IN2 = OUT	IN1 - IN2 = OUT	*LAD и FBD*
IN1 + OUT = OUT	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	Деление	
IN1 * IN2 = OUT	IN1 / IN2 = OUT	*LAD и FBD*
IN1 * OUT = OUT	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

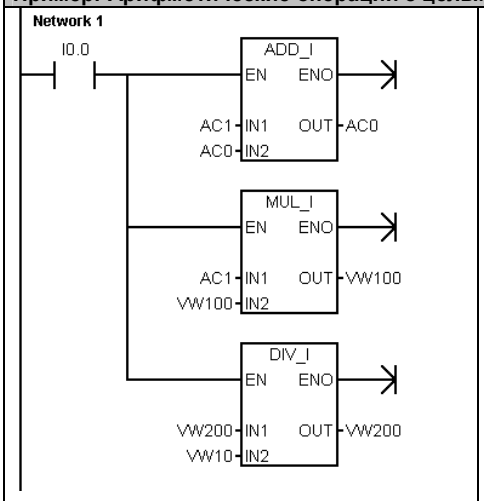
- SM1.0 (ноль)
- SM1.1 (переполнение, во время операции получено недопустимое значение или обнаружен недопустимый входной параметр)
- SM1.2 (отрицательное число)
- SM1.3 (деление на ноль)

Таблица 6–37. 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

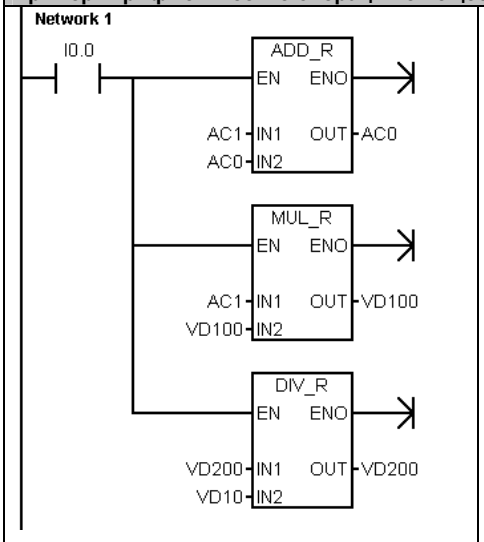


Network 1

LD	I0.0
+I	AC1, AC0
*I	AC1, VW100
/I	VW10, VW200

Сложение				Умножение				Деление			
40	+	60	= 100	40	*	20	= 800	4000	/	40	= 100
AC1		AC0	AC0	AC1		VW100	VW100	VW200		VW10	VW200

Пример: Арифметические операции с вещественными числами

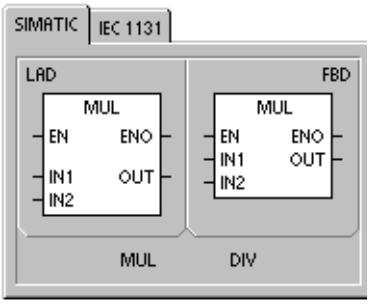


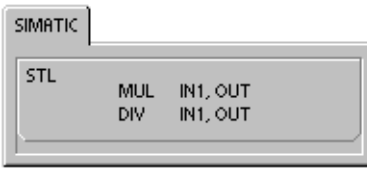
Network 1

LD	I0.0
+R	AC1, AC0
*R	AC1, VD100
/R	VD10, VD200

Сложение				Умножение				Деление			
4000.0	+	6000.0	= 10000.0	400.0	*	200.0	= 80000.0	4000.0	/	41.0	= 97.5609
AC1		AC0	AC0	AC1		VD100	VD100	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)

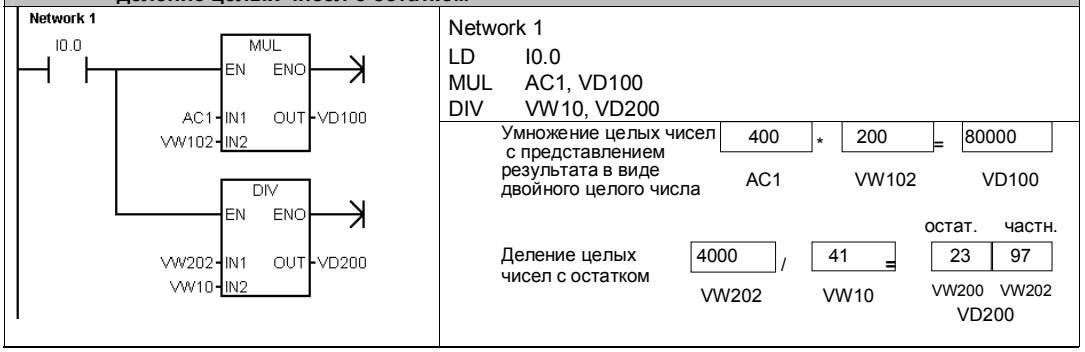
Биты специальной памяти, на которые действует команда

- SM1.0 (ноль)
- SM1.1 (переполнение)
- SM1.2 (отрицательное число)
- SM1.3 (деление на ноль)

Таблица 6–38. 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

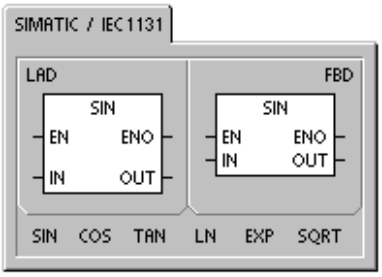
Пример: Умножение целых чисел с представлением результата в виде двойного целого числа и деление целых чисел с остатком



Примечание: VD100 содержит VW100 и VW102, а VD200 содержит VW200 и VW202.

Числовые функции

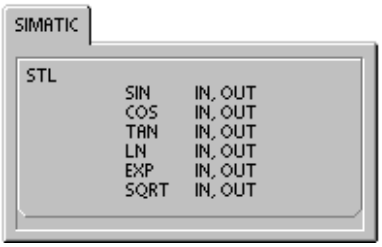
Синус, косинус и тангенс



Команды вычисления **синуса, косинуса** и **тангенса** вычисляют тригонометрическую функцию угловой величины IN и помещают результат в OUT. Входной угол задается в радианах.

SIN (IN) = OUT COS (IN) = OUT TAN (IN) = OUT

Для преобразования угла из градусов в радианы: Используйте MUL_R (*R) для умножения угла в градусах на 1.745329E-2 (примерно на π/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.

LN (IN) = OUT EXP (IN)= 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.

SQRT (IN)= OUT

Для получения других корней:

5 в кубе = 5^3 = EXP(3*LN(5)) = 125
Кубический корень из 125 = 125^(1/3) = EXP((1/3)*LN(125))= 5
Квадратный корень из 5 в кубе = 5^(3/2) = EXP(3/2*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 6–39. 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

Увеличение на 1

IN + 1 = OUT *LAD и FBD*
OUT + 1 = OUT *STL*

Уменьшение на 1

IN - 1 = OUT *LAD и 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.

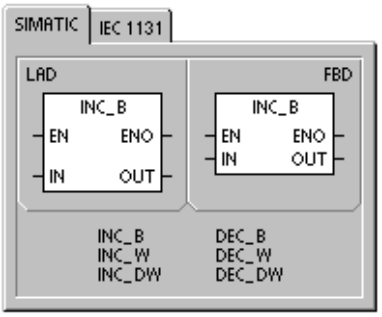
Операции увеличения двойного слова на 1 (INCD) увеличения и уменьшения двойного слова на 1 (DECD) учитывают знак.

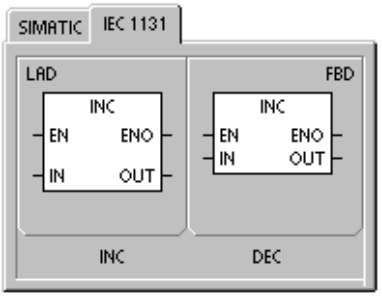
Сбойные состояния, устанавливающие ENO = 0:

- 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





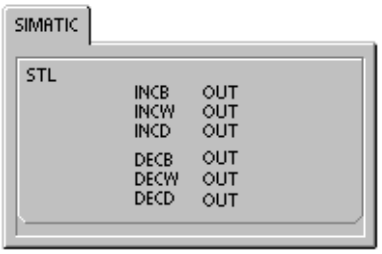
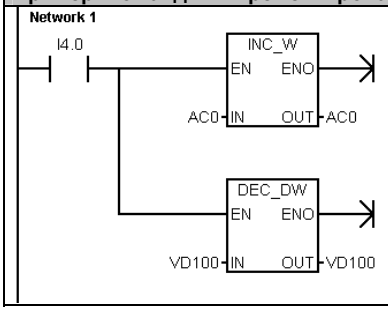


Таблица 6–40. Допустимые операнды для команд инкрементирования и декрементирования

Входы/выходы	Типы данных	Операнды
IN	BYTE INT DINT	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 INT DINT	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD 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: Команды инкрементирования и декрементирования

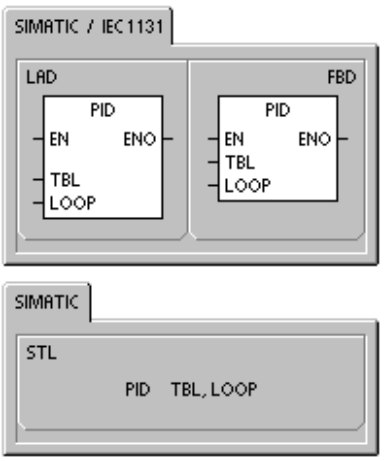



```
Network 1
LD      I4.0
INCW    AC0
DECD    VD100
```

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.

Таблица 6–41. 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 32 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].

PID-алгоритм

В установившемся 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)
- $M_{initial}$ - 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	=	$K_c * e_n$	+	$K_I * \sum_1^n e_n + M_{нач}$	+	$K_D * (e_n - e_{n-1})$
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. 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 пропорциональная 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:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_n + PV_{n-1})$$

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 физическое 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 для задания начального регулирующего воздействия при организации перехода от ручного управления к автоматическому с помощью команды PID-реггулятор (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)**	*если расчетное значение M_n > 1.0*
или					
MX	**=**	**- (MP_n + MD_n)**			*если расчетное значение M_n < 0.0*
где:	MX	- значение скорректированного смещения			
	MP_n	- значение пропорциональной составляющей регулирующего воздействия в момент квантования n			
	MD_n	- значение дифференциальной составляющей регулирующего воздействия в момент квантования n			
	M_n	- значение регулирующего воздействия в момент квантования n			

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 36 байтов и формат, показанный в таблице 6–42.

Таблица 6–42. Таблица контура регулирования

Смещение	Поле	Формат	Тип	Описание
0	Регулируемая переменная (PV_n)	Двойное слово - REAL	In	Содержит регулируемую переменную, которая должна быть масштабирована между 0.0 и 1.0.
4	Заданное значение (SP_n)	Двойное слово - REAL	In	Содержит заданное значение, которое должно быть масштабировано между 0.0 и 1.0.
8	Регулирующее воздействие (M_n)	Двойное слово - REAL	In/Out	Содержит расчетное регулирующее воздействие, масштабированное между 0.0 и 1.0.
12	Коэффициент усиления (K_C)	Двойное слово - REAL	In	Содержит усиление, являющееся коэффициентом пропорциональности. Может быть положительным или отрицательным числом.
16	Период квантования (T_S)	Двойное слово - REAL	In	Содержит период квантования в секундах. Должен быть положительным числом.
20	Постоянная времени интегрирования (T_I)	Двойное слово - REAL	In	Содержит постоянную времени интегрирования в минутах. Должна быть положительным числом.
24	Постоянная времени воздействия по производной (T_D)	Двойное слово - REAL	In	Содержит постоянную времени воздействия по производной в минутах. Должна быть положительным числом.
28	Смещение (MX)	Двойное слово - REAL	In/Out	Содержит значение смещения или интегральной суммы в пределах от 0.0 до 1.0.
32	значение регулируемой переменной (PV_{n-1})	Двойное слово - REAL	In/Out	Содержит предыдущее значение регулируемой переменной, сохраняемое после последнего выполнения команды PID-регулятор.

Пример программы с PID-регулятором

В этом примере в резервуаре с водой должно поддерживаться постоянное давление воды. Вода постоянно забирается из резервуара с различной скоростью. Для добавления воды в резервуар со скоростью, обеспечивающей надлежащее давление воды, а также не допускающей опустошения резервуара, используется насос, снабженный приводом с переменной скоростью.

Заданным значением для этой системы является установка уровня воды, эквивалентная наполнению резервуара на 75%. Регулируемая переменная поставляется поплавковым измерительным устройством, показывающим, насколько полон резервуар. Выдаваемое им значение может находиться в пределах от 0% при пустом резервуаре до 100% при полном резервуаре. Регулирующим воздействием является величина скорости насоса, которая может меняться от 0% до 100% от максимальной скорости.

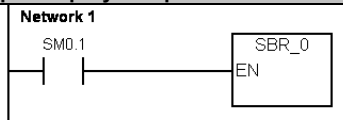
Заданное значение определяется заранее и непосредственно вводится в таблицу контура регулирования. Регулируемая переменная поставляется как однополярная аналоговая величина из поплавкового измерительного устройства. Регулирующее воздействие записывается как однополярная аналоговая величина, используемая для управления скоростью насоса. Диапазон аналогового входа и аналогового выхода равен 32 000.

В этом примере используется только пропорциональный и интегральный регулятор. Коэффициент усиления контура регулирования и постоянные времени определены из инженерных расчетов и могут быть настроены для достижения оптимального управления. Рассчитанные значения коэффициента усиления и постоянных времени: K_C = 0,25, T_S = 0,1 секунды и T_I = 30 минут.

Скорость насоса управляется вручную, пока резервуар не будет заполнен на 75%, затем открывается клапан, позволяющий воде вытекать из резервуара. Одновременно насос переключается из ручного режима в автоматический. Для переключения из ручного режима в автоматический используется цифровой вход. Этот вход (I0.0) обеспечивает переключение между ручным и автоматическим режимом: 0 = ручной режим, а 1 = автоматический режим. В ручном режиме скорость насоса записывается оператором в VD108 как вещественное число в диапазоне от 0.0 до 1.0.

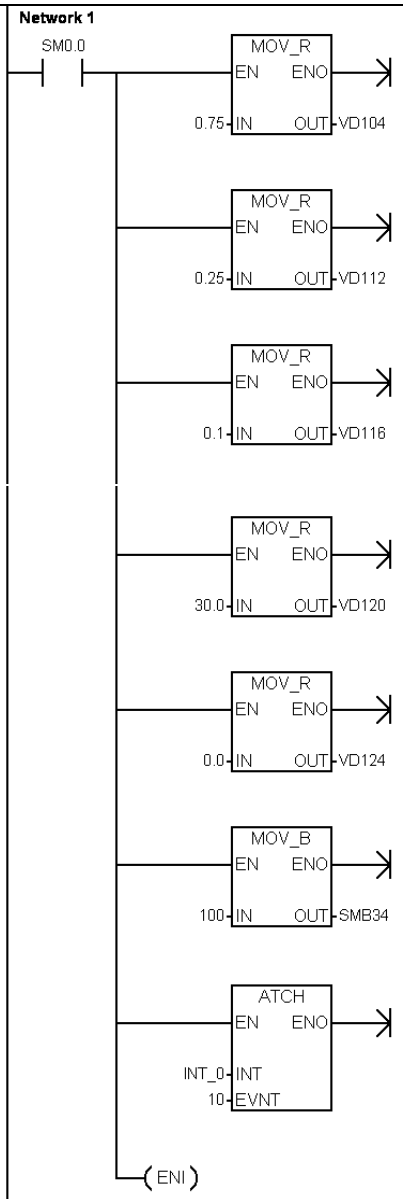
Пример: PID-регулятор

MAIN




```
Network 1      //В первом цикле
               //вызвать подпрограмму инициализации
LD      SM0.1
CALL    SBR_0
```

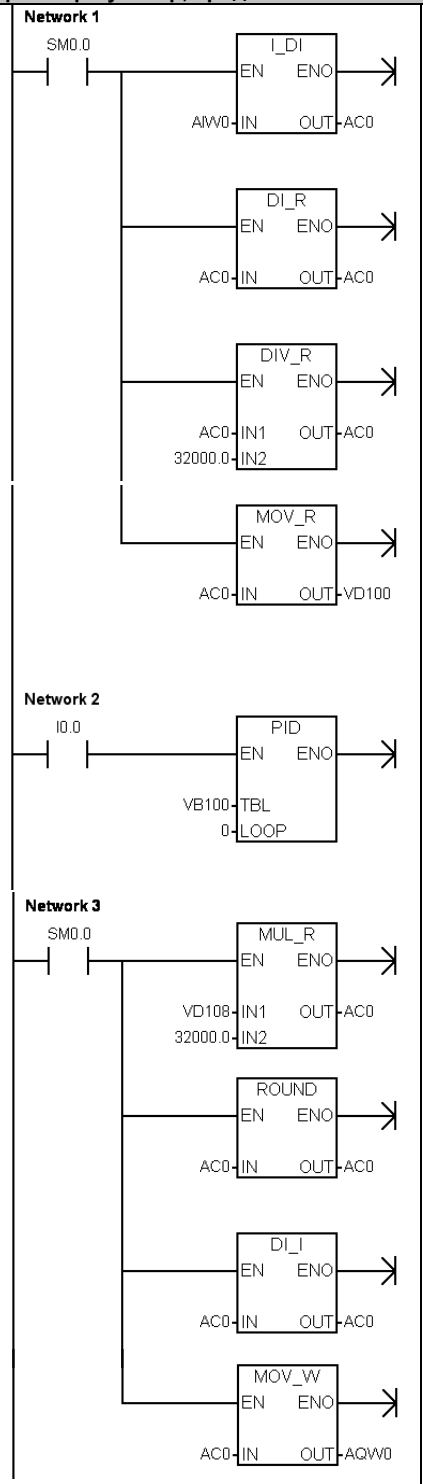
SBR 0




```
Network 1 //Загрузить параметры PID и назначить
          //программу обработки прерываний PID:
          //1. Загрузить заданное значение = 75%
          //   емкости резервуара.
          //2. Загрузить коэффициент усиления = 0,25.
          //3. Загрузить период квантования = 0,1 с.
          //4. Загрузить постоянную времени
          //   интегрирования = 30 мин.
          //5. Исключить действие по производной.
          //6. Установить интервал (100 мс) для
          //   управляемого временем прерывания
          //   INT_0.
          //7. Вызывать PID-регулятор с помощью
          //   управляемого временем прерывания.
          //8. Разблокировать прерывания
LD      SM0.0
MOVR    0.75, VD104
MOVR    0.25, VD112
MOVR    0.1, VD116
MOVR    30.0, VD120
MOVR    0.0, VD124
MOVB    100, SMB34
ATCH    INT_0, 10
ENI
```

Пример: PID-регулятор, продолжение

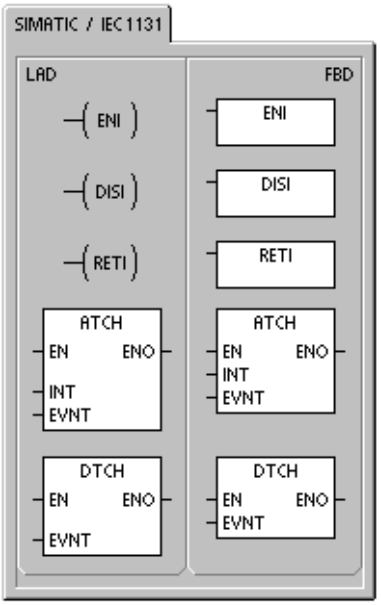
INT0




```
Network 1 //Преобразовать PV в нормализованное
          //вещественное число:
          //1. Преобразовать целое в двойное целое.
          //2. Преобразовать двойное целое в
          //   вещественное число.
          //3. Нормализовать это значение.
          //4. Сохранить нормализованное значение PV
          //   в таблице контура регулирования.
LD      SM0.0
ITD     AIW0, AC0
DTR     AC0, AC0
/R      32000.0, AC0
MOVR    AC0, VD100

Network 2 //Выполнить регулирование при включенном
          //автоматическом режиме.
LD      I0.0
PID     VB100, 0

Network 3 //Преобразовать регулирующее
          //воздействие Mn в целое число.
          //Mn – униполярная величина, которая не
          //может быть отрицательной.
          //1. Передать регулирующее воздействие
          //   в аккумулятор.
          //2. Масштабировать значение в
          //   аккумуляторе.
          //3. Преобразовать вещественное число в
          //   двойное целое.
          //4. Преобразовать двойное целое в
          //   целое.
          //5. Записать это значение в аналоговый
          //   выход.
LD      SM0.0
MOVR    VD108, AC0
*R      32000.0, AC0
ROUND   AC0, AC0
DTI     AC0, AC0
MOVW    AC0, AQW0
```

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.

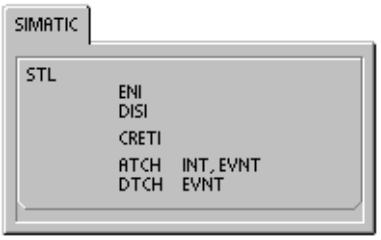
Когда вы переходите в режим RUN, прерывания первоначально заблокированы. Находясь в режиме RUN, вы можете разблокировать все прерывания, выполнив команду разблокирования прерываний. Выполнение команды блокирования прерываний запрещает обработку прерываний, однако активные прерывающие события и далее 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 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.

Таблица 6–43. Допустимые операнды для команд назначения и отсоединения прерываний

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 226 и CPU 226XM:* 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 (DTCH). Команда отсоединения возвращает прерывание в неактивное или игнорируемое состояние. Таблица 6–44 перечисляет различные типы прерывающих событий.

Таблица 6–44. Прерывающие события

Событие	Описание		CPU 221 CPU 222	CPU 224	CPU 226 CPU 226XM
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=PV (текущее значение = предустановленному)	Да	Да	Да
13	HSC1	CV=PV (текущее значение = предустановленному)		Да	Да
14	HSC1	Изменение направления		Да	Да
15	HSC1	Внешний сброс		Да	Да
16	HSC2	CV=PV (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 CT=PT	Да	Да	Да
22	Таймер T96	Прерывание CT=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
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 исполняется 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 6–45 приведены некоторые указания и ограничения по использованию программ обработки прерываний в вашей программе.

Таблица 6–45. 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 препятствую прерыванию последовательностей команд, использующих разделяемые ячейки памяти.

- 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 препятствую прерыванию последовательностей команд, использующих разделяемые ячейки памяти.
- 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 правильный 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 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 на 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 заполнится 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 исполняется, 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 6–46.

Таблица 6–46. Максимальное количество записей на очередь прерываний

Очередь	CPU 221, CPU 222, CPU 224	CPU 226 и CPU 226XM
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 6–47. Вам следует использовать эти биты только в программе обработки прерывания, так как они сбрасываются, когда очередь опустошается и управление возвращается главной программе.

В табл. 6–48 представлены все прерывающие события, их приоритеты и соответствующие номера событий.

Таблица 6–47. 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 6–48. Порядок приоритетов для прерывающих событий

Событие	Описание		Группа приоритета	Приоритет в группе
8	Порт 0	Символ принят	Коммуникации *Наивысший приоритет*	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 при завершении PTO	Дискретные прерывания *Средний приоритет*	0
20	PLS1	Прерывание при завершении PTO		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	Прерывание, управляемое временем 0	SMB34	Прерывания, управляемые временем *Низший приоритет*	0
11	Прерывание, управляемое временем 1	SMB35		1
21	Таймер T32	Прерывание CT=PT		2
22	Таймер T96	Прерывание CT=PT		3

Пример: Прерывания		

M A I N	Network 1
	SM0.1 — ATCH (EN, ENO); INT_0 – INT; 1 – EVNT
	(ENI)
	Network 2
	SM5.0 — DTCH (EN, ENO); 1 – EVNT
	Network 3
	M5.0 — (DISI)
		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.
		LD SM0.1
		ATCH INT_0, 1
		ENI
		Network 2 //Если обнаружена ошибка ввода/ввода,
		//заблокировать прерывание по падающему фронту на I0.0.
		//Этот сегмент не обязателен.
		LD SM5.0
		DTCH 1
		Network 3 //Если M5.0 установлен,
		//заблокировать все прерывания.
		LD M5.0
		DISI

I N T 0	Network 1
	SM5.0 — (RETI)
		Network 1 //Программа обработки прерываний по падающему фронту
		//на I0.0:
		//Условный возврат при ошибке ввода/вывода.
		LD SM5.0
		CRETI

Пример: Прерывание, управляемое временем, для чтения значения управляемого входа		
M A I N	Network 1 SM0.1 — SBR_0 (EN)	Network 1 // В первом цикле вызвать подпрограмму 0. LD SM0.1 CALL SBR_0

S B R 0	Network 1
	SM0.0 — MOV_B (EN, ENO); 100 – IN; OUT – SMB34
	ATCH (EN, ENO); INT_0 – INT; 10 – EVNT
	(ENI)
		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.
		LD SM0.0
		MOVB 100, SMB34
		ATCH INT_0, 10
		ENI

INT 0	Network 1 SM0.0 — MOV_W (EN, ENO); AIW4 – IN; OUT – VW100	Network 1 //считывать значение AIW4 каждые 100 мс LD SM0.0 MOVW AIW4, VW100

Логические операции

Операции инвертирования

Инвертирование байта, слова и двойного слова

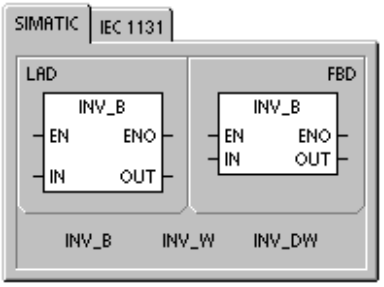
Команды инвертирования байта (INVB), слова (INVW) и двойного слова (INVD) образуют дополнение входа IN до единицы и загружают результат по адресу OUT.

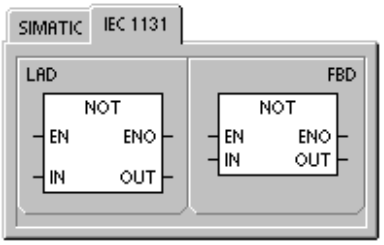
Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Связанные с командой биты специальной памяти:

- SM1.0 (ноль)





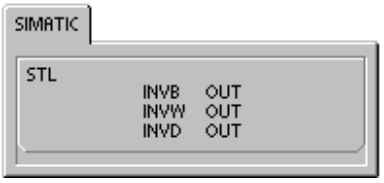
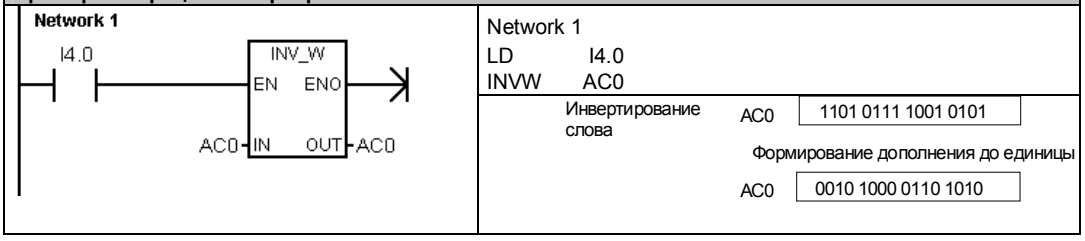


Таблица 6–49. Допустимые операнды для команд инвертирования

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

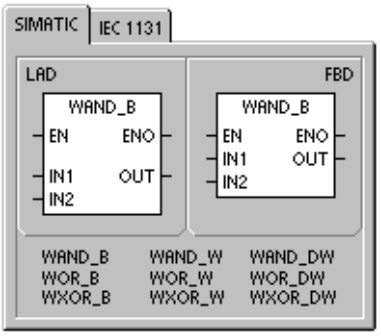
Пример: Операция инвертирования



Поразрядные логические операции И, ИЛИ и исключающее ИЛИ

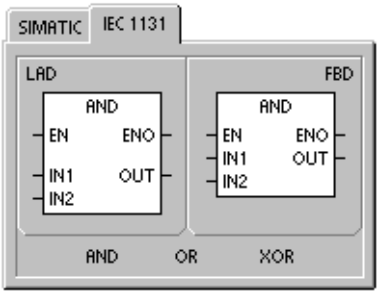
Поразрядное логическое И с байтами, словами и двойными словами

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.



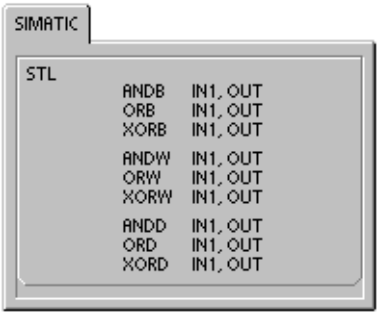
Поразрядное логическое ИЛИ с байтами, словами и двойными словами

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)

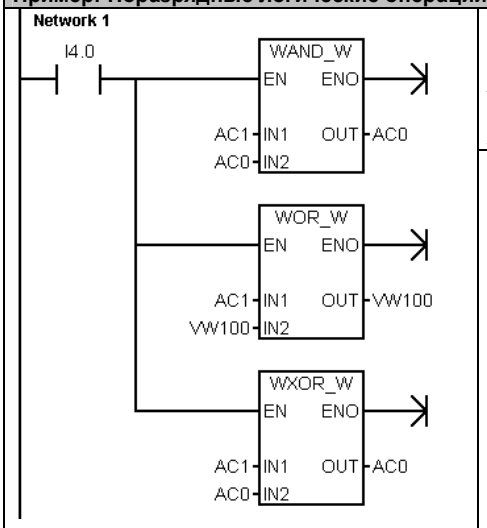
Связанные с командой биты специальной памяти:

- SM1.0 (ноль)

Таблица 6–50. 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

Пример: Поразрядные логические операции И, ИЛИ и исключающее ИЛИ



Network 1

LD	I4.0
ANDW	AC1, AC0
ORW	AC1, VW100
XORW	AC1, AC0

Поразрядное И

AC1 0001 1111 0110 1101

AND

AC0 1101 0011 1110 0110

равно

AC0 0001 0011 0110 0100

Поразрядное ИЛИ

AC1 0001 1111 0110 1101

OR

VW100 1101 0011 1010 0000

равно

VW100 1101 1111 1110 1101

Поразрядное Исключающее ИЛИ

AC1 0001 1111 0110 1101

XOR

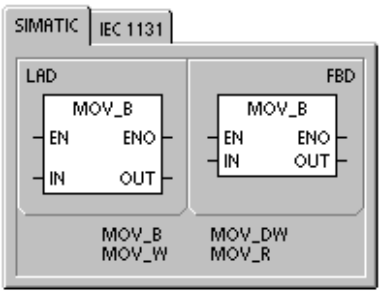
AC0 0001 0011 0110 0100

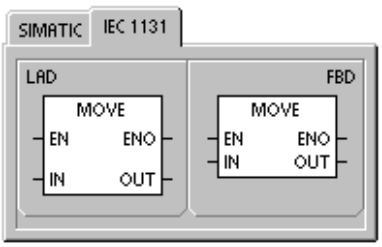
равно

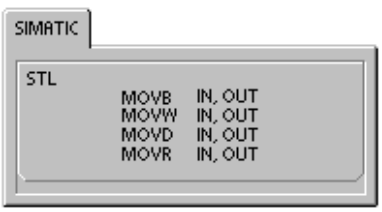
AC0 0000 1100 0000 1001

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.

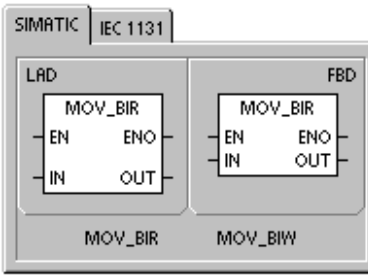
Сбойные состояния, устанавливающие ENO = 0

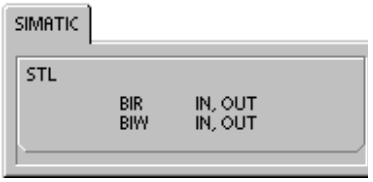
- 0006 (косвенный адрес)

Таблица 6–51. 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, AC, HC, &IB, &QB, &VB, &MB, &SB, &T, &C, *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 6–52. 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 6–53. 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

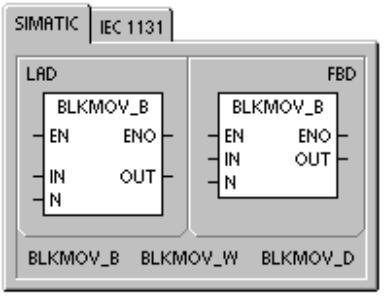
Групповая пересылка байтов, слов, двойных слов

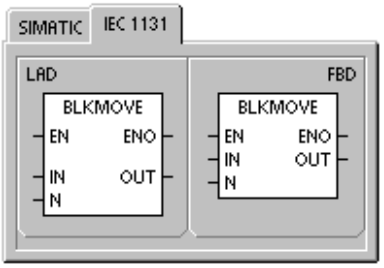
Команды групповой пересылки байтов (BMB), слов (BMW) и двойных слов (BMD) пересылают указанное количество (N) байтов, слов или двойных слов, начиная с входного адреса IN, в новую область, начинающуюся с выходного адреса OUT.

N находится в диапазоне от 1 до 255.

Сбойные состояния, устанавливающие ENO = 0

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





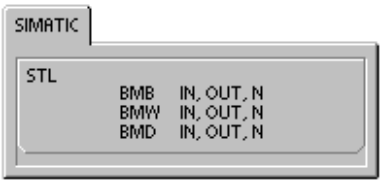
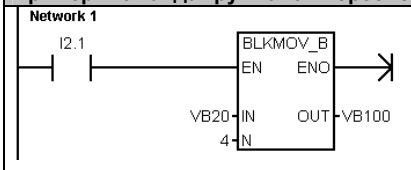


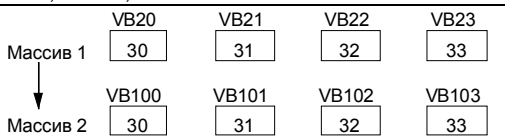
Таблица 6–54. Допустимые операнды для команд групповой пересылки

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



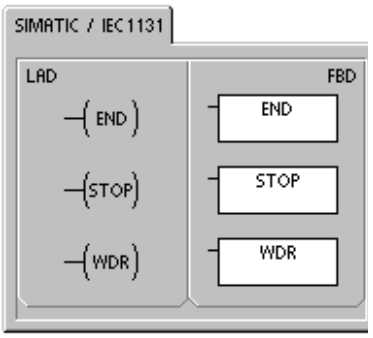

```
Network 1      //Переслать массив 1 (от VB20 до VB23)
               //в массив 2 (от VB100 до VB103)
LD     I2.1
BMB    VB20, VB100, 4
```



Команды управления программой

Условное завершение

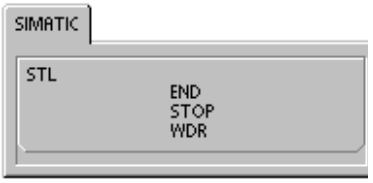
Команда условного завершения (END) завершает текущий цикл в зависимости от результата предшествующей логической операции. Вы можете использовать команду условного завершения в главной программе, но ее нельзя использовать в подпрограммах и программах обработки прерываний.



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 Freeport)
- 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.

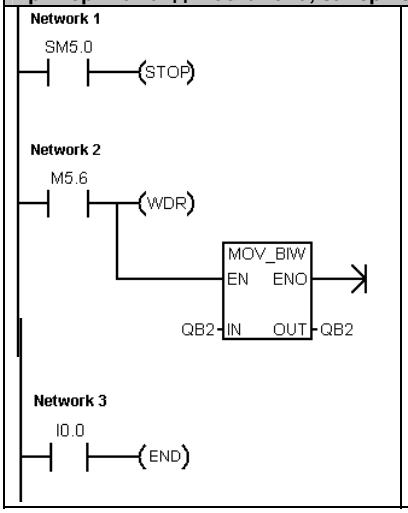
Совет

Команду сброса времени контроля 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 времени контроля, вы должны также использовать команду непосредственной записи для одного выходного байта (QB) на каждый цифровой модуль вывода, чтобы сбросить контроль времени модуля расширения.

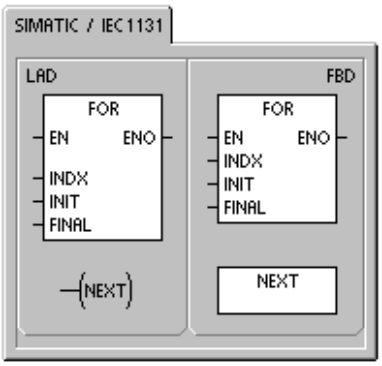
Если вы используете команду сброса времени контроля, чтобы можно было выполнить программу с большим временем цикла, то перевод переключателя режимов работы в положение STOP вызывает переход S7–200 в состояние STOP в течение 1,4 секунды.

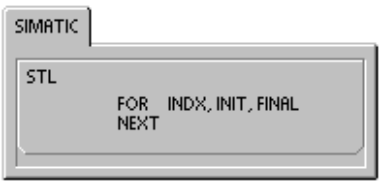
Пример: Команды останова, завершения и сброса времени контроля




```
Network 1      //При обнаружении ошибки ввода/вывода
               //перейти принудительно в режим STOP.
LD     SM5.0
STOP

Network 2      //Если M5.6 включен, сделать возможным
               //продление цикла:
               //1. Сбросить время контроля для S7–200.
               //2. Сбросить контроль времени для первого
               //   модуля вывода.
LD     M5.6
WDR
BIW    QB2, QB2

Network 3      //Если I0.0 включен, завершить текущий цикл.
LD     I0.0
END
```

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 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 выполнят 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.

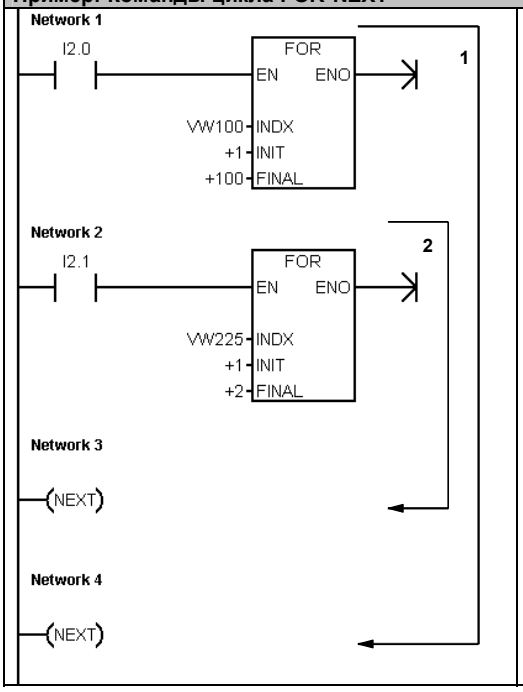
Если начальное значение больше конечного, то цикл не выполняется. После каждого исполнения команд между FOR и NEXT значение INDX увеличивается, а результат сравнивается с конечным значением. Если INDX больше конечного значения, то цикл завершается.

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 6–55. 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

Пример: Команды цикла FOR-NEXT




```
Network 1      //Когда включается I2.0, внешний цикл
               //(стрелка 1) выполняется 100 раз
LD     I2.0
FOR    VW100, +1, +100

Network 2      //Внутренний цикл (стрелка 2)
               //выполняется дважды для каждого
               //исполнения внешнего цикла,
               //если включен I2.1.
LD     I2.1
FOR    VW225, +1, +2

Network 3      //Конец цикла 2.
NEXT

Network 4      // Конец цикла 1 .
NEXT
```

Команды перехода

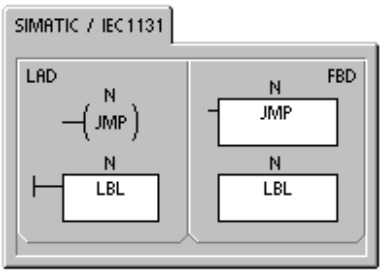
Команда перехода на метку (JMP) осуществляет переход к указанной метке N внутри программы.

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.



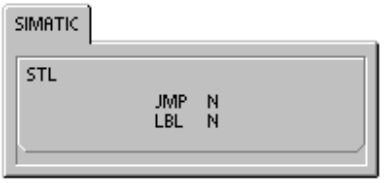
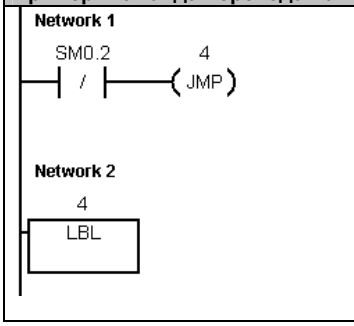
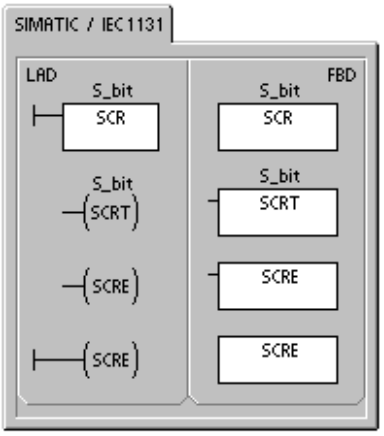


Таблица 6–56. Допустимые операнды для команд перехода		
Входы/выходы	**Типы данных**	**Операнды**
N	WORD	Константа (от 0 до 255)

Пример: Команда перехода на метку	
	Network 1 //Если сохраняемые данные не потеряны, перейти на //метку LBL 4 LDN SM0.2 JMP 4 Network 2 LBL 4

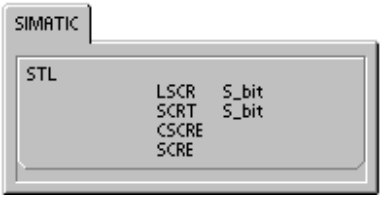
Команды для реле управления очередностью (SCR)



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 SCR. 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 6–57. Допустимые операнды для команд реле управления очередностью

Входы/выходы	Типы данных	Операнды
S_bit	BOOL	S

На рис. 6–31 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.

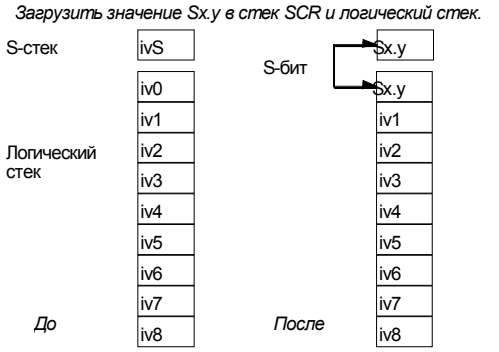
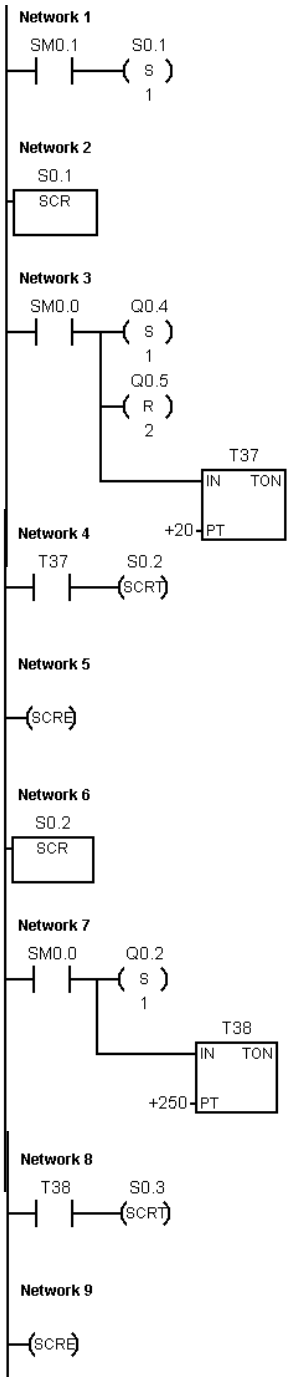


Рис. 6–31. Влияние LSCR на логический стек

- 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).

Пример: Операции для реле управления очередностью




```
Network 1       //В первом цикле активизировать состояние 1.
LD      SM0.1
S       S0.1, 1

Network 2       //Начало области управления для состояния 1.
LSCR    S0.1

Network 3       //Управление сигналами для улицы 1:
                //1. Установка: Включить красный свет.
                //2. Сброс: Выключить желтый свет и включить
                //   зеленый.
                //3. Запустить 2-секундный таймер.
LD      SM0.0
S       Q0.4, 1
R       Q0.5, 2
TON     T37, +20

Network 4       //После 2-секундной задержки переход к
                //состоянию 2.
LD      T37
SCRT    S0.2

Network 5       //Конец области действия SCR для состояния 1.
SCRE

Network 6       //Начало области управления для состояния 2.
LSCR    S0.2

Network 7       //Начало области управления для состояния 2:
                //1. Установка: Включить зеленый свет.
                //2. Запустить 25-секундный таймер.
LD      SM0.0
S       Q0.2, 1
TON     T38, +250

Network 8       //После 25-секундной задержки переход к
                //состоянию 3.
LD      T38
SCRT    S0.3

Network 9       //Конец области действия SCR для состояния 2.
SCRE
```

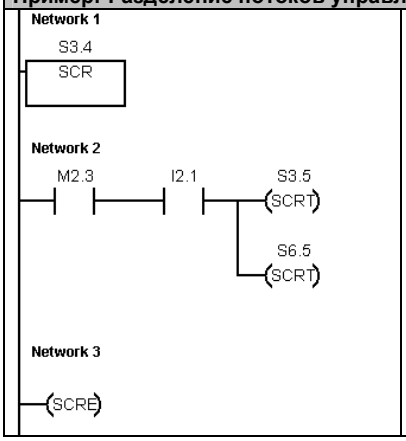
Разделение потоков управления

Во многих приложениях единый поток последовательных состояний должен быть разделен на два или более различных потоков. Когда поток управления разделяется на несколько потоков, все выходящие потоки должны активизироваться одновременно. Это показано на рис. 6–32.

Рис. 6–32. Разделение потока управления

Разделение потоков управления может быть реализовано в программе SCR путем использования нескольких команд SCRT, активизируемых одним и тем же условием перехода, как показано в следующем примере.

Пример: Разделение потоков управления




```
Network 1      //Начало области управления для состояния L.
LSCR     S3.4

Network 2
LD       M2.3
A        I2.1
SCRT     S3.5          //Переход к состоянию M
SCRT     S6.5          //Переход к состоянию N

Network 3      //Конец области SCR для состояния L.
SCRE
```

Слияние потоков управления

Ситуация, аналогичная разделению потоков управления, возникает, когда два или более потоков последовательных состояний должны быть объединены в один поток. Когда несколько потоков объединяются в один поток, говорят, что они сливаются. Когда несколько потоков сливаются в один поток, все входящие потоки должны быть завершены до того, как будет выполняться следующее состояние. На рис. 6–33 изображено слияние двух потоков управления.

Слияние потоков управления может быть реализовано в программе SCR путем создания перехода от состояния L к состоянию L’ и перехода от состояния M к состоянию M’. Когда биты SCR, представляющие L’ и M’, установлены, состояние N может быть активизировано, как показано в следующем примере.

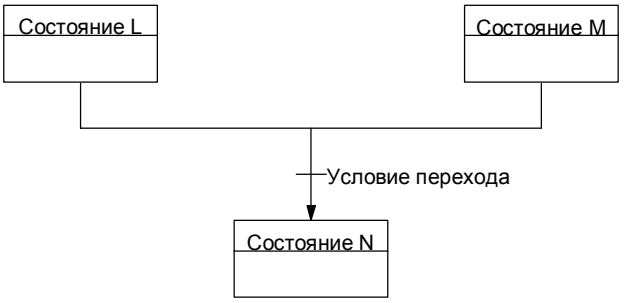
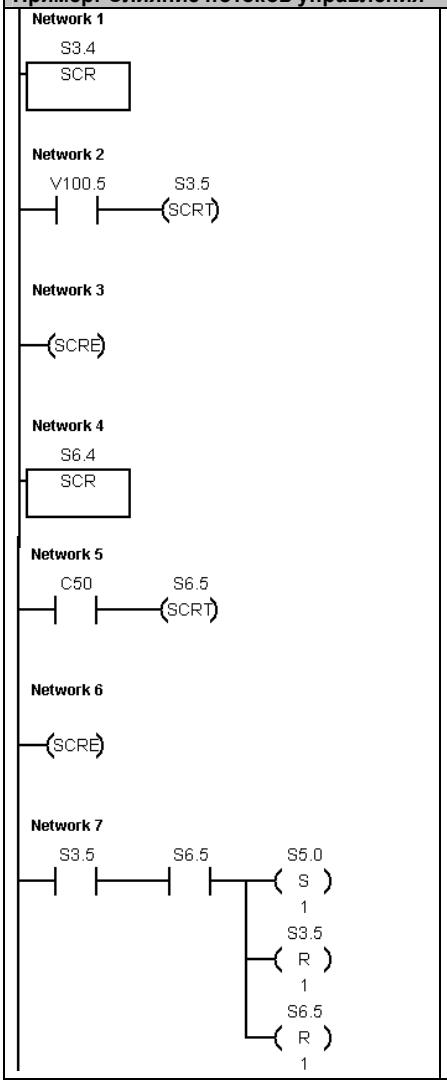


Рис. 6–33. Слияние потоков управления

Пример: Слияние потоков управления

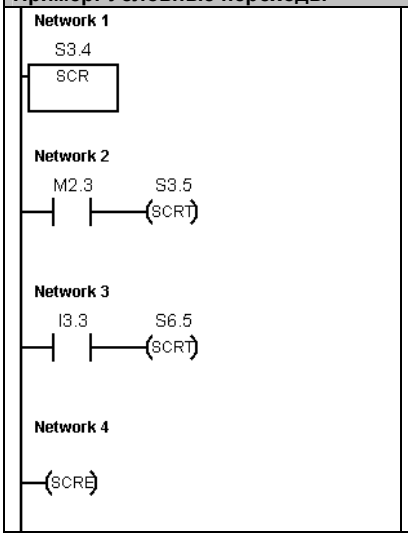



```
Network 1       // Начало области управления для состояния L
LSCR    S3.4

Network 2       //Переход к состоянию L'
LD      V100.5
SCRT    S3.5

Network 3       //Конец области SCR для состояния L
SCRE

Network 4       //Начало области управления для состояния M
LSCR    S6.4

Network 5       //Переход к состоянию M'
LD      C50
SCRT    S6.5

Network 6       //Конец области SCR для состояния M
SCRE

Network 7       //Когда активизированы и состояние L', и
                //состояние M':
                //1. Активизировать состояние N (S5.0)
                //2. Сбросить состояние L' (S3.5)
                //3. Сбросить состояние M' (S6.5)
LD      S3.5
A       S6.5
S       S5.0, 1
R       S3.5, 1
R       S6.5, 1
```

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. 6–34, где представлена эквивалентная программа SCR.

Рис. 6–34. Разделение потока управления в зависимости от условия перехода

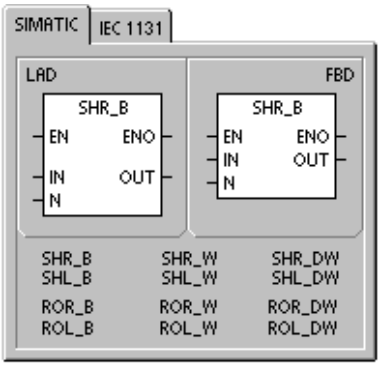
Пример: Условные переходы




```
Network 1      //Начало области управления для состояния L
LSCR     S3.4

Network 2      //Переход к состоянию M
LD       M2.3
SCRT     S3.5

Network 3      //Переход к состоянию N
LD       I3.3
SCRT     S6.5

Network 4      //Конец области SCR для состояния L
SCRE
```

Команды сдвига и циклического сдвига

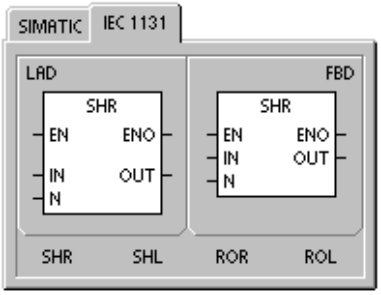
Команды сдвига вправо и сдвига влево



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.



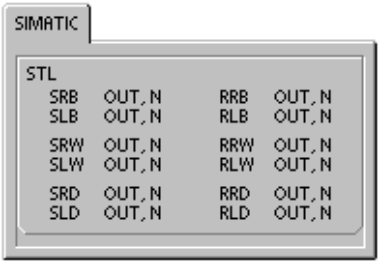
Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

Связанные с командой биты специальной памяти:

- 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)

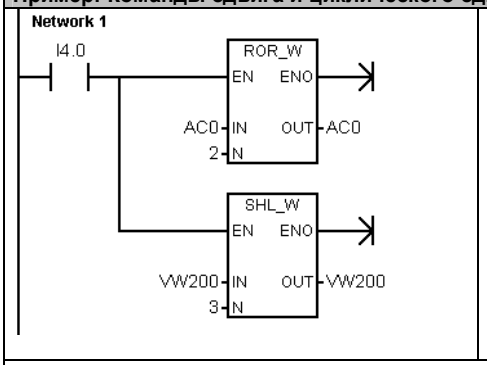
Связанные с командой биты специальной памяти:

- SM1.0 (ноль)
- SM1.1 (переполнение)

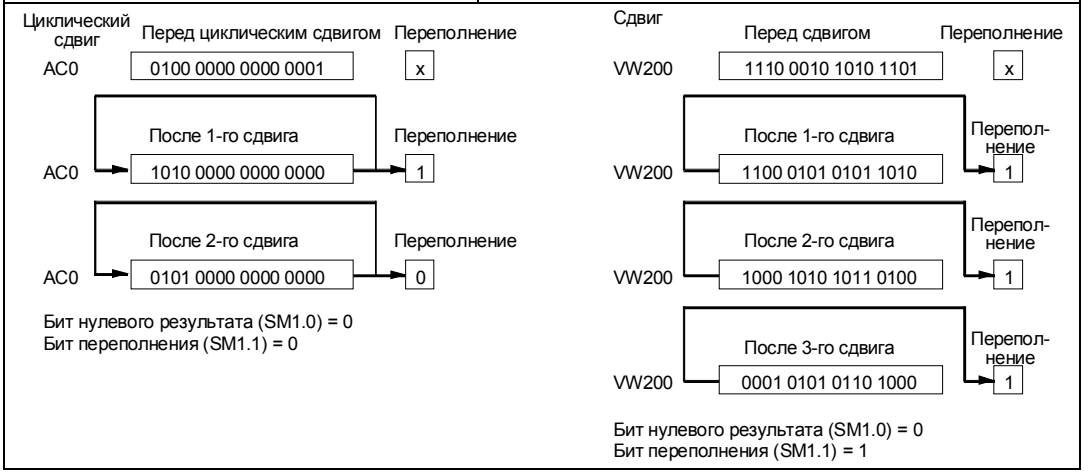
Таблица 6–58. Допустимые операнды для команд сдвига и циклического сдвига

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
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа

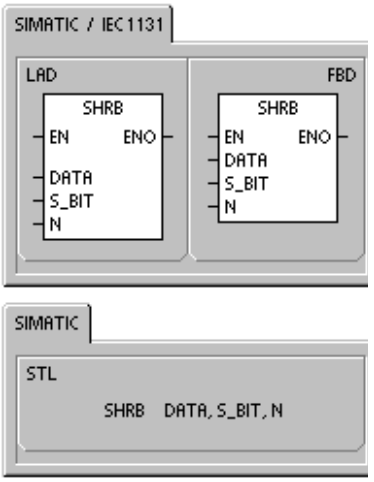
Пример: Команды сдвига и циклического сдвига




```
Network 1
LD      I4.0
RRW     AC0, 2
SLW     VW200, 3
```



Команда перемещения значения бита в регистр сдвига



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)

Связанные с командой биты специальной памяти:

- SM1.1 (переполнение)

Таблица 6–59. 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 = *[(байт параметра S_BIT) + ([N] - 1 + (бит параметра S_BIT)) / 8] . [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.

MSB.b = V33 + ([14] - 1 +4)/8
= V33 + 17/8
= V33 + 2 с остатком 1
= V35.1

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. 6–35 показан сдвиг битов при отрицательном и положительном значении N.

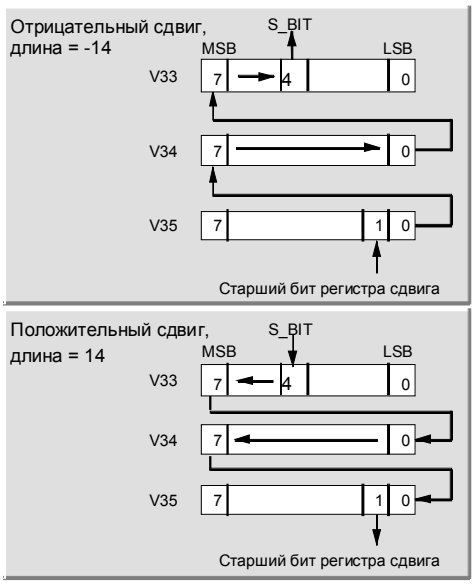
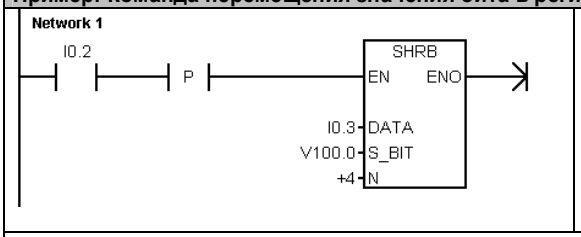


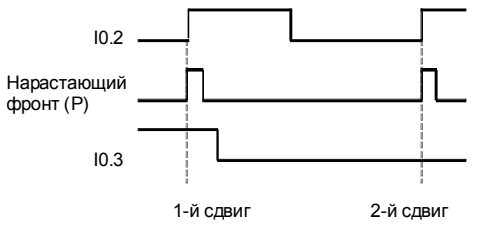
Рис. 6–35. Ввод и удаление данных из регистра сдвига

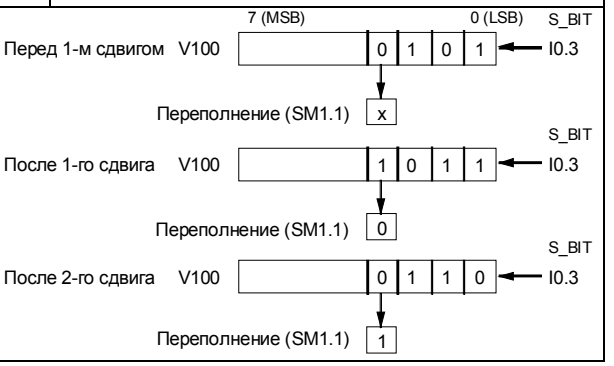
Пример: Команда перемещения значения бита в регистр сдвига




```
Network 1
LD      I0.2
EU
SHRB    I0.3, V100.0, +4
```

Временная диаграмма





Команда перестановки байтов в слове

Команда перестановки байтов меняет местами старший и младший байты слова IN.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)

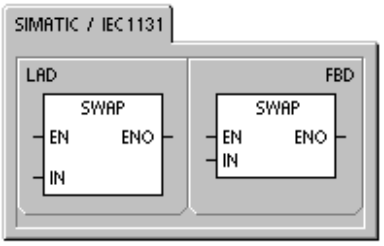
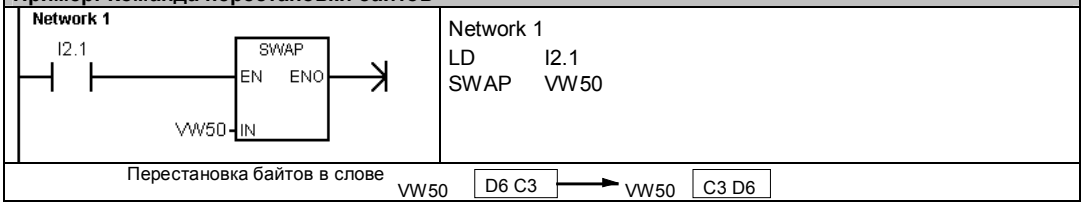


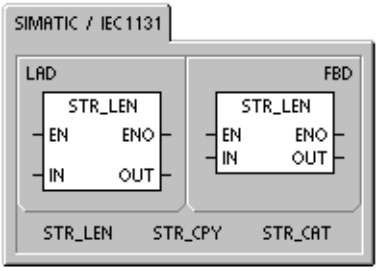
Таблица 6–60. Допустимые операнды для команды перестановки байтов

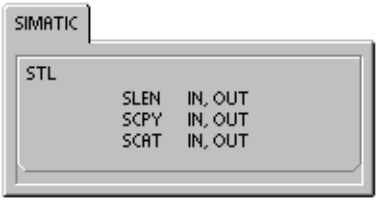
Входы/выходы	Типы данных	Операнды
IN	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW,AC, *VD, *LD, *AC

Пример: Команда перестановки байтов



Строковые команды





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 оказывают воздействие на ENO.

Сбойные состояния, устанавливающие ENO = 0

- 0006 (косвенный адрес)
- 0091 (ошибка области)

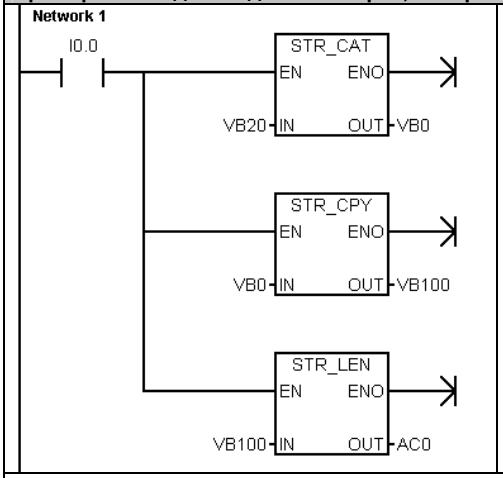
Таблица 6–61. Допустимые операнды для команды определения длины строки

Входы/выходы	Типы данных	Операнды
IN	BYTE (строка)	VB, LB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Таблица 6–62. Допустимые операнды для команд копирования строки и соединения строк

Входы/выходы	Типы данных	Операнды
IN, OUT	BYTE (строка)	VB, LB, *VD, *LD, *AC

Пример: Команды соединения строк, копирования строки и определения длины строки




```
Network 1 //1. Присоединить строку с адресом VB20
          //   к строке с адресом VB0
          //2. Скопировать строку с адресом VB0
          //   в новую строку с адресом VB100
          //3. Получить длину строки, начинающейся
          //   по адресу VB100
LD        I0.0
SCAT      VB20, VB0
STRCPY    VB0, VB100
STRLEN    VB100, AC0
```

Перед выполнением программы

VB0						VB6
6	'H'	'E'	'L'	'L'	'O'	' '

VB20					VB25
5	'W'	'O'	'R'	'L'	'D'

После выполнения программы

VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

VB100											VB111
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

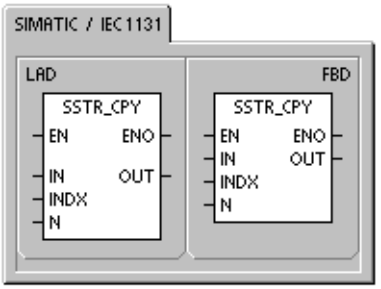
AC0
11

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 области)
- 009B (индекс=0)



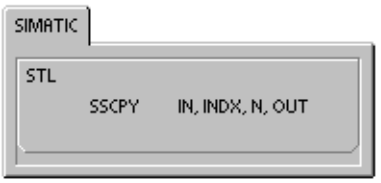
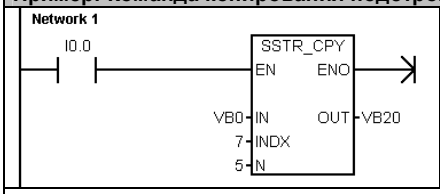


Таблица 6–63. Допустимые операнды для команды копирования подстроки из строки

Входы/выходы	Типы данных	Операнды
IN, OUT	BYTE (строка)	VB, LB, *VD, *LD, *AC
INDX, N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, константа

Пример: Команда копирования подстроки




```
Network 1      //Начиная с седьмого символа в строке,
               //находящейся по адресу VB0, скопировать
               //5 символов в новую строку по адресу VB20

LD      I0.0
SSCPY   VB0, 7, 5, VB20
```

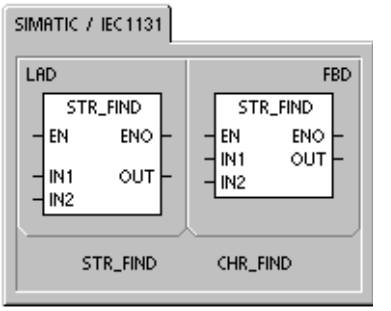
Перед выполнением программы

VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

После выполнения программы

VB20					VB25
5	'W'	'O'	'R'	'L'	'D'

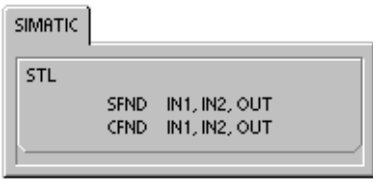
Поиск строки в строке



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 области)
- 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

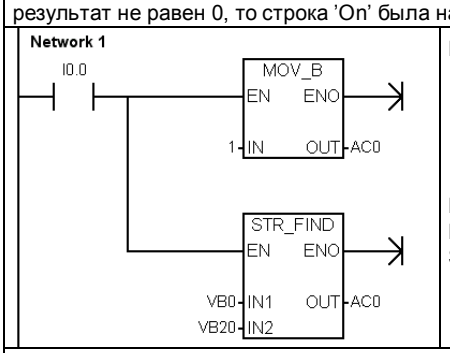
- 0006 (косвенный адрес)
- 0091 (ошибка области)
- 009B (индекс=0)

Таблица 6–64. Допустимые операнды для команд поиска строки в строке и первого символа в строке

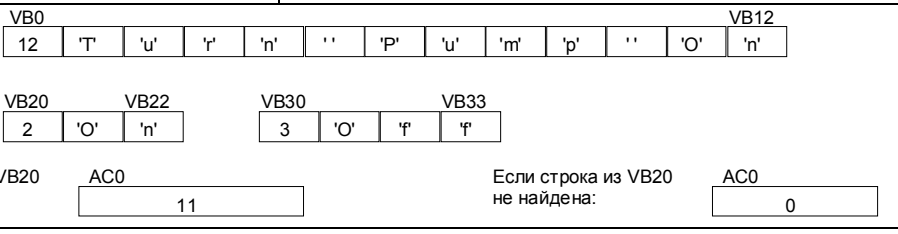
Входы/выходы	Типы данных	Операнды
IN1, IN2	BYTE (строка)	VB, LB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

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).

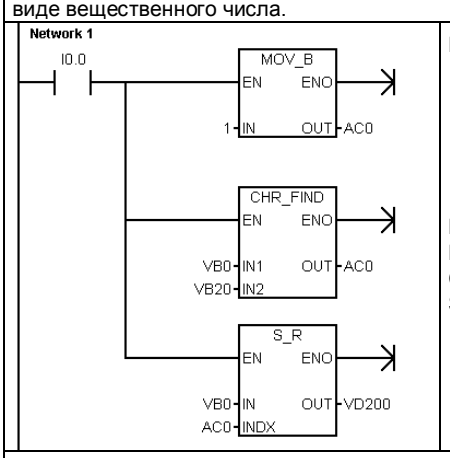



```
Network 1      //1. Установить AC0 в 1.
               //   (AC0 используется как параметр OUT.)
               //2. Искать в строке, находящейся по адресу
               //   VB0, строку, находящуюся по адресу
               //   VB20 (’On [Включить]’), начиная с первой
               //   позиции (AC0=1).
LD      I0.0
MOVB    1, AC0
SFND    VB0, VB20, AC0
```



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      //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											VB11
11	'T'	'e'	'm'	'p'	' '	' '	'9'	'8'	'.'	'6'	'F'

VB20												VB32
12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'-'

Начальная позиция температуры, хранящейся в VB0:

AC0

7

Значение температуры в виде вещественного числа:

VD200

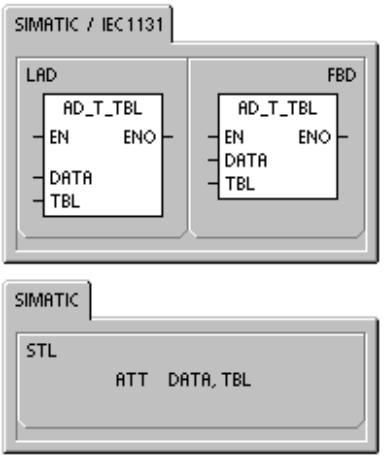
98.6

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)

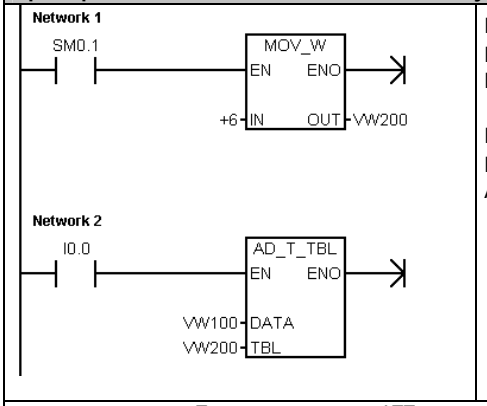
Связанные с командой биты специальной памяти:

- SM1.4 устанавливается в 1, если вы пытаетесь переполнить таблицу

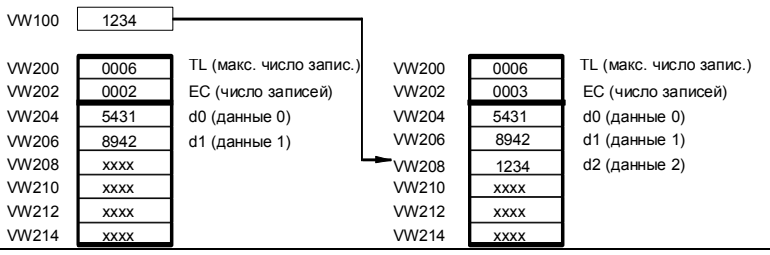
Таблица 6–65. Допустимые операнды для табличных команд

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

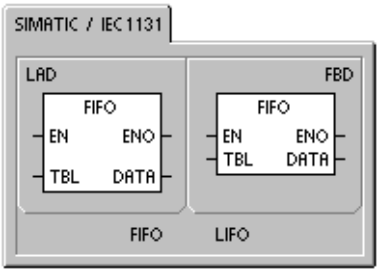



```
Network 1        //Загрузить максимальную длину таблицы
LD       SM0.1
MOVW     +6, VW200

Network 2
LD       I0.0
ATT      VW100, VW200
```



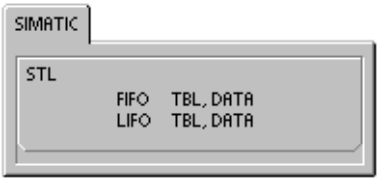
Удаление первой записи из таблицы и удаление последней записи из таблицы

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)

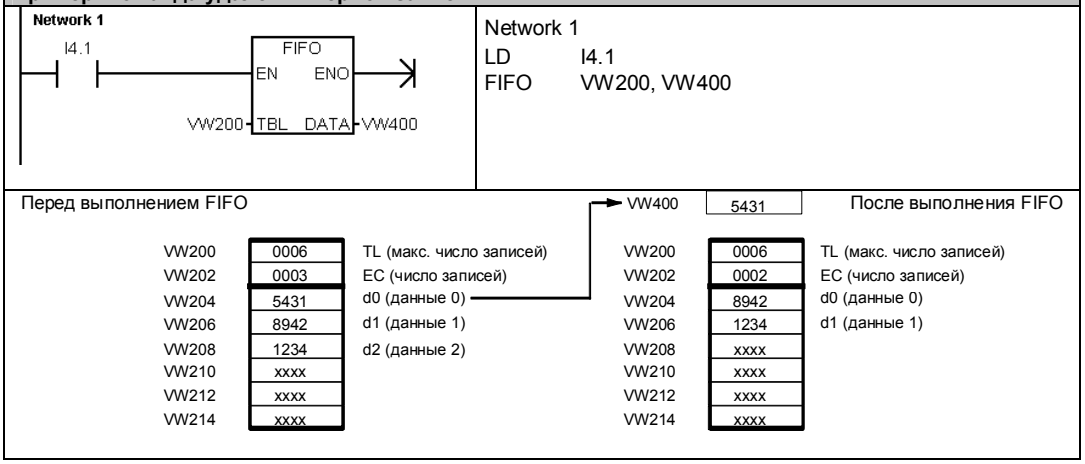
Связанные с командой биты специальной памяти:

- SM1.5 устанавливается в 1, если вы пытаетесь удалить запись из пустой таблицы

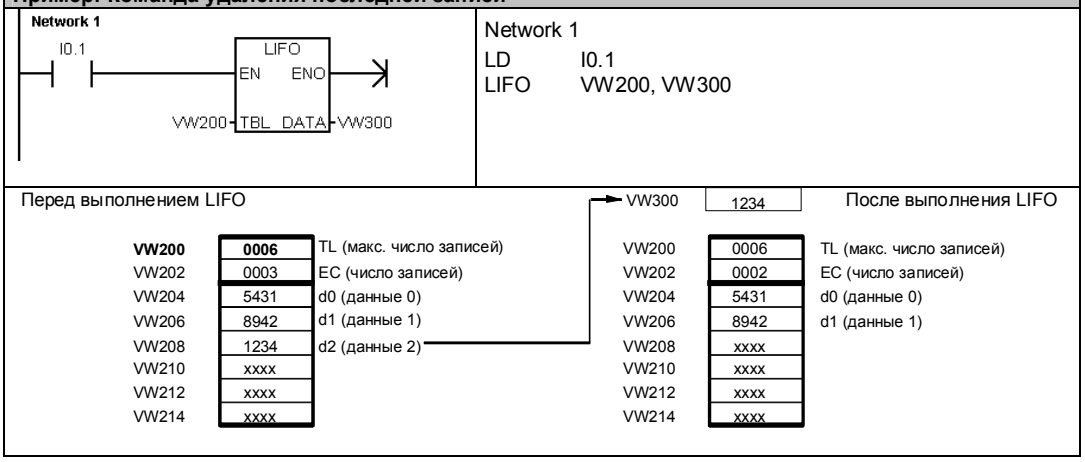
Таблица 6–66. Допустимые операнды для команд удаления первой и последней записи

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



Пример: Команда удаления последней записи



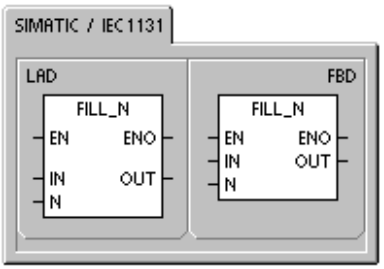
Заполнение памяти комбинацией двоичных разрядов

Команда заполнения памяти (FILL) заполняет область памяти, начинающуюся с адреса OUT, N словами по образцу, расположенному 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)



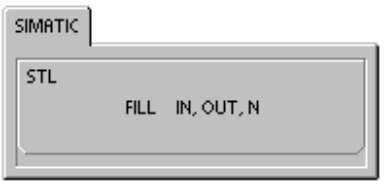
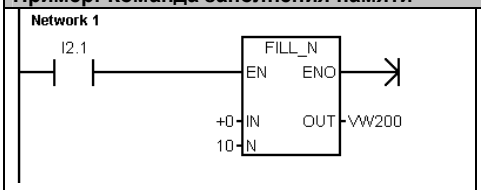


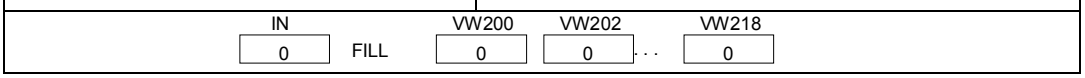
Таблица 6–67. Допустимые операнды для команды заполнения памяти

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

Пример: Команда заполнения памяти

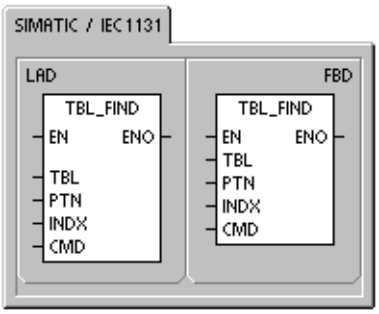



```
Network 1
LD      I2.1
FILL    +0, VW200, 10
```



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 соответствующая запись, найдена, то INDX указывает на эту запись в таблице. Для нахождения следующей записи INDX должен быть увеличен, прежде чем команда Поиск в таблице будет вызвана снова. Если запись, удовлетворяющая условию поиска, не найдена, то INDX имеет значение, равное количеству записей.

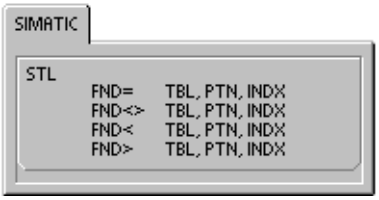


Таблица может иметь до 100 записей. Записи данных (область поиска) нумеруются от 0 до максимального значения, равного 99.

Сбойные состояния, устанавливающие ENO = 0

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

Таблица 6–68. 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 чем (<), 4: больше чем (>)

Совет

Когда вы используете команду поиска в таблицах, созданных с помощью команд ATT, LIFO и 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. 6–36.

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

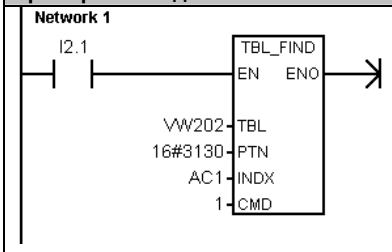
VW200	0006	TL (макс. число записей)
VW202	0006	EC (число записей)
VW204	xxxx	d0 (данные 0)
VW206	xxxx	d1 (данные 1)
VW208	xxxx	d2 (данные 2)
VW210	xxxx	d3 (данные 3)
VW212	xxxx	d4 (данные 4)
VW214	xxxx	d5 (данные 5)

Формат таблицы для TBL_FIND

VW202	0006	EC (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. 6–36. Различие в форматах таблиц у команды "Поиск в таблице" и команд ATT, LIFO и FIFO

Пример: Команда поиска значения в таблице




```
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 [0] — Для поиска с самой верхней записи в таблице AC1 должен быть установлен в 0

Поиск в таблице

AC1 [2] — AC1 содержит номер записи данных, которая соответствует критерию поиска.

AC1 [3] — Перед поиском остальных записей в таблице увеличить INDX на 1

Поиск в таблице

AC1 [4] — AC1 содержит номер записи данных, которая соответствует второму совпадению, найденному в таблице (d4)

AC1 [5] — Перед поиском остальных записей в таблице увеличить INDX на 1.

Поиск в таблице

AC1 [6] — AC1 содержит значение, равное количеству записей. Вся таблица была просмотрена в безуспешных поисках еще одного совпадения.

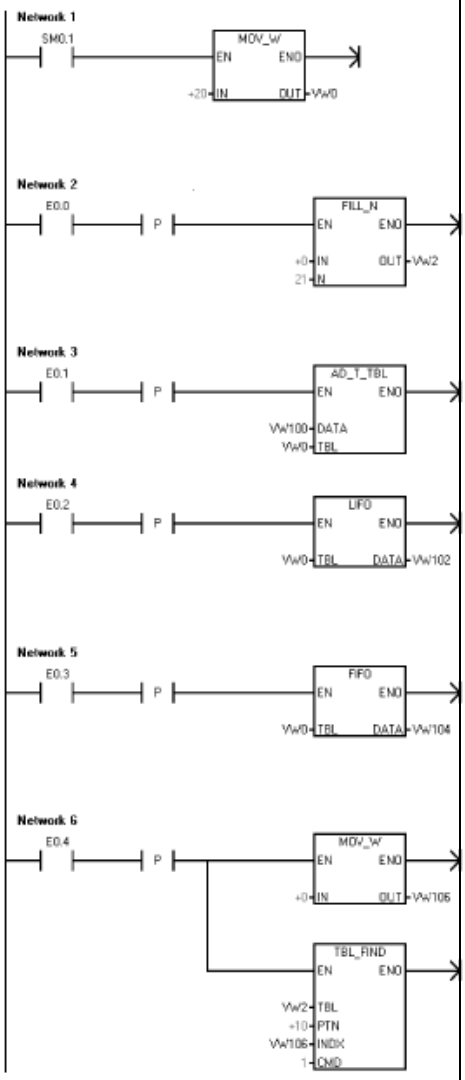
AC1 [0] — Перед новым поиском в таблице INDX должен быть установлен в 0.

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.

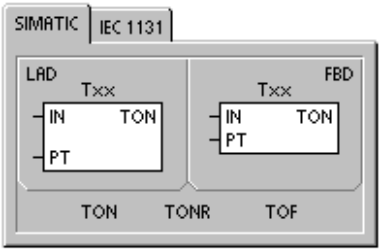
Для поиска в таблице индекс (VW106) перед выполнением поиска должен быть установлен в 0. Если найдено совпадение, индекс будет указывать номер соответствующей записи, а если совпадение не обнаружено, то индекс будет содержать текущее количество записей для таблицы (VW2).

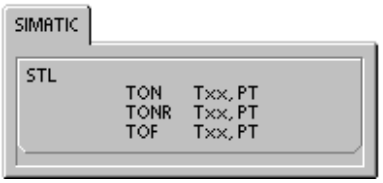



```
Network 1      //Создать таблицу с 20 записями, начиная с
               //адреса 4.
               //1. В 1-ом цикле определит максимальную
               //   длину таблицы.
LD             SM0.1
MOVW           +20, VW0

Network 2      //Сбросить таблицу входом I0.0
               //При нарастающем фронте на I0.0
               //заполнить адреса, начиная с VW2,
               //значениями "+0" .
LD        I0.0
EU
FILL      +0, VW2, 21
Network 3      //Записать значение в таблицу с помощью
               //входа I0.1
               //При нарастающем фронте на I0.1
               //скопировать в таблицу значение из адреса
               //VW100.
LD        I0.1
EU
ATT       VW100, VW0
Network 4      //Прочитать первую запись таблицы с
               //помощью входа I0.2
               //Переслать последнюю запись таблицы в
               //VW102.
               //Это уменьшает количество записей.
               //При нарастающем фронте на I0.2
               //переслать последнюю запись таблицы в
               //VW102
LD        I0.2
EU
LIFO      VW0, VW102
Network 5      //Прочитать последнюю запись таблицы с
               //помощью входа I0.3
               //Переслать первую запись таблицы в
               //VW102.
               //Это уменьшает количество записей.
               //При нарастающем фронте на I0.0
               //переслать первую запись таблицы в
               //VW104
LD        I0.3
EU
FIFO      VW0, VW104
Network 6      //Искать в таблице первый адрес, имеющий
               //значение 10.
               //1. При нарастающем фронте на I0.4
               //   сбросить указатель на индекс.
               //2. Найти в таблице запись, равную 10.
LD        I0.4
EU
MOVW      +0, VW106
FND=      VW2, +10, VW106
```

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 (Txx) определяет его разрешающую способность.

Таймер с задержкой выключения

Таймер с задержкой выключения (TOF) используется для задержки выключения выхода на фиксированный интервал времени после выключения входа. Номер таймера (Txx) определяет его разрешающую способность.

Таблица 6–69. 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 (Txx). Например, у вас не могут одновременно использоваться TON T32 и TOF T32.

Как показано в таблице 6–70, 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 накапливания 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 6–70. 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 и уловки

Советы и уловки (Tips and Tricks) 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 времени.

- 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 6–71, номер таймера определяет его разрешающую способность.

Совет
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.

Таблица 6–71. 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

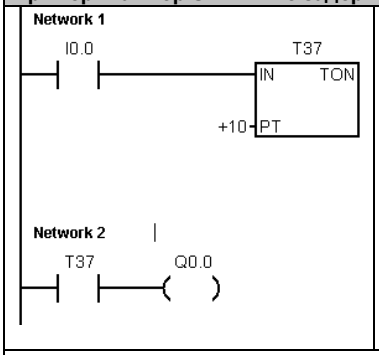
Влияние разрешающей способности на работу таймера

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 сканирования. Текущее значение и бит таймера остаются неизменными на протяжении цикла, и интервалы времени, накопленные в течение цикла, добавляются к текущему значению на начало цикла сканирования.

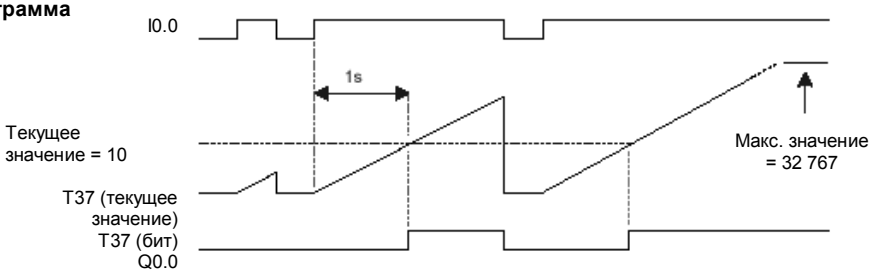
У таймера с разрешающей способностью 100 мс бит таймера и текущее значение обновляются, когда выполняется команда; поэтому, чтобы таймер сохранял правильное значение времени, обратите внимание на то, чтобы ваша программа выполняла команду для 100–миллисекундного таймера только один раз за цикл.

Пример: Таймер SIMATIC с задержкой включения



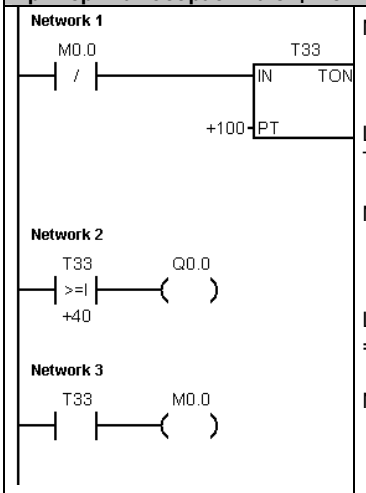

```
Network 1      //100-миллисекундный таймер T37 заканчивает отсчет
               //времени по истечении (10 x 100 мс = 1 с)
               //I0.0 установлен = активизировать T37,
               //I0.0 сброшен = деактивизировать и сбросить T37
LD     I0.0
TON    T37, +10

Network 2      //Бит T37 управляется таймером T37
LD     T37
=      Q0.0
```

Временная диаграмма



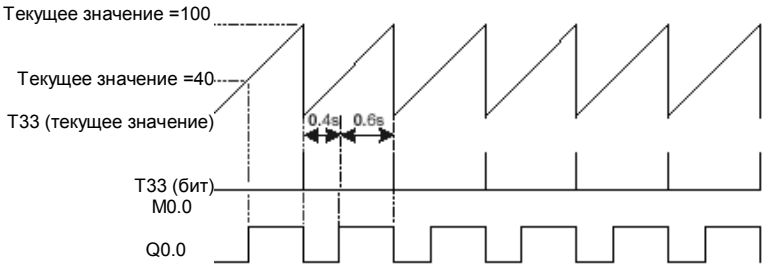
Совет
Чтобы гарантировать, что выход самосбрасывающегося таймера включался на время цикла каждый раз, когда таймер достигает предустановленного значения, вы должны использовать на разрешающем входе размыкающий контакт вместо бита таймера.

Пример: Самосбрасывающийся таймер SIMATIC с задержкой включения

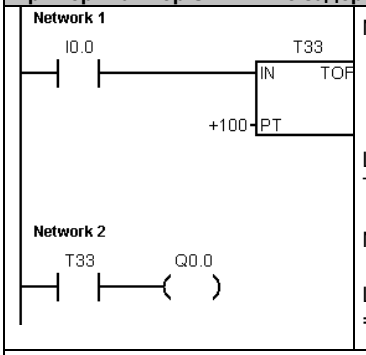



```
Network 1      //10-миллисекундный таймер T33 заканчивает отсчет
               //времени по истечении (100 x 10 мс = 1 с)
               //Импульс M0.0 слишком быстр для наблюдения в
               //представлении статуса программы
LDN     M0.0
TON     T33, +100

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
```

Временная диаграмма

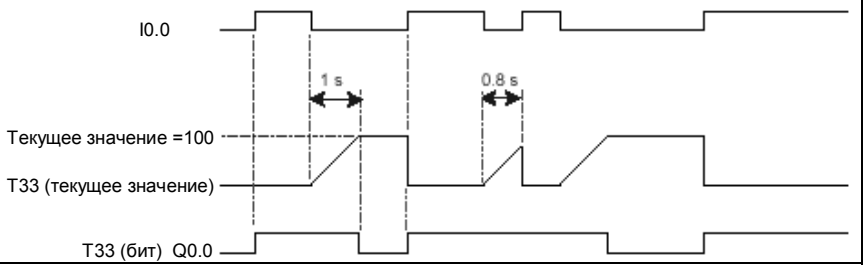


Пример: Таймер SIMATIC с задержкой выключения

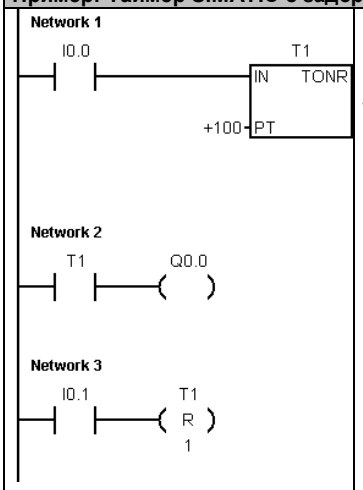



```
Network 1      //10-миллисекундный таймер T33 заканчивает отсчет
               //времени по истечении (100 x 10 мс = 1 с)
               //При выключении I0.0 T33 активизируется
               // При включении I0.0 T33 деактивизируется и
               //сбрасывается
LD      I0.0
TOF     T33, +100

Network 2      //Таймер T33 управляет Q0.0 через контакт таймера
               //T33
LD      T33
=       Q0.0
```

Временная диаграмма

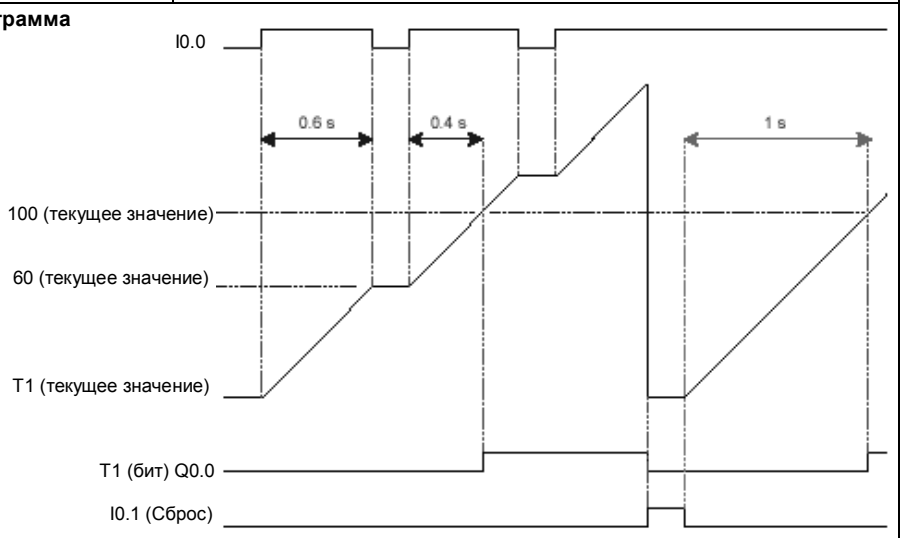


Пример: Таймер SIMATIC с задержкой включения с запоминанием

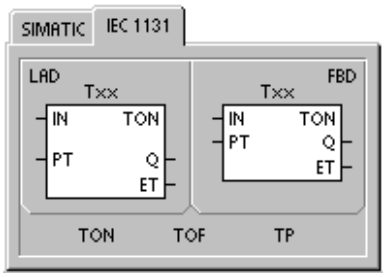



```
Network 1      //10-миллисекундный таймер TONR T1 заканчивает
               //отсчет времени при PT=(100 x 10 мс=1с)
LD     I0.0
TONR   T1, +100

Network 2      //Бит T1 управляется таймером T1.
               //Включить Q0.0, после того как таймер проработает
               //в целом 1 секунду
LD     T1
=      Q0.0

Network 3      //Таймеры TONR должны сбрасываться командой
               //сброса с адресом T.
               //Сбросить таймер T1 (текущее значение и бит), когда
               //I0.1 включен.
LD     I0.1
R      T1, 1
```

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 6–72. 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.

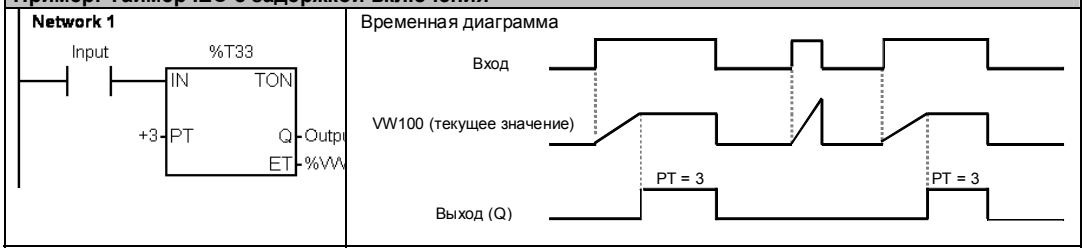
- Команда TON ведет отсчет времени до предварительно заданного значения с момента, когда разрешающий вход (IN) становится истинным. Когда истекшее время (ET) равно заданному времени (PT), выходной бит таймера (Q) включается. Выходной бит сбрасывается, когда разрешающий вход выключается. Когда заданное 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.

Каждое отсчет текущего значения кратен базе времени. Например, значение 50 в 10–миллисекундном таймере представляет величину 500 мс. Таймеры IEC (TON, TOF и TP) имеются с тремя разрешающими способностями. Разрешающая способность определяется номером таймера, как показано в таблице 6–73.

Таблица 6–73. Разрешающая способность таймеров IEC

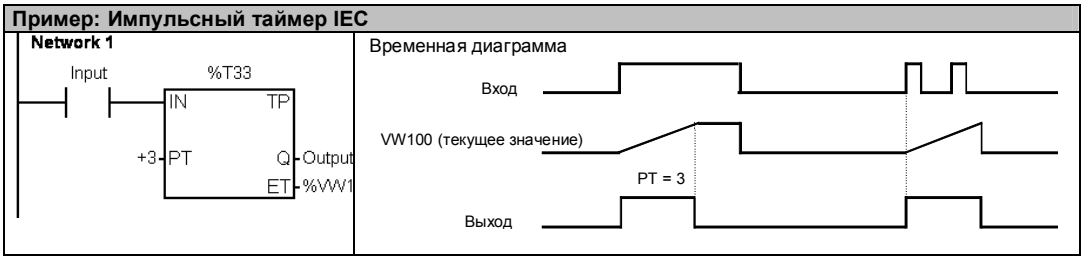
Разрешающая способность	Максимальное значение		Номер таймера
1 мс	32,767 с	(0,546 минуты)	T32, T96
10 мс	327,67 с	(5,46 минуты)	T33 – T36, T97 – T100
100 мс	3276,7 с	(54,6 минуты)	T37 – T63, T101 – T255

Пример: Таймер IEC с задержкой включения



Пример: Таймер IEC с задержкой выключения

Пример: Импульсный таймер IEC

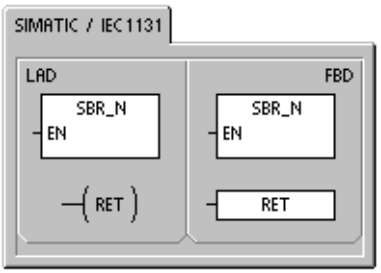


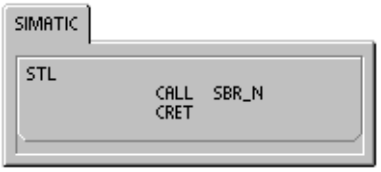
Операции с подпрограммами

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.

Подпрограмма не может быть помещена 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 6–74. Допустимые операнды для команды вызова подпрограммы

Входы/выходы	Типы данных	Операнды
SBR_N	WORD	Константа *для CPU 221, CPU 222, CPU 224 и CPU 226*: от 0 до 63 *для CPU 226XM:* от 0 до 127

Совет
STEP 7-Micro/WIN 32 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.

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 6–75. 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 Под текущей записью появится место для записи еще одного параметра выбранного типа.

Таблица 6–75. 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.
TEMP	Локальная память, не используемая для передаваемых параметров, может использоваться для временного хранения данных внутри подпрограммы.

Как показано на рис. 6–37, 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.
- INT, DINT: Эти типы данных определяют входной или выходной параметр со знаком размером 2 или 4 байта соответственно.

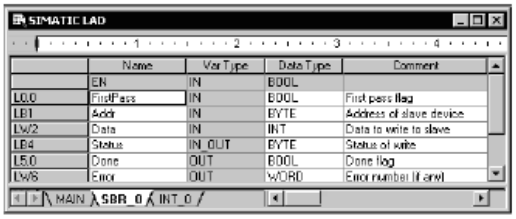


Рис. 6–37. Таблица локальных переменных

- REAL: Этот тип данных определяет число с плавающей точкой IEEE однократной точности (4 байта).
- 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

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.

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.

Только STL:

```
Network 1
LD      I0.0
CALL    SBR_0, I0.1, VB10, I1.0, &VB100, *AC1, VD200
```

Для правильного отображения в LAD и FBD:

```
Network 1
LD      I0.0
=       L60.0
LD      I0.1
=       L63.7
LD      L60.0
CALL    SBR_0, L63.7, VB10, I1.0, &VB100, *AC1, VD200
```

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.

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	Network 1 SM0.1 — SBR_0 EN	Network 1 //В первом цикле вызвать подпрограмму 0 для //инициализации. LD SM0.1 CALL SBR_0
S B R 0	Network 1 M14.3 — (RET) Network 2 SM0.0 — MOV_B EN ENO 10 — IN OUT — VB0	Network 1 //Вы можете использовать условный возврат // для выхода из подпрограммы перед //последним сегментом. LD M14.3 CRET Network 2 //Этот сегмент будет пропущен, если M14.3 //установлен. LD SM0.0 MOVB 10, VB0

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 детской игрой.

В этой главе

Основы обмена данными в сети для S7–200 208

Выбор протокола связи для вашей сети 211

Инсталляция и деинсталляция интерфейсов для обмена данными 216

Построение сети 218

Создание протоколов, определяемых пользователем, при свободно программируемом обмене данными 222

Использование в сети модемов и STEP 7-Micro/WIN 224

Для опытных пользователей 228

Основы обмена данными в сети для S7–200

Использование в сети Master- и Slave-устройств

S7–200 поддерживает несимметричную сеть с master- и slave-устройствами и может работать в сети как в качестве ведущего (master) устройства, так и в качестве ведомого (slave) устройства, тогда как STEP 7-Micro/WIN всегда является master-устройством.

Совет
Если вы используете Windows NT и кабель PC/PPI, то в сети не может быть других master-устройств.

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 (двухточечный обмен данными).

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 сете 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, например, STEP 7, то вам, возможно, потребуется изменить сетевой адрес для STEP 7-Micro/WIN.

Как показано на рис. 7–1, 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.
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 (Transmission rate) для STEP 7-Micro/WIN.

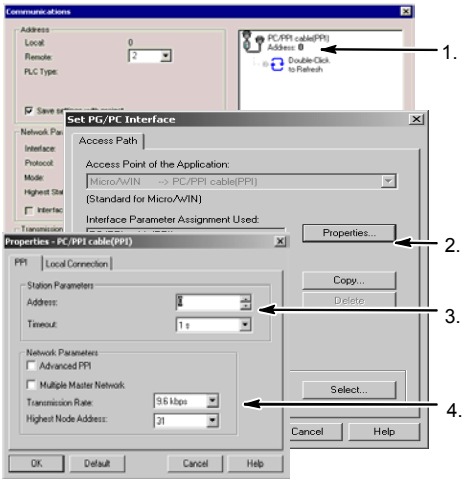


Рис. 7–1. Настройка STEP 7-Micro/WIN

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. Выбрав параметры для S7–200, вы должны загрузить в S7–200 системный блок.

По умолчанию скорость передачи для каждого порта S7–200 составляет 9,6 кБод, а сетевой адрес равен 2.

Как показано на рис. 7–2, 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.

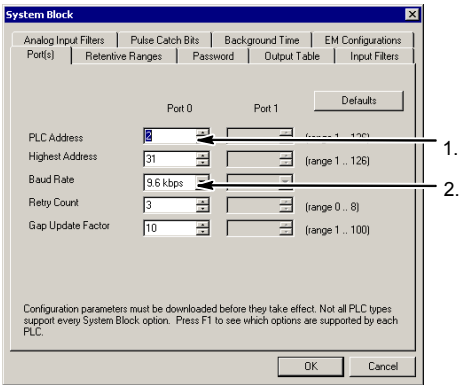


Рис. 7–2. Настройка CPU S7–200

Установка удаленного адреса

Перед тем как вы сможете загрузить измененные настройки в S7–200, вы должны установить коммуникационный порт (COM) STEP 7-Micro/WIN и удаленный адрес S7–200 таким образом, чтобы они соответствовали текущим настройкам удаленного S7–200. См. рис. 7–3.

После того как вы загрузили измененные настройки, вы должны снова настроить порт COM (если имеются отличия от настроек, использованных для удаленного S7–200). Для вызова диалогового окна Communications [Обмен данными] щелкните на пиктограмме Communications на навигационной панели или выберите команду меню **View > Component >Communications [Вид > Компонент > Обмен данными]**.

1. Выберите удаленный адрес.
2. Обеспечьте, чтобы параметры (скорость передачи) для COM-порта, порта удаленного S7–200 и кабеля PC/PPI совпадали. В противном случае обмен данными потерпит неудачу.

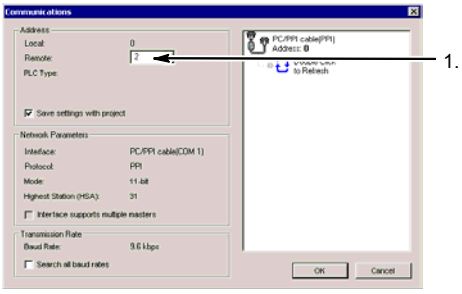


Рис. 7–3. Настройка CPU S7–200

Поиск CPU S7–200 в сети

Вы можете искать и идентифицировать CPU S7–200, присоединенные к сети. Вы можете также вести поиск в сети при определенной скорости передачи или на всех скоростях, когда вы хотите идентифицировать модули S7–200.

Если вы используете кабель PC/PPI, то STEP 7-Micro/WIN может вести поиск только при скоростях 9,6 кБод и 19,2 кБод. Для платы CP STEP 7-Micro/WIN ведет поиск со скоростями 9,6 кБод, 19,2 кБод и 187,5 кБод. Поиск начинается при скорости, установленной в данный момент.

1. Откройте диалоговое окно Communications [Обмен данными] и дважды щелкните на пиктограмме Refresh [Обновить], чтобы начать поиск.
2. Для поиска на всех скоростях активизируйте триггерную кнопку Search All baud rates [Искать на всех скоростях передачи].

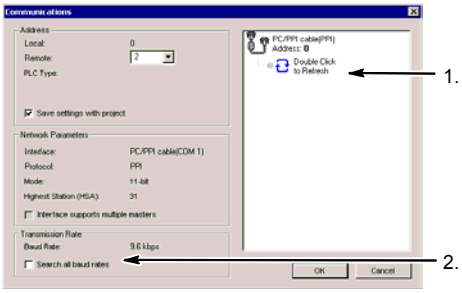


Рис. 7–4. Поиск CPU в сети

Выбор протокола связи для вашей сети

CPU S7–200 поддерживают одну или несколько из следующих возможностей обмена данными, которые позволяют вам настроить свою сеть на производительность и функциональные возможности, требуемые вашим приложением:

- 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.

Протокол PPI

PPI является протоколом типа master–slave [ведущий-подчиненный]: master-устройства посылают запросы slave-устройствам, а slave-устройства реагируют. См. рис. 7–5. 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.

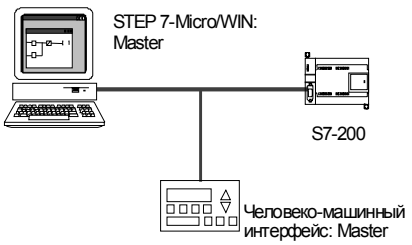


Рис. 7–5. Сеть PPI

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.

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.

Вы можете использовать протокол PPI для обмена данными со всеми CPU S7–200. Для обмена данными с EM 277 вы должны активизировать расширенный 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	4
	Порт 1	9,6 кБод, 19,2 кБод или 187,5 кБод	4
Модуль EM 277		9,6 кБод до 12 МБод	6 на модуль

Протокол MPI

MPI допускает обмен данными типа master–master и master–slave. См. рис. 7–6. 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.

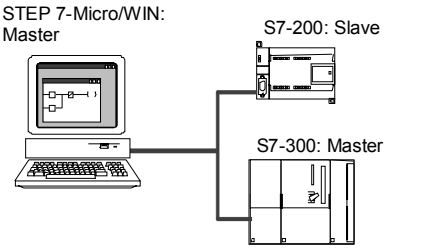


Рис. 7–6. Сеть MPI

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.

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. 7–7. 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.

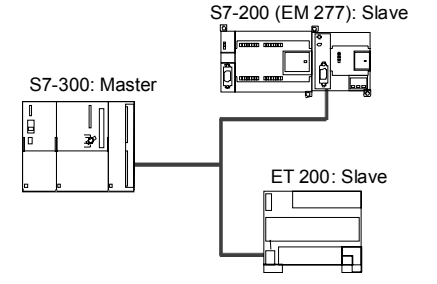


Рис. 7–7. Сеть PROFIBUS

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.

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 PC/PPI или через плату коммуникационного процессора (CP), установленную в устройстве программирования.

В примере сети, представленном в верхней части рис. 7–8, master-устройством сети является устройство программирования (STEP 7-Micro/WIN). В примере сети, представленном в нижней части рис. 7–8, 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.

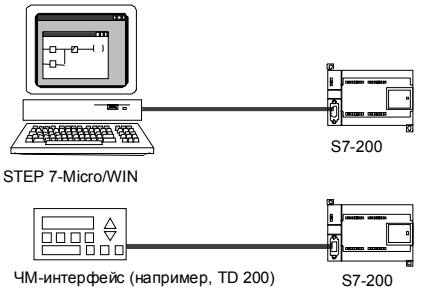


Рис. 7–8. Сеть PPI с одним master-устройством

В конфигурации PPI с одним master-устройством вы настраиваете STEP 7-Micro/WIN на использование протокола PPI: выберите single–master [одно master-устройство], multi–master [несколько master-устройств] или PPI Advanced [расширенный двухточечный интерфейс].

Сеть PPI с несколькими master-устройствами

На рис. 7–9 показан пример сети с несколькими master-устройствами и одним slave-устройством. Устройство программирования (STEP 7-Micro/WIN) использует плату CP или кабель PC/PPI, а 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. CPU S7–200 является slave-устройством.

Чтобы в сети несколько master-устройств могли обращаться к одному slave-устройству, вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master). Оптимальным решением здесь является PPI Advanced.

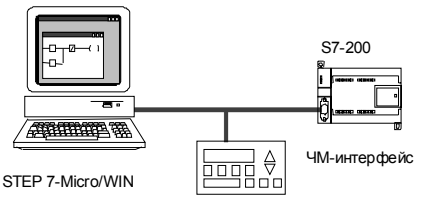


Рис. 7–9. Несколько master-устройств с одним slave-устройством

На рис. 7–10 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.

Для сети PPI с нескольким master- и slave-устройствами вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master). Оптимальным решением здесь является PPI Advanced.

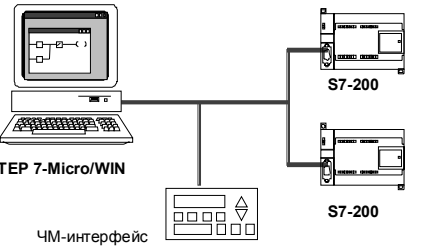


Рис. 7–10. Несколько master- и slave-устройств

Сложные сети PPI

На рис. 7–11 показан пример сети, использующей несколько master-устройств с обменом данными между равноправными узлами.

STEP 7-Micro/WIN и устройство 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).

Для такого рода сложных сетей PPI вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master). Оптимальным решением здесь является PPI Advanced.

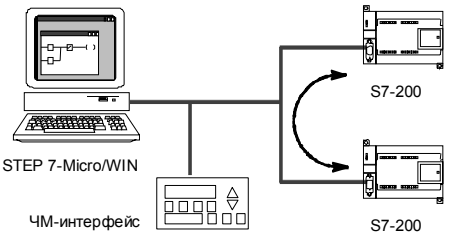


Рис. 7–11. Обмен данными между равноправными узлами

На рис. 7–12 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 этой сети вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master). Оптимальным решением здесь является PPI Advanced.

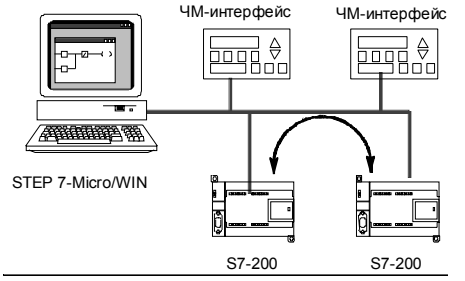


Рис. 7–12. Устройства ЧМ-интерфейса и обмен данными между равноправными узлами

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. 7–13, 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 скоростей свыше 19,2 кБод STEP 7-Micro/WIN должен быть присоединен через плату коммуникационного процессора (CP).

Для обмена данными с модулями CPU S7–200 вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master). Оптимальным решением здесь является PPI Advanced.

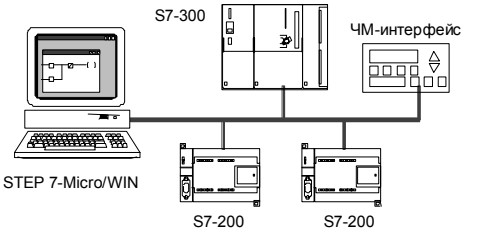


Рис. 7–13. Скорости передачи до 187,5 кБод

Сети со скоростями передачи свыше 187,5 кБод (до 12 МБод)

При скоростях передачи свыше 187,5 кБод CPU S7–200 должен использовать для присоединения к сети модуль EM 277. См. рис. 7–14. 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 все CPU S7–200 через EM 277. Для обмена данными с EM 277 вы должны установить в STEP 7-Micro/WIN протокол PPI и активизировать драйвер сети с несколькими ведущими узлами (multi-master).

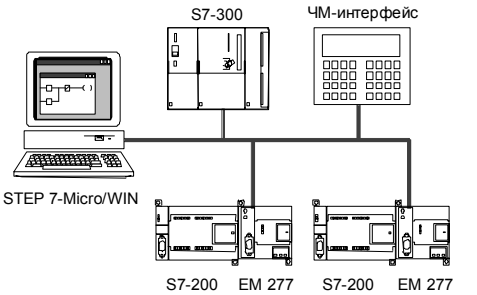


Рис. 7–14. Скорости передачи свыше 187,5 кБод

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. 7–15 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.

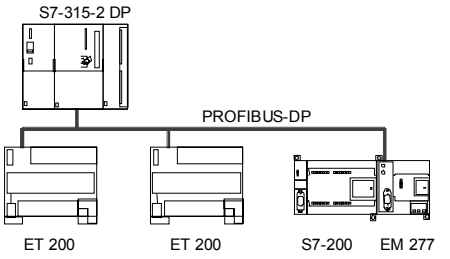


Рис. 7–15. Сеть с S7–315-2 DP

Сети со STEP 7-Micro/WIN и ЧМ-интерфейсом

На рис. 7–16 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 19,2 кБод.

Вы настраиваете 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, то выберите универсальный профиль (DP/FMS) для всех master-устройств PROFIBUS.

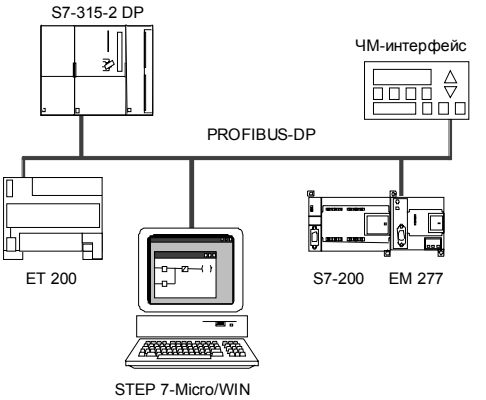


Рис. 7–16. Сеть PROFIBUS

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/Uninstallling 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/Uninstallling 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].

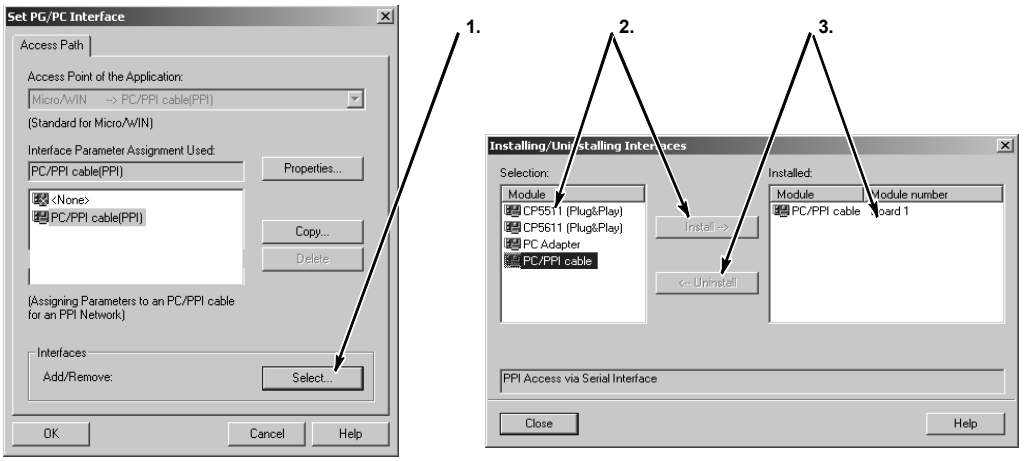


Рис. 7–17. Диалоговые окна Set PG/PC Interface и Installing/Uninstalling Interfaces

Совет

Специальная информация для инсталляции аппаратуры под Windows NT

Инсталляция аппаратных модулей под Windows NT слегка отличается от инсталляции аппаратных модулей под Windows 95. Даже если для обеих операционных систем используются одни и те же аппаратные модули, инсталляция под Windows NT требует более подробных знаний об аппаратуре, которую вы хотите установить. Windows 95 пытается автоматически настроить системные ресурсы для вас, а под Windows NT этого не происходит. Windows NT лишь предлагает вам значения по умолчанию. Эти значения могут как соответствовать, так и не соответствовать конфигурации аппаратуры. Вы можете легко изменить эти параметры, чтобы они соответствовали требуемым настройкам системы.

После того как вы инсталлировали аппаратуру, выделите ее в окне списка Installed [Инсталлировано] и щелкните на кнопке Resources [Ресурсы]. После этого появится диалоговое окно Resources. В этом диалоговом окне вы можете изменить системные настройки для аппаратуры, которую вы фактически установили. Если эта кнопка недоступна (серая), то вам больше ничего не нужно делать.

Здесь вам, возможно, потребуется обратиться к руководству для вашей аппаратуры, чтобы выяснить настройки для каждого из параметров, перечисленных в диалоговом окне. Возможно, вам придется испытать несколько различных прерываний, чтобы правильно установить обмен данными.

Изменение настроек порта вашего компьютера для работы в режиме PPI с несколькими ведущими узлами (Multi–Master)

Если вы используете кабель PC/PPI с операционной системой, которая поддерживает конфигурацию PPI с несколькими ведущими узлами (Multi–Master) (Windows NT не поддерживает этот режим), 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 OK 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 7–4, максимальная длина сегмента сети определяется двумя факторами: потенциальной развязкой (использованием 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 7–4. 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 завершение 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. 7–18), 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, даже если ему не присвоен сетевой адрес.

Рис. 7–18. Пример сети с повторителями

Выбор сетевого кабеля

В сетях S7–200 используются стандартные кабели RS–485 в виде витой пары. В таблице 7–5 приведены технические данные сетевого кабеля. К сегменту сети можно присоединить до 32 устройств.

Таблица 7–5. Общие технические данные сетевого кабеля

Технические данные	Описание
Тип кабеля	Экранированная витая пара
Сопротивление контура	≤115 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 ±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 7–6 показан разъем, обеспечивающий физическое соединение с коммуникационным портом, и описано назначение его контактов.

Таблица 7–6. Назначение контактов коммуникационного порта S7–200

Разъем	Номер контакта	Сигнал PROFIBUS	Порт 0/Порт 1
Конт. 1, Конт. 6, Конт. 5, Конт. 9	1	Экран	Заземляющий контакт
	2	Обратный провод 24 В	Общий провод логики
	3	RS–485, сигнал B	RS–485, сигнал B
	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 A	RS–485, сигнал A
	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 7–6) 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. 7–19 показано типовое присоединение для кабельных разъемов.

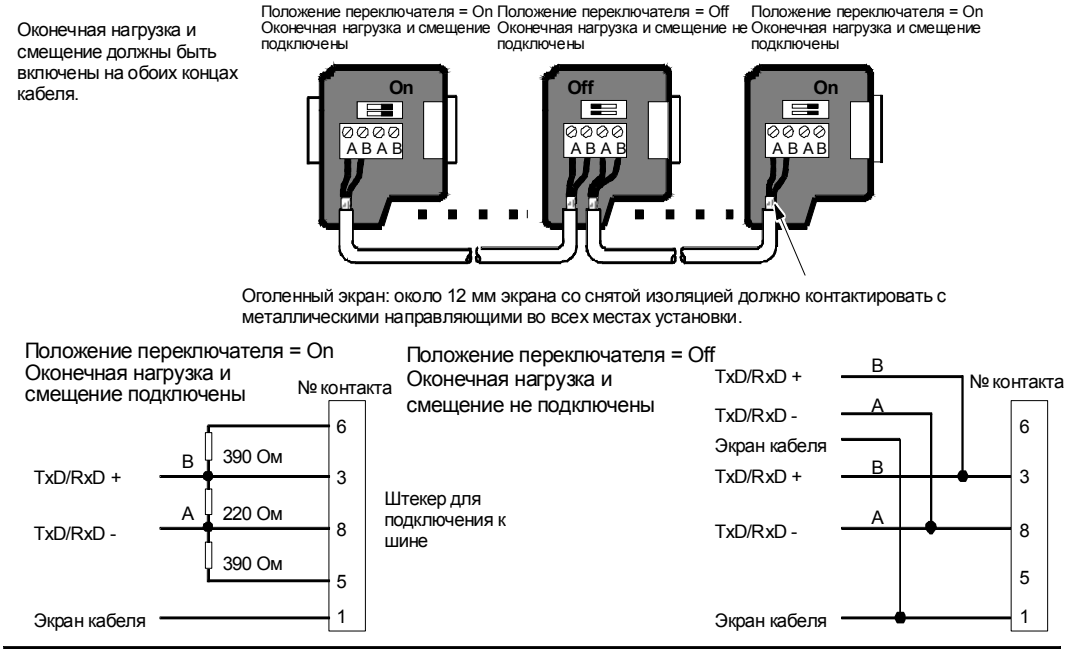


Рис. 7–19. Присоединение сетевого кабеля

Выбор платы CP или кабеля PC/PPI для вашей сети

Как показано в таблице 7–7, STEP 7-Micro/WIN поддерживает несколько плат CP, которые дают возможность устройству программирования (вашему компьютеру или устройству программирования SIMATIC) работать в сети в качестве master-устройства.

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. Кабель PC/PPI также позволяет активизировать несколько master-устройств.

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.

Осторожно

Использование преобразователя RS–485/RS–232 без гальванической развязки может повредить порт RS–232 вашего компьютера.

Кабель PC/PPI фирмы Siemens (номер для заказа 6ES7 901-3BF21-0XA0) обеспечивает гальваническую развязку между портом RS–485 на CPU S7–200 и портом RS–232, используемым для подключения на вашем компьютере. Если вы не пользуетесь кабелем PC/PPI фирмы Siemens, то вы должны позаботиться о гальванической развязке для порта RS–232 вашего компьютера.

Таблица 7–7. Платы CP и протоколы, поддерживаемые STEP 7-Micro/WIN

Конфигурация	Скорость передачи	Протокол
Кабель PC/PPI[1] Подключен к порту COM на устройстве программирования	9,6 кБод или 19,2 кБод	PPI
CP 5511 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
MPI встроенный порт на устройстве программирования SIMATIC или плата CP для вашего компьютера (плата ISA)	от 9,6 кБод до 12 МБод	PPI, MPI и PROFIBUS

[1] Кабель PC/PPI обеспечивает гальваническую развязку между портом RS–485 (на CPU S7–200) и портом RS–232, 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 вида 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, примите во внимание следующее:

- 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

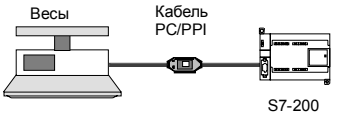
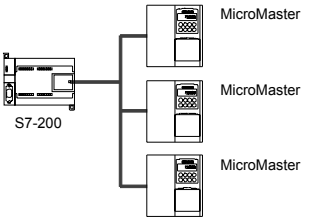
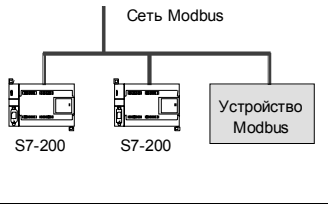
При свободно программируемом обмене данными (Freeport) ваша программа может управлять коммуникационным портом CPU S7–200. Режим свободно программируемого обмена данными можно использовать для реализации определяемых пользователем протоколов для обмена данными со многими типами интеллектуальных устройств. 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-прота, 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 7–8. Использование режима свободно программируемого обмена данными

Конфигурация сети	Описание	
Использование режима свободно программируемого обмена данными через соединение RS–232		Пример: Использование S7–200 с электронными весами, имеющими порт RS–232. ■ Кабель PC/PPI соединяет порт RS–232 на весах с портом RS–485 на CPU S7–200. ■ CPU S7–200 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		Пример: Использование S7–200 с приводами SIMODRIVE MicroMaster. ■ STEP 7-Micro/WIN предоставляет в распоряжение библиотеку USS. ■ CPU S7–200 является master-устройством, а приводы – slaves-устройствами. Советы и уловки *Пример программы, использующей USS, вы найдете в советах и уловках (Tips and Tricks) на компакт-диске с документацией. Совет 28*
Создание программы пользователя, эмулирующей slave-устройство в другой сети		Пример: Соединение CPU S7–200 с сетью Modbus. ■ Программа пользователя в S7–200 эмулирует slave-устройство Modbus. ■ STEP 7-Micro/WIN предоставляет в распоряжение библиотеку Modbus. Советы и уловки *Пример программы, использующей Modbus, вы найдете в советах и уловках (Tips and Tricks) на компакт-диске с документацией. Совет 41.*

Использование кабеля PC/PPI и режима свободно программируемого обмена данными с устройствами RS–232

Кабель PC/PPI и функции свободно программируемого обмена данными можно использовать для связи CPU S7–200 со многими устройствами, совместимыми со стандартом RS–232.

Кабель PC/PPI находится в режиме передачи, когда данные передаются из порта RS–232 в порт RS–485. Кабель PC/PPI находится в режиме приема, когда он бездействует или передает данные из порта RS–485 в порт RS–232. Кабель из режима приема в режим передачи немедленно после обнаружения символов в линии передачи RS–232.

Кабель PC/PPI поддерживает скорости передачи между 1200 Бод и 115,2 кБод. Для настройки кабеля на нужную скорость передачи используются DIP-перключатели на корпусе кабеля PC/PPI. В таблице 7–9 Бод 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 7–9.

Таблица 7–9. Время переключения и настройки

Скорость передачи	Время переключения	Настройки (1 = наверху)
от 38400 до 115200	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

Если вы работаете с кабелем PC/PPI 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 PC/PPI достаточное время для переключения из режима передачи в режим приема, чтобы можно было передавать данные из порта RS–485 в порт RS–232.

Использование в сети модемов и STEP 7-Micro/WIN

STEP 7-Micro/WIN версии 3.2 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 Modems Properties, выберите тип, наиболее близкий к вашему, или обратитесь к изготовителю вашего модема, чтобы получить файлы с конфигурацией модема для Windows.

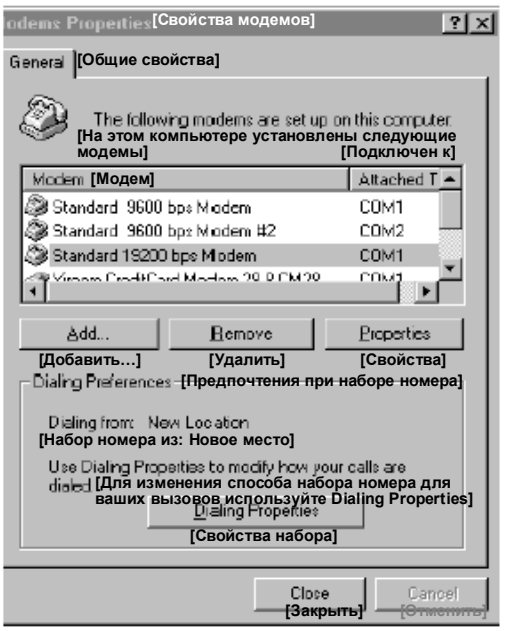


Рис. 7–20. Настройка локального модема

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. 7–21.

1. Дважды щелкните на символе в диалоговом окне Communication [Связь] (см. рис. 7-22).
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. Снова откройте диалоговое окно Communications [Связь] и дважды щелкните на пиктограмме для соединения через модем.
4. В появившемся окне Modem Connection [Соединение через модем] (рис. 7-21) 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.

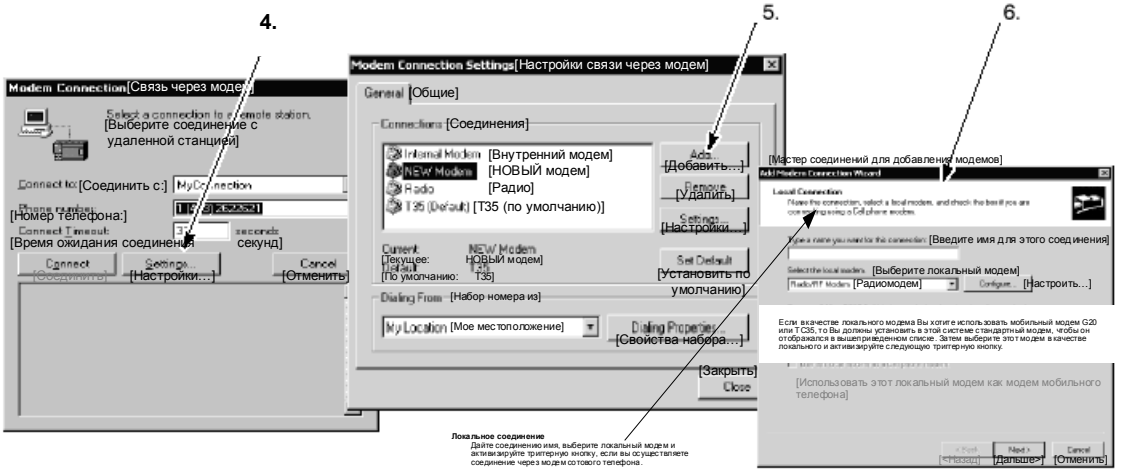


Рис. 7–21. Добавление соединения через модем

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.

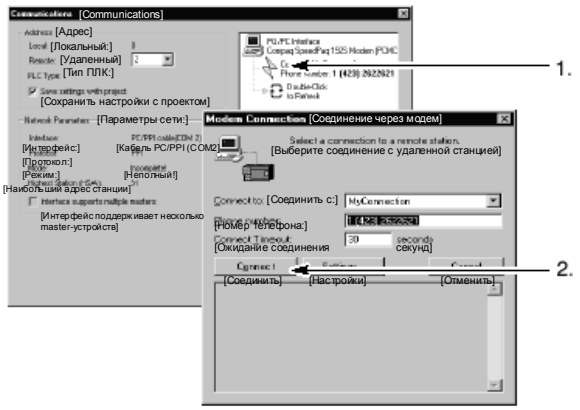


Рис. 7–22. Соединение с S7–200

Настройка удаленного модема

Мастер конфигурирования модемного модуля

Удаленный модем – это модем, присоединенный к S7–200. Если этот модем является модемным модулем EM 241, то конфигурирование 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 конфигурирования 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.

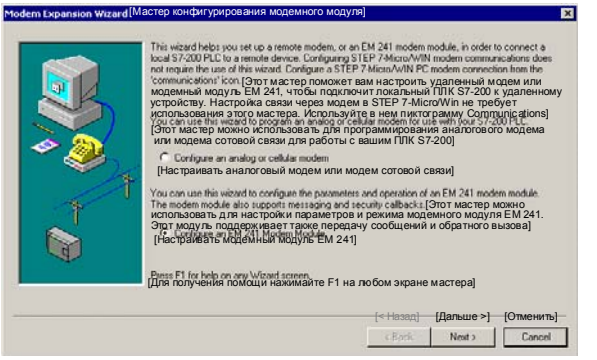


Рис. 7–23. Мастер конфигурирования модемного модуля

Использование модема с кабелем PC/PPI

Кабель PC/PPI можно использовать для соединения коммуникационного порта RS–232 модема с CPU S7–200. См. рис. 7–24. Переключатели 1, 2 и 3 на кабеле PC/PPI устанавливают скорость передачи. Переключатель 4 выбирает 10–битовый или 11–битовый протокол PPI. Переключатель 5 выбирает режим Data Communications Equipment [Аппаратура передачи данных] (DCE) или Data Terminal Equipment [Терминальное оборудование] (DTE). Переключатель 6 (если имеется) выбирает режим сигнала RTS [готовность к передаче] для порта RS–232 кабеля PC/PPI.

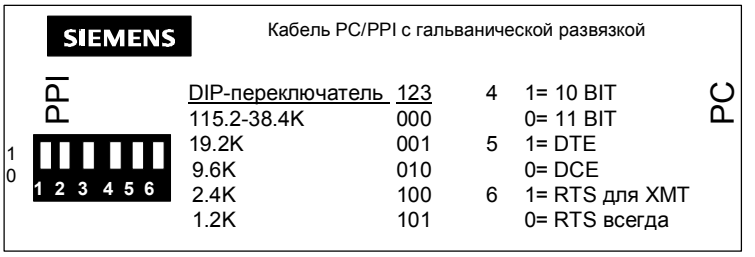


Рис. 7–24. Настройки для кабеля PC/PPI

Чтобы ПК мог управлять модемом, модемы обычно используют сигналы управления RS–232 (например, RTS, CTS и DTR). Если вы используете модем с кабелем PC/PPI, то вы должны настроить модем для работы без этих сигналов. Команды, с помощью которых настраивается ваш модем, вы найдете в документации к вашему модему.

Переключатель 4 кабеля PC/PPI выбирает 10– битовый или 11–битовый режим для протокола PPI. Используйте переключатель 4 только в том случае, если S7–200 соединен через модем со STEP 7-Micro/WIN. В противном случае установите переключатель 4 для работы в 11–битовом режиме, чтобы обеспечить надлежащую работу с другими устройствами.

Переключатель 5 кабеля PC/PPI позволяет установить порт RS–232 кабеля в режим DCE или DTE. Если вы используете кабель PC/PPI со STEP 7-Micro/WIN, или кабель PC/PPI соединен с компьютером, установите кабель PC/PPI в режим DCE. Если вы используете кабель PC/PPI с модемом (который является устройством DCE), установите кабель PC/PPI в режим DTE. Это устраняет необходимость устанавливать адаптер безмодемного режима между кабелем PC/PPI и модемом. В зависимости от имеющегося на модеме разъема, вам, возможно, еще понадобится переходник 9/25 контактов.

Переключатель 6 кабеля PC/PPI выбирает режим сигнала RTS на разъеме RS–232. Выбор «RTS for XMT [RTS для XMT]» приводит к тому, что сигнал RTS активен, когда S7–200 ведет передачу через порт RS–485, и сигнал RTS не активен, когда S7–200 принимает данные. Выбор «RTS Always [RTS всегда]» приводит к тому, что сигнал RTS всегда активен в порту RS–232 кабеля PC/PPI, независимо от того, ведет ли S7–200 передачу или прием. Переключатель 6 влияет на сигнал RTS только в том случае, если кабель PC/PPI установлен в режим DTE.

На рис. 7–25 показано назначение контактов для обычного адаптера модема.

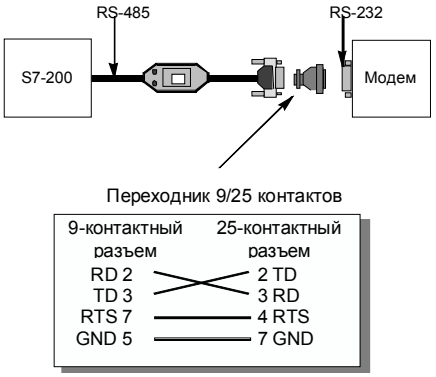


Рис. 7–25. Назначение контактов адаптера

В таблице 7–10 приведены номера и функции контактов для портов RS–485 и RS–232 кабеля PC/PPI в режиме DTE. В таблице 7–11 приведены номера и функции контактов для портов RS–485 и RS–232 кабеля PC/PPI в режиме DCE. Кабель PC/PPI поставляет сигнал RTS только тогда, когда он находится в режиме DTE.

Таблица 7–10. Назначение контактов штепсельных разъемов RS–485 и RS–232 в режиме DTE

Назначение контактов разъема RS–485		**Назначение контактов разъема RS–232 в режиме DTE**[1]	
Контакт	**Описание сигнала**	**Контакт**	**Описание сигнала**
1	Земля (логическая земля RS-485)	1	*Не используется:* Распознавание носителя информации (DCD)
2	Обратный провод 24 В (логическая земля RS–485)	2	Прием данных (RD) (вход в кабель PC/PPI)
3	Сигнал B (RxD/TxD+)	3	Передача данных (TD) (выход из кабеля PC/PPI)
4	RTS (уровень TTL)	4	*Не используется:* Терминал готов к передаче данных (DTR)
5	Земля (логическая земля RS-485)	5	Земля (логическая земля RS–232)
6	NC (не соединен)	6	*Не используется:* Набор данных готов (DSR)
7	Питание 24 В	7	Запрос на передачу (RTS) (устанавливается переключателем)
8	Сигнал A (RxD/TxD-)	8	*Не используется:* Готовность к приему (CTS)
9	Выбор протокола	9	*Не используется:* Кольцевой индикатор (RI)

[1] Для модемов, возможно, потребуется переходник розетка/вилка, а также 9/25 контактов

Таблица 7–11. Назначение контактов штепсельных разъемов RS–485 и RS–232 в режиме DCE

Назначение контактов разъема RS–485		**Назначение контактов разъема RS–232 в режиме DCE**	
Контакт	**Описание сигнала**	**Контакт**	**Описание сигнала**
1	Земля (логическая земля RS-485)	1	*Не используется:* Распознавание носителя информации (DCD)
2	Обратный провод 24 В (логическая земля RS–485)	2	Прием данных (RD) (выход из кабеля PC/PPI)
3	Сигнал B (RxD/TxD+)	3	Передача данных (TD) (вход в кабель PC/PPI)
4	RTS (уровень TTL)	4	*Не используется:* Терминал готов к передаче данных (DTR)
5	Земля (логическая земля RS-485)	5	Земля (логическая земля RS–232)
6	NC (не соединен)	6	*Не используется:* Набор данных готов (DSR)
7	Питание 24 В	7	*Не используется:* Запрос на передачу (RTS)
8	Сигнал A (RxD/TxD-)	8	*Не используется:* Готовность к приему (CTS)
9	Выбор протокола	9	*Не используется:* Кольцевой индикатор (RI)

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 каждый второй раз, когда он получает маркер. 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. 7–26 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).

Советы и уловки

Описание сетей с передачей маркера вы найдете в советах и уловках (Tips and Tricks) на компакт-диске с документацией. См. Совет 42.

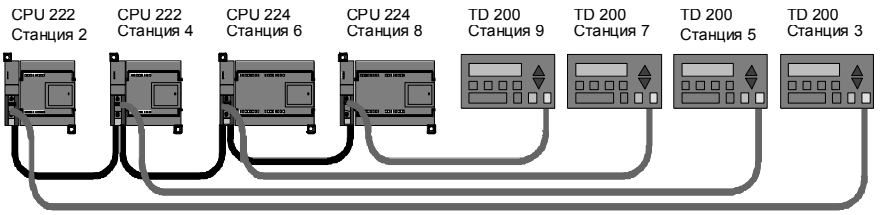


Рис. 7–26. Пример сети с передачей маркера

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. 7–26, 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 7–12 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 7–12. 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. 7–27, 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 соединения для STEP 7-Micro/WIN и одно соединения 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, например, PPI Advanced.

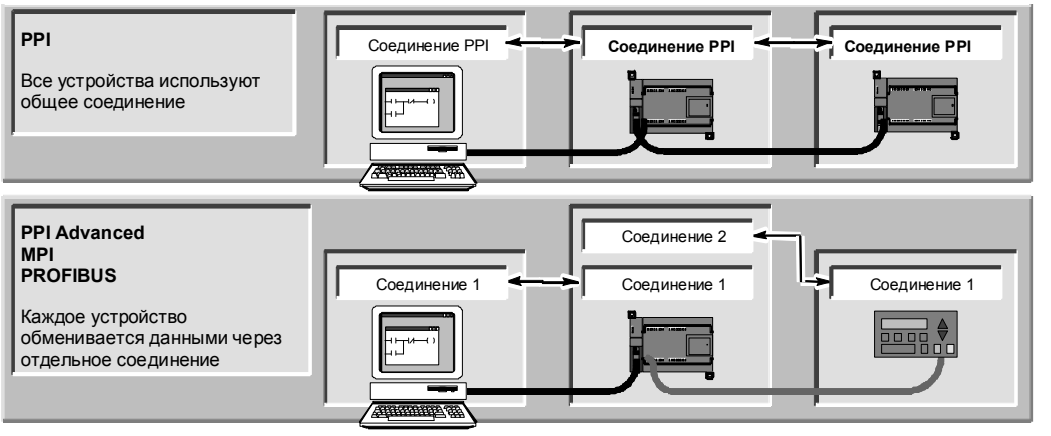


Рис. 7–27. Управление коммуникационными соединениями

Как показано в таблице 7–13, 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 7–13. Характеристики модулей CPU S7–200 и EM 277

Модуль	Скорость передачи	Соединения	Поддерживаемые протоколы
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] Для порта 0 и порта 1 CPU S7–200 использование MPI и PROFIBUS возможно только в том случае, если производится обмен данными с S7–200, являющимся slave-устройством.

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. 7–28 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.

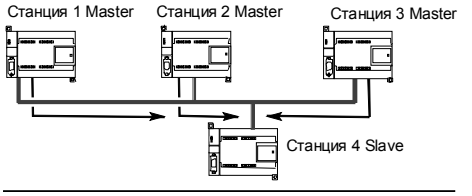


Рис. 7–28. Коммуникационный конфликт

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. 7–29. 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.

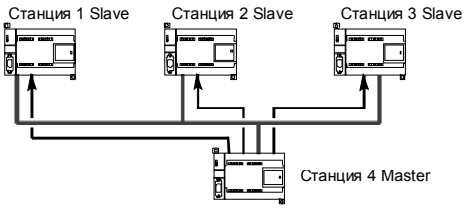


Рис. 7–29. Устранение конфликта

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 7–14. 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 7–14.

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 отношению к времени 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 7–14, целевое время оборота маркера для этой сети равно 0,613 секунды. Однако, в соответствии с данными о производительности, приведенными в таблице 7–12, 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-устрпойств 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 включении 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 7–12, 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 7–12 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	=	0,63 с
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 7–14.) 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 maste-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.

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 236
Отображение состояния программы 238
Использование таблицы состояний для контроля и изменения данных в S7–200 239
Принудительное присваивание значений 240
Исполнение программы в течение определенного количества циклов 240
Указания по устранению неисправностей аппаратуры 241

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.

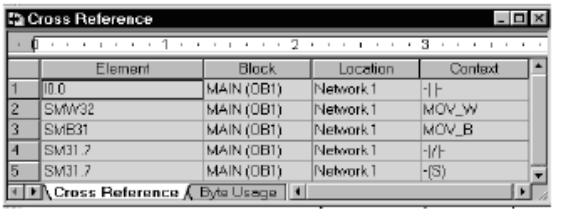
Использование таблицы перекрестных ссылок для проверки ссылок в вашей программе

Перекрестные ссылки

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 указывает для каждого операнда (столбец Element) программный блок (столбец Block), сегмент (network) или строку (line) (в столбце location [местоположение]) и команду (в столбце context), где он используется.

Вы имеете возможность переключаться между абсолютным и символическим представлением, чтобы изменять отображение всех операндов.



Cross Reference

	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	-| |-
2	SMW32	MAIN (OB1)	Network 1	MOV_W
3	SMB31	MAIN (OB1)	Network 1	MOV_B
4	SM31.7	MAIN (OB1)	Network 1	-|/|-
5	SM31.7	MAIN (OB1)	Network 1	-(S)

Cross Reference / Byte Usage

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 224, начиная с версии 1.10, и CPU 226, начиная с версии 1.00, 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 пункт Cross Reference [Перекрестные ссылки] в меню View [Вид]. 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.
- Если вы удалили команду назначения прерывания, но не удалили программу обработки прерываний, то S7–200 продолжает выполнять программу обработки прерываний до выключения S7–200 или его перевода в STOP. Аналогично, при удалении команды отсоединения прерывания продолжают выполняться до выключения S7–200 или его перевода 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.

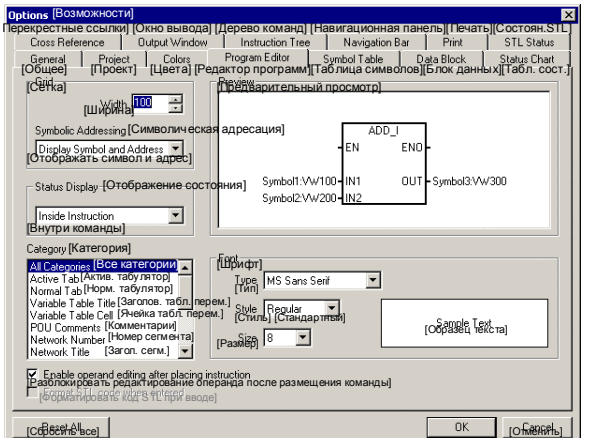


Рис. 8–2. Возможности отображения состояния

Отображение состояния 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]**, а затем выберите закладку STL Status [Состояние STL]. См. рис. 8–3.

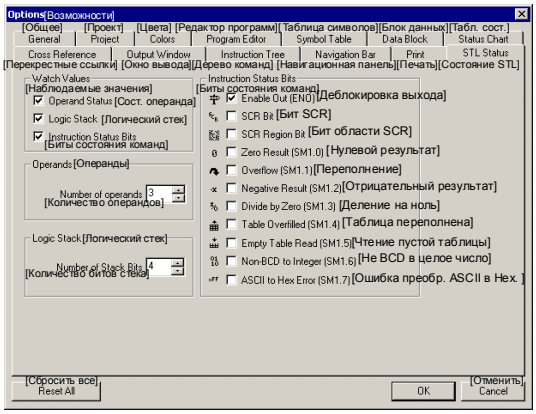
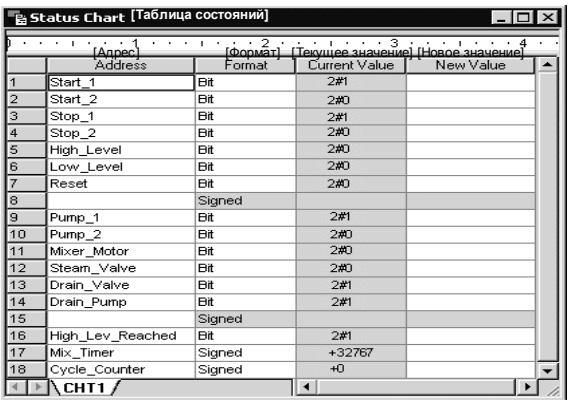


Рис. 8–3. Возможности отображения состояния STL

Использование таблицы состояний для наблюдения 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.



	Address	Format	Current Value	New Value
1	Start_1	Bit	2#1	
2	Start_2	Bit	2#0	
3	Stop_1	Bit	2#1	
4	Stop_2	Bit	2#0	
5	High_Level	Bit	2#0	
6	Low_Level	Bit	2#0	
7	Reset	Bit	2#0	
8		Signed		
9	Pump_1	Bit	2#1	
10	Pump_2	Bit	2#0	
11	Mixer_Motor	Bit	2#0	
12	Steam_Valve	Bit	2#0	
13	Drain_Valve	Bit	2#1	
14	Drain_Pump	Bit	2#1	
15		Signed		
16	High_Lev_Reached	Bit	2#1	
17	Mix_Timer	Signed	+32767	
18	Cycle_Counter	Signed	+0	

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 ЭСППЗУ 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. Непосредственно 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.

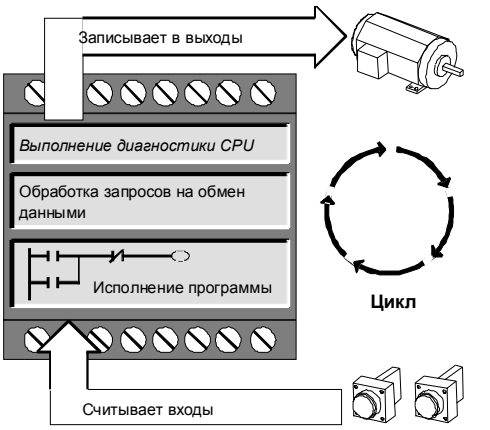


Рис. 8–5. Цикл S7–200

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	■ Управляемое устройство вызвало бросок напряжения, который повредил выход ■ Ошибка в программе пользователя ■ Плохо или неправильно подключена проводка ■ Чрезмерная нагрузка ■ Принудительный режим работы выхода	■ 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)	В следующем списке приведены наиболее часто встречающиеся коды ошибок и их причины: ■ Ошибка программирования пользователя - 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)	Прочитайте код фатальной ошибки и обратитесь к Приложению C за информацией о типе ошибки: ■ В случае ошибок программирования проверьте использование команд FOR, NEXT, JMP, LBL и сравнения. ■ В случае электрических помех: - 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	■ Сгорел предохранитель ■ Перепутана полярность проводов 24 В ■ Неправильное напряжение	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.
Неустойчивая работа, связанная с силовыми устройствами	■ Плохое заземление ■ Прокладка проводки внутри шкафа управления ■ Слишком малое время задержки для входных фильтров	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.
Сеть связи повреждена при подключении к внешнему устройству. Поврежден или порт на компьютере, или порт на S7–200, или кабель PC/PPI.	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.	■ 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. За номерами для заказа оборудования S7–200 обратитесь к Приложению E.
Другие коммуникационные проблемы (STEP 7-Micro/WIN)	Обратитесь к главе 7 за информацией об обмене данными в сети.	
Обработка ошибок	Обратитесь к Приложению C за информацией о кодах ошибок.	

Создание программы для модуля позиционирования

9

Модуль позиционирования EM 253 – это специальный функциональный модуль S7–200, генерирующий последовательности импульсов, используемых для управления скоростью вращения и положением шаговых двигателей или серводвигателей без обратной связи. Он обменивается данными с S7–200 через шину расширения ввода/вывода и появляется в конфигурации входов/выходов как интеллектуальный модуль с восемью цифровыми выходами.

На основе информации, хранящейся в памяти переменных S7–200, модуль позиционирования генерирует последовательности импульсов, необходимые для управления движением.

Для упрощения управлением позиционированием в ваших приложениях STEP 7-Micro/WIN предоставляет в ваше распоряжение мастер управления позиционированием (Position Control wizard), позволяющий полностью сконфигурировать модуль позиционирования за несколько минут. STEP 7-Micro/WIN имеет также панель управления, позволяющую управлять, наблюдать и тестировать процессы позиционирования.

В этой главе

Функции модуля позиционирования	244
Конфигурирование модуля позиционирования	246
Команды позиционирования, созданные мастером управления позиционированием	257
Примеры программ для модуля позиционирования	269
Наблюдение за модулем позиционирования с помощью панели управления EM 253	274
Коды ошибок для модуля и команд позиционирования	276
Для опытных пользователей	278

Функции модуля позиционирования

Модуль позиционирования предоставляет функциональные возможности и мощности, необходимые для одноосного управления позиционированием без обратной связи:

- Скоростное управление с диапазоном от 12 импульсов в секунду до 200 000 импульсов в секунду
- Поддержка толчкообразного (S-образная кривая) и линейного ускорения и замедления
- Настраиваемая измерительная система, позволяющая вводить данные в физических единицах измерения (например, дюймы или сантиметры) или в виде количества импульсов
- Настраиваемая компенсация люфта
- Поддержка абсолютного, относительного и ручного методов управления позиционированием
- Непрерывное функционирование
- До 25 профилей перемещения, имеющих до 4 изменений скорости на профиль
- Четыре различных режима поиска опорной точки с выбором начального направления поиска и направления приближения в конце перемещения для каждой последовательности
- Сменные клеммные блоки для полевой проводки для облегчения монтажа и демонтажа

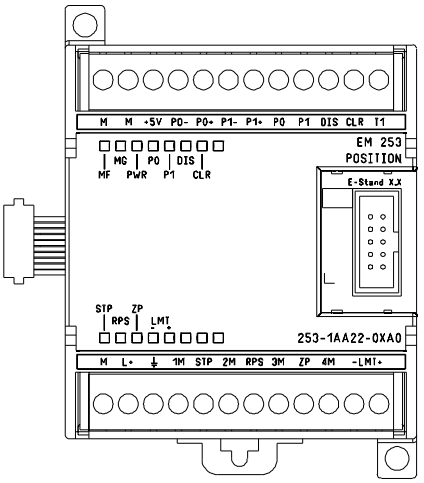


Рис. 9–1. Модуль позиционирования EM 253

STEP 7-Micro/WIN используется для создания информации о конфигурациях и профилях, используемой модулем позиционирования. Эта информация загружается в S7–200 с вашими программными блоками. Так как вся информация, необходимая для управления позиционированием, хранится в S7–200, то вы можете заменить модуль позиционирования без перепрограммирования и перенастройки модуля.

S7–200 резервирует 8 битов выходного регистра образа процесса (область памяти Q) для интерфейса с модулем позиционирования. Ваша прикладная программа в S7–200 использует эти биты для управления функционированием модуля позиционирования. Эти 8 выходных битов не связаны с физическими полевыми выходами модуля позиционирования.

Модуль позиционирования предоставляет в распоряжение пять цифровых входов и четыре цифровых выхода, обеспечивающих интерфейс с вашим приложением. См. таблицу 9–1. Эти входы и выходы являются локальными для модуля позиционирования. В приложении A приведены подробные технические данные для модуля позиционирования, включая схемы соединений для подключения модуля позиционирования к некоторым часто используемым устройствам управления приводами и усилительным устройствам.

Таблица 9–1. Входы и выходы модуля позиционирования

Сигнал	Описание
STP	Вход STP заставляет модуль остановить текущее перемещение. Желаемый режим для STP можно выбрать внутри мастера управления позиционированием.
RPS	Вход RPS (Reference Point Switch – переключатель опорной точки) устанавливает опорную точку или начальное положение для абсолютных операций перемещения.
ZP	Вход ZP (Zero Pulse - нулевой импульс) помогает установить опорную точку или начальное положение. Обычно устройство управления двигателем или усилитель выдает один нулевой импульс на оборот двигателя.
LMT+ LMT-	Входы LMT+ и LMT- устанавливают максимальные пределы для перемещения. Мастер управления позиционированием позволяет настроить работу входов LMT+ и LMT-.
P0 P1 P0+, P0- P1+, P1-	P0 и P1 – это импульсные выходы, организованные с помощью транзисторов с открытым стоком, которые управляют движением и направлением движения двигателя. P0+, P0- и P1+, P1 – это дифференциальные импульсные выходы, которые предоставляют такие же функциональные возможности, как P0 и P1 соответственно, но обеспечивают более высокое качество сигнала. Транзисторные выходы с открытым стоком и дифференциальные выходы активны все одновременно. Выбор тех или иных импульсных выходов производится в зависимости от требований к интерфейсу, предъявляемых устройством управления двигателем или усилителем.
DIS	DIS – это транзисторный выход, используемый для активизации и деактивизации устройства управления двигателем или усилителя.
CLR	CLR – это транзисторный выход, используемый для очистки регистра счета сервоимпульсов.

Программирование модуля позиционирования

STEP 7-Micro/WIN предоставляет в распоряжение простые в использовании инструментальные средства для конфигурирования и программирования позиционного модуля. Действуйте следующим образом:

1. Сконфигурируйте модуль позиционирования. В STEP 7-Micro/WIN имеется мастер управления позиционированием (Position Control wizard) для создания таблицы конфигурации и профилей и команд позиционирования. На стр. 246 вы найдете дальнейшую информацию о конфигурировании модуля позиционирования.
2. Протестируйте работу модуля позиционирования. В STEP 7-Micro/WIN имеется панель управления EM 253 для проверки подключения проводки к входам и выходам, конфигурации модуля позиционирования и функционирования профилей перемещений. На стр. 274 вы найдете дальнейшую информацию о панели управления EM 253.
3. Создайте программу, подлежащую исполнению в S7–200. Мастер управления позиционированием автоматически генерирует команды позиционирования, которые вы вставляете в свою программу. На стр. 257 вы найдете дальнейшую информацию о командах позиционирования. Вставьте в свою программу следующие команды:
   - Для активизации модуля позиционирования вставьте команду POSx_CTRL. Используйте бит SM0.0 (всегда включено), чтобы обеспечить исполнение этой команды в каждом цикле.
   - Для перемещения двигателя в определенное положение используйте команду POSx_GOTO или POSx_RUN. Команда POSx_GOTO осуществляет перемещение к месту, указанному входами вашей программы. Команда POSx_RUN исполняет профили перемещения, созданные с помощью мастера управления позиционированием.
   - Для использования при перемещении абсолютных координат вы должны установить нулевую позицию для своего приложения. Для установки нулевой позиции используйте команду POSx_RSEEK или POSx_LDPOS.
   - Остальные 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.

Совет
Дальнейшую информацию о подключении модуля позиционирования к различным устройствам управления шаговыми двигателями вы найдете в Приложении A.

Совет
Для согласования с настройками по умолчанию в мастере управления позиционированием установите DIP-перключатели на устройстве управления шаговым двигателем на 10 000 импульсов на оборот.

Конфигурирование модуля позиционирования

Управление позиционированием

Вы должны создать для модуля позиционирования таблицу конфигурации и профилей, чтобы модуль мог управлять перемещением в вашем приложении. Мастер управления позиционированием делает процесс конфигурирования быстрым и легким, ведя вас шаг за шагом по процессу конфигурирования. Подробную информацию о таблице конфигурации и профилей вы найдете под заголовком "Для опытных пользователей" на стр. 278.

Мастер управления позиционированием дает возможность создавать таблицу конфигурации и профилей в режиме offline. Вы можете создать конфигурация без подключения к CPU S7–200 с установленным модулем позиционирования.

Для запуска мастера управления позиционированием ваш проект должен быть скомпилирован и настроен на режим символической адресации.

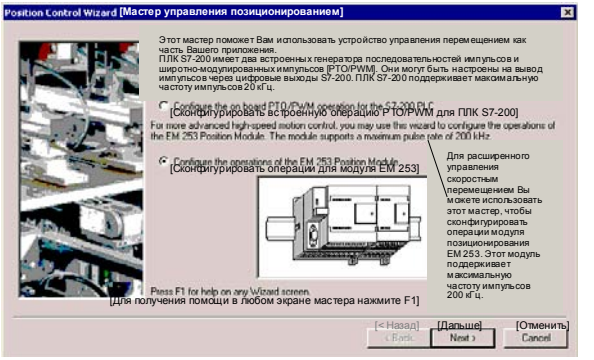


Рис. 9–2. Мастер управления позиционированием

Для запуска мастера управления позиционированием щелкните на пиктограмме Tools [Инструментальные средства], находящейся на навигационной панели, а затем дважды щелкните на пиктограмме Position Control Wizard [Мастер управления позиционированием], или выберите команду меню **Tools > Position Control Wizard [Инструментальные средства > Управление позиционированием]**.

Мастер управления позиционированием позволяет сконфигурировать работу модуля позиционирования или функции PTO/PWM команды вывода импульсов. После того как вы выберете опцию для модуля позиционирования и щелкнете на кнопке Next [Дальше], мастер проведет вас по шагам, необходимым для конфигурирования модуля позиционирования.

Ввод положения модуля позиционирования

Вы должны определить параметры для модуля и набор профилей перемещения для своего приложения, введя тип модуля и его положение. Мастер управления позиционированием упрощает эту задачу, автоматически считывая положение интеллектуального модуля. Вам нужно только щелкнуть на кнопке Read Modules [Прочитать модули].

Для CPU S7–200 с ПЗУ версий, меньшей 1.2, вы должны установить интеллектуальный модуль рядом с CPU, чтобы мастер управления позиционированием мог сконфигурировать модуль.

Выбор вида измерения

Вы должны выбрать систему измерения, которая должна использоваться при конфигурировании. Вы можете выбирать между физическими единицами и импульсами. Если вы выбираете импульсы, то вам не нужно указывать никакой другой информации. Если вы выбираете физические единицы, то вы должны ввести следующие данные: количество импульсов, необходимо для совершения одного оборота двигателя (см. спецификацию для своего двигателя или привода), базовую единицу измерения (например, дюйм, фут, миллиметр или сантиметр) и величину перемещения (или «единицы») на один оборот двигателя.

В STEP 7-Micro/WIN имеется панель управления EM253, позволяющая изменять количество единиц на оборот после конфигурирования позиционного модуля.

Если вы позднее изменяете систему измерения, то вы должны удалить всю конфигурацию, включая команды, сгенерированные мастером управления позиционированием. Затем вы должны ввести данные, согласованные с новой системой измерения.

Редактирование конфигураций входов и выходов, установленных по умолчанию

В мастере управления позиционированием имеются расширенные возможности, позволяющие просматривать и редактировать конфигурации входов и выходов, установленные по умолчанию для модуля позиционирования:

- В закладке Input Active Levels [Уровни активизации входов] можно изменять настройки для уровня активизации. Если уровень установлен на High [Высокий], то считывается логическая 1, когда через вход протекает ток. Если уровень установлен на Low [Низкий], то логическая 1 считывается, когда ток через вход не протекает. Уровень логической 1 всегда интерпретируется как условие активности. Светодиоды горят, когда ток протекает через вход, независимо от уровня активизации. (По умолчанию активен высокий уровень).

- В закладке Input Filter Times [Времена входного фильтра] вы можете указать время задержки (в диапазоне от 0,20 до 12,80 мс) для фильтрации входов STP, RPS, LMT+ и LMT-. Эта задержка помогает отфильтровать шум во входной проводке, который может вызвать непредвиденные изменения состояний входов. (Умолчание = 6,4 мс)
- В закладке Pulse and Directional Outputs [Импульсные выходы и выходы направления] вы можете указать метод управления направлением. Сначала вы должны указать полярность выходов.

Выбор положительной полярности

Для приложения, использующего положительную полярность, выберите для настройки своего привода и ориентации вашего приложения один из следующих методов (см. рис. 9–3):

- Модуль позиционирования выдает импульсы из выхода P0 для положительного направления вращения и из выхода P1 для отрицательного направления вращения.
- Модуль позиционирования выдает импульсы из выхода P0. Модуль включает выход P1 для положительного направления вращения и выключает выход P1 для отрицательного направления вращения. (Это настройка по умолчанию.)

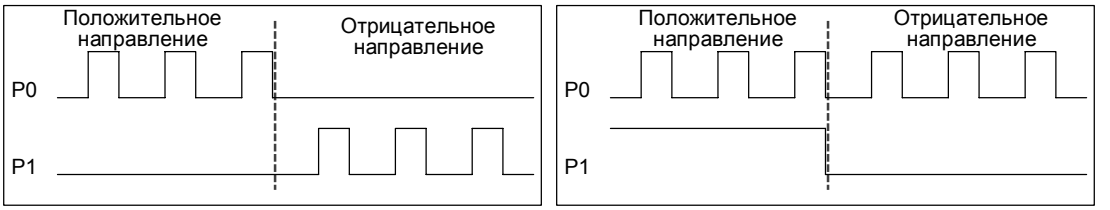


Рис. 9–3. Возможности вращения при положительной полярности

Выбор отрицательной полярности

Для приложения, использующего отрицательную полярность, выберите для настройки своего привода и ориентации вашего приложения один из следующих методов (см. рис. 9–4):

- Модуль позиционирования выдает импульсы из выхода P0 для отрицательного направления вращения и из выхода P1 для положительного направления вращения.
- Модуль позиционирования выдает импульсы из выхода P0. Модуль выключает выход P1 для положительного направления вращения и включает выход P1 для отрицательного направления вращения.

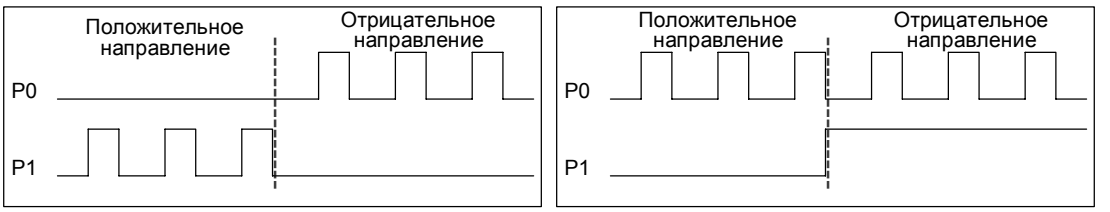


Рис. 9–4. Возможности вращения при отрицательной полярности

Настройка реакции модуля на физические входы

Вы должны указать, как модуль позиционирования реагирует на выключатели LMT+ и LMT- и на вход STP: нет действия (игнорировать входное условие), останов с задержкой (по умолчанию) или немедленный останов.

Предупреждение

Управляющие устройства могут выходить из строя в условиях, когда безопасность не обеспечивается, что может привести к непредсказуемому поведению управляемого оборудования. Такие непредсказуемые действия могут привести к гибели или тяжким телесным повреждениям персонала и/или к материальному ущербу.

Функции ограничения и останова в модуле позиционирования реализованы с помощью электронной логики, которая не обеспечивает такого уровня защиты, как электромеханические управляющие устройства. Поэтому позаботьтесь о функции аварийного отключения, электромеханических или резервных предохранительных устройствах, действующих независимо от модуля позиционирования и CPU S7–200.

Ввод максимальной скорости вращения и скорости вращения при пуске и останове

Вы должны указать для своего приложения максимальную скорость (MAX_SPEED) и скорость при пуске и останове (SS_SPEED):

- MAX_SPEED: Введите значение для оптимальной рабочей скорости вашего приложения с учетом возможного вращающего момента вашего двигателя. Вращающий момент, необходимый для привода нагрузки, определяется трением, инерцией и временами ускорения и замедления. Мастер управления позиционированием рассчитывает и отображает минимальную скорость вращения, которой может управлять модуль позиционирования для указанного значения MAX_SPEED.
- SS_SPEED: Введите значение в рамках возможностей вашего двигателя приводить в движение нагрузку при малых скоростях. Если значение SS_SPEED слишком мало, то двигатель и нагрузка могут совершать колебания или перемещаться короткими прыжками в начале и конце пути. Если значение SS_SPEED слишком велико, то двигатель может терять импульсы при запуске, а нагрузка может увлекать двигатель за собой при попытке останова.

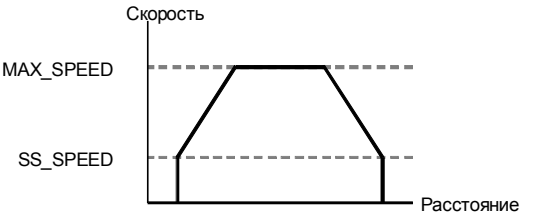


Рис. 9–5. Максимальная скорость и скорость при пуске и останове

В спецификациях двигателей скорость вращения при пуске и останове при определенной нагрузке указывается различными способами. Обычно значение SS_SPEED составляет от 5 до 15% от значения MAX_SPEED. Значение SS_SPEED должно быть больше, чем минимальная скорость, указанная в спецификации для MAX_SPEED.

Для правильного выбора скоростей для вашего приложения обратитесь к спецификации своего двигателя. На рис. 9–6 типичная нагрузочная кривая (вращающий момент/скорость) двигателя.

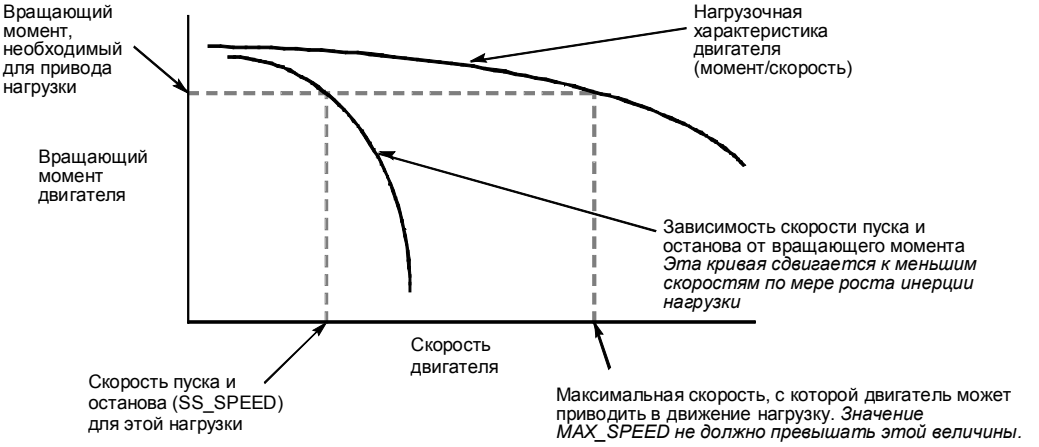


Рис. 9–6. Типичная нагрузочная кривая для двигателя

Ввод параметров стартстопного режима

Команда JOG, реализующая работу в стартстопном режиме, используется для ручного перемещения инструмента в желаемое место. С помощью мастера управления позиционированием устанавливаются следующие параметры стартстопного режима:

- JOG_SPEED: Скорость двигателя при стартстопном режиме – максимальная скорость, которая может быть получена, команда JOG активна.
- JOG_INCREMENT: Расстояние, на которое перемещается инструмент при кратковременном применении команды JOG.

На рис. 9–7 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 расстояние, указанное в параметре JOG_INCREMENT, со скоростью, определяемой параметром SS_SPEED. Если команда JOG все еще активна по истечении 0,5 секунды, то модуль позиционирования ускоряется до скорости, указанной в параметре JOG_SPEED. Движение продолжается до завершения команды JOG. Затем модуль позиционирования замедляется до остановки. Команду JOG можно активизировать из панели управления EM 253 или с помощью команды позиционирования.

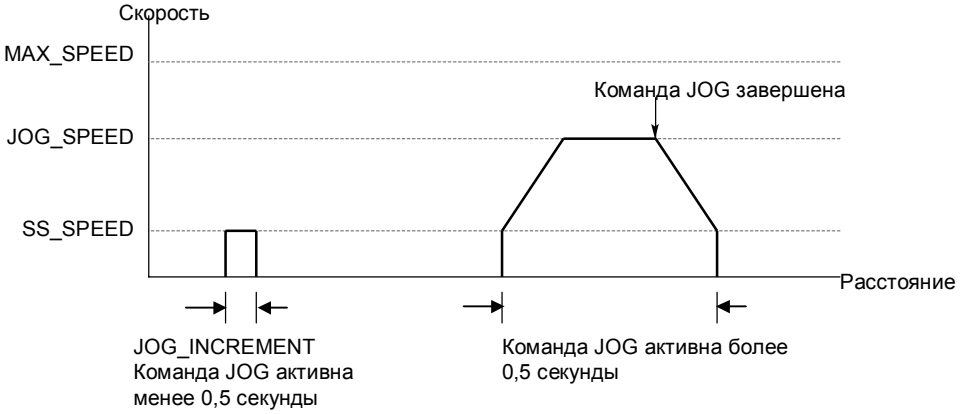


Рис. 9–7. Представление стартстопного режима

Ввод времен ускорения и замедления

В качестве части конфигурации модуля позиционирования устанавливаются времена ускорения и замедления. По умолчанию каждое из этих времен устанавливается равным 1 секунде. Обычно двигатели могут работать менее 1 секунды. Следующие времена задаются в миллисекундах:

- ACCEL_TIME: Время, необходимое для ускорения двигателя от SS_SPEED до MAX_SPEED.
  Умолчание = 1000 мс
- DECEL_TIME: Время, необходимое для замедления двигателя от MAX_SPEED до SS_SPEED.
  Умолчание = 1000 мс

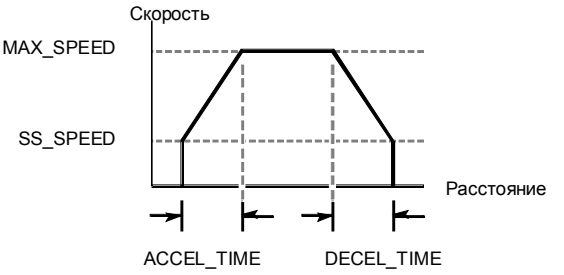


Рис. 9–8. Времена ускорения и замедления

Совет
Времена ускорения и замедления двигателя определяются методом проб и ошибок. Сначала с помощью мастера управления позиционированием введите большее значение. Затем во время тестирования своего приложения вы можете изменять нужным образом это значение в панели управления EM 253. Оптимизируйте эти установки для приложения, постепенно уменьшая эти времена, пока двигатель не начнет останавливаться.

Ввод времени компенсации толчков

Компенсация толчков обеспечивает плавное управление позиционированием путем уменьшения толчков (изменений скорости) на участках ускорения и замедления профиля перемещения. См. рис. 9–9. Уменьшение толчков улучшает характеристики слежения. Компенсация толчков известна также как «профилирование S-кривой». Компенсация толчков может применяться только к простым одношаговым профилям. Эта компенсация применяется равным образом на начальном и конечном участках кривых ускорения и замедления. Компенсация толчков не применяется к первому и последнему шагу между нулевой скоростью и SS_SPEED.

Компенсация толчков определяется вводом значения времени (JERK_TIME). Это время, необходимое для изменения ускорения от нуля до максимального значения, определяемого параметрами MAX_SPEED, SS_SPEED и ACCEL_TIME, или, соответственно, DECEL_TIME. Увеличение времени компенсации толчка обеспечивает плавную работу при меньшем увеличении времени цикла, чем то, что было бы получено простым увеличением ACCEL_TIME и DECEL_TIME. Нулевое значение означает, что компенсация не производится.

(Умолчание = 0 мс)

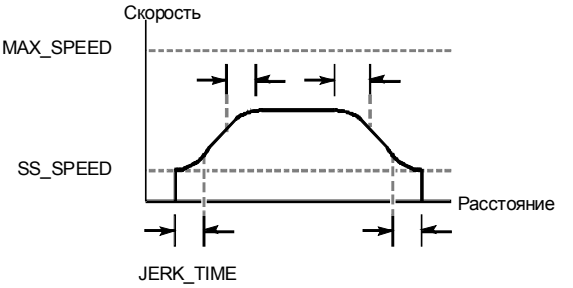


Рис. 9–9. Компенсация толчка

Совет
Хорошим начальным значением для JERK_TIME является 40% от ACCEL_TIME.

Конфигурирование опорной точки и параметров поиска

Если в вашем приложении перемещения задаются от некоторого абсолютного положения, то вы должны установить нулевую позицию, которая фиксирует измерения положения относительно известной точки в физической системе. Один из способов действий состоит в том, что в вашей физической системе обеспечивается опорная точка (reference point, RP). Модуль позиционирования снабжен внешним входом – переключателем опорной точки (reference point switch, RPS), который используется при поиске RP.

Вы можете сконфигурировать параметры для поиска опорной точки (поиска RP), управляющие тем, как ваше приложение, обеспечивающее позиционирование, ищет RP. RP может находиться в центре активной зоны RPS, на краю активной зоны RPS или на расстоянии от края активной зоны RPS, определяемом заданным количеством фронтов на входе нулевого импульса (ZP). Для конфигурирования RP введите следующую информацию:

- Введите скорости поиска RP для двигателя:
  - RP_FAST – это начальная скорость, которую модуль использует при выполнении команды поиска RP. Обычно значение RP_FAST составляет приблизительно 2/3 от значения MAX_SPEED.
  - RP_SLOW – это скорость в конце приближения к RP. Малая скорость используется при приближении к RP, чтобы не проскочить мимо нее. Обычно значение RP_SLOW равно значению SS_SPEED.
- Введите начальное направление поиска (RP_SEEK_DIR) и конечное направление поиска (RP_APPR_DIR) для команды поиска RP. Эти направления указываются как отрицательное или положительное.
  - RP_SEEK_DIR – это начальное направление для операции поиска RP. Обычно это направление от рабочей области к окрестности RP. Важную роль в определении области, где производится поиск RP, играют конечные выключатели. Если при выполнении поиска RP встречается конечный выключатель, то это может привести к изменению направления, что позволяет продолжить поиск. (По умолчанию – отрицательное)
  - RP_APPR_DIR – это конечное направление приближения к RP. Для уменьшения люфта и обеспечения большей точности RP_APPR_DIR имеет то же направление, что и обычный рабочий цикл. (По умолчанию – положительное)

Мастер управления позиционированием располагает расширенными опциями опорной точки, которые позволяют указать смещение RP (RP_OFFSET), являющееся расстоянием между RP и нулевым положением. См. рис. 9–10. RP идентифицируется методом, при котором определяется точное положение относительно RPS. Для конфигурирования смещения RP вводятся следующие значения:

- RP_OFFSET: Расстояние от RP до нулевого положения физической измерительной системы. По умолчанию = 0
- Компенсация люфта: Расстояние, на которое должен переместиться двигатель, чтобы устранить люфт в системе при изменении направления. Компенсация люфта всегда имеет положительное значение. По умолчанию = 0

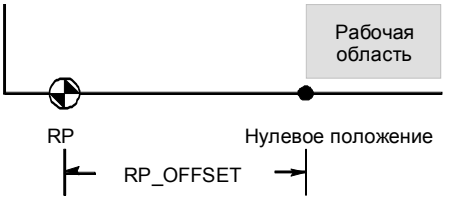


Рис. 9–10. Связь между RP и нулевым положением

Конфигурирование последовательности поиска RP

Вы можете сконфигурировать последовательность, которую модуль позиционирования использует для поиска опорной точки. На рис. 9–11 показана упрощенная диаграмма установленной по умолчанию последовательности поиска опорной точки. Для последовательности поиска RP вы можете выбрать следующие возможности:

- Режим поиска RP 0: Последовательность поиска RP не выполняется
- Режим поиска RP 1: RP находится в том месте, где вход RPS становится активным при приближении со стороны рабочей области (по умолчанию)
- Режим поиска RP 2: RP находится в центре активной области входа RPS.
- Режим поиска RP 3: RP находится вне активной области входа RPS. RP_Z_CNT указывает, сколько отсчетов входа ZP (нулевые импульсы) должно быть получено после активизации входа RPS.
- Режим поиска RP 4: RP, как правило, находится внутри активной области входа RPS. RP_Z_CNT указывает, сколько отсчетов входа ZP (нулевые импульсы) должно быть получено после активизации входа RPS.

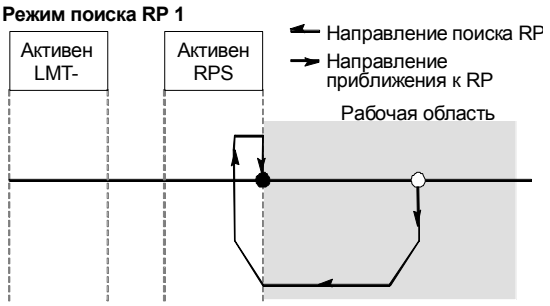


Рис. 9–11. Принятая по умолчанию последовательность поиска RP (упрощено)

Совет
Область 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 генерирует ошибку.

Дальнейшую информацию о различных последовательностях поиска RP для модуля позиционирования вы найдете на рисунках 9–14 - 9–17 на страницах 254 - 255.

Конфигурирование профилей перемещения для модуля позиционирования

Профиль – это заранее определенное описание перемещения, состоящее из одной или нескольких скоростей, с которыми осуществляется перемещение от начальной точки к конечной. Для использования модуля вам нет необходимости определять профиль. Мастер управления позиционированием предоставляет в ваше распоряжение подпрограмму (POSx_GOTO), с помощью которой вы можете управлять перемещениями.

- Количество профилей: Вы можете выбрать не более 25 профилей.
- Адрес для командного байта: Вы должны ввести адрес в области памяти выходов (Q) для командного байта модуля позиционирования. Описание нумерации входов/выходов вы найдете на рис. 4–10 на стр. 31.
- Адрес для таблицы конфигурации и профилей: Вы должны ввести начальный адрес в памяти для таблицы конфигурации и профилей, в которой хранятся данные конфигурации для модуля позиционирования и данные для всех профилей. Данные конфигурации для модуля позиционирования требует 92 байтов памяти переменных, а каждый профиль требует 34 байтов памяти переменных. Например, количество памяти, необходимой для таблицы конфигурации и профилей модуля позиционирования с одним профилем, равно 126 байтам памяти переменных.

  Мастер управления позиционированием может предложить адрес свободной области в памяти переменных нужного размера.

Определение профиля перемещения

Мастер управления позиционированием располагает средствами для определения профиля перемещения, с помощью которых вы определяете все профили перемещения для своего приложения. Для каждого профиля вы выбираете режим работы и определяете характеристики каждого отдельного шага для профиля. Мастер управления позиционированием дает также возможность определить для каждого профиля символическое имя, которое просто вводится при определении профиля. После конфигурирования профиля вы можете сохранить конфигурацию и распечатать копию параметров.

Выбор режима работы для профиля

Конфигурирование профиля производится в соответствии с режимом работы: абсолютное позиционирование, относительное позиционирование, непрерывное вращение с одной скоростью или непрерывное вращение с двумя скоростями. На рис. 9–12 показаны различные режимы работы.

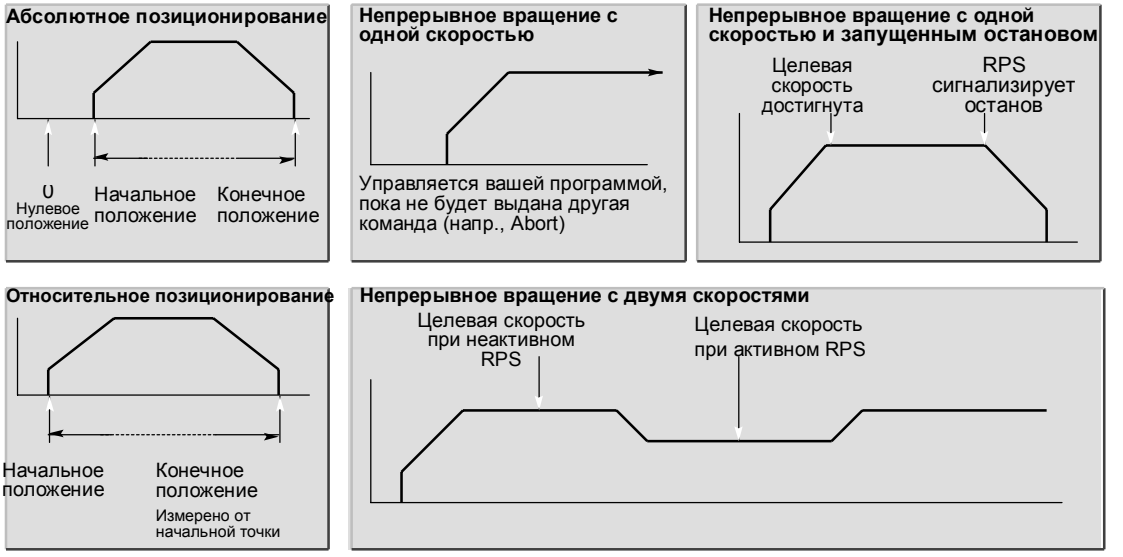


Рис. 9–12. Выбор режимов для модуля позиционирования

Формирование шагов для профиля

Шаг – это фиксированное расстояние, на которое перемещается инструмент, включая расстояние, покрываемое при ускорении и замедлении. Каждый профиль может содержать до 4 отдельных шагов.

Для каждого шага вы определяете целевую скорость и конечное положение. Если вы определяете несколько шагов, просто щелкайте на кнопке New Step [Новый шаг] и вводите информацию для каждого шага профиля. На рис. 9–13 показаны четыре возможных профиля; но возможны и другие комбинации.

Щелкнув на кнопке Plot Step [Представить шаг графически], вы получите графическое представление шага, рассчитанное мастером управления перемещением. Это позволяет легко контролировать и редактировать каждый шаг в интерактивном режиме.

Рис. 9–13. Примеры профилей перемещения

Завершение конфигурирования модуля позиционирования

Сконфигурировав режим работы модуля позиционирования, щелкните на кнопке Finish [Закончить]. После этого мастер управления позиционированием выполняет следующие задачи:

- Вставляет таблицу конфигурации и профилей для модуля в блок данных вашей программы для S7–200
- Создает глобальную таблицу символов для параметров перемещения
- Вставляет подпрограммы с командами перемещения в программный блок проекта, чтобы их можно было использовать в вашем приложении

Вы можете снова запустить мастер управления позиционированием, если вы хотите изменить информацию о конфигурации или профиле.

Совет
Так как мастер управления позиционированием производит изменения в программном блоке, блоке данных и системном блоке, то убедитесь, что все три блока загружены в CPU S7–200. В противном случае у модуля позиционирования, возможно, не будет всех программных компонентов, необходимых для надлежащего функционирования.

Режимы поиска RP, поддерживаемые модулем позиционирования

На следующих рисунках представлены диаграммы различных вариантов для отдельных режимов поиска RP.

- На рис. 9–14 показаны два варианта для режима поиска RP 1. В этом режиме положение RP определяется, в том месте, где вход RPS становится активным при приближении со стороны рабочей области.
- На рис. 9–15 показаны два варианта для режима поиска RP 2. В этом режиме положение RP определяется в центре активной области входа RPS.
- На рис. 9–16 показаны два варианта для режима поиска RP 3. В этом режиме положение RP определяется указанным количеством нулевых импульсов (ZP) вне активной области входа RPS.
- На рис. 9–17 показаны два варианта для режима поиска RP 4. В этом режиме положение RP определяется указанным количеством нулевых импульсов (ZP) внутри активной области входа RPS.

Для каждого режима четыре комбинации направления поиска RP и приближения к RP. (Показаны только две из этих комбинаций.) эти комбинации определяют модель для операции поиска RP. Для каждой из этих комбинаций имеется четыре различных начальных точки:

Рабочие области для каждой диаграммы были выбраны таким образом, чтобы движение от опорной точки к рабочей области требовало перемещения в том же направлении, что и приближение к RP. При таком выборе расположения рабочей области для первого перемещения к этой области после поиска опорной точки все люфты в приводном механизме устраняются.

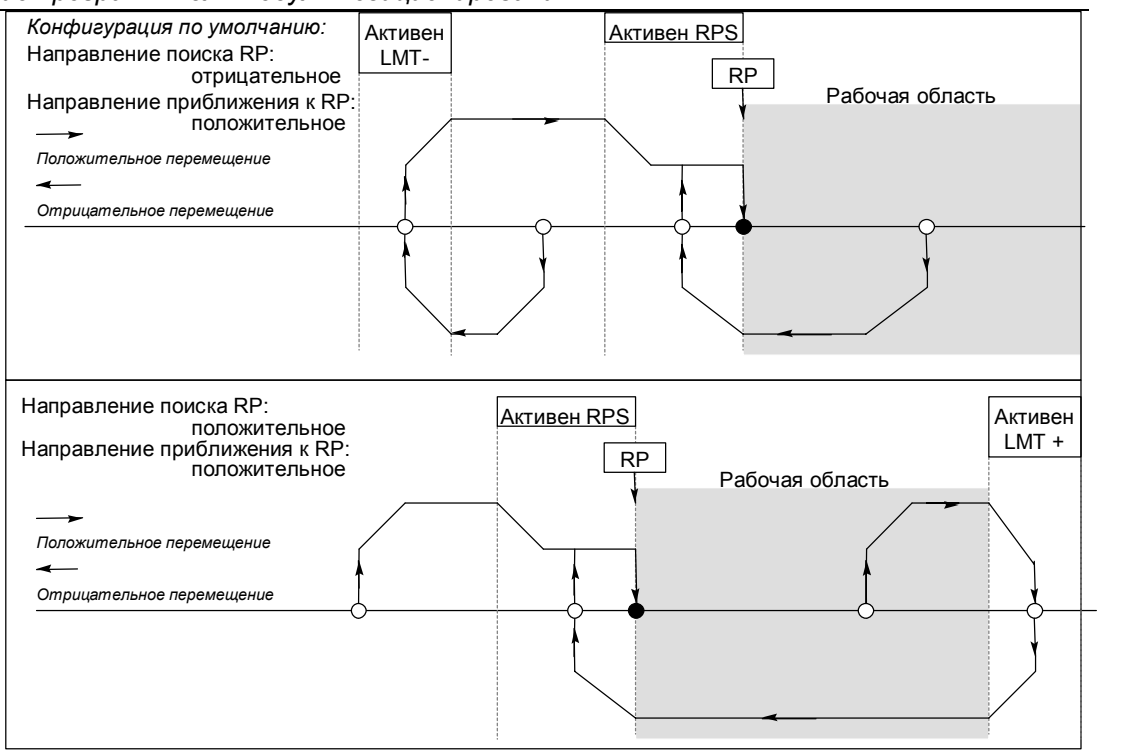


Рис. 9–14. Режим поиска RP 1

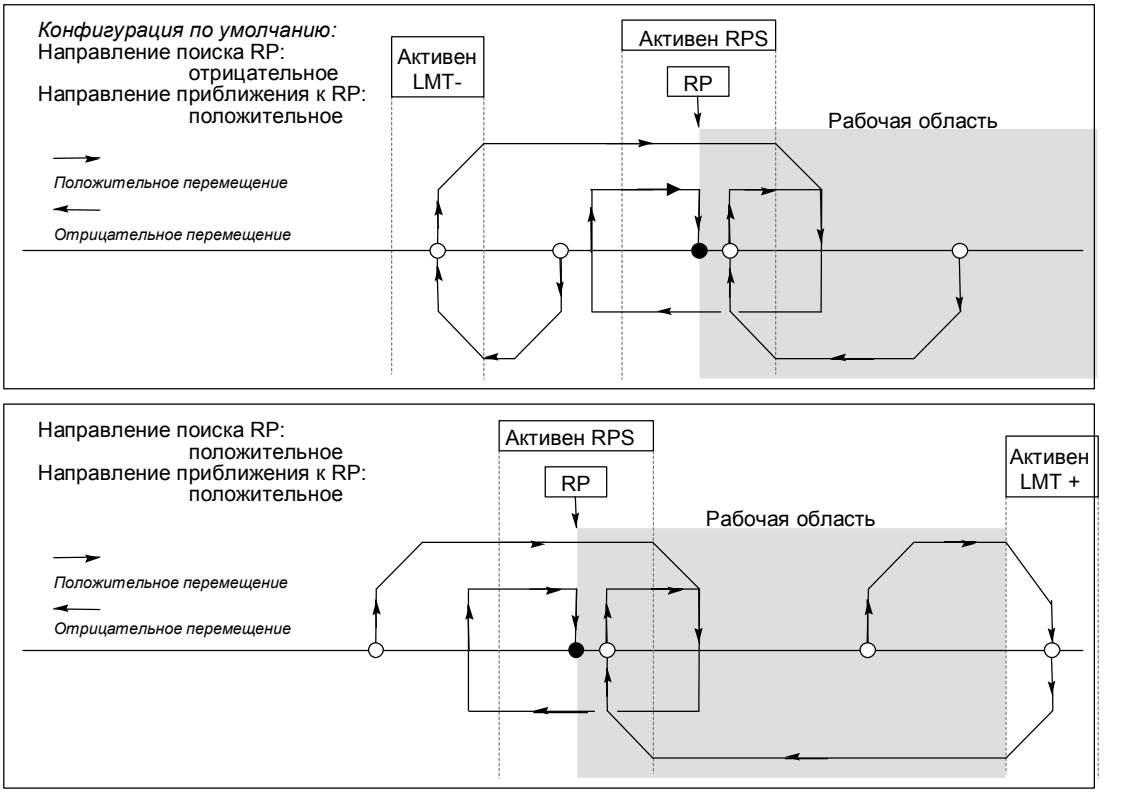


Рис. 9–15. Режим поиска RP 2

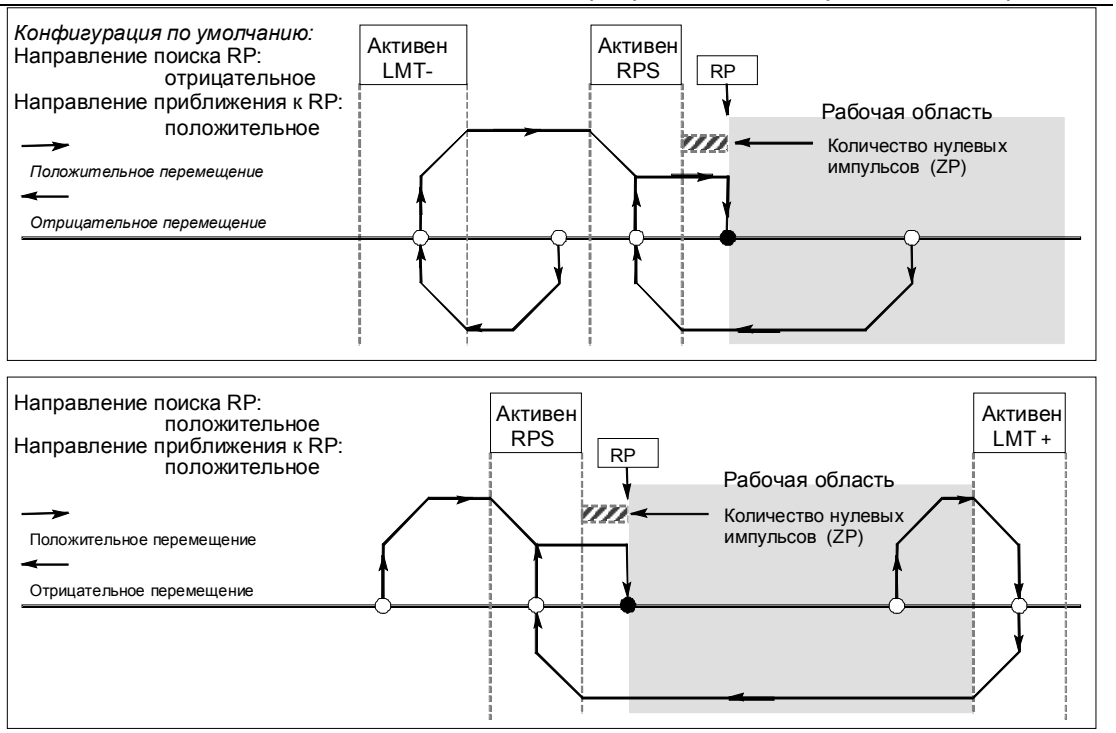


Рис. 9–16. Режим поиска RP 3

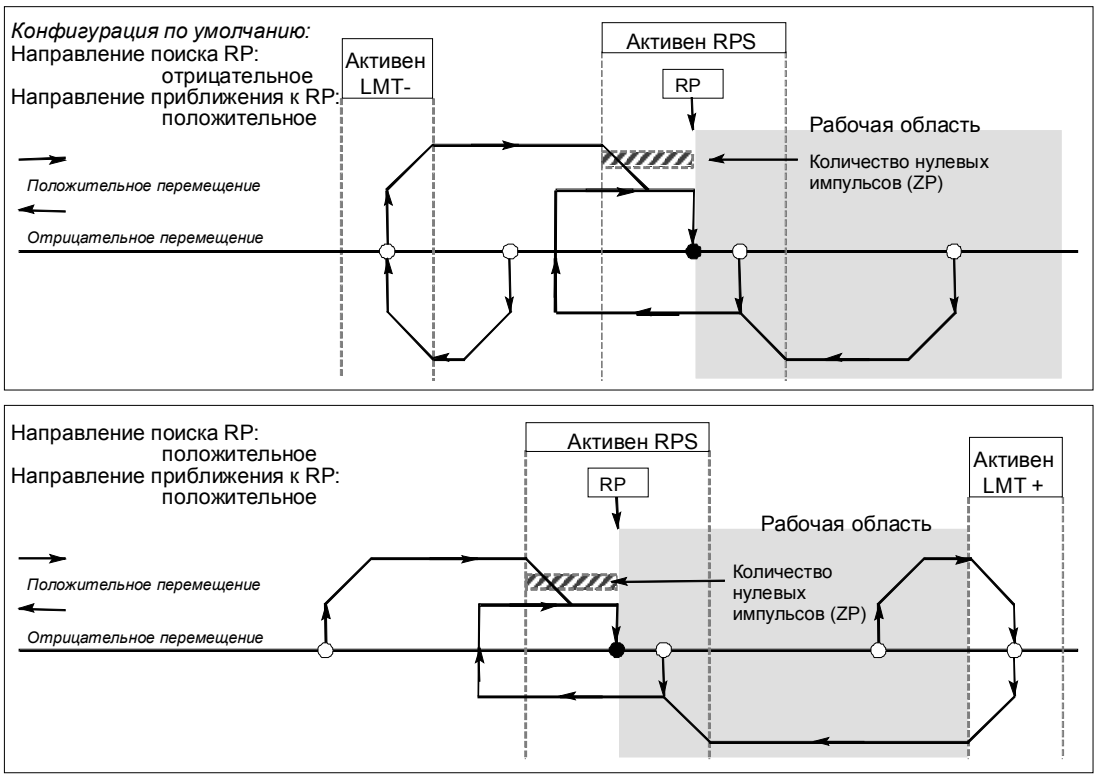


Рис. 9–17. Режим поиска RP 4

Выбор положения рабочей области для устранения люфта

На рис. 9–18 показано положение рабочей области относительно опорной точки (RP), области активности RPS и конечных выключателей (LMT+ и LMT-) для направления приближения, при котором устраняется люфт. Во второй части рисунка рабочая область расположена так, что люфт не устраняется. На рис. 9–18 показан режим поиска RP 3. Такое же расположение рабочей области возможно, хотя и не рекомендуется, для каждой из последовательностей поиска в остальных режимах поиска RP.

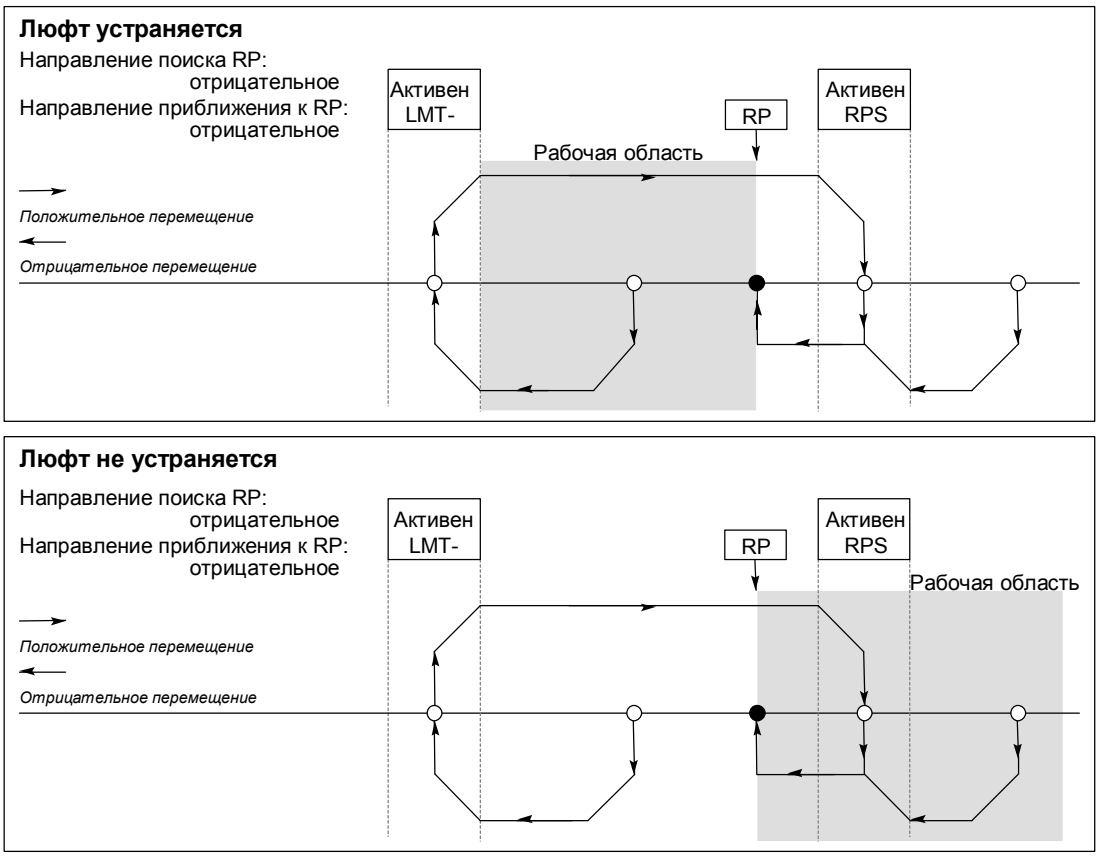


Рис. 9–18. Расположение рабочей области с устранением и без устранения люфта

Команды позиционирования, созданные мастером управления перемещением

Мастер управления позиционированием облегчает управление модулем позиционирования путем создания уникальных подпрограмм с командами на основе установленной вами позиции модуля и выбранных вами вариантов конфигурации. Каждая команда позиционирования имеет префикс "POS*x*_", где *x* указывает позицию модуля. Так как каждая команда позиционирования является подпрограммой, то 11 команд позиционирования используют 11 подпрограмм.

Совет
Команды позиционирования увеличивают потребность вашей программы в памяти не более чем на 1700 байтов. Для уменьшения необходимого объема памяти вы можете удалить неиспользуемые команды позиционирования. Для восстановления удаленных команд позиционирования просто снова запустите мастер управления позиционированием.

Указания по использованию команд позиционирования

Вы должны учитывать, что в каждый данный момент времени может быть активна только одна команда позиционирования.

Команды POSx_RUN и POSx_GOTO могут выполняться в программе 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, когда модуль позиционирования завершит перемещение.

Мастер управления позиционированием автоматически конфигурирует значения для параметров скорости (Speed и C_Speed) и параметров положения (Pos или C_Pos) в соответствии с выбранной системой измерения. Для импульсов эти параметры имеют значения типа DINT. Для физических единиц измерения параметры имеют значения типа REAL в выбранных вами единицах измерения. Например: выбор сантиметров (см) сохраняет параметры положения в виде величин типа REAL в сантиметрах, а параметры скорости в виде величин типа REAL в сантиметрах в секунду (см/сек).

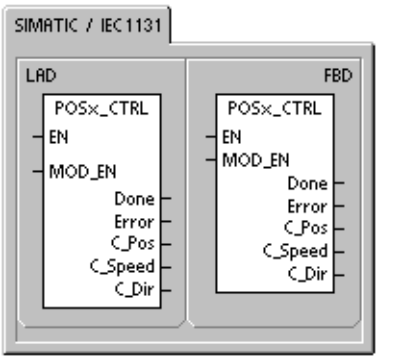
Для конкретных задач управления позиционированием требуются следующие команды позиционирования:

- Вставьте команду POSx_CTRL в свою программу и используйте контакт SM0.0 для ее исполнения в каждом цикле.
- Для задания перемещения к некоторому абсолютному положению вы должны сначала использовать команду POSx_RSEEK или POSx_LDPOS для установки нулевого положения.
- Для движения к определенному месту в соответствии с входами вашей программы используйте команду POSx_GOTO.
- Для выполнения профилей перемещения, сконфигурированных вами с помощью мастера управления позиционированием, используйте команду POSx_RUN.

Другие команды позиционирования являются необязательными.

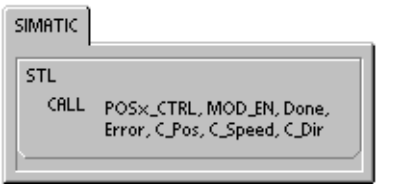
Команда POSx_CTRL

Команда POSx_CTRL (управление) активизирует и инициализирует модуль позиционирования, автоматически давая этому модулю команду на загрузку таблицы конфигурации и профилей при каждом переходе S7–200 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 (включен всегда) в качестве входа для параметра EN.

Параметр EN должен быть включен, чтобы другие команды позиционирования могли управлять модулем позиционирования. Если параметр EN выключается, то модуль позиционирования прерывает исполнение всех обрабатываемых команд.



Выходные параметры команды POSx_CTRL выдают текущее состояние модуля позиционирования.

Параметр Done [Готово] включается, когда модуль позиционирования завершает выполнение команды.

Параметр Error [Ошибка] содержит результат операции. В таблице 9–13 на стр. 276 определены коды ошибок.

Параметр C_Pos указывает на текущую позицию модуля. В зависимости от выбранной единицы измерения это или количество импульсов (DINT), или количество физических единиц измерения (REAL).

Параметр C_Speed дает текущее значение скорости модуля. Если вы настроили систему измерения модуля позиционирования на импульсы, то C_Speed является значением типа DINT, содержащим количество импульсов в секунду. Если вы настроили систему измерения на физические единицы, то C_Speed является значением типа REAL, содержащим количество выбранных физических единиц в секунду (REAL).

Параметр C_Dir указывает текущее направление вращения двигателя.

Таблица 9–2. Параметры команды POSx_CTRL

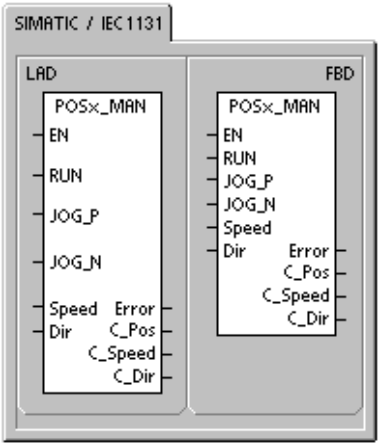
Входы/выходы	Тип данных	Операнды
MOD_EN	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done, C_Dir	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

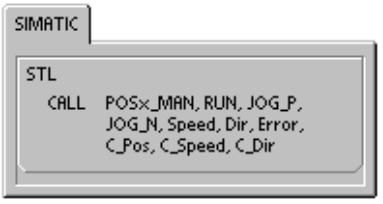
Совет

Модуль позиционирования считывает таблицу конфигурации и профилей только при включении питания или получении команды на загрузку конфигурации.

- Если для изменения конфигурации вы используете мастер управления позиционированием, то команда POSx_CTRL автоматически заставляет модуль позиционирования загружать таблицу конфигурации и профилей при каждом переходе CPU S7–200 в режим RUN.
- Если для изменения конфигурации вы используете панель управления EM 253, то щелчок на кнопке Update Configuration [Обновить конфигурацию] заставляет модуль позиционирования загрузить новую таблицу конфигурации и профилей.
- Если для изменения конфигурации вы используете другой метод, то вы тоже должны дать модулю позиционирования команду на новую загрузку конфигурации, чтобы загрузить таблицу конфигурации и профилей. В противном случае модуль позиционирования продолжает использовать старую таблицу конфигурации и профилей.

Команда POSx_MAN





Команда POSx_MAN (ручной режим) переводит модуль позиционирования в ручной режим. В этом режиме двигатель может работать с различными скоростями или в стартстопном режиме в положительном или отрицательном направлении. Когда команда POSx_MAN активизирована, допустимы только команды POSx_CTRL и POSx_DIS.

В каждый данный момент можно активизировать только один из входов RUN, JOG_P или JOG_N.

При активизации параметра RUN (запуск/останов) модуль позиционирования получает команду на ускорение до заданной скорости (параметр Speed) в заданном направлении (параметр Dir). Значение параметра Speed можно изменять во время работы двигателя, но параметр Dir должен оставаться постоянным. При деактивизации параметра RUN модуль позиционирования получает команду на замедление, пока двигатель не остановится.

При активизации параметра JOG_P (стартстопный режим в положительном направлении) или JOG_N (стартстопный режим в отрицательном направлении) модуль позиционирования получает команду на перемещение в стартстопном режиме в положительном или отрицательном направлении. Если параметр JOG_P или JOG_N остается активным в течение менее 0,5 секунды, то модуль позиционирования выдает импульсы для перемещения на расстояние, указанное в JOG_INCREMENT. Если параметр JOG_P или JOG_N остается активным в течение 0,5 секунды или дольше, то модуль позиционирования начинает ускоряться до скорости, указанной в JOG_SPEED.

Параметр Speed определяет скорость, когда активизирован параметр RUN. Если вы настроили измерительную систему модуля позиционирования на импульсы, то скорость является величиной типа DINT, измеряемой в импульсах в секунду. Если вы настроили измерительную систему модуля позиционирования на физические единицы измерения, то скорость является величиной типа REAL, измеряемой в выбранных единицах в секунду. Этот параметр можно изменять во время работы двигателя.

Параметр Dir определяет направление перемещения, когда активизирован параметр RUN. Вы не можете изменять это значение, когда параметр RUN активизирован.

Параметр Error содержит результат этой операции. В таблице 9–13 на стр. 276 определены коды ошибок.

Параметр C_Pos указывает на текущую позицию модуля. В зависимости от выбранной единицы измерения это или количество импульсов (DINT), или количество физических единиц измерения (REAL).

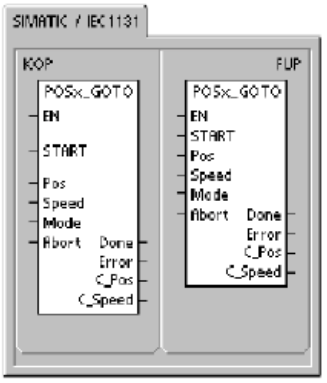
Параметр C_Speed содержит текущее значение скорости модуля. В зависимости от выбранных единиц измерения эта величина является или числом импульсов в секунду (DINT), или числом физических единиц измерения в секунду (REAL).

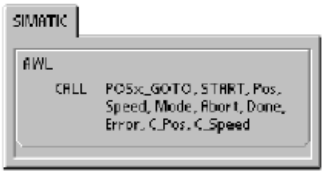
Параметр C_Dir указывает текущее направление вращения двигателя.

Таблица 9–3. 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 дает указание модулю позиционирования на переход в желаемое место.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду GOTO в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда GOTO посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда GOTO, используйте для включения параметра START элемент распознавания фронта.

Параметр Pos содержит значение, которое указывает, нужно ли переместиться к определенному месту (для абсолютного перемещения), или нужно переместиться на определенное расстояние (для относительного перемещения). В зависимости от выбранных единиц измерения это значение является или количеством импульсов (DINT), или количеством физических единиц (REAL).

Параметр Speed определяет максимальную скорость для данного перемещения. В зависимости от выбранных единиц измерения это значение является или количеством импульсов в секунду (DINT), или количеством физических единиц в секунду (REAL).

Параметр Mode выбирает вид перемещения:
- 0 - Абсолютное позиционирование
- 1 - Относительное позиционирование
- 2 - Непрерывное вращение с одной скоростью в положительном направлении
- 3 - Непрерывное вращение с одной скоростью в отрицательном направлении

Параметр Done включается, когда модуль позиционирования завершает выполнение этой команды.

Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

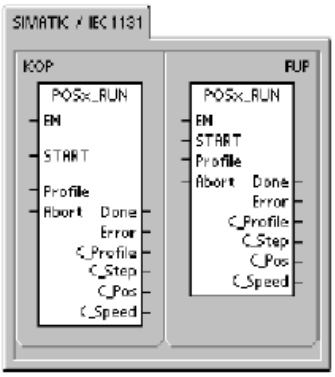
Параметр C_Pos содержит текущее положение модуля. В зависимости от единиц измерения это значение является или количеством импульсов (DINT), или количеством физических единиц (REAL).

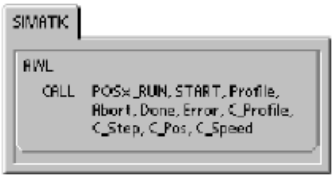
Параметр C_Speed содержит текущее значение скорости модуля. В зависимости от единиц измерения это значение является или количеством импульсов в секунду (DINT), или количеством физических единиц в секунду (REAL).

Таблица 9–4. Параметры для 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 (выполнить профиль) заставляет модуль позиционирования выполнить операцию перемещения в соответствии с определенным профилем, хранящимся в таблице конфигурации и профилей.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду RUN в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда RUN посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

Параметр Profile [Профиль] содержит номер или символическое имя профиля перемещения. Вы можете использовать также расширенные команды перемещения (118 – 127). Дальнейшую информацию о командах перемещения вы найдете в таблице 9–19 на стр. 284.

Включение параметра Abort заставляет модуль позиционирования остановить текущий профиль и замедляться, пока двигатель не остановится.

Параметр Done включается, когда модуль завершает выполнение этой команды.

Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Параметр C_Profile содержит профиль, исполняемый в данный момент модулем позиционирования.

Параметр C_Step содержит исполняемый в данный момент шаг профиля.

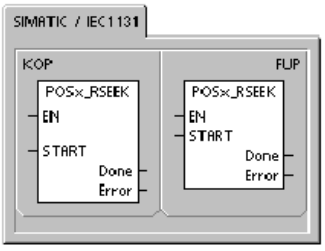
Параметр C_Pos указывает на текущую позицию модуля. В зависимости от единиц измерения это значение является или количеством импульсов (DINT), или количеством физических единиц (REAL).

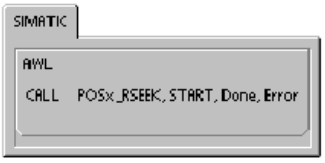
Параметр C_Speed содержит текущее значение скорости модуля. В зависимости от единиц измерения это значение является или количеством импульсов в секунду (DINT), или количеством физических единиц в секунду (REAL).

Таблица 9–5. Параметры для 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 (искать положение опорной точки) инициирует операцию поиска опорной точки в соответствии с методом поиска, указанным в таблице конфигурации и профилей. Когда модуль позиционирования определяет местоположение опорной точки, и перемещение прекращается, модуль позиционирования загружает значение параметра RP_OFFSET в текущую позицию и генерирует импульс длительностью 50 миллисекунд на выходе CLR.

Значение по умолчанию для RP_OFFSET равно 0. Для изменения значения RP_OFFSET можно использовать мастер управления позиционированием, панель управления EM253 или команду POSx_LDOFF (загрузить смещение).

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду RSEEK в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда RSEEK посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

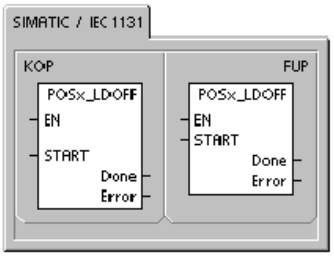
Параметр Done включается, когда модуль завершает выполнение этой команды.

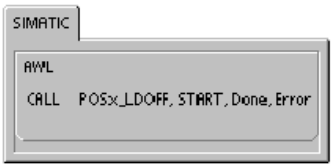
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Таблица 9–6. Параметры для 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, которое находится в другом месте, чем опорная точка.

Перед выполнением этой команды вы сначала должны определить положение опорной точки. Вы должны также переместить станок в исходное положение. Получив команду LDOFF, модуль позиционирования вычисляет смещение между исходным (текущим) положением и положением опорной точки. Модуль позиционирования затем сохраняет вычисленное смещение в параметре RP_OFFSET и устанавливает текущее положение в 0. Это устанавливает исходное положение в качестве нулевого.

В том случае, если двигатель "теряет" свое положение (например, при потере питания или ручном изменении положения двигателя), вы можете использовать команду POSx_RSEEK для автоматического восстановления нулевого положения.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду LDOFF в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда LDOFF посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

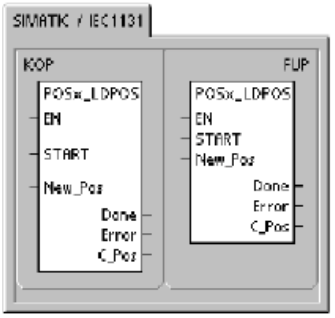
Параметр Done включается, когда модуль завершает выполнение этой команды.

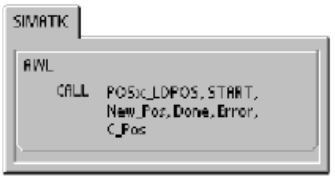
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Таблица 9–7. Параметры для 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 позицию) заменяет текущее значение положения в модуле позиционирования новым значением. Эту команду можно использовать также для того, чтобы установить новое нулевое положение для любой команды абсолютного перемещения.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду LDPOS в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда LDPOS посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

Параметр New_Pos поставляет новое значение, которым должно быть заменено текущее значение положения, сообщаемое модулем позиционирования и используемое для абсолютных перемещений. В зависимости от единиц измерения эта величин является количеством импульсов (DINT) или количеством физических единиц (REAL).

Параметр Done включается, когда модуль завершает выполнение этой команды.

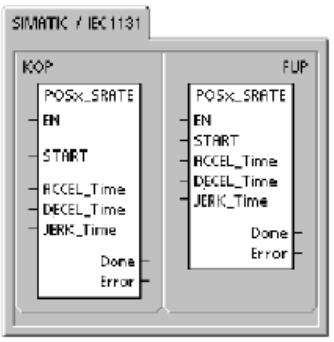
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Параметр C_Pos указывает на текущую позицию модуля. В зависимости от единиц измерения это значение является или количеством импульсов (DINT), или количеством физических единиц (REAL).

Таблица 9–8. Параметры для 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) заставляет модуль позиционирования изменить времена ускорения, замедления и компенсации толчка.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START копирует новые значения времени в таблицу конфигурации и профилей и посылает SRATE в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда SRATE посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

Параметры ACCEL_Time, DECEL_Time и JERK_Time определяют новые времена ускорения, замедления и компенсации толчка в миллисекундах (мс).

Параметр Done включается, когда модуль завершает выполнение этой команды.

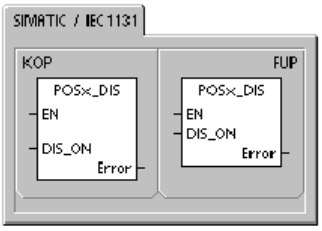
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

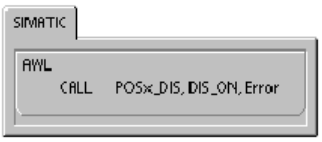
Таблица 9–9. Параметры для команды POSx_SRATE

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 включает и выключает выход DIS модуля позиционирования. Это позволяет использовать выход DIS для деактивизации и активизации устройства управления двигателем. Если вы используете выход DIS в модуле позиционирования, то эта команда может вызываться каждый цикл или только в тех случаях, когда вам нужно изменить значение выхода DIS.





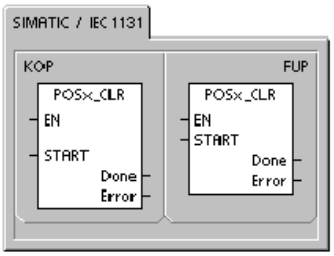
Когда бит EN включается для активизации команды, параметр DIS_ON управляет выходом DIS модуля позиционирования. Подробную информацию о выходе DIS вы найдете в таблице 9–1 на стр. 244 или в технических данных модуля позиционирования в Приложении A.

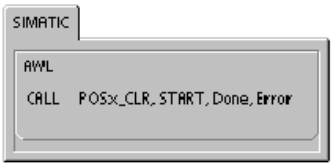
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Таблица 9–10. Параметры для 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) заставляет модуль позиционирования сгенерировать импульс длительностью 50 мс на выходе CLR.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду CLR в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда CLR посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

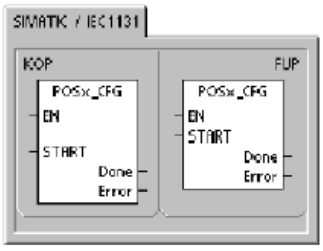
Параметр Done включается, когда модуль завершает выполнение этой команды.

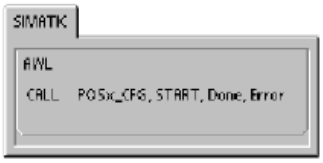
Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Таблица 9–11. Параметры для 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 (перезагрузить конфигурацию) заставляет модуль позиционирования считать конфигурационный блок по адресу, определяемому указателем таблицы конфигурации и профилей. Модуль позиционирования затем сравнивает новую конфигурацию с существующей и выполняет все необходимые изменения в настройке или новые расчеты.

Команда активизируется включением бита EN. Обеспечьте, чтобы бит EN оставался включенным, пока бит DONE не сигнализирует, что выполнение команды завершено.

Включение параметра START посылает команду CFG в модуль позиционирования. В каждом цикле, когда параметр START включен, и модуль позиционирования в данный момент времени не занят, команда CFG посылается в модуль позиционирования. Чтобы гарантировать, что послана только одна команда, используйте для включения параметра START элемент распознавания фронта.

Параметр Done включается, когда модуль завершает выполнение этой команды.

Параметр Error содержит результат этой команды. Определения кодов ошибок вы найдете в таблице 9–13 на стр. 276.

Таблица 9–12. Параметры для команды POSx_CFG

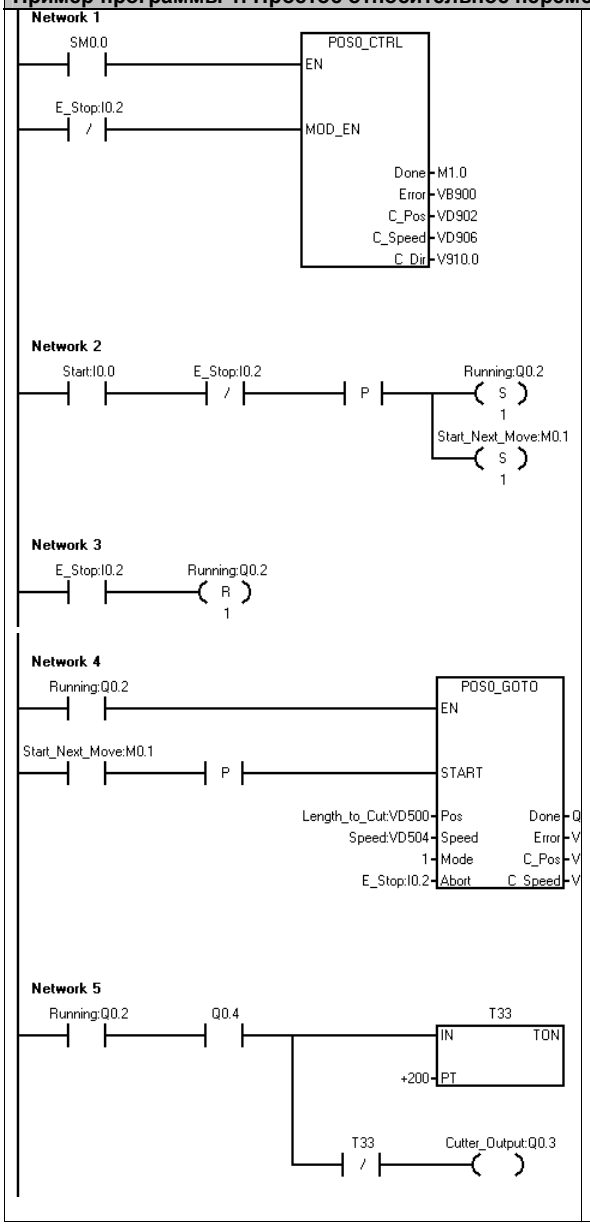
Входы/выходы	Тип данных	Операнды
START	BOOL	I, Q, V, M, SM, S, T, C, L, поток сигнала
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Примеры программ для модуля позиционирования

В первом примере программы показано простое относительное перемещение, использующее команды POSx_CTRL и POSx_GOTO для выполнения операции раскроя по длине. Эта программа не требует ни режима поиска RP, ни профиля перемещения, а длина может быть измерена в импульсах или физических единицах. Введите длину (VD500) и целевую скорость (VD504). Когда I0.0 (Start) включается, станок запускается. Когда включается I0.1 (Stop), станок заканчивает текущую операцию и останавливается. Когда включается I0.2 (E_Stop), станок прерывает любое перемещение и немедленно останавливается.

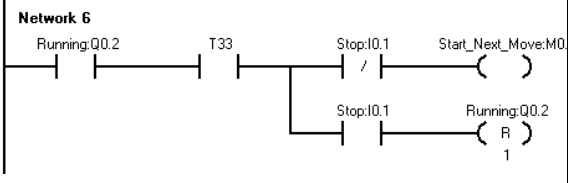
Во второй программе приводится пример применения команд POSx_CTRL, POSx_RUN, POSx_RSEEK и POSx_MAN. Вы должны сконфигурировать режим поиска RP и профиль перемещения.

Пример программы 1: Простое относительное перемещение (раскрой по длине)



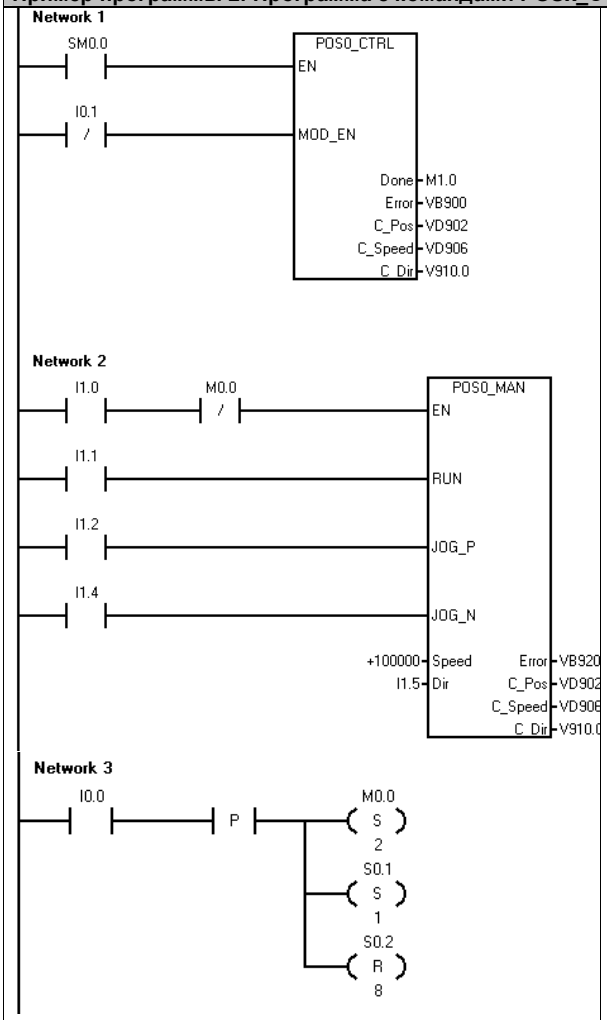

```
Network 1      //Команда управления (модуль в слоте 0).
LD     SM0.0
=      L60.0
LDN    I0.2
=      L63.7
LD     L60.0
CALL   POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906,
       V910.0

Network 2      //Команда запуска переводит станок в
               //автоматический режим
LD     I0.0
AN     I0.2
EU
S      Q0.2, 1
S      M0.1, 1

Network 3      //E_Stop: немедленный останов и
               //выключение автоматического режима.
LD     I0.2
R      Q0.2, 1

Network 4      //Перемещение к определенной точке:
               //Ввод длины для отрезания.
               //Ввод целевой скорости в Speed.
               //Установка режима на 1 (режим
               //относительного перемещения).
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

Network 5      //При нахождении в заданном положении
               //включить режущий инструмент на 2 сек.,
               //чтобы выполнить отрезание.
LD     Q0.2
A      Q0.4
TON    T33, +200
AN     T33
=      Q0.3
```

Пример программы 1: Простое относительное перемещение (раскрой по длине), продолжение



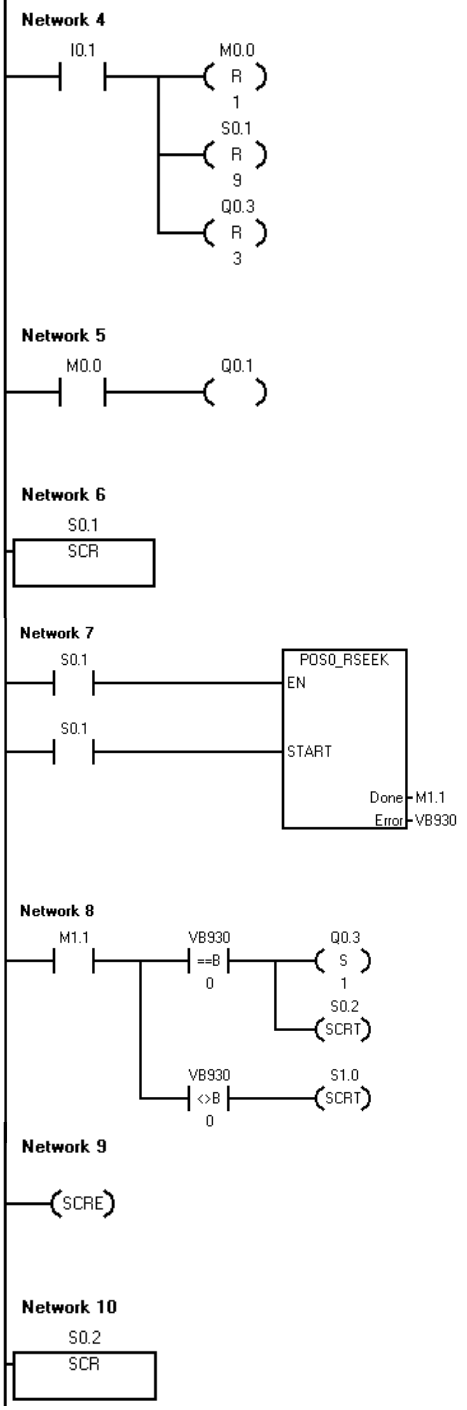

```
Network 6      //Новый запуск после отрезания,
               //если не активен Stop.
LD     Q0.2
A      T33
LPS
AN     I0.1
=      M0.1
LPP
A      I0.1
R      Q0.2, 1
```

Пример программы 2: Программа с командами POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN



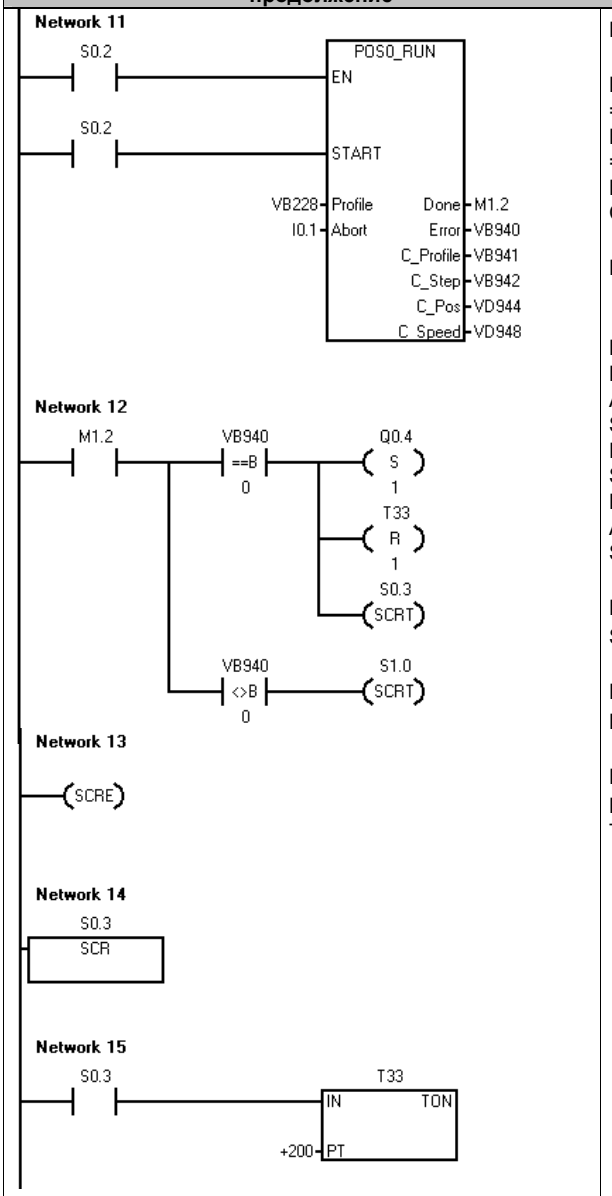

```
Network 1      //Активизация модуля позиционирования
LD     SM0.0
=      L60.0
LDN    I0.1
=      L63.7
LD     L60.0
CALL   POS0_CTRL, L63.7, M1.0, VB900, VD902,
       VD906, V910.0

Network 2      //Ручной режим, если не установлен
               //автоматический
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,
       I1.5, VB920, VD902, VD906, V910.0

Network 3      //Активизация автоматического режима
LD     I0.0
EU
S      M0.0, 2
S      S0.1, 1
R      S0.2, 8
```

Пример программы 2: Программа с командами POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение



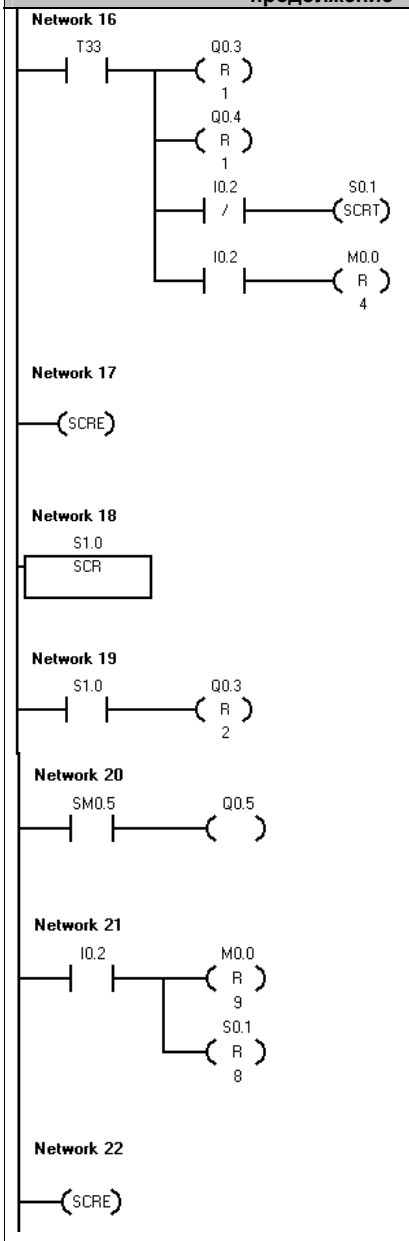

```
Network 4       //Аварийный останов
                //Деактивизировать модуль и
                //автоматический режим
LD      I0.1
R       M0.0, 1
R       S0.1, 9
R       Q0.3, 3

Network 5       //При нахождении в автоматическом
                //режиме:
                //включить световой сигнал "Работа"
LD      M0.0
=       Q0.1

Network 6
LSCR    S0.1

Network 7       //Найти опорную точку (RP)
LD      S0.1
=       L60.0
LD      S0.1
=       L63.7
LD      L60.0
CALL    POS0_RSEEK, L63.7, M1.1, VB930

Network 8       //При нахождении в опорной точке (RP):
                //Закрепить материал и
                //перейти к следующему шагу.
LD      M1.1
LPS
AB=     VB930, 0
S       Q0.3, 1
SCRT    S0.2
LPP
AB<>    VB930, 0
SCRT    S1.0

Network 9
SCRE

Network 10
LSCR    S0.2
```

Пример программы 2: Программа с командами POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение




```
Network 11      //С помощью профиля 1 перейти в
                //заданное положение.
LD      S0.2
=       L60.0
LD      S0.2
=       L63.7
LD      L60.0
CALL    POS0_RUN, L63.7, VB228, I0.1, M1.2, VB940,
        VB941, VB942, VD944, VD948
Network 12      //при достижении заданного положения
                //включить отрезное устройство и
                //перейти к следующему шагу.
LD      M1.2
LPS
AB=     VB940, 0
S       Q0.4, 1
R       T33, 1
SCRT    S0.3
LPP
AB<>    VB940, 0
SCRT    S1.0

Network 13
SCRE

Network 14      //Ждать завершения отрезания
LSCR    S0.3

Network 15
LD      S0.3
TON     T33, +200
```

Пример программы 2: Программа с командами POSx_CTRL, POSx_RUN, POSx_SEEK и POSx_MAN, продолжение




```
Network 16      //Если STOP не включен, по окончании
                //отрезания запустить снова.
LD      T33
LPS
R       Q0.3, 1
R       Q0.4, 1
AN      I0.2
SCRT    S0.1
LPP
A       I0.2
R       M0.0, 4
Network 17
SCRE

Network 18
LSCR    S1.0

Network 19      //Сбросить выходы.
LD      S1.0
R       Q0.3, 2

Network 20      //Световой сигнал ошибки мигает.
LD      SM0.5
=       Q0.5

Network 21      //Выйти из программы обработки
                //ошибок, если STOP включен.
LD      I0.2
R       M0.0, 9
R       S0.1, 8

Network 22
SCRE
```

Наблюдение за модулем позиционирования с помощью панели управления EM 253

Для оказания помощи в разработке решения задачи позиционирования STEP 7-Micro/WIN предоставляет в ваше распоряжение панель управления EM 253. Использование закладок Operation [Работа], Configuration [Конфигурация] и Diagnostics [Диагностика] облегчает наблюдение и управление работой модуля позиционирования на этапах запуска и тестирования процесса разработки.

В панели управления EM 253 вы можете проверить, правильно ли подключен модуль позиционирования, настроить данные конфигурации и проверить профили перемещения.

Отображение и управление работой модуля позиционирования

Закладка Operation [Работа] панели управления дает возможность оказывать воздействие на работу модуля позиционирования. Панель управления отображает текущую скорость, текущее положение и текущее направление модуля позиционирования. Вы можете также видеть состояние светодиодов входов и выходов (за исключением импульсных светодиодов).

Панель управления позволяет воздействовать на модуль позиционирования, изменяя скорость и направление, останавливая и запуская перемещение и работая с инструментом в стартстопном режиме (если перемещение остановлено).

Кроме того, вы можете генерировать следующие команды перемещения:

- Активизация ручного режима. Эта команда делает возможным позиционирование инструмента с помощью элементов ручного управления.
- Выполнение профиля перемещения. Эта команда позволяет выбрать профиль, подлежащий исполнению. Панель управления отображает состояние профиля, выполняемого модулем позиционирования.
- Поиск опорной точки. Эта команда ищет опорную точку с помощью сконфигурированного режима поиска.

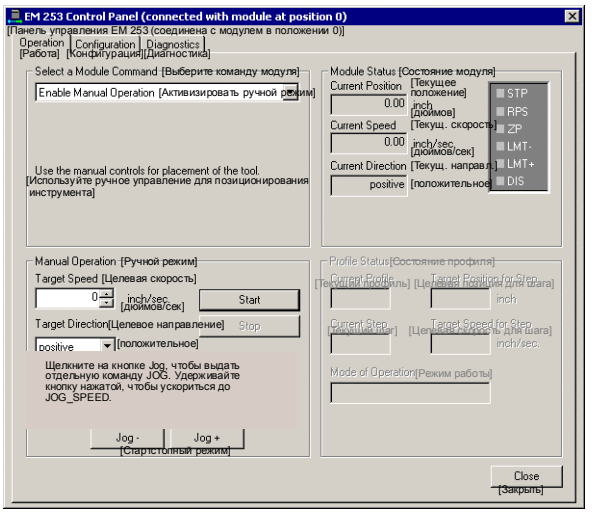


Рис. 9–19. Закладка Operation панели управления EM 253

- Загрузка смещения опорной точки. После того как вы сдвинули инструмент с помощью элементов ручного управления в новое нулевое положение, загрузите смещение опорной точки.
- Новая загрузка текущего положения. 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.
- Вывод импульса на выходе CLR. Эта команда генерирует импульс длительностью 50 мс на выходе CLR модуля позиционирования.
- "Заучивание" профиля перемещения. Эта команда позволяет сохранить целевую позицию и скорость для профиля перемещения и шага при ручном позиционировании инструмента. Панель управления отображает состояние профиля, выполняемого модулем позиционирования.
- Загрузка конфигурации модуля. Эта команда загружает новую конфигурацию, заставляя модуль позиционирования считывать конфигурационный блок из памяти переменных S7–200.
- Перемещение к абсолютной позиции. Эта команда позволяет вам переместиться в указанное положение с целевой скоростью. Перед использованием этой команды вы уже должны установить нулевое положение.
- Выполнение относительного перемещения. Эта команда позволяет вам переместиться на указанное расстояние от текущего положения с целевой скоростью. Вы можете ввести положительное или отрицательное расстояние.
- Сброс командного интерфейса. Эта команда очищает командный байт для модуля позиционирования и устанавливает бит Done [Выполнено]. Используйте эту команду, если модуль позиционирования, по-видимому, не реагирует на команды.

Отображение и изменение конфигурации модуля позиционирования

Закладка Configuration [Конфигурация] панели управления дает возможность просматривать и изменять настройки конфигурации для модуля позиционирования, хранящиеся в блоке данных S7–200.

После изменения настроек конфигурации просто щелкните на кнопке, чтобы обновить эти настройки в проекте STEP 7-Micro/Win и в блоке данных S7–200.

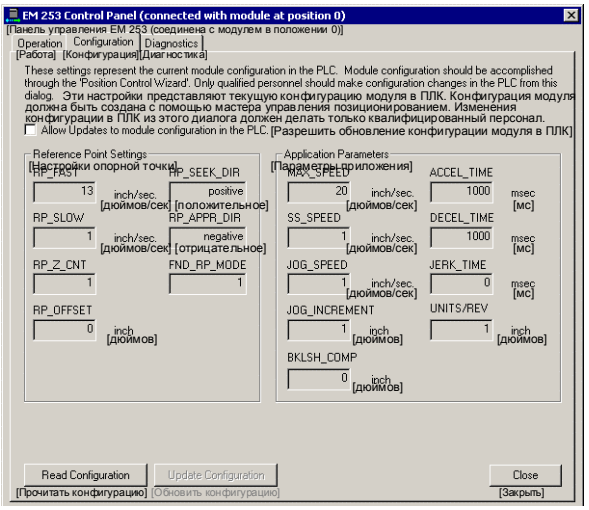


Рис. 9–20. Закладка Configuration панели управления EM 253

Отображение диагностической информации для модуля позиционирования

Закладка Diagnostics [Диагностика] панели управления дает возможность просматривать диагностическую информацию, относящуюся к модулю позиционирования.

Вы можете просматривать конкретную информацию о модуле позиционирования, например, положение модуля в цепочке входов/выходов, тип модуля и номер версии ПЗУ, а также выходной байт, используемый в качестве командного байта для этого модуля.

Панель управления отображает все ошибочные состояния, возникшие в результате выполнения команд. Ошибочные состояния для команд вы найдете в таблице 9–13 на стр. 276.

Вы можете также просматривать любые ошибочные состояния, сообщаемые модулем позиционирования. Ошибочные состояния для модуля вы найдете в таблице 9–14 на стр. 277.

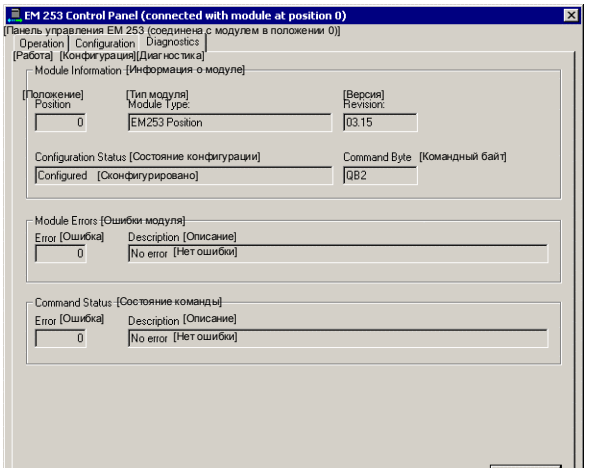


Рис. 9–21. Закладка Diagnostics панели управления EM 253

Коды ошибок для модуля и команд позиционирования

Таблица 9–13. Коды ошибок команд

Код ошибки	Описание
0	Нет ошибки
1	Исполнение прекращено пользователем
2	Ошибка конфигурации Для просмотра кодов ошибок пользуйтесь закладкой Diagnostics [Диагностика] панели управления EM 253
3	Недопустимая команда
4	Исполнение прекращено из-за неправильной конфигурации. Для просмотра кодов ошибок пользуйтесь закладкой Diagnostics [Диагностика] панели управления EM 253
5	Исполнение прекращено из-за отсутствия напряжения у пользователя
6	Исполнение прекращено из-за отсутствия определения опорной точки
7	Исполнение прекращено из-за активности входа STP
8	Исполнение прекращено из-за активности входа LMT-
9	Исполнение прекращено из-за активности входа LMT+
10	Исполнение прекращено из-за проблем при выполнении перемещения
11	Для указанного профиля нет сконфигурированного блока профилей
12	Недопустимый режим работы
13	Режим работы, не поддерживается для этой команды
14	Недопустимое количество шагов в блоке профилей
15	Недопустимое изменение направления
16	Недопустимое расстояние
17	Запуск RPS произошел до достижения целевой скорости
18	Недостаточная ширина области активности RPS
19	Скорость вне диапазона
20	Недостаточное расстояние для выполнения желаемого изменения скорости
21	Недопустимое положение
22	Неизвестно нулевое положение
23 –127	Резерв
128	Модуль позиционирования не может обработать эту команду: или модуль позиционирования занят другой командой, или для этой команды не было стартового импульса
129	Ошибка модуля позиционирования: ■ Местоположение модуля позиционирования или адрес в области памяти выходов, сконфигурированный мастером управления позиционированием, не совпадает с фактическим положением или адресом в памяти ■ Другие ошибочные состояния вы найдете в SMB8 – SMB21 (идентификатор модуля ввода/вывода и регистр ошибок)
130	Модуль позиционирования не активизирован
131	Модуль позиционирования недоступен: ошибка модуля или модуль не активизирован (См. состояние команды POSx_CTRL)

Таблица 9–14. Коды ошибок модуля

Код ошибки	Описание
0	Нет ошибки
1	Нет напряжения у пользователя
2	Отсутствует конфигурационный блок
3	Ошибка указателя конфигурационного блока
4	Размер конфигурационного блока превышает доступную память переменных
5	Недопустимый формат конфигурационного блока
6	Указано слишком много профилей
7	Недопустимая спецификация STP_RSP
8	Недопустимая спецификация LMT-_RPS
9	Недопустимая спецификация LMT+_RPS
10	Недопустимая спецификация FILTER_TIME
11	Недопустимая спецификация MEAS_SYS
12	Недопустимая спецификация 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 конфигурации и профилей

Мастер управления позиционированием был разработан, чтобы облегчит применение позиционирования путем автоматической генерации информации о конфигурации и профилях на основе ответов, даваемых вами о вашей системе позиционирования. Информация из таблицы конфигурации и профилей предназначена опытным пользователям, желающим создавать свои собственные программы управления позиционированием.

Таблица конфигурации и профилей находится в области памяти переменных S7–200. Как показано в таблице 9–15, настройки конфигурации хранятся в следующих видах информации:

- Конфигурационный блок содержит информацию, используемую при подготовке модуля к выполнению команд позиционирования.
- Интерактивный блок поддерживает непосредственную настройку параметров перемещения программой пользователя.
- Каждый профильный блок описывает заранее определенное перемещение, подлежащее выполнению модулем позиционирования. Вы можете сконфигурировать до 25 профильных блоков.

Совет
Для создания большего количества профилей перемещения, чем 25, вы можете обмениваться таблицами конфигурации и профилей путем изменения значения, хранящегося в указателе таблицы конфигурации и профилей.

Таблица 9–15. Таблица конфигурации и профилей

Смещение	Имя	Описание функции	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	Количество допустимых шагов на профиль (1 байт)	--
10	PROFILES	Количество профилей от 0 до 25 (1 байт)	--
11	Резерв	Установлен на 0x0000	--
13	IN_OUT_CFG	Указывает на использование входов и выходов модуля (1 байт) MSB 7 6 5 4 3 2 1 LSB 0 P/D POL 0 0 STP RPS LMT- LMT+ P/D Этот бит указывает на использование P0 и P1. Положительная полярность (POL = 0): 0 – Импульсы на P0 для вращения в положительном направлении Импульсы на P1 для вращения в отрицательном направлении 1 – Импульсы на P0 для вращения P1 управляет направлением вращения (0 – положительное, 1 – отрицательное) Отрицательная полярность (POL = 1): 0 – Импульсы на P0 для вращения в положительном направлении Импульсы на P1 для вращения в отрицательном направлении 1 – Импульсы на P0 для вращения P1 управляет направлением вращения (0 – положительное, 1 – отрицательное) POL Этот бит устанавливает соглашение о полярности для P0 и P1. (0 – положительная полярность; 1 – отрицательная полярность) STP Этот бит управляет уровнем активности для входа останова. RPS Этот бит управляет уровнем активности для входа RPS LMT- Этот бит управляет уровнем активности для входа границы отрицательного перемещения LMT+ Этот бит управляет уровнем активности для входа границы положительного перемещения 0 – активен высокий уровень 1 – активен низкий уровень	--

Таблица 9–15. Таблица конфигурации и профилей, продолжение

Смещение	Имя	Описание функции	Тип
14	STP_RSP	Определяет реакцию привода на вход STP (1 байт) 0 Нет реакции. Игнорировать состояние этого входа. 1 Замедлиться до останова и показать, что вход STP активен. 2 Завершить импульсы и показать вход STP 3 – 255 Резерв (ошибка, если указано это значение)	--
15	LMT-_RSP	Определяет реакцию привода на вход для отрицательной границы перемещения (1 байт) 0 Нет реакции. Игнорировать состояние этого входа. 1 Замедлиться до останова и показать, что граница достигнута. 2 Завершить импульсы и показать, что граница достигнута. 3 – 255 Резерв (ошибка, если указано это значение)	--
16	LMT+_RSP	Определяет реакцию привода на вход для положительной границы перемещения (1 байт) 0 Нет реакции. Игнорировать состояние этого входа. 1 Замедлиться до останова и показать, что граница достигнута. 2 Завершить импульсы и показать, что граница достигнута. 3 – 255 Резерв (ошибка, если указано это значение)	--
17	FILTER_TIME	Указывает время фильтрации для входов STP, LMT-, LMT+ и RPS (1 байт) MSB 7 6 5 4 3 2 1 LSB 0 STP, LMT-, LMT+ | **RPS** '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 это значение указано)	--
18	MEAS_SYS	Указывает систему измерения (1 байт) 0 Импульсы (скорость измеряется в импульсах в секунду, а положение – в импульсах). Значения хранятся как данные типа DINT. 1 Физические единицы (скорость измеряется в выбранных единицах в секунду, а положение – в выбранных единицах). Значения хранятся как данные одинарной точности типа REAL. от 2 до 255 Резерв (ошибка, если указано это значение)	--
19	--	Резерв (установлен на 0)	--
20	PLS/REV	Указывает количество импульсов на оборот двигателя (4 байта) Действителен, только если MEAS_SYS установлен в 0.	DINT
24	UNITS/REV	Указывает количество физических единиц на оборот двигателя (4 байта) Действителен, только если MEAS_SYS установлен в 1.	REAL
28	UNITS	Зарезервировано для STEP 7-Micro/WIN для хранения единицы, определяемой пользователем (строка символов) (4 байта)	--
32	RP_CFG	Определяет конфигурацию для поиска опорной точки (1 байт) MSB 7 6 5 4 3 2 1 LSB 0 | | 0 0 | MODE RP_ADDR_DIR RP_SEEK_DIR RP_SEEK_DIR Этот бит определяет начальное направление для поиска опорной точки. (0 – положительное направление, 1 – отрицательное направление) RP_APPR_DIR Этот бит определяет направление приближения при завершении поиска опорной точки. (0 – положительное направление, 1 – отрицательное направление) MODE Определяет метод поиска опорной точки. '0000' Поиск опорной точки деактивирован. '0001' Опорная точка – это место, где становится активным вход RPS. '0010' Опорная точка находится в середине активной области входа RPS. '0011' Опорная точка находится вне активной области входа RPS. '0101' – '1111' Резерв (ошибка, если эти значения указаны).	--
33	--	Резерв (установлен на 0)	--
34	RP_Z_CNT	Количество импульсов входа ZP, используемых для определения опорной точки (4 байта)	DINT

Таблица 9–15. Таблица конфигурации и профилей, продолжение

Смещение	Имя	Описание функции	Тип
38	RP_FAST	Высокая скорость для поиска RP: MAX_SPD или меньше (4 байта)	DINT REAL
42	RP_SLOW	Медленная скорость для поиска RP: максимальная скорость, с которой двигатель может немедленно остановиться (4 байта)	DINT REAL
46	SS_SPEED	Скорость пуска и останова. (4 байта) Это максимальная скорость, до которой двигатель может мгновенно разогнаться 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 недействительны.	DINT REAL
50	MAX_SPEED	Максимальная рабочая скорость двигателя (4 байта)	DINT REAL
54	JOG_SPEED	Скорость при стартстопном режиме. MAX_SPEED или менее (4 байта)	
58	JOG_INCREMENT	Это расстояние (или количество импульсов), на которое осуществляется перемещение в ответ на отдельный стартстопный импульс. (4 байта)	DINT REAL
62	ACCEL_TIME	Время, необходимое для ускорения с минимальной до максимальной скорости, в миллисекундах (4 байта)	DINT
66	DECEL_TIME	Время, необходимое для замедления с максимальной до минимальной скорости, в миллисекундах (4 байта)	DINT
70	BKLSH_COMP	Компенсация люфта: расстояние, используемое для компенсации люфта в системе при изменении направления (4 байта)	DINT REAL
74	JERK_TIME	Время, в течение которого компенсация толчка применяется к начальному и конечному участкам кривой ускорения/замедления (S-кривая). Значение 0 деактивизирует компенсацию толчка. Время компенсации толчка дается в миллисекундах. (4 байта)	DINT
Интерактивный блок			
78	MOVE_CMD	Устанавливает режим работы (1 байт) 0 Абсолютное позиционирование 1 Относительное позиционирование 2 Непрерывная работа с фиксированной скоростью при положительном направлении вращения 3 Непрерывная работа с фиксированной скоростью при отрицательном направлении вращения 4 Ручное управление скоростью, положительное направление вращения 5 Ручное управление скоростью, отрицательное направление вращения 6 Непрерывная работа с фиксированной скоростью при положительном направлении вращения и запущенном останове (вход RPS сигнализирует останов) 7 Непрерывная работа с фиксированной скоростью при отрицательном направлении вращения и запущенном останове (вход RPS сигнализирует останов) 8 –255 Резерв (ошибка, если указано это значение)	--
79	--	Резерв. Установлен на 0	--
80	TARGET_POS	Целевая позиция для этого перемещения (4 байта)	DINT REAL
84	TARGET_SPEED	Целевая скорость для этого перемещения (4 байта)	DINT REAL
88	RP_OFFSET	Абсолютное положение опорной точки (4 байта)	DINT REAL
Профильный блок 0			
92 (+0)	STEPS	Количество шагов в этой последовательности перемещения (1 байт)	--
93 (+1)	MODE	Устанавливает режим работы для этого профильного блока (1 байт) 0 Абсолютное позиционирование 1 Относительное позиционирование 2 Непрерывная работа с фиксированной скоростью при положительном направлении вращения 3 Непрерывная работа с фиксированной скоростью при отрицательном направлении вращения 4 Резерв (ошибка, если указано это значение) 5 Резерв (ошибка, если указано это значение) 6 Непрерывная работа с фиксированной скоростью при положительном направлении вращения и запущенном останове (RPS сигнализирует останов) 7 Непрерывная работа с фиксированной скоростью при отрицательном направлении вращения и запущенном останове (вход RPS сигнализирует останов) 8 Непрерывная работа с двумя скоростями при положительном направлении вращения (RPS устанавливает скорость) 9 Непрерывная работа с двумя скоростями при отрицательном направлении вращения (RPS устанавливает скорость) 10 –255 Резерв (ошибка, если указано это значение)	--

Таблица 9–15. Таблица конфигурации и профилей, продолжение

Смещение	Имя		Описание функции	Тип
94 (+2)	0	POS	Положение, в которое нужно перейти на шаге 0 (4 байта)	DINT REAL
98 (+6)		SPEED	Целевая скорость для шага 0 (4 байта)	DINT REAL
102 (+10)	1	POS	Положение, в которое нужно перейти на шаге 1 (4 байта)	DINT REAL
106 (+14)		SPEED	Целевая скорость для шага 1 (4 байта)	DINT REAL
110 (+18)	2	POS	Положение, в которое нужно перейти на шаге 2 (4 байта)	DINT REAL
114 (+22)		SPEED	Целевая скорость для шага 2 (4 байта)	DINT REAL
118 (+26)	3	POS	Положение, в которое нужно перейти на шаге 3 (4 байта)	DINT REAL
122 (+30)		SPEED	Целевая скорость для шага 3 (4 байта)	DINT REAL
Профильный блок 1				
126 (+34)	STEPS		Количество шагов в этой последовательности перемещения (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 0 (4 байта)	DINT REAL
132 (+40)		SPEED	Целевая скорость для шага 0 (4 байта)	DINT REAL
...	...	...	...	...

Адреса специальной памяти для модуля позиционирования

S7–200 выделяет 50 байтов специальной памяти (SM) каждому интеллектуальному модулю с учетом физического расположения модуля в системе входов/выходов. См. таблицу 9–16. Когда модуль обнаруживает ошибочное состояние или изменение в состоянии данных, то он обновляет эти адреса в SM-памяти. Первый модуль обновляет SMB200 – SMB249 так, как это необходимо для сообщения об ошибочном состоянии, второй модуль обновляет SMB250 – SMB299 и т.д.

Таблица 9–16. Байты специальной памяти SMB200 – SMB549

SM-байты для интеллектуального модуля:						
Слот 0	**Слот 1**	**Слот 2**	**Слот 3**	**Слот 4**	**Слот 5**	**Слот 6**
SMB200 – SMB249	SMB250 – SMB299	SMB300 – SMB349	SMB350 – SMB399	SMB400 – SMB449	SMB450 – SMB499	SMB500 – SMB549

В таблице 9–17 показана структура области данных SM, выделенной для интеллектуального модуля. Определение дается для случая, когда интеллектуальный модуль находится в слоте 0 системы входов/выходов.

Таблица 9–17. Определение области специальной памяти для позиционного модуля EM 253

Адрес SM	Описание
SMB200 – SMB215	Имя модуля (16 символов ASCII). SMB200 является первым символом: «Положение EM253»
SMB216 – SMB219	Номер версии программного обеспечения (4 символа ASCII). SMB216 является первым символом.
SMW220	Код ошибки для модуля. Описание кодов ошибок вы найдете в таблице 9–14.
SMB222	Состояние входов/выходов. Отражает состояние входов и выходов модуля. MSB 7 6 5 4 3 2 1 LSB 0 DIS | 0 | 0 | STP | LMT- | LMT+ | RPS | ZP DIS Деактивизировать выходы 0 = поток сигнала отсутствует 1 = поток сигнала STP Вход останова 0 = поток сигнала отсутствует 1 = поток сигнала LMT- Вход для границы отрицательного перемещения 0 = поток сигнала отсутствует 1 = поток сигнала LMT+ Вход для границы положительного перемещения 0 = поток сигнала отсутствует 1 = поток сигнала RPS Вход переключателя опорной точки 0 = поток сигнала отсутствует 1 = поток сигнала ZP Вход нулевого импульса 0 = поток сигнала отсутствует 1 = поток сигнала
SMB223	Мгновенное состояние модуля. Отражает состояние конфигурации модуля и направления вращения. MSB 7 6 5 4 3 2 1 LSB 0 0 | 0 | 0 | 0 | 0 | OR | R | CFG OR Целевая скорость вращения вне диапазона 0 = в диапазоне 1 = вне диапазона R Направление вращения 0 = положительное 1 = отрицательное CFG Модуль сконфигурирован 0 = не сконфигурирован 1 = сконфигурирован
SMB224	CUR_PF – байт, указывающий на профиль, который исполняется в данный момент времени.
SMB225	CUR_STP – байт, указывающий на шаг, который исполняется в данный момент времени в профиле.
SMD226	CUR_POS – значение в формате двойного слова, показывающее текущее положение модуля.
SMD230	CUR_SPD – значение в формате двойного слова, показывающее текущую скорость модуля.
SMB234	Результат выполнения команды. Описание кодов ошибок вы найдете в табл. 9-13. Ошибочные состояния с кодами больше 127 создаются подпрограммами, которые были созданы мастером. MSB 7 6 LSB 0 D | ERROR D Бит завершения 0= операция выполняется 1= операция завершена (устанавливается модулем во время инициализации)
SMB235 – SMB244	Резерв
SMB245	Смещение для первого байта выходов Q, используемого в качестве командного интерфейса этого модуля. Для удобства пользователя S7–200 автоматически задает это смещение, и модулю оно не требуется.
SMD246	Указатель на адрес в памяти переменных таблицы конфигурации и профилей. Указатель на область, отличную от памяти переменных, недействителен. Модуль позиционирования контролирует этот адрес, пока он получает ненулевое значение указателя.

Описание командного байта для модуля позиционирования

Модуль позиционирования предоставляет один байт цифровых выходов, который используется в качестве командного байта. На рис. 9–22 показано определение командного байта. В таблице 9–18 приведены определения командного кода (Command_code).

Запись в командный байт, при которой бит R изменяется с 0 на 1, интерпретируется модулем как новая команда.

Если модуль обнаруживает переход в состояние незанятости (бит R переходит в состояние 0), когда команда активна, то выполняемая операция прерывается и, если движение продолжается, то выполняется замедление до останова.

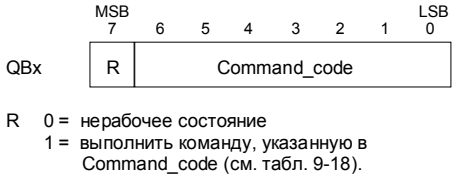


Рис. 9–22. Определение командного байта

После завершения операции модуль должен перейти в состояние незанятости, прежде чем будет принята новая команда. Если операция прервана, то модуль должен завершить все замедления, прежде чем будет принята новая команда. Любое изменение в значении Command_code, когда команда активна, игнорируются.

Реакция модуля позиционирования на изменение режима работы S7–200 или на ошибочное состояние определяется воздействием, которое S7–200 оказывает на цифровые выходы в соответствии с его функцией:

- *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 выбрать состояние, в которое цифровые выходы должны принять после перехода в STOP, или указать, чтобы выходы сохранили свое последнее состояние.
    - *Если бит R при переходе в STOP выключается:* Модуль позиционирования замедляет все текущие перемещения до останова.

Таблица 9–18. Определения Command_code

Command_code	Команда	
от 000 0000 до 000 1111	от 0 до 24	Выполнить перемещение, указанное в профильных блоках с 0 по 24
от 100 0000 до 111 0101	от 25 до 117	Резерв (ошибка, если указано это значение)
111 0110	118	Активизировать выход DIS
111 0111	119	Деактивизировать выход DIS
111 1000	120	Импульс на выходе CLR
111 1001	121	Вновь загрузить текущее положение
111 1010	122	Выполнить перемещение, указанное в интерактивном блоке
111 1011	123	Получить информацию о смещении опорной точки
111 1100	124	Стартстопный режим при положительном направлении вращения
111 1101	125	Стартстопный режим при отрицательном направлении вращения
111 1110	126	Поиск опорной точки
111 1111	127	Вновь загрузить конфигурацию

    - *Если бит R при переходе в STOP включается:* Модуль позиционирования завершает все текущие команды. Если ни одна команда не обрабатывается, то модуль позиционирования выполняет команду, которая указана в битах Command_code.
    - *Если бит R удерживается 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 останова.

Модуль позиционирования реализует контроль времени, который выключает выходы, если потеряна связь с S7–200. По истечении времени контроля модуль позиционирования замедляет все текущие перемещения до останова.

Если обнаружена фатальная ошибка в аппаратуре или ПЗУ модуля, то модуль позиционирования устанавливает выходы P0, P1, DIS и CLR в неактивное состояние.

Таблица 9–19. Команды позиционирования

Команда	Описание
Команды 0 – 24: *Выполнить перемещения, указанные в профильных блоках с 0 по 24*	Когда эта команда выполняется, модуль позиционирования осуществляет перемещение, указанное в поле MODE профильного блока, указанного в части Command_code команды. ■ В режиме 0 (абсолютное позиционирование) профильный блок перемещения определяет от одного о четырех шагов, причем каждый шаг содержит как положение (POS), так и скорость (SPEED), которые описывают сегмент перемещения. Спецификация POS представляет абсолютное положение, отсчитываемое от положения, называемого опорной точкой. Направление перемещения определяется соотношением между текущим положением и положением первого шага в профиле. В многошаговом перемещении изменение направления перемещения запрещено и приводит к сообщению об ошибке. ■ В режиме 1 (относительное позиционирование) профильный блок перемещения определяет от одного о четырех шагов, причем каждый шаг содержит как положение (POS), так и скорость (SPEED), которые описывают сегмент перемещения. Знак положения (POS) определяет направление перемещения. В многошаговом перемещении изменение направления перемещения запрещено и приводит к сообщению об ошибке. ■ В режимах 2 и 3 (режимы непрерывной работы с фиксированной скоростью) указание положения (POS) игнорируется, и модуль ускоряется до скорости, указанной в поле SPEED первого шага. Режим 2 используется для положительного направления вращения, а режим 3 используется для отрицательного направления вращения. Перемещение прекращается, когда командный байт переходит в состояние незанятости. ■ В режимах 6 и 7 (режимы непрерывной работы с фиксированной скоростью и запущенным остановом) модуль ускоряется до скорости, указанной в поле SPEED первого шага. Если вход RPS становится активным, перемещение останавливается после прохождения расстояния, указанного в поле POS первого шага. (Расстояние, указанное в поле, должно включать в себя и путь торможения.) Если поле POS равно нулю, когда вход RPS становится активным, модуль позиционирования замедляется до останова. Режим 6 используется для положительного направления вращения, а режим 7 используется для отрицательного направления вращения. ■ В режимах 8 и 9 двоичное значение входа RPS выбирает одну из двух скоростей, указанных в профильном блоке в первых двух шагах. - 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 используется для отрицательного направления вращения. Величина SPEED управляет скоростью перемещения. Значение POS в этом режиме игнорируется.
Команда 118 *Активизирует выход DIS*	Когда эта команда выполняется, модуль позиционирования активизирует выход DIS.
Команда 119 *Деактивизирует выход DIS*	Когда эта команда выполняется, модуль позиционирования деактивизирует выход DIS.
Команда 120 *Импульс на выходе CLR*	Когда эта команда выполняется, модуль позиционирования генерирует импульс длительностью 50 миллисекунд на выходе CLR.
Команда 121 *Снова загружает текущее положение*	Когда эта команда выполняется, модуль позиционирования присваивает текущему положению значение, находящееся в поле TARGET_POS интерактивного блока.
Команда 122 *Выполнить перемещение, указанное в интерактивном блоке*	Когда эта команда выполняется, модуль позиционирования осуществляет перемещение, указанное в поле MOVE_CMD интерактивного блока. ■ В режимах 0 и 1 (режимы абсолютного и относительного позиционирования) выполняется одношаговое перемещение на основе информации о целевой скорости и положении, указанной в полях TARGET_SPEED и TARGET_POS интерактивного блока. ■ В режимах 2 и 3 (режимы непрерывной работы с фиксированной скоростью) задание положения игнорируется, а модуль позиционирования ускоряется до скорости, указанной в поле TARGET_SPEED интерактивного блока. Перемещение прекращается, когда командный байт переходит в состояние незанятости. ■ 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 непрерывно контролирует это положение и соответствующим образом реагирует, когда значение скорости изменяется.

Таблица 9–19. Команды позиционирования, продолжение

Команда	Описание
Команда 123 *Получение информации о смещении опорной точки*	Когда эта команда выполняется, модуль позиционирования устанавливает нулевое положение, отличное от местонахождения опорной точки. Перед выдачей этой команды вы должны определить положение опорной точки, а также, используя стартстопный режим, установить станок в исходное положение. Получив эту команду, модуль позиционирования вычисляет смещение между исходным (текущим) положением и положением опорной точки и записывает вычисленное смещение в поле RP_OFFSET интерактивного блока. Текущее положение затем устанавливается в 0, делая исходное положение нулевым. В случае если шаговый двигатель "забывает" свое положение (например, в результате потери питания или ручной перестановки шагового двигателя в новое положение), может быть выдана команда поиска опорной точки, чтобы автоматически восстановить нулевое положение.
Команда 124 *Стартстопный режим в положительном направлении*	С помощью этой команды вы можете выдавать импульсы вручную для перемещения шагового двигателя в положительном направлении. Если команда остается активной в течение менее 0,5 секунды, то модуль позиционирования выдает импульсы до тех пор, пока не будет пройдено расстояние, указанное в JOG_INCREMENT. Если команда остается активной в течение 0,5 секунды или дольше, модуль позиционирования ускоряется до скорости, указанной в JOG_SPEED. Когда обнаруживается переход в состояние незанятости, модуль позиционирования замедляется до останова.
Команда 125 *Стартстопный режим в отрицательном направлении*	С помощью этой команды вы можете выдавать импульсы вручную для перемещения шагового двигателя в отрицательном направлении. Если команда остается активной в течение менее 0,5 секунды, то модуль позиционирования выдает импульсы до тех пор, пока не будет пройдено расстояние, указанное в JOG_INCREMENT. Если команда остается активной в течение 0,5 секунды или дольше, модуль позиционирования ускоряется до скорости, указанной в JOG_SPEED. Когда обнаруживается переход в состояние незанятости, модуль позиционирования замедляется до останова.
Команда 126 *Поиск положения опорной точки*	Когда эта команда выполняется, модуль позиционирования начинает поиск опорной точки, используя заданный метод поиска. Когда опорная точка найдена, и движение прекращено, модуль позиционирования загружает значение, считанное из поля RP_OFFSET интерактивного блока, в текущее положение и выдает на выходе CLR импульс длительностью 50 миллисекунд.
Команда 127 *Снова загрузить конфигурацию*	Когда эта команда выполняется, модуль позиционирования считывает указатель на таблицу конфигурации и профилей из соответствующего адреса в специальной памяти, а затем считывает конфигурационный блок из адреса, указываемого указателем на таблицу конфигурации и профилей. Модуль позиционирования сравнивает только что полученные конфигурационные данные с существующей конфигурацией модуля и выполняет все необходимые изменения в настройке или новые расчеты. Все буферизованные профили отбрасываются.

Описание буфера профилей модуля позиционирования

Модуль позиционирования сохраняет необходимые для исполнения данные максимум для 4 профилей в буферной памяти (кэше). Когда модуль позиционирования получает команду на выполнение профиля, он проверяет, хранится ли запрошенный профиль в буферной памяти. Если необходимые для исполнения профиля данные находятся в буферной памяти, то модуль позиционирования немедленно выполняет профиль. Если необходимые для исполнения профиля данные не находятся в буферной памяти, то модуль позиционирования считывает данные профильного блока из таблицы конфигурации и профилей в S7–200 и рассчитывает необходимые для исполнения профиля данные перед его выполнением.

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 конфигурации и профилей в S7–200 и рассчитывает данные, необходимые для выполнения перемещения.

Новое конфигурирование модуля позиционирования удаляет все необходимые для исполнения данные, хранящиеся в буферной памяти.

Создание собственных команд управления позиционированием

Мастер управления позиционированием создает команды позиционирования для управления работой модуля позиционирования; но вы можете также создавать свои собственные команды. Следующий сегмент кода STL дает пример того, как можно создавать свои собственные команды управления для модуля позиционирования.

В этом примере используется CPU 224 модуля S7–200 с модулем позиционирования, расположенным в слоте 0. Модуль позиционирования конфигурируется при запуске. CMD_STAT – это символ для SMB234, CMD – символ для QB2 и NEW_CMD – 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 – символ для QB2
                //NEW_CMD – символ для профиля.
                //
                //1. Очистить бит Done модуля позиционирования.
                //2. Очистить командный байт модуля позиционирования.
                //3. Выдать новую команду.
                //4. Ждать выполнения команды.
LD      SM0.0
MOVB    0, CMD_STAT
BIW     0, CMD
BIW     NEW_CMD, CMD
SCRT    State_1

Network 3
SCRE

Network 4       // Ждать завершения команды.
LSCR    State_1

Network 5       //Если команда завершена без ошибок, перейти в состояние незанятости.
LDB=    CMD_STAT, 16#80
SCRT    Idle_State

Network 6       // Если команда завершена с ошибкой, перейти в состояние обработки ошибок.
LDB>    CMD_STAT, 16#80
SCRT    Error_State

Network 7
SCRE
```

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 конфигурирования модемного модуля (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	288
Использование мастера конфигурирования модемного модуля	294
Обзор команд для модемного модуля и ограничений	298
Команды для модемного модуля	299
Пример программы для модемного модуля	303
CPU S7–200, поддерживающие интеллектуальные модули	303
Биты специальной памяти для модемного модуля	304
Для опытных пользователей	306
Формат телефонных номеров для передачи сообщений	308
Формат текстовых сообщений	309
Формат сообщений для передачи данных CPU	310

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

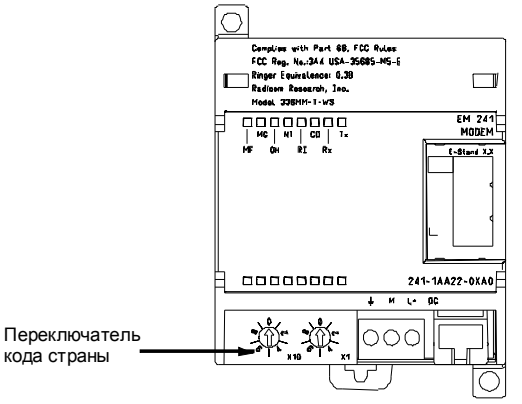
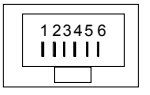


Рис. 10–1. Модемный модуль EM 241

Для настройки модемного модуля можно использовать мастер конфигурирования модемного модуля (Modem Expansion wizard) STEP 7-Micro/WIN. Технические данные модемного модуля вы найдете в Приложении A.

Международный интерфейс с 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.

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 конфигурирования модемного модуля 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 цифровые и текстовые пейджинговые сообщения и сообщения 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 конфигурирования модемного модуля. Мастер конфигурирования модемного модуля 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 конфигурирования модемного модуля и сохраняются в конфигурационном блоке модемного модуля. Затем конфигурационный блок загружается в блок данных в CPU S7–200. Мастер конфигурирования модемного модуля 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 конфигурирования модемного модуля 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 конфигурирования модемного модуля. 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 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 конфигурирования модемного модуля. 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 всех телефонных номеров

Режим обратного вызова выбирается проверкой соответствующей опции в мастере конфигурирования модемного модуля, 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 конфигурирования модемного модуля во время конфигурирования обратных вызовов выбором опции «Enable callbacks to any phone number [Активизировать обратные вызовы для всех 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 конфигурирования модемного модуля 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 конфигурирования модемного модуля, 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 номер 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: ■ 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 конфигурирования модемного модуля

Запустите мастер конфигурирования модемного модуля 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 конфигурирования модемного модуля выберите опцию Configure an EM 241 Modem module [Сконфигурировать модемный модуль EM 241] и щелкните на Next> [Дальше >].
2. Мастеру конфигурирования модемного модуля 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 конфигурирования модемного модуля 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 значением с плавающей точкой.
   - Если формат данных является целым, а количество цифр справа от десятичной точки не равно нулю, то целое значение отображается как шкалированное 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 добавит 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 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, кода 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 конфигурирования модемного модуля создает конфигурационный блок для модемного модуля и требует от пользователя, чтобы он ввел начальный адрес в памяти, где хранится 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> [Дальше>].
16. Теперь мастер 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 для модемного модуля и ограничений

Мастер конфигурирования модемного модуля 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 конфигурирования модемного модуля, 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

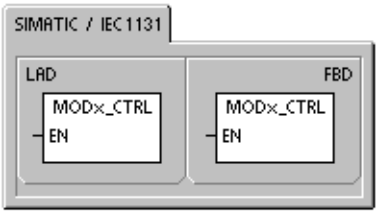
Для использования команд для модемного модуля в своей программе для S7–200 действуйте следующим образом:

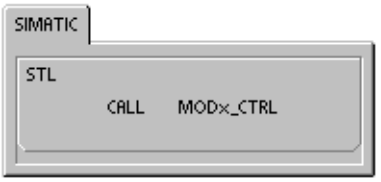
1. С помощью мастера конфигурирования модемного модуля создайте конфигурационную таблицу модемного модуля.
2. Вставьте в свою программу команду MODx_CTRL и используйте контакт SM0.0 для выполнения ее в каждом цикле.
3. Вставьте команду MODx_MSG для каждого сообщения, которое вы хотите послать.
4. Вставьте 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.





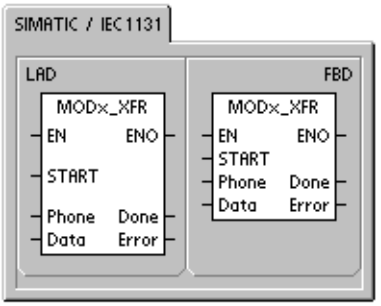
Команда MODx_XFR

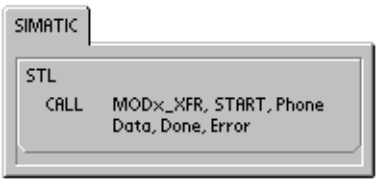
Команда MODx_XFR (Передача данных) используется, чтобы заставить модемный модуль прочитать или записать данные в другой CPU S7–200 или устройство Modbus. Эта команда требует от 20 до 30 секунд от момента подачи сигнала на вход START до установки бита Done [Готово].

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 конфигурирования модемного модуля.

Параметр Data [Данные] – это номер одной из определенных передач данных. Вы можете использовать здесь символическое имя, которое вы присвоили передаче данных с помощью мастера конфигурирования модемного модуля.





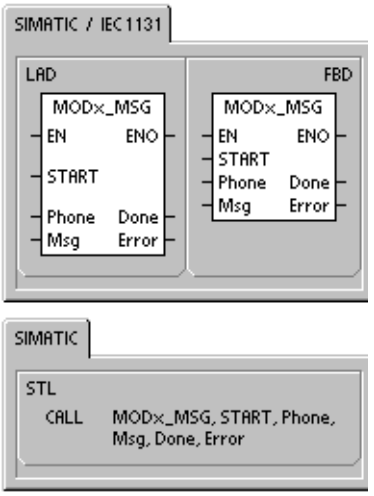
Done – это бит, который включается, когда модемный модуль завершает передачу данных.

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 конфигурирования модемного модуля.

Параметр Msg – это номер одного из определенных сообщений. Вы можете использовать здесь символическое имя, которое вы присвоили сообщению при определении сообщения с помощью мастера конфигурирования модемного модуля.

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

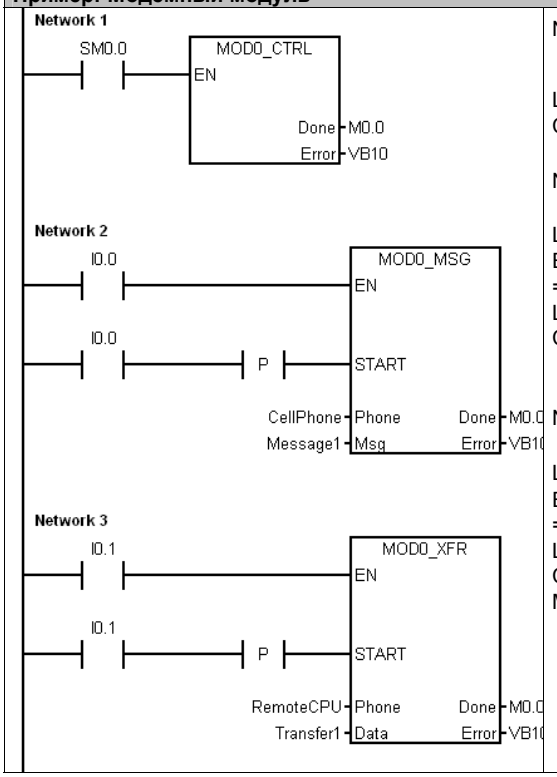
Таблица 10–8. Значения ошибок, возвращаемые командами MODx_MSG и MODx_XFR

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

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: ■ Местоположение модемного модуля или адрес в памяти выходов Q, сконфигурированные с помощью мастера конфигурирования модемного модуля, не совпадает с текущим положением или адресом в памяти ■ См. SMB8 –SMB21 (идентификатор модуля ввода/вывода и регистр ошибок)

Пример программы для модемного модуля

Пример: Модемный модуль




```
Network 1      // Вызов подпрограммы MOD0_CTRL
               // в каждом цикле.

LD     SM0.0
CALL   MOD0_CTRL, M0.0, VB10

Network 2      // Передача сообщения на сотовый
               // телефон.
LD     I0.0
EU
=      L63.7
LD     I0.0
CALL   MOD0_MSG, L63.7, CellPhone, Message1,
       M0.0, VB10

Network 3      // Передача данных на удаленный
               // CPU.
LD     I0.1
EU
=      L63.7
LD     I0.1
CALL   MOD0_XFR, L63.7, RemoteCPU, Transfer1,
M0.0,
       VB10
```

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 версии 1.10 или выше	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relay
CPU 224 версии 1.10 или выше	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relay
CPU 226 версии 1.00 или выше	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relay
CPU 226XM версии 1.00 или выше	CPU 226XM DC/DC/DC
	CPU 226XM 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 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 MSB 7 6 5 4 3 2 1 LSB 0 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).

Таблица 10–11. Адреса специальной памяти для модемного модуля EM 241, продолжение

Адрес SM- памяти	**Описание**
SMB226	Результат команды пользователя MSB 7 6 5 … LSB 0 D | 0 | ERROR D - бит завершения операции; 0 – операция выполняется 1 – операция завершена ERROR: описание кода ошибки, см. таблицу 10-8
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.

Для опытных пользователей

Обзор конфигурационной таблицы

Мастер конфигурирования модемного модуля был спроектирован таким образом, чтобы упростить применения модема путем автоматической генерации конфигурационной таблицы на основе ваших ответов о системе. Информация, содержащаяся в конфигурационной таблице, предназначена для опытных пользователей, которые хотели бы создавать свои собственные программы управления модемным модулем и формировать свои собственные сообщения.

Конфигурационная таблица расположена в области памяти переменных S7–200. В таблице 10–12 первый столбец содержит байтовое смещение относительно адреса, на который показывает указатель на конфигурационную область в специальной памяти. Информация в конфигурационной таблице делится на четыре раздела.

- Конфигурационный блок содержит информацию для конфигурирования модуля.
- Блок телефонных номеров для обратного вызова содержит заранее определенные телефонные номера для функции обратного вызова.
- Блок телефонных номеров для сообщений содержит телефонные номера, используемые для передачи сообщений или передач данных CPU.
- Блок сообщений содержит заранее определенные сообщения для передачи в службы передачи сообщений.

Таблица 10–12. Конфигурационная таблица для модемного модуля

Конфигурационный блок	
Байтовое смещение	**Описание**
от 0 до 4	Идентификатор модуля – Пять символов ASCII, используемых для установления соответствия между конфигурационной таблицей и интеллектуальным модулем. Версия 1.00 модемного модуля EM 241 ожидает идентификатора «M241A».
5	Длина конфигурационного блока – в настоящее время 24.
6	Длина телефонного номера для обратного вызова – Допустимые значения от 0 до 40.
7	Длина телефонного номера для передачи сообщений – Допустимые значения от 0 до 120.
8	Количество телефонных номеров обратного вызова – Допустимые значения от 0 до 250.
9	Количество телефонных номеров для передачи сообщений Допустимые значения от 0 до 250.
10	Количество сообщений – Допустимые значения от 0 до 250.
11, 12	Резерв (2 байта)
13	Этот байт содержит деблокирующие биты для поддерживаемых функций. MSB 7 6 5 4 3 2 1 LSB 0 PD CB PW MB BD 0 0 0 PD - 0 = тональный набор 1 = импульсный набор CB - 0 = обратный вызов заблокирован 1 = обратный вызов разрешен PW - 0 = пароль заблокирован 1 = пароль разрешен MB - 0 = разблокирован протокол PPI 1 = разблокирован протокол Modbus BD - 0 = "слепой" набор заблокирован 1 = слепой набор разрешен Биты 2, 1 и 0 модулем игнорируются
14	Резерв
15	Попытки – Эта величина указывает, сколько раз модем пытается набрать номер и послать сообщение прежде чем выдать ошибку. Значение 0 препятствует модему набирать номер.
от 16 до 23	Пароль – Восемь символов ASCII

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 - 8 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, S, SM, V, T, C и AI. 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 числа, 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 в120 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).

В этой главе

Требования для использования протокола USS 312
Расчет времени, необходимого для обмена данными с приводом 313
Использование команд USS 314
Команды для протокола USS 315
Примеры программ для протокола USS 322
Коды ошибок выполнения команд USS 323
Подключение и наладка привода MicroMaster серии 3 324
Подключение и наладка привода MicroMaster серии 4 327

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.

  Для назначения порта 0 протоколу USS или протоколу 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 226, CPU 226XM 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 3450 байт. В зависимости от конкретно используемых команд USS программы поддержки этих команд могут увеличить программу управления минимум на 2150 байт и максимум на 3450 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 получения 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. Сконфигурируйте параметры приводов, чтобы согласовать их с используемыми в программе скоростью передачи и адресом.

Рис. 11–1. Выделение памяти переменных для библиотеки команд

Пояснение к рисунку: Inhalte einfügen – Вставить содержимое; Speicher für Bibliothek – Память для библиотеки

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.

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.

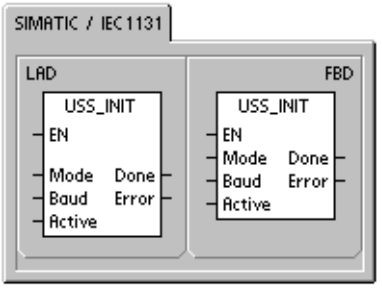
Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое соединено при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано.

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

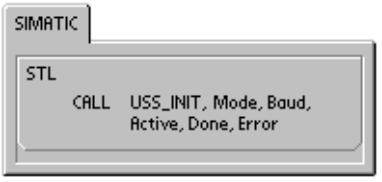
Команда USS_INIT

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 деблокирует этот протокол, а входное значение, равное 0, назначает порт 0 протоколу PPI и блокирует протокол USS.

Параметр Baud устанавливает скорость передачи на 1200, 2400, 4800, 9600, 19200, 38400, 57600 или 115200 Бод.

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.

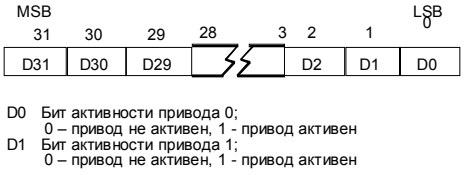
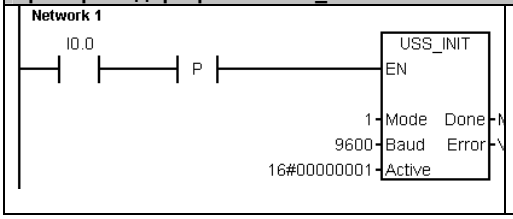


Рис. 11–2. Формат параметра для активного привода

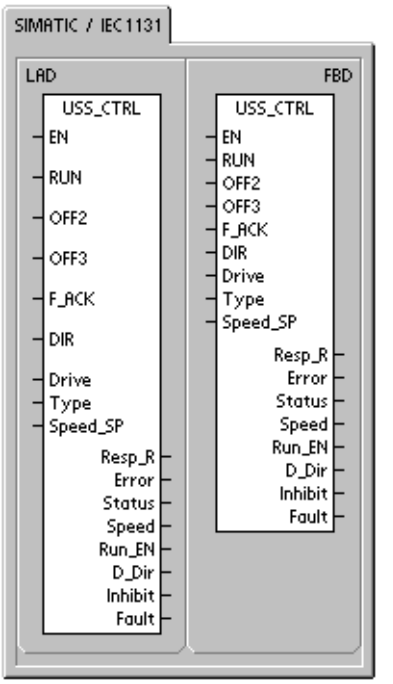
Когда команда USS_INIT завершается, включается выход Done. Выходной байт Error содержит результат выполнения команды. Таблица 11–6 определяет возможные ошибочные состояния, которые могут возникать в результате выполнения команды.

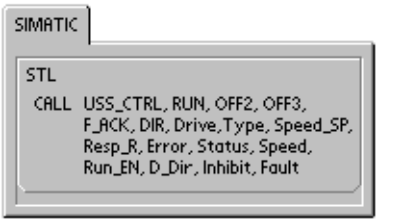
Пример: Подпрограмма USS_INIT




```
Network 1
LD      I0.0
EU
CALL    USS_INIT, 1, 9600, 16#00000001, M0.0, VB10
```

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 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 разблокировать команду USS_CTRL. Эта команда всегда должна быть разблокирована.

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 [подтверждение неисправности]) используется для того, чтобы подтвердить 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.

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.

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

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 ошибочные состояния, которые могут возникнуть в результате выполнения команды.

Status – это необработанное 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 [деблокировка RUN]) показывает, работает привод (1) или остановлен (0).

D_Dir показывает направление вращения привода.

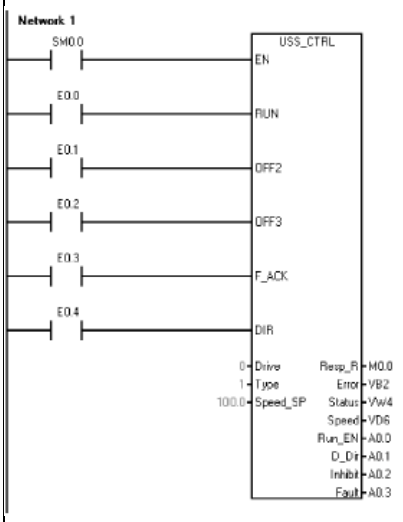
Inhibit показывает состояние бита запрета в приводе (0 - не запрещен, 1 - запрещен). Для сброса бита запрета, нужно выключить бит Fault, и входы RUN, OFF2 и OFF3 тоже должны быть выключены.

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

Пример: Подпрограмма USS_CTRL



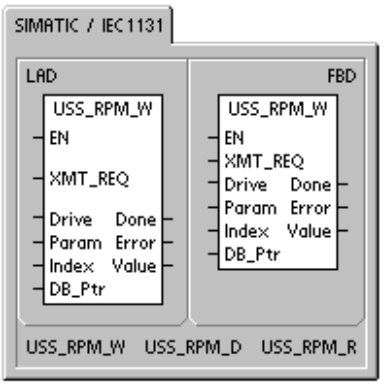
Отображается только в STL:

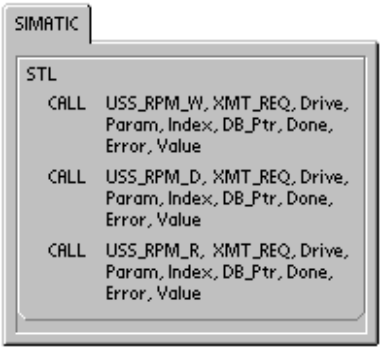
```
Network 1      //Блок управления для привода  0
LD      SM0.0
CALL    USS_CTRL, I0.0, I0.1, I0.2, I0.3, I0.4, 0, 1, 100.0, M0.0,
VB2, VW4, VD6, Q0.0, Q0.1, Q0.2, Q0.3
```

Отображается в LAD или FBD:

```
Network 1      //Блок управления для привода 0
LD      SM0.0
=       L60.0
LD       I0.0
=       L63.7
LD      I0.1
=       L63.6
LD      I0.2
=        L63.5
LD      I0.3
=       L63.4
LD      I0.4
=       L63.3
LD      L60.0
CALL    USS_CTRL, L63.7, L63.6, L63.5, L63.4, L63.3, 0, 1, 100.0,
M0.0, VB2, VW4, VD6, Q0.0, Q0.1, Q0.2, Q0.3
```

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 ошибочные состояния, которые могут возникнуть 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:

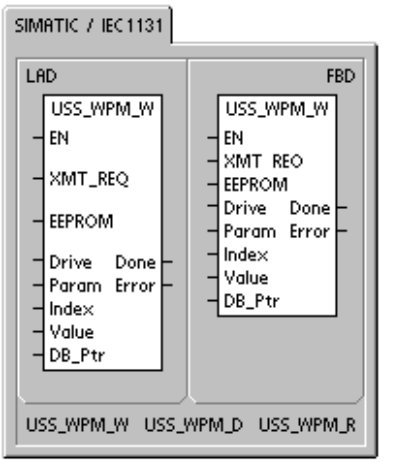
- Команда USS_WPM_W записывает не имеющий знака параметр в формате слова.
- Команда USS_WPM_D записывает не имеющий знака параметр в формате двойного слова.
- Команда USS_WPM_R записывает параметр в формате числа с плавающей точкой.

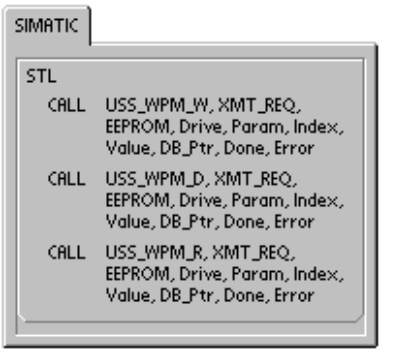
В каждый конкретный момент времени может быть активна только одна команда чтения (USS_RPM_x) или записи (USS_WPM_x).

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 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

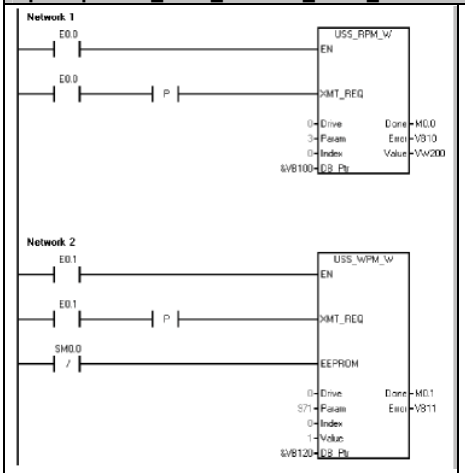
Предостережение

Когда вы используете команду USS_WPM_x для обновления набора параметров в ЭСППЗУ, вы должны гарантировать, что не превышается 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.

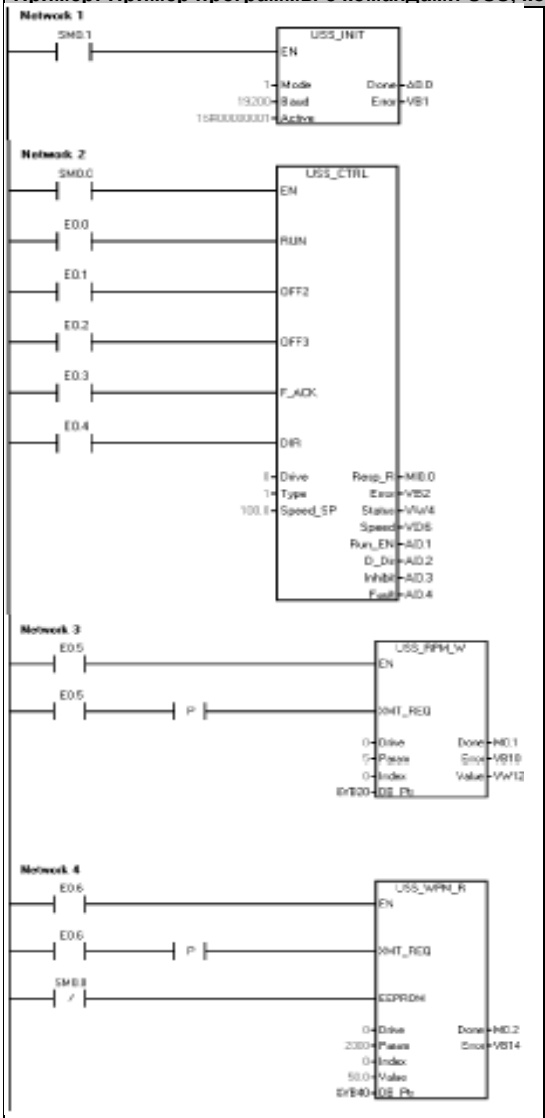
Пример: USS_RPM_x и USS_WPM_x




```
Network 1       //Оба контакта должны иметь один и тот же адрес.
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

Network 2       //Оба контакта должны иметь один и тот же адрес
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
```

Примеры программ для протокола USS

Пример: Пример программы с командами USS, которая правильно отображается в STL




```
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, индекс 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

Примечание:  Этот код STL не компилируется в
             LAD или FBD.
```

Коды ошибок выполнения команд USS

Таблица 11–6. Коды ошибок выполнения команд USS

Коды ошибок	Описание
0	Ошибок 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 разъеме. Рекомендуется соединить клемму 2-0V на приводе MicroMaster с массой.

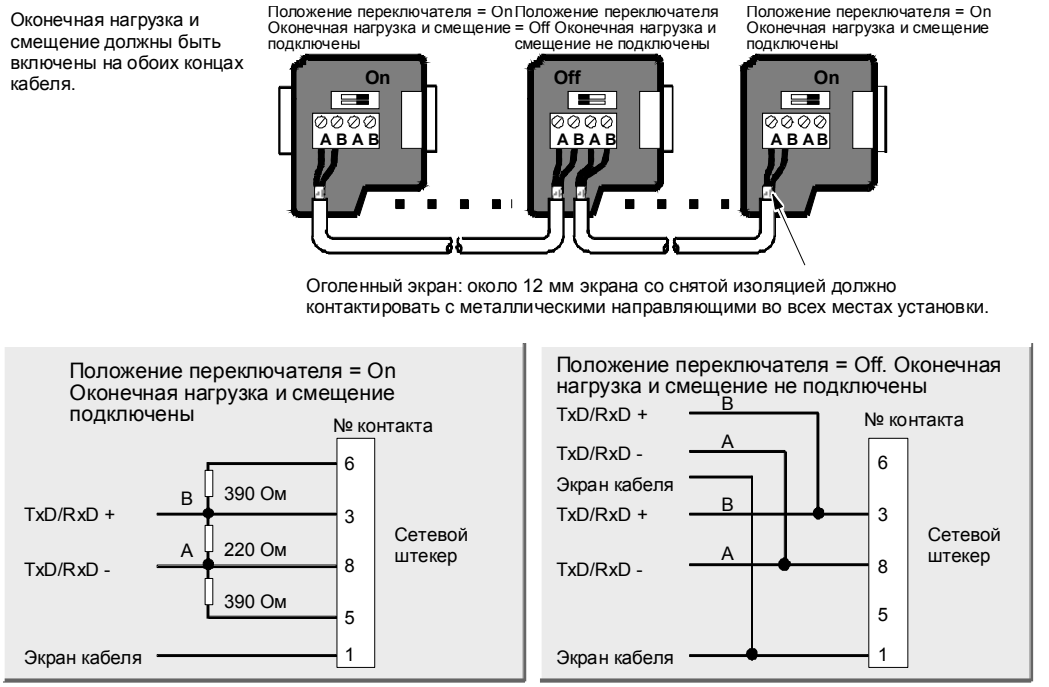


Рис. 11–5. Оконечная нагрузка и смещение сетевого кабеля

Настройка привода MicroMaster 3

Перед подключением привода к S7–200 вы должны убедиться, что привод имеет указанные ниже системные параметры. Параметры устанавливаются с помощью вспомогательной клавиатуры привода:

1. Восстановите заводскую настройку параметров привода (не обязательно). Нажмите клавишу P: на дисплее появится P000. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P944. Нажмите клавишу P, чтобы ввести параметр.

   P944=1

2. Разрешите доступ для чтения и записи ко всем параметрам. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P009. Нажмите клавишу P, чтобы ввести параметр.

   P009=3

3. Проверьте параметры настройки двигателя вашего привода. Параметры настройки изменяются соответственно используемому двигателю (двигателям). Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появятся настройки двигателя для вашего привода. Нажмите клавишу P, чтобы ввести параметр.

   P081=Номинальная частота двигателя (Гц)
   P082=Номинальная скорость двигателя (оборотов в минуту)
   P083=Номинальный ток двигателя (A)
   P084=Номинальное напряжение двигателя (В)
   P085=Номинальная мощность двигателя (кВт/л.с.)

4. Установите режим управления “локальный/дистанционный”. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P910. Нажмите клавишу P, чтобы ввести параметр.

   P910=1 Режим управления “дистанционный”

5. Установите скорость передачи последовательного интерфейса RS-485. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P092. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий скорости передачи вашего последовательного интерфейса RS-485. Для ввода нажмите клавишу P.

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 (значение по умолчанию). Изменения в настройке параметров сохраняются в периоды, когда питание снимается.

13. Отображение режима. Нажмите P, чтобы выйти из режима задания параметров.

Присоединение 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 имеющих винтов, предназначенных для работы USS. Для подключения S7–200 можно использовать стандартные 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).

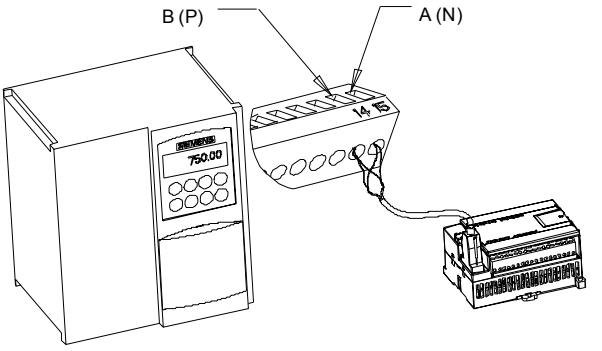


Рис. 11–6. Присоединение клеммного блока MM420

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, модель 6SE6420.

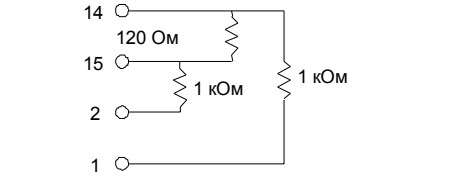


Рис. 11–7. Пример подключения оконечной нагрузки и смещения

Настройка привода MM4

Перед подключением привода к S7–200 вы должны убедиться, что привод имеет указанные ниже системные параметры. Параметры устанавливаются с помощью вспомогательной клавиатуры привода:

1. Восстановите заводскую настройку параметров привода (не обязательно):
   P0010=30
   P0970=1

   Если вы пропускает 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, установить 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 COM-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.

В этой главе

Требования для использования протокола Modbus 330
Инициализация и время исполнения протокола Modbus 330
Адресация Modbus 331
Использование команд протокола Modbus Slave 332
Команды протокола Modbus Slave 333

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), связанными 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 MaxAI команды MBUS_INIT указывает максимальное количество входных регистров(AIW), к которым может обращаться master-устройство Modbus.

Параметр MaxHold команды MBUS_INIT указывает максимальное количество регистров временного хранения информации (памяти переменных), к которым Modbus разрешен доступ master-устройству Modbus.

Дальнейшую информацию, необходимую для настройки ограничений памяти для Modbus Slave, вы найдете в описании команды MBUS_INIT.

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.

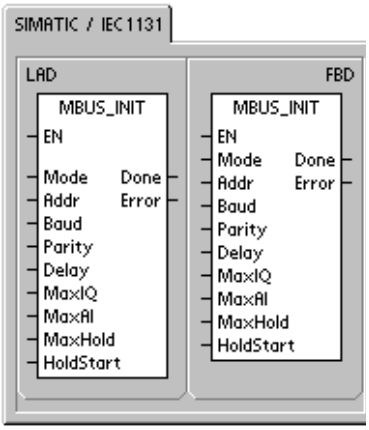
Команды протокола Modbus Slave поддерживают протокол Modbus RTU. Эти команды используют свободно программируемый обмен данными S7–200 для поддержки наиболее часто используемых функций Modbus. Поддерживаются следующие функции Modbus:

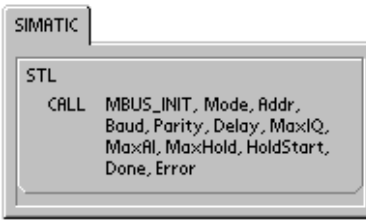
Таблица 12–2. Функции, поддерживаемые протоколом Modbus Slave

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 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.

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 стандартное условие Modbus 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 часто, установите задержку на значение от 10 до 100 миллисекунд. Значение параметра Delay может находиться в переделах от 0 до 32767 миллисекунд.

Параметр MaxIQ устанавливает количество входов и выходов, предоставляемых в распоряжением 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:

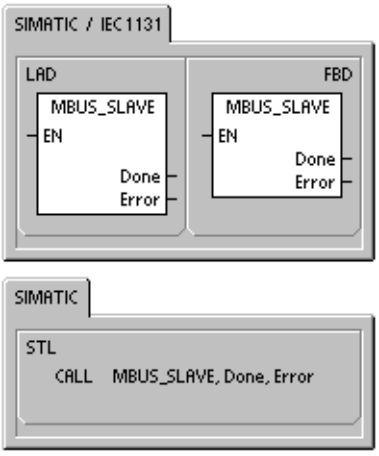
- 0 для CPU 221
- 16 для CPU 222
- 32 для CPU 224, 226 и 226XM

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.

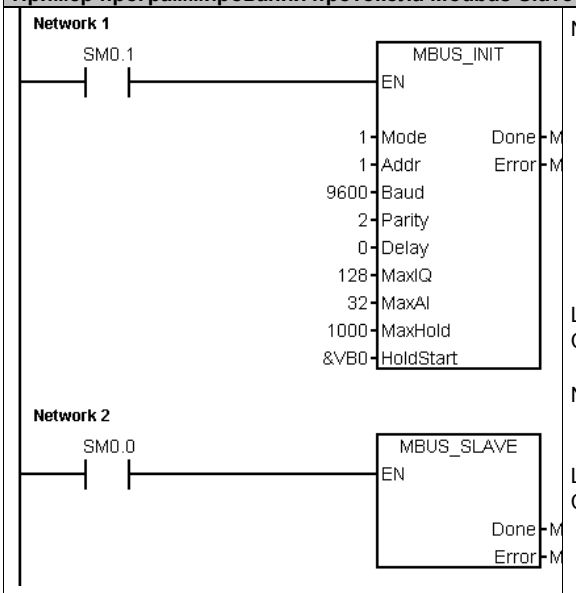
Таблица 12–4. Параметры для команды MBUS_SLAVE

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

Пример программирования протокола Modbus Slave




```
Network 1
          //Инициализировать протокол Modbus
          //Slave в первом цикле.
          //Установить адрес slave-устройства в 1,
          //установить для порта 0 скорость
          //передачи 9600 Бод с контролем на
          //четность, разрешить всем доступ ко
          //всем значениям I, Q и AI, разрешить
          // доступ к 1000 регистров для
          //временного хранения информации
          //(2000 байтов), начиная с VB0.
LD        SM0.1
CALL      MBUS_INIT,1,1,9600,2,0,128,32,1000,
          &VB0,M0.1,MB1
Network 2
          //Выполнять протокол Modbus Slave
          // в каждом цикле.
LD        SM0.0
CALL      MBUS_SLAVE,M0.2,MB2
```

Технические данные

A

В этой главе

Общие технические данные 338
Технические данные CPU 340
Технические данные цифровых модулей расширения 346
Технические данные аналоговых модулей расширения 351
Технические данные модулей расширения для термопар и термометров сопротивления (RTD) 361
Технические данные модуля EM 277 PROFIBUS-DP 373
Технические данные модемного модуля EM 241 385
Технические данные модуля позиционирования EM 253 387
Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2) 393
Дополнительные съемные модули 395
Кабель расширения входов-выходов 395
Кабель PC/PPI 396
Имитаторы входов 398

Общие технические данные

Соответствие стандартам

Национальные и международные стандарты, перечисленные ниже, использовались для определения рабочих характеристик и проведения испытаний семейства изделий S7-200. Таблица A–1 определяет конкретное соответствие этим стандартам.

- Директива Европейского сообщества (CE) по устройствам низкого напряжения 73/23/EEC
  EN 61131-2: Программируемые контроллеры – Требования к оборудованию
- Директива Европейского сообщества (CE) по электромагнитной совместимости 89/336/EEC

  Стандарты по электромагнитному излучению
  EN 50081-1: Жилая, коммерческая зоны и легкая промышленность
  EN 50081-2: 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 взаимные исследования]: FM класс I, раздел 2, группы опасных помещений A, B, C и D, T4A. Класс I, зона 2, IIC, T4.

Удостоверения о допуске к эксплуатации на море

В следующей таблице приведены морские агентства и номера соответствующих сертификатов на момент печати данного руководства.

Информацию о текущих допусках к эксплуатации вы можете получить в местных представительствах 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-8071
Bureau Veritas [Бюро Веритас] (BV)	09051 / A2 BV
Nippon Kaiji Kyokai [Ниппон Кэйдзи Кёкай, Япония] (NK)	A-534

Технические данные

Все CPU S7–200 и модули расширения соответствуют техническим данным, перечисленным в таблице A–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.

Срок службы электрической части реле

На рис. A–1 показаны типовые эксплуатационные данные, задаваемые поставщиками реле. Фактическая характеристика может отличаться в зависимости от вашего конкретного приложения.

Внешний защитный контур, подобранный в соответствии с нагрузкой, продлит срок службы контактов.

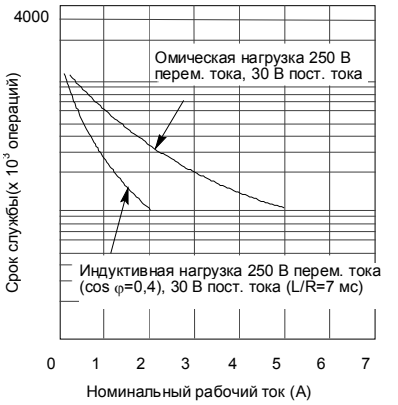


Рис. A–1. Срок службы электрической части реле

Таблица A–1. Технические данные

Условия окружающей среды — Транспортировка и хранение	
IEC 68-2-2, испытание Bb, сухое тепло и IEC 68-2-1, испытание Ab, холод	от –40° C до +70° C
IEC 68-2-30, испытание Db, влажное тепло	от 25° C до 55° C, влажность 95%
IEC 68-2-31, опрокидывание	100 мм, 4 падения, без упаковки
IEC 68-2-32, 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° C до 55° C при горизонтальной установке, от 0° C до 45° C при вертикальной установке Влажность 95% без конденсации
IEC 68-2-14, испытание Nb	от 5° C до 55° C, 3° C в минуту
IEC 68-2-27, механический удар	15 G, импульс 11 мс, 6 ударов по каждой из 3 осей
IEC 68-2-6, 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	Предохраняет пальцы от контакта с высоким напряжением, как проверено стандартными щупами. Внешняя защита требуется для предохранения от пыли, грязи, воды и инородных предметов диаметром менее 12,5 мм.
Электромагнитная совместимость – Помехоустойчивость согласно EN61000-6-2[1]	
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	от 80 МГц до 1 ГГц 10, В/м, модуляция 80% сигналом 1 кГц
EN 61000-4-4, быстрые переходные помехи	2 кВ, 5 кГц при наличии цепи связи с системными напряжениями питания переменного и постоянного тока 2 кВ, 5 кГц при наличии соединительной клеммы с цифровыми 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 (цепи 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 EN50081 -1[2] и -2	
EN 55011, класс A, 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 A, группа 1, излучаемое [1] 30 МГц - 230 МГц 230 МГц - 1 ГГц	 30 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 30 м 37 дБ (мкВ/м) квазипиковое; измеренное на расстоянии 30 м
EN 55011, класс B, 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 B, 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. 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

1 Блок должен устанавливаться на заземленной металлической раме с присоединением “земли” S7-200 непосредственно к металлу монтажной арматуры. Кабели прокладываются вдоль металлических опор.

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.

Технические данные CPU

Таблица A–2. Номера для заказа CPU

Номер для заказа	Модель CPU	Блок питания CPU (номинальное напряжение)	Входы CPU	Выходы CPU	Съемный клеммный блок
6ES7 211-0AA22-0XB0	CPU 221	24 В пост. тока	6 x 24 В пост. тока	4 x 24 В пост. тока	Нет
6ES7 211-0BA22-0XB0	CPU 221	от 120 до 240 В перем. тока	6 x 24 В пост. тока	4 x реле	Нет
6ES7 212-1AB22-0XB0	CPU 222	24 В пост. тока	8 x 24 В пост. тока	6 x 24 В пост. тока	Нет
6ES7 212-1BB22-0XB0	CPU 222	от 120 до 240 В перем. тока	8 x 24 В пост. тока	6 x реле	Нет
6ES7 214-1AD22-0XB0	CPU 224	24 В пост. тока	14 x 24 В пост. тока	10 x 24 В пост. тока	Да
6ES7 214-1BD22-0XB0	CPU 224	от 120 до 240 В перем. тока	14 x 24 В пост. тока	10 x реле	Да
6ES7 216-2AD22-0XB0	CPU 226	24 В пост. тока	24 x 24 В пост. тока	16 x 24 В пост. тока	Да
6ES7 216-2BD22-0XB0	CPU 226	от 120 до 240 В перем. тока	24 x 24 В пост. тока	16 x реле	Да
6ES7 216-2AF22-0XB0	CPU 226XM	24 В пост. тока	24 x 24 В пост. тока	16 x 24 В пост. тока	Да
6ES7 216-2BF22-0XB0	CPU 226XM	от 120 до 240 В перем. тока	24 x 24 В пост. тока	16 x реле	Да

Таблица A–3. Общие технические данные CPU

Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Имеющиеся напряжения пост. тока +5 В, +24 В	
6ES7 211-0AA22-0XB0	CPU 221 DC/DC/DC 6 входов/ 4 выхода	90 x 80 x 62	270 г	3 Вт	0 мА	180 мА
6ES7 211-0BA22-0XB0	CPU 221 AC/DC/Relay 6 входов/ 4 реле	90 x 80 x 62	310 г	6 Вт	0 мА	180 мА
6ES7 212-1AB22-0XB0	CPU 222 DC/DC/DC 8 входов/ 6 выходов	90 x 80 x 62	270 г	5 Вт	340 мА	180 мА
6ES7 212-1BB22-0XB0	CPU 222 AC/DC/Relay 8 входов/ 6 реле	90 x 80 x 62	310 г	7 Вт	340 мА	180 мА
6ES7 214-1AD22-0XB0	CPU 224 DC/DC/DC 14 входов/ 10 выходов	120.5 x 80 x 62	360 г	7 Вт	660 мА	280 мА
6ES7 214-1BD22-0XB0	CPU 224 AC/DC/Relay 14 входов/ 10 реле	120.5 x 80 x 62	410 г	10 Вт	660 мА	280 мА
6ES7 216-2AD22-0XB0	CPU 226 DC/DC/DC 24 входа/16 выходов	196 x 80 x 62	550 г	11 Вт	1000 мА	400 мА
6ES7 216-2BD22-0XB0	CPU 226 AC/DC/Relay 24 входа/16 реле	196 x 80 x 62	660 г	17 Вт	1000 мА	400 мА
6ES7 216-2AF22-0XB0	CPU 226XM DC/DC/DC 24 входа/16 выходов	196 x 80 x 62	550 г	11 Вт	1000 мА	400 мА
6ES7 216-2BF22-0XB0	CPU 226XM AC/DC/Relay 24 входа/16 реле	196 x 80 x 62	660 г	17 Вт	1000 мА	400 мА

Таблица A–4. Технические данные CPU

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Память					
Размер программы пользователя (ЭСППЗУ)	2048 слов		4096 слов	4096 слов	8192 слова
Данные пользователя (ЭСППЗУ)	1024 слова (сохраняются при отключении питания)		2560 слов (сохраняются при отключении питания)	2560 слов (сохраняются при отключении питания)	5120 слов (сохраняются при отключении питания)
Буферизация (мощный конденсатор) (возможна батарейка)	Тип. 50 час. (мин. 8 час. при 40°C) Тип. 200 дней		Тип. 190 час. (мин. 120 час. при 40°C) Тип. 200 дней		
Входы/выходы					
Встроенные цифровые входы и выходы	6 входов/4 выхода	8 входов/6 выходов	14 входов/10 выходов	24 входа/16 выходов	
Цифровые входы и выходы (образ процесса)	256 (128 входов/128 выходов)				
Аналоговые входы и выходы (образ процесса)	Нет	32 (16 входов/16 выходов)	64 (32 входов/32 выходов)		
Макс. количество модулей расширения	Нет	2 модуля	7 модулей		
Макс. количество интеллектуальных модулей	Нет	2 модуля	7 модулей		
Входы для захвата импульсов	6	8	14		
Скоростные счетчики однофазные двухфазные	Всего 4 счетчика 4 при 30 кГц 2 при 20 кГц		Всего 6 счетчиков 6 при 30 кГц 4 при 20 кГц		
Импульсные выходы	2 при 20 кГц (только выходы постоянного тока)				

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
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	0,37 мкс на команду				
Часы реального времени	Дополнительный съемный модуль		Встроенные		
Дополнительные съемные модули	Память, батарейка и часы реального времени		Память и		
Встроенные средства для обмена данными					
Порты	Порт 1 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 повторителем для гальванической развязки: 1000 м до 187,5 кБод, 1200 м до 38,4 кБод Без повторителя для 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)				

Таблица A–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	Макс. нагрузка
CPU 221	80 мА	450 мА	30/15 мА при 120/240 В перем. тока	120/60 мА при 120/240 В перем. тока
CPU 222	85 мА	500 мА	40/20 мА при 120/240 В перем. тока	140/70 мА при 120/240 В перем. тока
CPU 224	110 мА	700 мА	60/30 мА при 120/240 В перем. тока	200/100 мА при 120/240 В перем. тока
CPU 226/CPU 226XM	150 мА	1050 мА	80/40 мА при 120/240 В перем. тока	320/160 мА при 120/240 В перем. тока
Толчок тока при включении	10 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 V Slow Blow		2 А, 250 V Slow Blow	
Питание датчиков 24 В пост. тока				
Напряжение датчиков	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			
Пульсации/помехи	Наводки от входного напряжения		Менее 1 В от пика к пику	
Электрическая развязка (датчиков с логикой)	Нет развязки			

Таблица A–6. Технические данные входов CPU

Общие данные	Вход 24 В пост. тока
Тип	Обычная полярность/обратная полярность (IEC тип 1 – обычная полярность)
Номинальное напряжение	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
Логика 0 (макс.)	5 В пост. тока при 1 мА
Входная задержка	Настраивается (от 0,2 до 12,8 мс) CPU 226, CPU 226XM: входы I1.6 – I2.7 имеют фиксированную задержку (4,5 мс)
Присоединение 2-проводного датчика близости (Bero) Допустимый ток утечки (макс.)	1 мА

Таблица A–6. Технические данные входов CPU, продолжение

Общие данные	Вход 24 В пост. тока	
Электрическая развязка (полевых устройств с логикой) Оптическая (гальваническая) Потенциально развязанные группы	Да 500 В перем. тока в течение 1 минуты См. коммутационную схему	
Частота на входе скоростного счетчика (макс.) Логика 1 = от 15 до 30 В пост. тока Логика 1 = от 15 до 26 В пост. тока	Однофазный счетчик 20 кГц 30 кГц	Двухфазный счетчик 10 кГц 20 кГц
Одновременно включенные входы	Все при 55° C	
Длина кабеля (макс.) экранированный не экранированный	500 м для обычных входов, 50 м для входов скоростных счетчиков 300 м для обычных входов	

Таблица A–7. Технические данные выходов CPU

Общие данные	Выход 24 В пост. тока	Релейный выход
Тип	Канальный полевой униполярный МОП-транзистор [1]	Сухой магнитоуправляемый слаботочный контакт
Номинальное напряжение	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 30 В пост. тока или от 5 до 250 В перем. тока
Ток перегрузки (макс.)	8 А в течение 100 мс	7 А при замкнутых контактах
Логика 1 (мин.)	20 В пост. тока при максимальном 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 модуль (макс.)	6 А	10 А
Ток утечки (макс.)	10 мкА	-
Ламповая нагрузка (макс.)	5 Вт	30 Вт пост. тока; 200 Вт перем. тока
Индуктивное напряжение на клеммах	L+ минус 48 В пост. тока, мощность потерь 1 Вт	-
Сопротивление в состоянии "включено" (контакт)	0,3 Ом макс.	0,2 Ом (макс., когда новый)
Электрическая развязка оптическая (гальваническая, полевое устройство с логикой) логика с контактом контакт с контактом сопротивление (логика с контактом) потенциально развязанные группы	 500 В перем. тока в течение 1 минуты - - - См. коммутационную схему	 - 1500 В перем. тока в течение 1 минуты 750 В перем. тока в течение 1 минуты 100 МОм См. коммутационную схему
Задержка выкл.-вкл./вкл.-выкл. (макс.) Процессы переключения (макс.)	2/10 мкс (Q0.0 и Q0.1) 15/100 мкс (все остальные) -	- 10 мс
Частота следования импульсов (макс.) Q0.0 и Q0.1	20 кГц	1 Гц
Срок службы механической части, циклов переключения	-	10 000 000 (без нагрузки)
Срок службы контактов	-	100 000 (при номинальной нагрузке)
Число одновременно включенных выходов	Все при 55° C	Все при 55° C
Параллельное включение двух выходов	Да	Нет
Длина кабеля (макс.) экранированный не экранированный	 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.

Коммутационные схемы

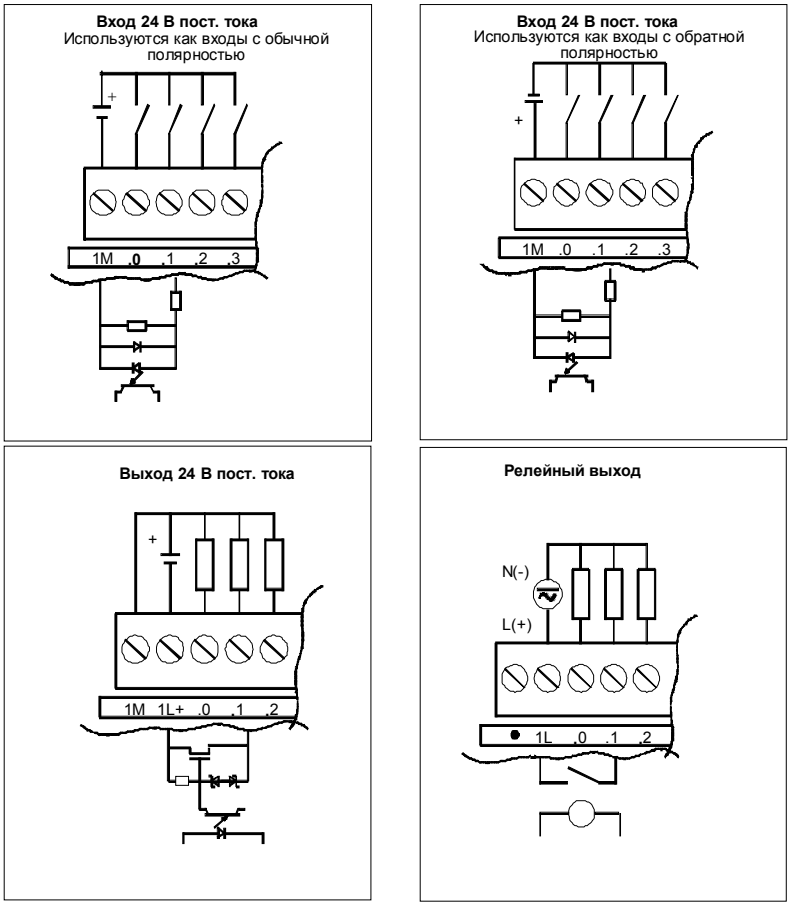


Рис. А–2. Входы и выходы CPU

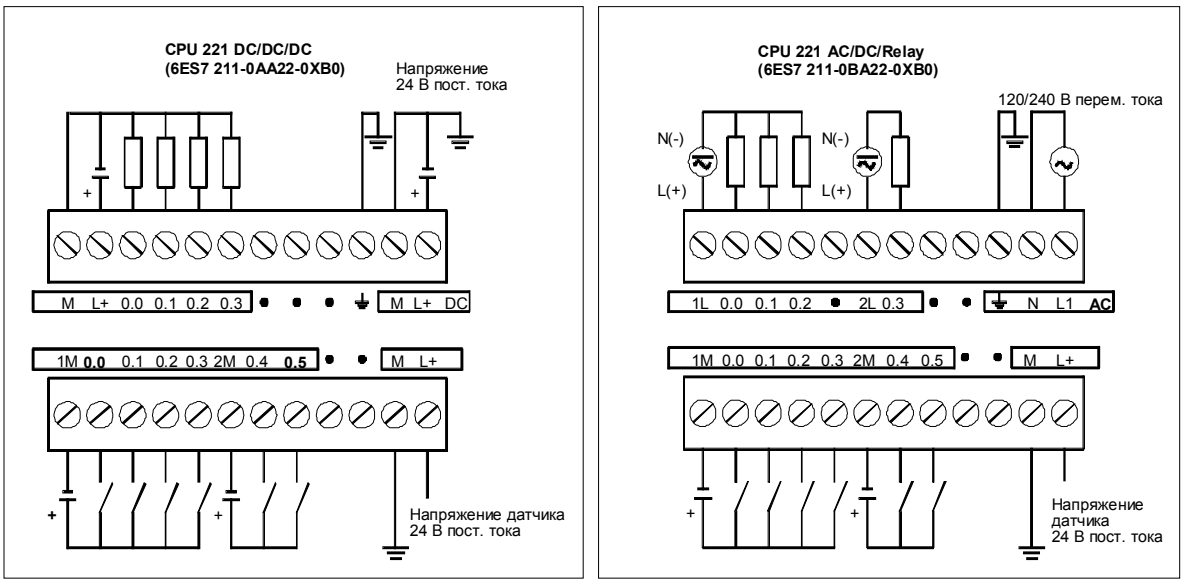


Рис. А–3. Коммутационные схемы CPU 221

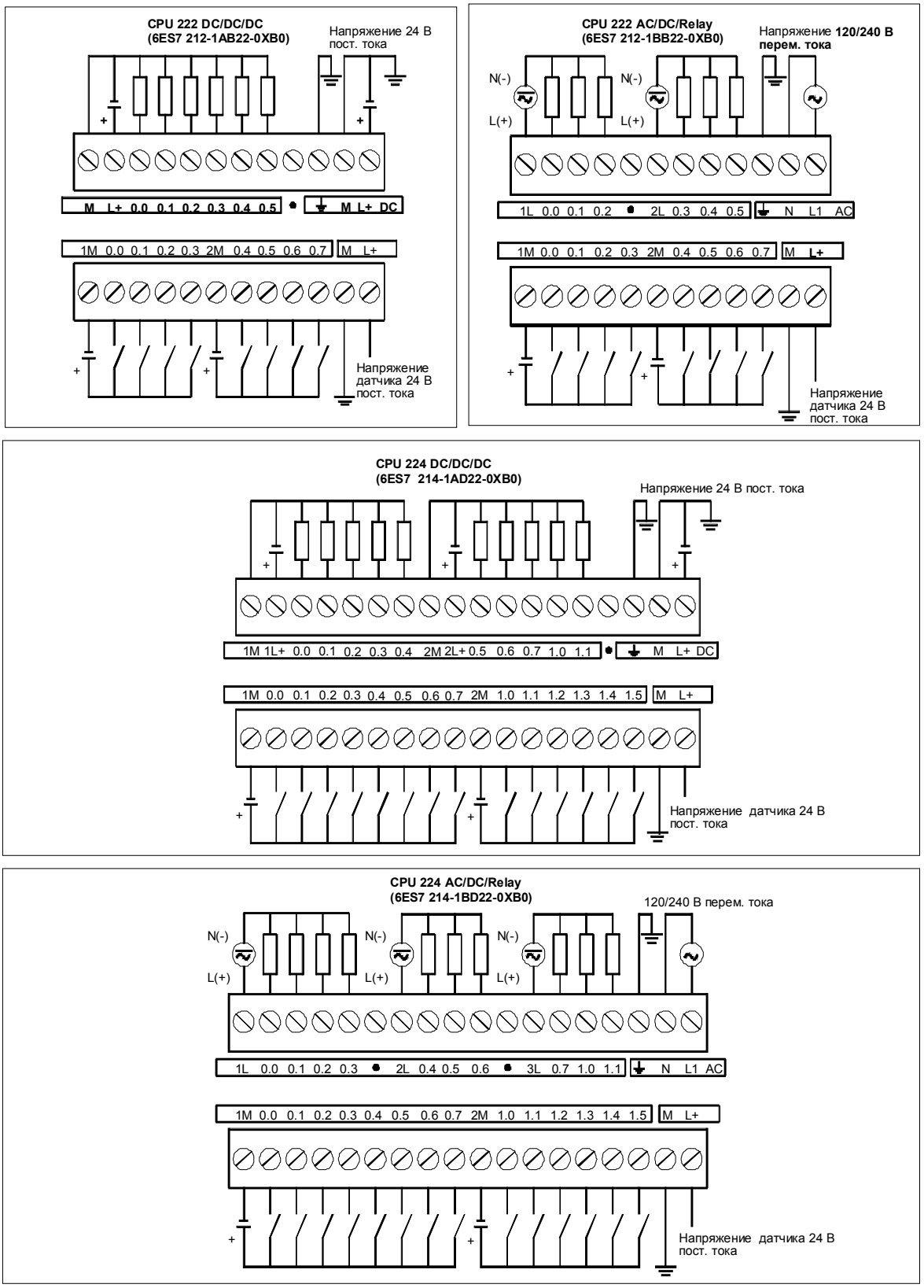
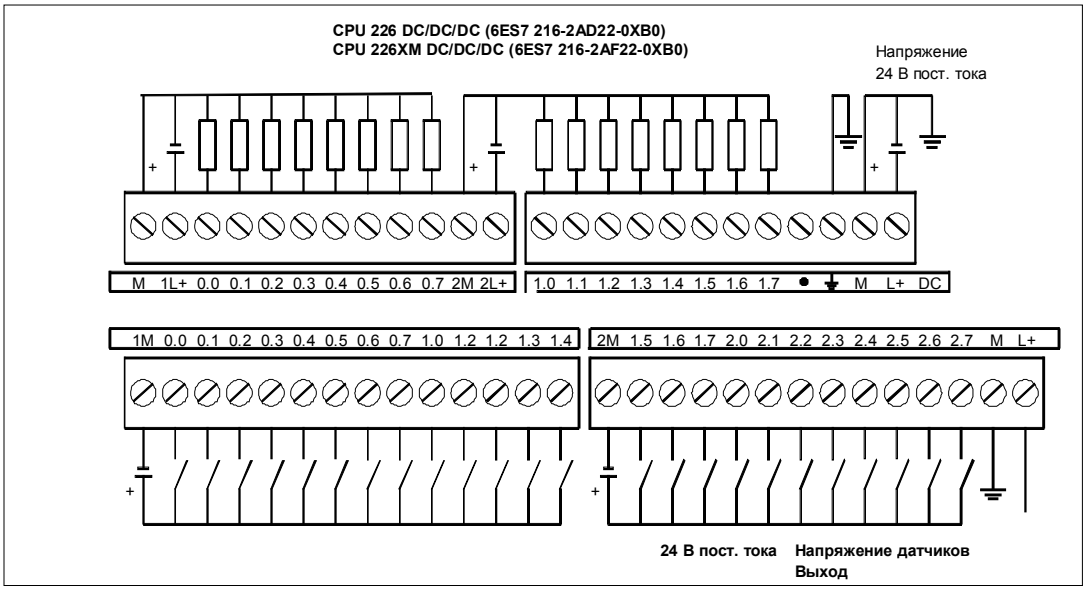


Рис. A–4. Коммутационные схемы CPU 222 и CPU 224



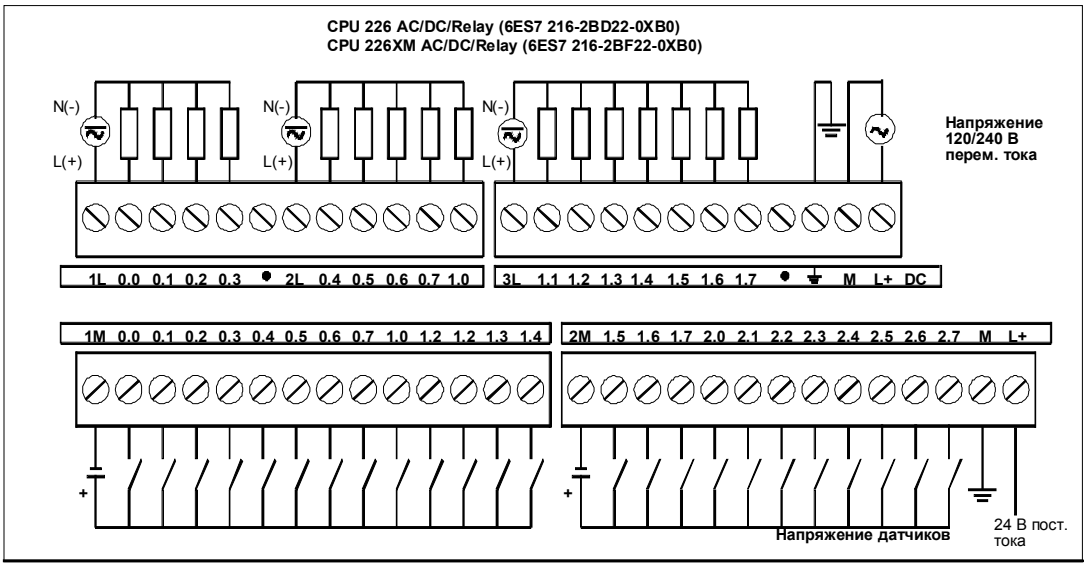


Рис. А–5. Коммутационные схемы CPU 226 и CPU 226XM

Таблица А–8. Назначение контактов коммуникационного порта S7–200

Разъем	Номер контакта	Сигнал PROFIBUS	Порт 0/Порт 1
Кон-такт 1 / Кон-такт 6 / Кон-такт 9 / Кон-такт 5	1	Экран	Заземление
	2	Обратный провод 24 В	Общий провод логики
	3	RS–485, сигнал B	RS–485, сигнал B
	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 A	RS–485, сигнал A
	9	Не используется	10-битовый протокол (вход)
	Корпус разъема	Экран	Заземление

Технические данные цифровых модулей расширения

Таблица A–9. Номера для заказа цифровых модулей расширения

Номер для заказа	Модуль расширения	Входы модуля расширения	Выходы модуля расширения	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 222-1BF22-0XA0	EM 222, цифровой вывод, 8 x 24 В пост. тока	-	8 x 24 В пост. тока	Да
6ES7 222-1HF22-0XA0	EM 222, цифровой вывод, 8 x реле	-	8 x реле	Да
6ES7 222-1EF22-0XA0	EM 222, цифровой вывод, 8 x 120/230 В перем. тока	-	8 x 120/230 В перем. тока	Да
6ES7 223-1BF22-0XA0	EM 223, 24 В пост. тока, цифровой ввод/вывод, 4 входа/4 выхода	4 x 24 В пост. тока	4 x 24 В пост. тока	Да
6ES7 223-1HF22-0XA0	EM 223, 24 В пост. тока, цифровой ввод/вывод, 4 входа/4 релейных выхода	4 x 24 В пост. тока	4 x реле	Да
6ES7 223-1BH22-0AX0	EM 223, 24 В пост. тока, цифровой ввод/вывод, 8 входов/8 выходов	8 x 24 В пост. тока	8 x 24 В пост. тока	Да
6ES7 223-1PH22-0XA0	EM 223, 24 В пост. тока цифровой ввод/вывод, 8 входов/8 релейных выходов	8 x 24 В пост. тока	8 x реле	Да
6ES7 223-1BL22-0XA0	EM 223, 24 В пост. тока цифровой ввод/вывод, 16 входов/16 выходов	16 x 24 В пост. тока	16 x 24 В пост. тока	Да
6ES7 223-1PL22-0XA0	EM 223, 24 В пост. тока цифровой ввод/вывод, 16 входов/16 релейных выходов	16 x 24 В пост. тока	16 x реле	Да

Таблица A–10. Общие технические данные цифровых 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 221-1BF22-0XA0	EM 221 DI 8 x 24 VDC	46 x 80 x 62	150 г	2 Вт	30 мА	-
6ES7 221-1EF22-0XA0	EM 221 DI 8 x AC 120/230 V	71.2 x 80 x 62	160 г	3 Вт	30 мА	-
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 Relays	46 x 80 x 62	170 г	2 Вт	40 мА	ВКЛ: 9 мА/выход, от 20,4 до 28,8 В пост. тока
6ES7 222-1EF22-0XA0	EM 222 DO 8 x AC 120/230 V	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	-
6ES7 223-1HF22-0XA0	EM 223 24 VDC 4 In/4 Relays	46 x 80 x 62	170 г	2 Вт	40 мА	ВКЛ: 9 мА/ выход, от 20,4 до 28,8 В пост. тока
6ES7 223-1BH22-0AX0	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 Relays	71.2 x 80 x 62	300 г	3 Вт	80 мА	ВКЛ: 9 мА/ выход, от 20,4 до 28,8 В пост. тока
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 Relays	137.3 x 80 x 62	400 г	6 Вт	150 мА	ВКЛ: 9 мА/ выход, от 20,4 до 28,8 В пост. тока

Таблица A–11. 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
Максимально допустимое длительное напряжение	30 В пост. тока	264 В перем. тока
Импульсное напряжение (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 реле близости (Bero) Допустимый ток утечки (макс.)	1 мА	1 мА перем. тока

Таблица A–11. Технические данные входов цифровых модулей расширения, продолжение

Общие данные	Вход 24 В пост. тока	Вход 120/230 В перем. тока (от 47 до 63 Гц)
Электрическая развязка оптическая (гальваническая, полевое устройство с логикой) Потенциально развязанные группы	500 В перем. тока в течение 1 минуты См. коммутационную схему	1500 В перем. тока в течение 1 минуты 1 вход
Количество одновременно включенных входов	Все при 55° C	Все при 55° C
Длина кабеля (макс.) экранированный не экранированный	 500 м 300 м	 500 м 300 м

Таблица A–12. Технические данные выходов цифровых модулей расширения

Общие данные	Выход 24 В пост. тока	Релейный выход	Выход 120/230 В перем. тока
Тип	Канальный полевой униполярный МОП-транзистор[1]	Сухой магнитоуправляемый 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	20,4 до 28,8 В пост. тока	от 5 до 30 В пост. тока или от 5 до 250 В перем. тока	от 40 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	-
Ток перегрузки (макс.)	8 А в течение 100 мс	7 А при замкнутых контактах	5 $А_{эфф}$ мс 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	-	-
Номинальный ток на один выход (макс.)	0,75 А	2.00 А	0,5 А перем. тока[3]
Номинальный ток на общий провод (макс.)	6 А	8 А	0,5 А перем. тока
Ток утечки (макс.)	10 мкА	-	1,1 $мА_{эфф}$ при 132 В перем. тока и 1,8 мА rhesus при 264 В перем. тока
Ламповая нагрузка (макс.)	5 Вт	30 Вт пост. тока/200 Вт перем. тока	60 Вт
Индуктивное напряжение на клеммах	L+ минус 48 В	-	-
Сопротивление в состоянии "включено" (контакт)	0,3 Ом (максимум)	0,2 Ом максимум, когда новый	410 Ом максимум, когда ток нагрузки меньше 0,05 А
Развязка гальваническая (полевое устройство с логикой) катушка с логикой катушка с контактом контакт с контактом сопротивление (катушка с контактом) потенциально развязанные группы	 500 В перем. тока в течение 1 минуты - - - - См. коммутационную схему	 - Нет 1500 В перем. тока в течение 1 минуты 750 В перем. тока в течение 1 минуты 100 МОммин., когда новый 4 входа/выхода	 1500 В перем. тока в течение 1 минуты - - - - 1 выход
Задержка выкл.-вкл./вкл.-выкл. (макс.) Процессы переключения (макс.)	 Макс. 50 мкс/200 мкс -	 - 10 мс	 0,2 мс + 1/2 цикла перем. тока -
Частота переключения (макс.)	-	1 Гц	10 Гц
Срок службы механической части, циклов переключения	-	10 000 000 (без нагрузки)	-
Срок службы контактов	-	100 000 (при номинальной нагрузке)	-
Количество одновременно включенных выходов	Все при 55 □C	Все при 55 □C	Все при 55 □C
Параллельное включение двух выходов	Да	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.

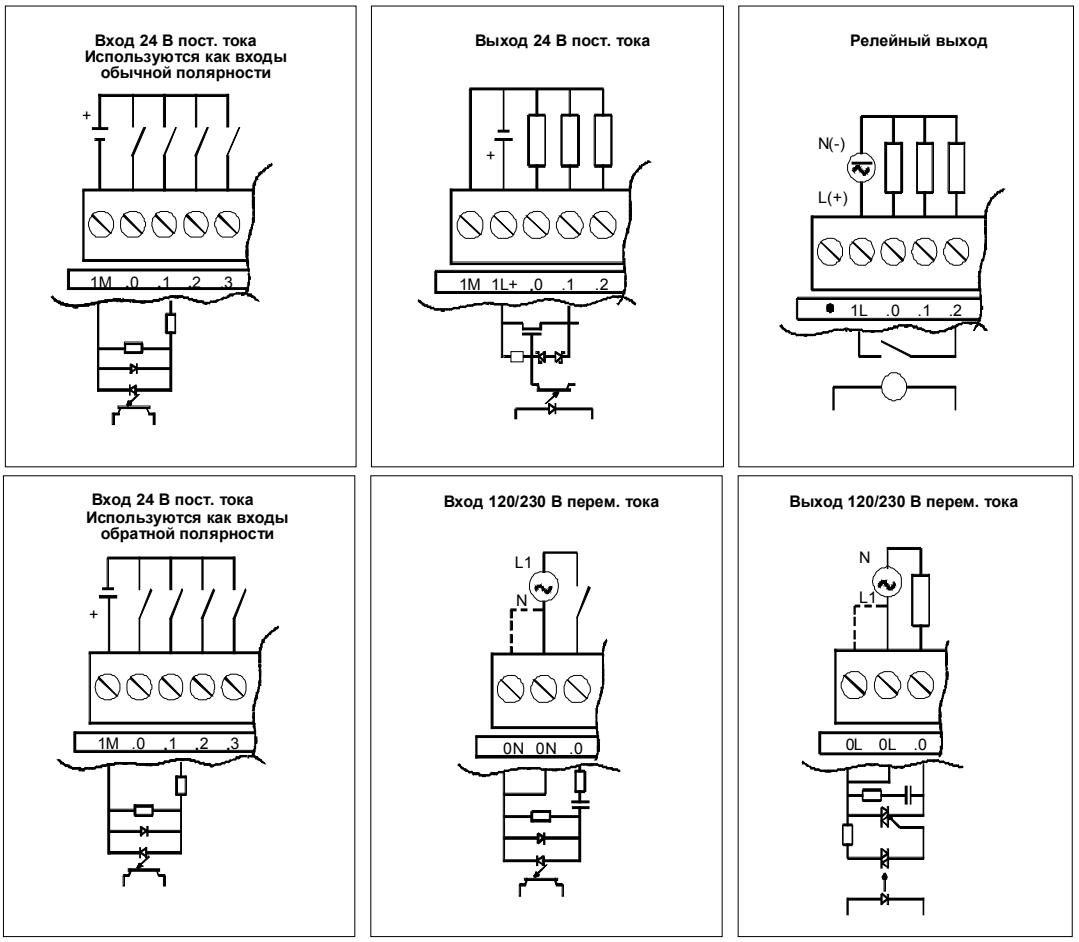


Рис. А–6. Входы и выходы цифровых модулей расширения S7–200

Коммутационные схемы

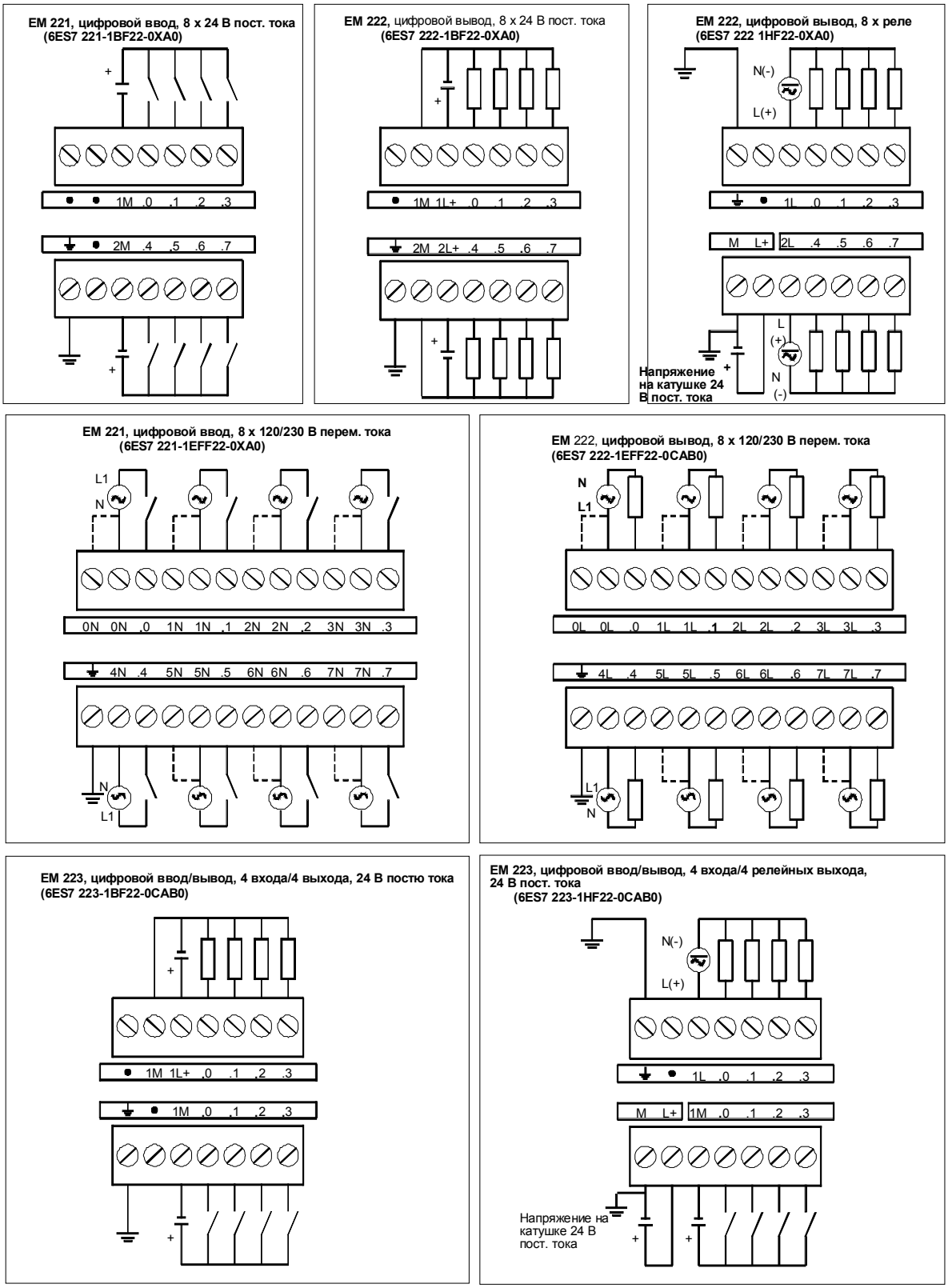


Рис. А–7. Коммутационные схемы для модулей расширения EM 221, EM 222 и EM 223

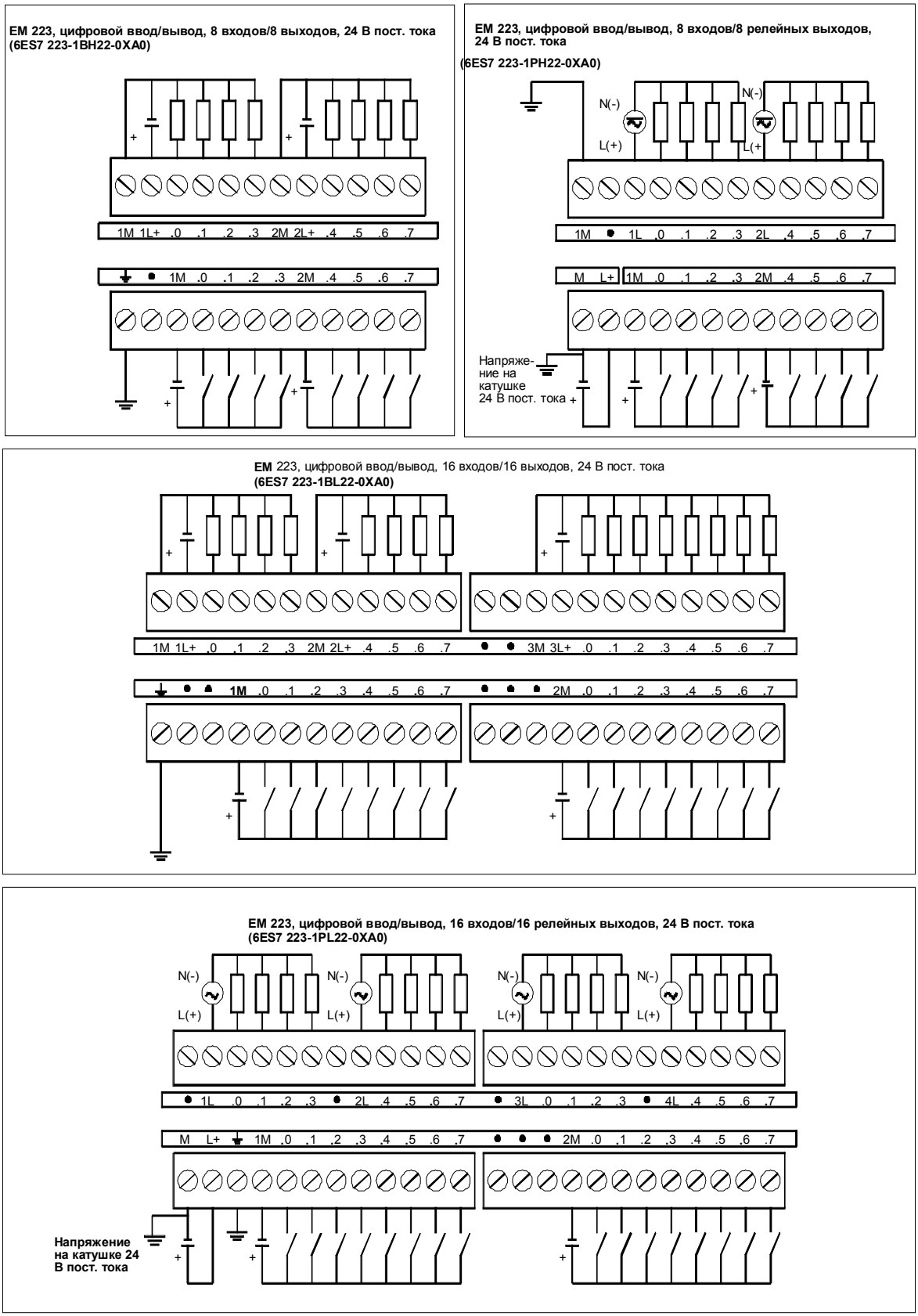


Рис. А–8. Коммутационные схемы для модулей расширения EM 223

Технические данные аналоговых модулей расширения

Таблица A–13. 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 232-0HB22-0XA0	EM 232, аналоговый вывод, 2 выхода	-	2	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 A–14. 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 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 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 A–15. Технические данные входов аналоговых модулей расширения

Общие данные	6ES7 231-0HC22-0XA0	6ES7 235-0KD22-0XA0
Формат слова данных Биполярный, полный диапазон Униполярный, полный диапазон	(См. рис. A–11) от -32000 до +32000 от 0 до 32000	(См. рис. A–11) от -32000 до +32000 от 0 до 32000
Полное сопротивление входа постоянного тока	Потенциальный вход ≥10 МОм, токовый вход 250 Ом	Потенциальный вход ≥10 МОм, токовый вход 250 Ом
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	12-битовое аналого-цифровое преобразование	12-битовое аналого-цифровое преобразование
Электрическая развязка (полевых устройств с логикой)	Нет	Нет
Тип входа	Дифференциальный	Дифференциальный
Диапазоны входных сигналов Напряжение (униполярное) Напряжение (биполярное) Ток	 от 0 до 10 В, от 0 до 5 В ±5 В, ± 2,5 В от 0 до 20 мА	 от 0 до 10 В, от 0 до 5 В от 0 до 1 В, от 0 до 500 мВ, от 0 до 100 мВ, от 0 до 50 мВ ±10 В, ±5 В, ±2.5 В, ±1 В, ±500 мВ, ±250 мВ, ±100 мВ, ±50 мВ, ±25 мВ от 0 до 20 мА
Входное разрешение Напряжение (униполярное) Напряжение (биполярное) Ток	См. таблицу A–18	См. таблицу A–19
Время аналого-цифрового преобразования	< 250 мкс	< 250 мкс
Переходная характеристика аналогового входа (реакция на единичный скачок)	1,5 мс до 95%	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	от 20,4 до 28,8

Таблица A–16. Технические данные входов аналоговых модулей расширения

Общие данные	6ES7 232-0HB22-0XA0	6ES7 235-0KD22-0XA0
Электрическая развязка (полевые устройства с логикой)	Нет	Нет
Диапазон сигнала		
Потенциальный выход	± 10 В	± 10 В
Токовый выход	от 0 до 20 мА	от 0 до 20 мА
Разрешение, полное		
Напряжение	12 битов	12 битов
Ток	11 битов	11 битов
Формат слова данных		
Напряжение	от -32000 до +32000	от -32000 до +32000
Ток	от 0 до +32000	от 0 до +32000
Точность		
Наихудший случай, от 0° до 55° C		
Потенциальный выход	± 2% полного диапазона	± 2% полного диапазона
Токовый выход	± 2% полного диапазона	± 2% полного диапазона
Типовая, 25° C		
Потенциальный выход	± 0,5% полного диапазона	± 0,5% полного диапазона
Токовый выход	± 0,5% полного диапазона	± 0,5% полного диапазона
Время установления сигнала		
Потенциальный выход	100 мкс	100 мкс
Токовый выход	2 мс	2 мс
Сопротивление		
Потенциальный выход	Минимальное 5000 Ом	Минимальное 5000 Ом
Токовый выход	Максимальное 500 Ом	Максимальное 500 Ом

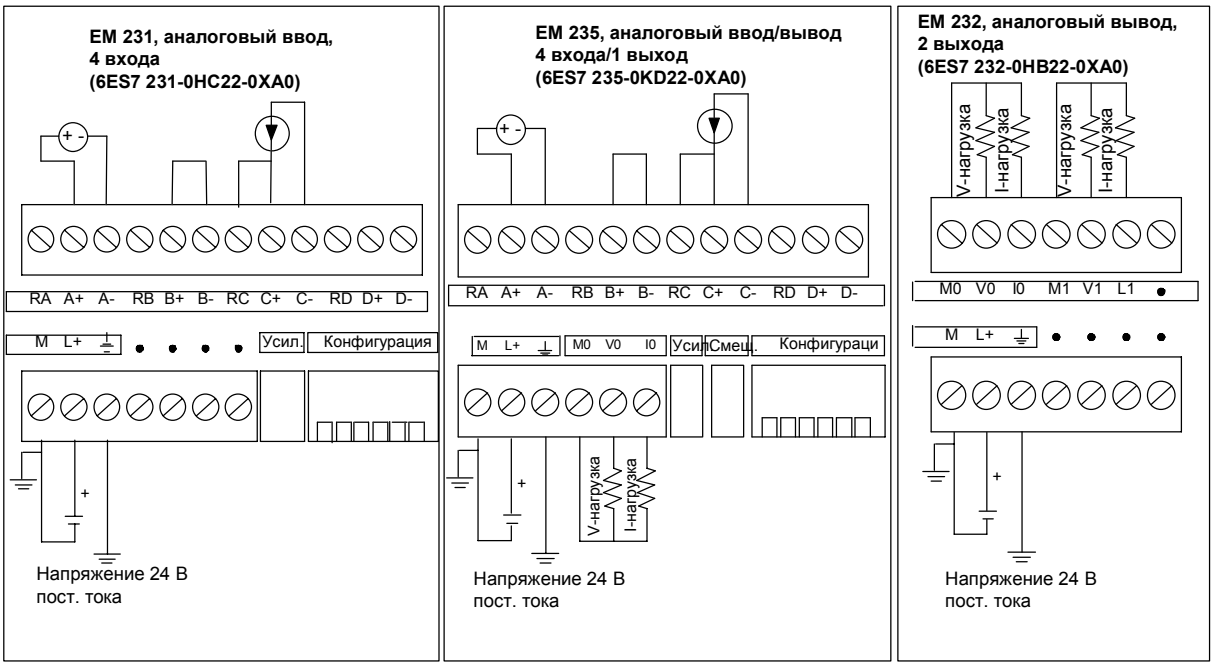


Рис. A–9. Коммутационные схемы для аналоговых модулей расширения

Светодиодные индикаторы аналоговых модулей

Светодиодные индикаторы аналоговых модулей показаны в таблице A–17.

Таблица A–17. Светодиодные индикаторы аналоговых модулей

Светодиодный индикатор	Включен	Выключен
24 В пост. тока	Нет неисправностей	Нет напряжения 24 В пост. тока

Калибровка входов

Калибровочные потенциометры воздействуют на каскад измерительного усилителя, который следует за аналоговым мультиплексором (см. схему соединений для входов EM 231 на рис. A–12 и EM 235 на рис. A–13). 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, сообщаемое 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. A–10, калибровочный потенциометр и конфигурационные двухпозиционные переключатели расположены в правой части нижней клеммного блока модуля.

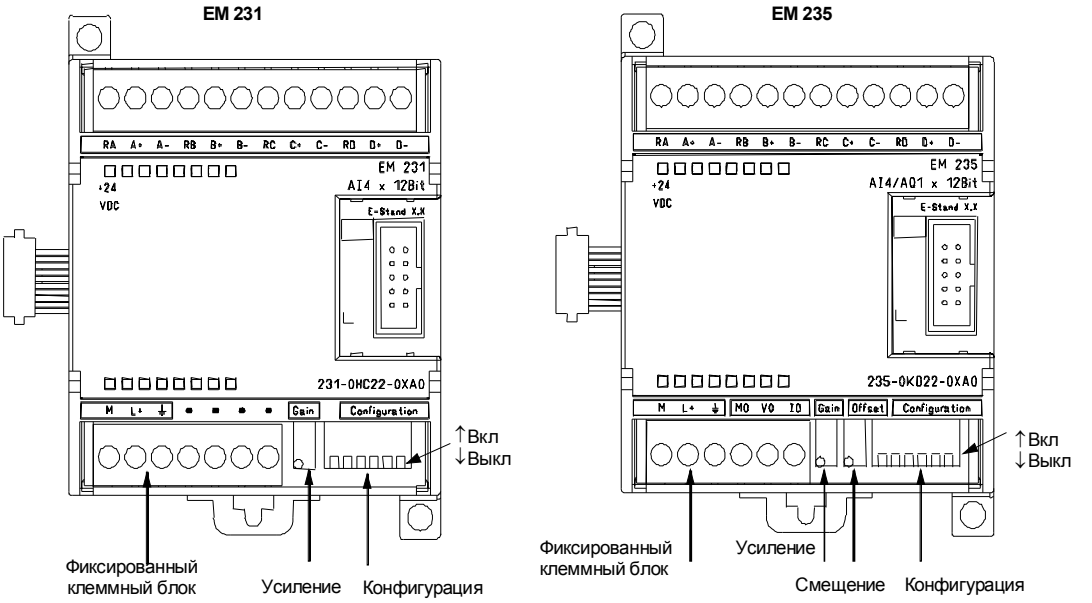


Рис. A–10. Расположение калибровочного потенциометра и конфигурационных двухпозиционных переключателей для EM 231 и EM 235

Конфигурирование EM 231

Таблица A–18 показывает, как конфигурировать модуль EM 231, используя конфигурационные двухпозиционные переключатели. Переключатели SW1, SW2 и SW3 выбирают диапазон аналогового входного сигнала. Все входы настраиваются на один и тот же диапазон аналогового входного сигнала. В этой таблице ВКЛ означает “замкнуто”, а ВЫКЛ означает “разомкнуто”.

Таблица A–18. Таблица положений конфигурационных переключателей EM 231 при выборе диапазона аналогового входного сигнала

Униполярный			Полный диапазон входного сигнала	Разрешающая способность
SW1	**SW2**	**SW3**		
ВКЛ	ВЫКЛ	ВКЛ	от 0 до 10 В	2,5 мВ
	ВКЛ	ВЫКЛ	от 0 до 5 В	1,25 мВ
			от 0 до 20 мА	5 мкА
Биполярный			**Полный диапазон входного сигнала**	**Разрешающая способность**
SW1	**SW2**	**SW3**		
ВЫКЛ	ВЫКЛ	ВКЛ	±5 В	2,5 мВ
	ВКЛ	ВЫКЛ	± 2,5 В	1,25 мВ

Конфигурирование EM 235

Таблица A–19 показывает, как конфигурировать модуль EM 235, используя конфигурационные 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 A–20 показывает, как выбрать тип сигнала “униполярный/биполярный” (переключатель SW6), усиление (переключатели SW4 и SW5) и ослабление (переключатели SW1, SW2 и SW3). В этих таблицах ВКЛ означает “замкнуто”, а ВЫКЛ означает “разомкнуто”.

Таблица A–19. Таблица положений конфигурационных переключателей EM 235 при выборе диапазона аналогового входного сигнала и разрешающей способности

Униполярный						Полный диапазон входного сигнала	Разрешающая способность
SW1	SW2	SW3	SW4	SW5	SW6		
ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВКЛ	от 0 до 50 мВ	12,5 мкВ
ВЫКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВКЛ	от 0 до 100 мВ	25 мкВ
ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	от 0 до 500 мВ	125 мкВ
ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	от 0 до 1 В	250 мкВ
ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	от 0 до 5 В	1,25 мВ
ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	от 0 до 20 мА	5 мкА
ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	от 0 до 10 В	2,5 мВ
Биполярный						**Полный диапазон входного сигнала**	**Разрешающая способность**
SW1	**SW2**	**SW3**	**SW4**	**SW5**	**SW6**		
ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	±25 мВ	12,5 мкВ
ВЫКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	±50 мВ	25 мкВ
ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	±100 мВ	50 мкВ
ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	±250 мВ	125 мкВ
ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	±500 мВ	250 мкВ
ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	±1 В	500 мкВ
ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	±2,5 В	1,25 мВ
ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	±5 В	2,5 мВ
ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	±10 В	5 мВ

Таблица A–20. Таблица положений конфигурационных переключателей EM 235 при выборе типа сигнала “униполярный/биполярный”, усиления и ослабления

Конфигурационные переключатели EM 235						Униполярный/биполярный	Усиление	Ослабление
SW1	SW2	SW3	SW4	SW5	SW6			
					ВКЛ	Униполярный		
					ВЫКЛ	Биполярный		
			ВЫКЛ	ВЫКЛ			x1	
			ВЫКЛ	ВКЛ			x10	
			ВКЛ	ВЫКЛ			x100	
			ВКЛ	ВКЛ			недопустимо	
ВКЛ	ВЫКЛ	ВЫКЛ						0,8
ВЫКЛ	ВКЛ	ВЫКЛ						0,4
ВЫКЛ	ВЫКЛ	ВКЛ						0,2

Формат слова данных для входов EM 231 EM 235

На рис. A–11 показано размещение 12-разрядного значения данных в слове аналоговых входов CPU.

Рис. A–11. Формат слова данных для входов EM 231 EM 235

Совет
12 битов данных аналого-цифрового преобразователя (АЦП) выравниваются в формате слова данных по левому краю. MSB – это знаковый бит: нуль указывает на положительное значение слова данных. В униполярном формате три концевых нуля заставляют слово данных изменяться на восемь единиц при каждом изменении значения АЦП на одну единицу. В биполярном формате четыре концевых 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

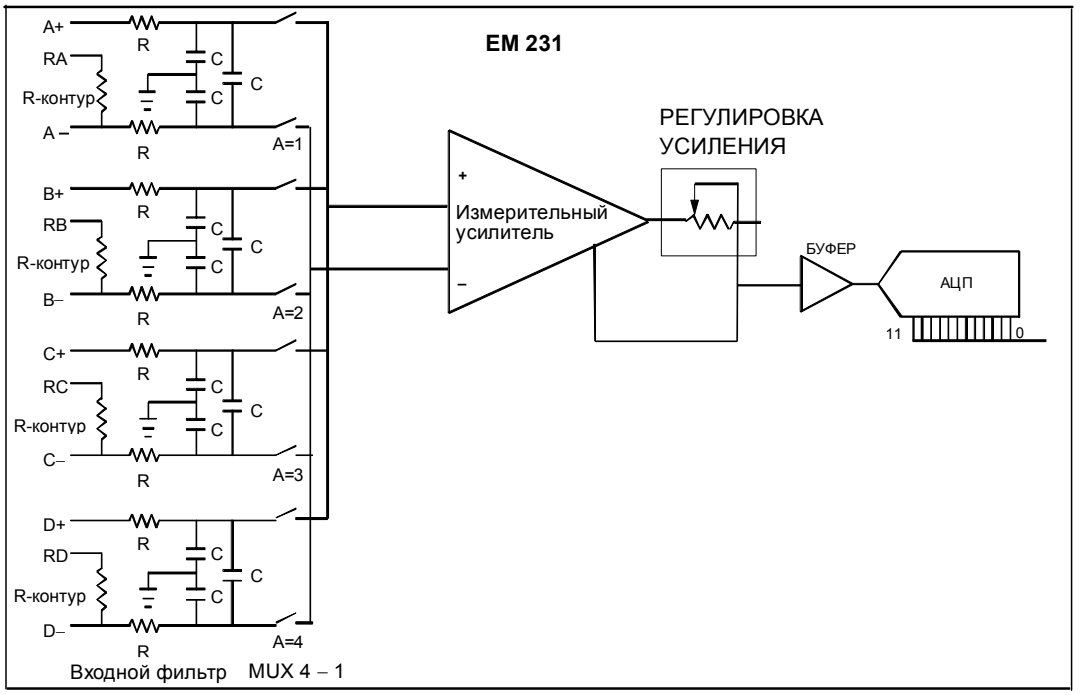


Рис. A–12. Схема соединений для входов EM 231

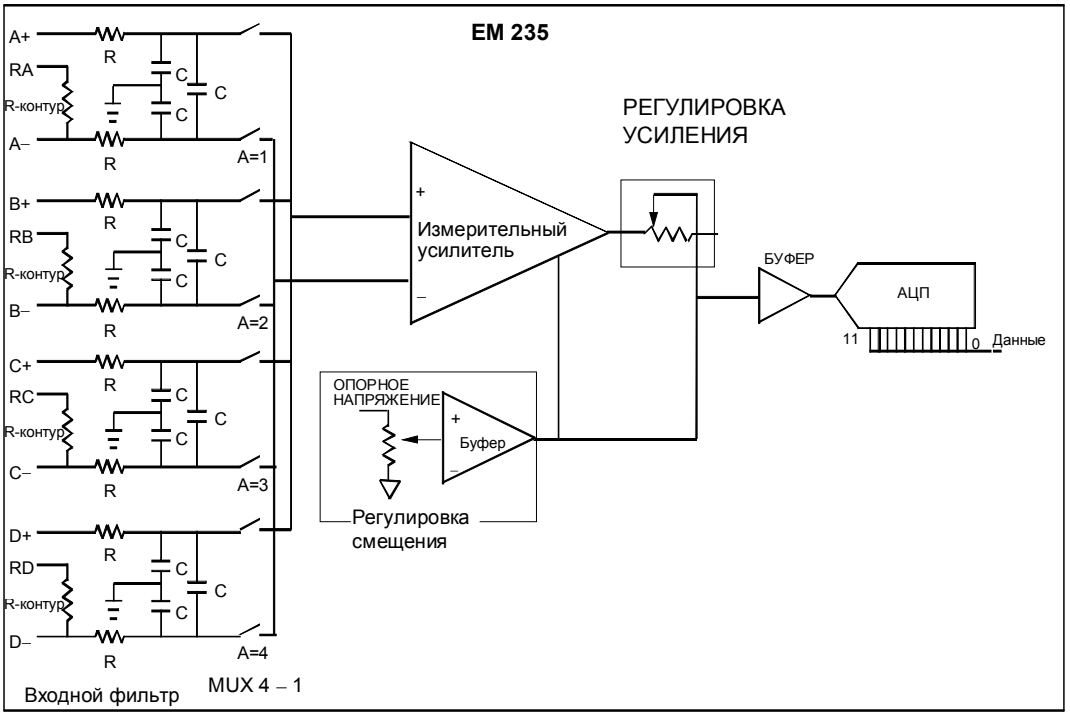


Рис. A–13. Схема соединений для входов EM 235

Формат слова данных для выходов EM 232 и EM 235

На рис. A–14 показано размещение 12-разрядного значения данных в слове аналоговых выходов CPU.

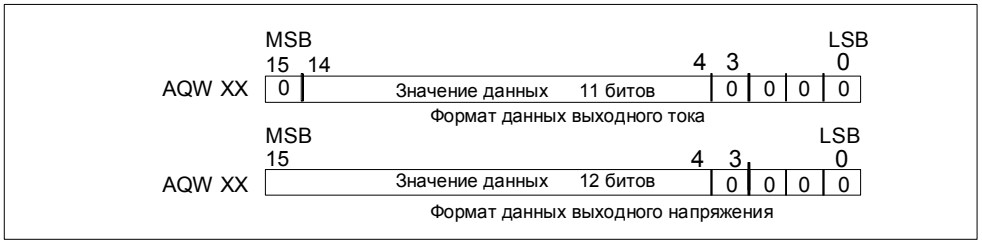


Рис. A–14. Формат слова данных для выходов EM 232 и EM 235

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

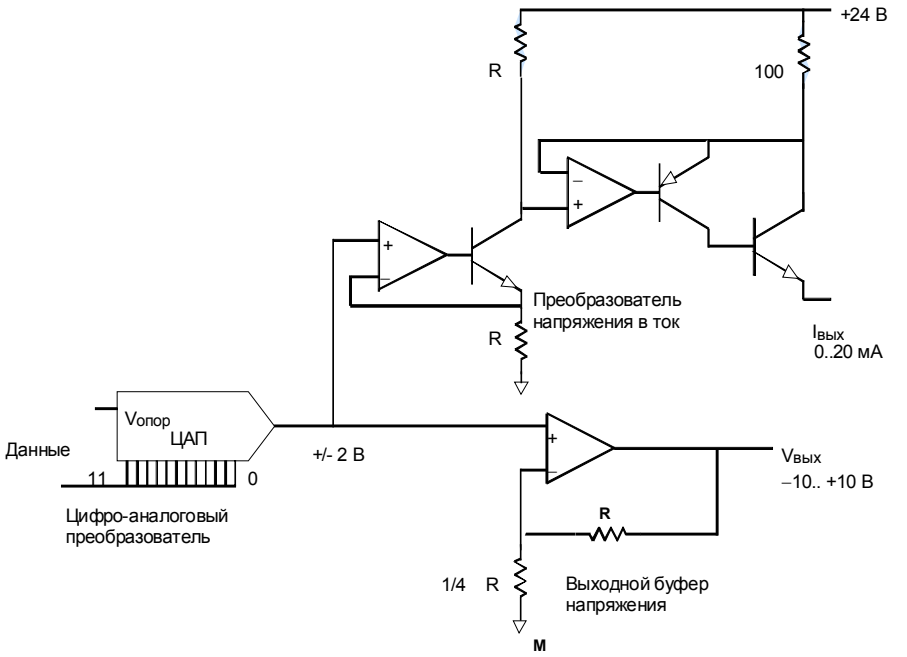


Рис. A–15. Схема соединений для выходов EM 232 и EM 235

Рекомендации по монтажу

Чтобы гарантировать точность и повторяемость, используйте следующие рекомендации:

- Обеспечьте, 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 для любых неиспользуемых каналов, как показано на рис. A–9.
- 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.

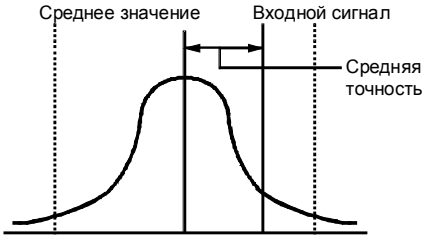
Совет
Не рекомендуется использовать модули расширения EM 231 и EM 235 с термопарами.

Описание и использование модуля аналогового 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 линеаризации 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, может наблюдаться соответствующее замедление реакции на изменения входного сигнала.



Границы повторяемости
(99 % всех отсчетов попадет в эти границы)

Рис. A–16. Определения точности

На рис. A–16 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 A–21 дает технические требования к повторяемости и среднюю точность в той мере, в какой это касается каждого из конфигурируемых 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 A–21. 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				
от 0 до 10 В				
± 2,5 В		± 48	± 0,05%	
± 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				
± 25 мВ	± 0,075%	± 48	± 0,25%	± 160
± 50 мВ			± 0,2%	± 128
± 100 мВ			± 0,1%	± 64
± 250 мВ			± 0,05%	± 32
± 500 мВ				
± 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 °C.

Технические данные модулей расширения для термопар и термометров сопротивления (RTD)

Таблица A–22. 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 231-7PB22-0XA0	EM 231, аналоговый ввод RTD, 2 входа	2 RTD	-	Нет

Таблица A–23. 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	87 мА	60 мА
6ES7 231-7PB22-0XA0	EM 231, аналоговый ввод, RTD, 2 входа	71,2 x 80 x 62	210 г	1,8 Вт	87 мА	60 мА

Таблица A–24. Технические данные модулей для термопар и термометров сопротивления (RTD)

Общие данные	6ES7 231-7PD22-0XA0 Термопара	6ES7 231-7PB22-0XA0 RTD
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 входов	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 RTD (выбрать один на модуль): Pt –100 Ом, 200 Ом, 500 Ом, 1000 Ом (при α = 3850 ‰, 3920 ‰, 3850,55 ‰, 3916 ‰, 3902 ‰) Pt -10000 Ом (α = 3850 ‰) Cu -9.035 Ом (α = 4720 ‰) Ni -10 Ом, 120 Ом , 1000 Ом (при α = 6720 ‰, 6178 ‰) R -150 Ом, 300 Ом, 600 Ом FS
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	405 мс (700 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
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 -27648 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)
Затухание входного фильтра	-3 дБ при 21 кГц	-3 дБ при 3,6 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 напряжений источника питания 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.

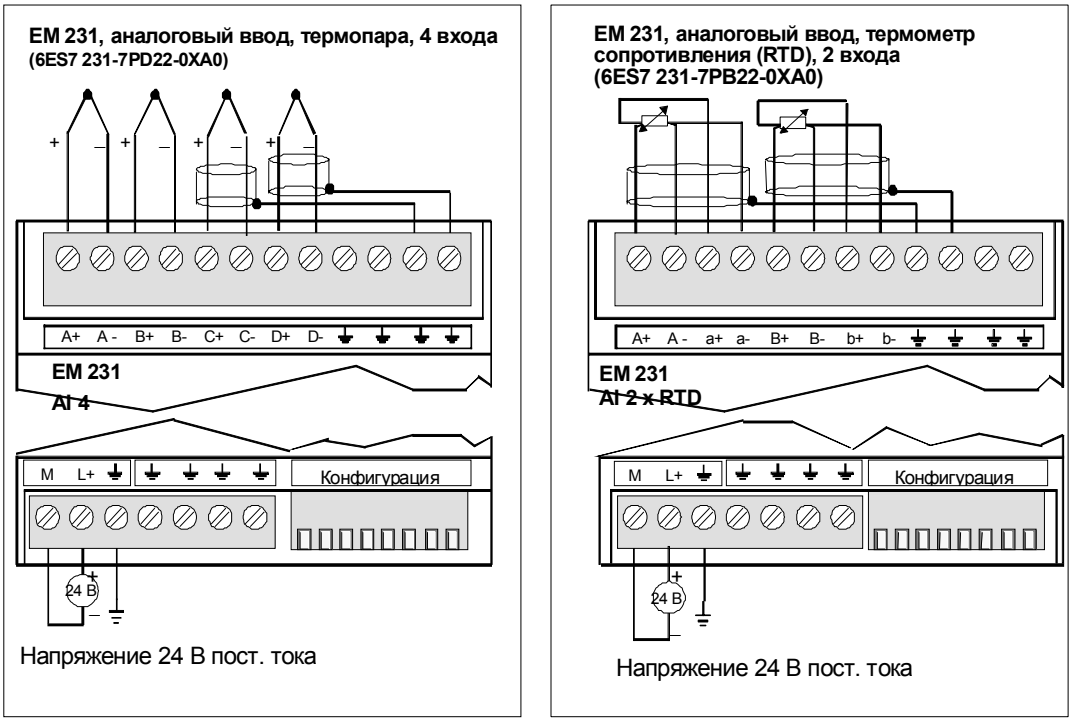


Рис. A–17. Обозначение клемм у модулей EM 231 Thermocouple и EM 231 RTD

Совместимость

Модули RTD и Thermocouple предназначены для работы с CPU 222, CPU 224, CPU 226 и CPU 226XM.

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, 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 линеаризация 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 линеаризуется с помощью таблиц для термопар.

Настройка модуля EM 231 Thermocouple

Конфигурационные двухпозиционные переключатели, расположенные в нижней части модуля, позволяют установить 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 A–25 показаны другие установки двухпозиционных переключателей.

Таблица A–25. Двухпозиционные переключатели для конфигурирования модуля EM 231 Thermocouple

Переключатели 1,2,3	Тип термопары	Установка	Описание
Переключатели 1, 2, 3 1 2 3 4* 5 6 7 8 Конфигурация ↑ 1 - Вкл ↓ 0 - Выкл *** Установите переключатель 4 в положение 0 (вниз)**	J (по умолчанию)	000	Переключатели 1 – 3 устанавливают тип термопары (или режим mV) для всех каналов модуля. Например, для термопары типа E Переключатель 1 = 0, Переключатель 2 = 1, Переключатель 3 = 1.
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
	+/-80 мВ	111	
Переключатель 5	**Проверка на обрыв провода – направление**	**Установка**	**Описание**
Переключатель 5 1 2 3 4 5 6 7 8 Конфигурация ↑ 1 - Вкл ↓ 0 - Выкл	Вверх по шкале (+3276,7 градуса)	0	0 указывает положительное направление при обрыве провода, 1 указывает отрицательное направление при обрыве провода
	Вниз по шкале (-3276,8 градуса)	1	
Переключатель 6	**Проверка на обрыв провода – активизация**	**Установка**	**Описание**
Переключатель 6 1 2 3 4 5 6 7 8 Конфигурация ↑ 1 - Вкл ↓ 0 - Выкл	Активизирована	0	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 блокирован. 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 проверка на обрыв провода.
	Деактивизирована	1	
Переключатель 7	**Шкала температур**	**Установка**	**Описание**
Переключатель 7 1 2 3 4 5 6 7 8 Конфигурация ↑ 1 - Вкл ↓ 0 - Выкл	Цельсий (_C)	0	Модуль EM 231 Thermocouple может сообщать температуру в градусах Цельсия или Фаренгейта. Преобразование градусов Цельсия в градусы Фаренгейта выполняется внутри модуля.
	Фаренгейт (_F)	1	
Переключатель 8	**Холодный спай**	**Установка**	**Описание**
Переключатель 8 1 2 3 4 5 6 7 8 Конфигурация ↑ 1 - Вкл ↓ 0 - Выкл	Компенсация холодного спая активизирована	0	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.
	Компенсация холодного спая деактивизирована	1	

Совет
- Источник тока контроля обрыва провода может создавать помехи сигналам от некоторых низковольтных источников, таких как имитаторы термопар.
- Входные напряжения, превышающие по абсолютной величине значение примерно 200 мВ, активизируют обнаружение обрыва провода даже тогда, когда источник тока для контроля обрыва провода блокирован.

Совет
- 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 снабжает ПЛК словами данных, отображающими температуру или состояния ошибки. 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 состояния ошибки и реагировать на них в соответствии с приложением. В таблице A–26 показаны индикаторы состояния EM 231 Thermocouple.

Таблица A–26. Индикаторы состояния EM 231Thermocouple

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. обратитесь к Приложению D)

[3] Диагностические ошибки вызывают ошибку конфигурирования модуля. Бит состояния для неисправности питания 24 В может быть установлен или не установлен перед ошибкой конфигурирования модуля.

Совет
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 мВ/80 мВ * 27648).

Все четыре канала обновляются каждые 405 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.

Совет
При использовании модуля EM 231 Thermocouple нужно отключить аналоговую фильтрацию в ПЛК. Аналоговая фильтрация может препятствовать своевременному обнаружению состояний ошибки.

Таблица A–27. 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
:	:								
27649	6C01							80,0029 мВ	
27648	6C00					↑		80 мВ	
:	:								
17680	4510		↑			1768,0 °C			NR
:	:								
13720	3598		1372,0 °C				↑		
:	:		перегрузка						
13000	32C8	↑	1300,0 °C				1300,0 °C		
:	:								
12000	2EE0	1200,0 °C			↑				
:	:								
10000	2710			↑	1000,0 °C				
:	:								
4000	0FA0			400,0 °C		400,0 °C			
:	:								
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,0029 мВ	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 мВ	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C отрицат. перегрузка	-0,1 °C	-0,0029 мВ	
:	:								
-500	FE0C					-50,0 °C			
-1500	FA24	-150,0 °C				↓			
:	:								
-2000	F830	отрицат. перегрузка	-200,0 °C						
:	:								
-2100	F7CC	-210,0 °C							
:	:		отрицат. перегрузка						
-2550	F60A			-255,0 °C	-255,0 °C				
:	:			отрицат. перегрузка	отрицат. перегрузка				
-2700	F574	↓	-270,0 °C	-270,0 °C	-270,0 °C		-270,0 °C		
:	:								
-27648	9400		↓	↓	↓		↓	-80 мВ	
-27649	93FF							-80,0029 мВ	
:	:								
-32512	8100							-94,071 мВ	UR
↓	↓							↓	↓
-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 A–28. 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 80,0029 мВ	OR
32140	7D90					3214,0 °F			
27649	6C01								
27648	6C00		↑			2764,8 °F		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				отрицат. перегрузка		
:	:								
-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		-454,0 °F		
:	:								
-27648	9400		↓	↓	↓		↓	-80 мВ	
-27649	93FF							-80,0029 мВ -94,071 мВ	OR
:	:								
-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.

Конфигурирование 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. A–18. Чтобы настройка двухпозиционных переключателей вступила в силу, вам нужно выключить, а затем включить ПЛК и/или источник питания потребителей 24 В.

Рис. A–18. Двухпозиционные переключатели для модуля EM 231 RTD

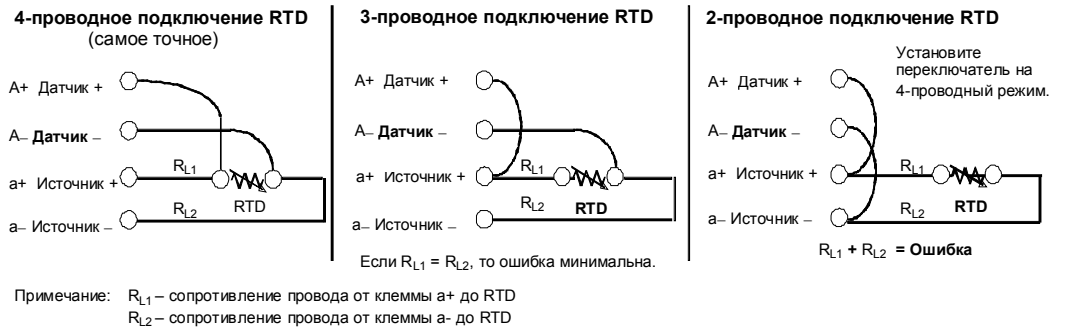
Выберите тип RTD, устанавливая двухпозиционные переключатели 1, 2, 3, 4 и 5 в соответствии с RTD, как показано в таблице A–29. Другие установки двухпозиционных переключателей показаны в таблице A–30.

Таблица A–29. Выбор типа RTD: двухпозиционные переключатели (SW) с 1 по 5

Тип RTD и альфа	SW1	SW2	SW3	SW4	SW5	Тип RTD и альфа	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	SPARE	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

Таблица A–30. Установка двухпозиционных переключателей RTD

Переключатель 6	Проверка на обрыв проводов	Установка	Описание
Переключатель 6 1 2 3 4 5 6 7 8 Конфигурация ↑1 - Вкл ↓0 - Выкл	Вверх по шкале (+3276,7 градусов)	0	Указывает положительное направление при обрыве провода
	Вниз по шкале (-3276,8 градусов)	1	Указывает отрицательное направление при обрыве провода
Переключатель 7	**Шкала температур**	**Установка**	**Описание**
Переключатель 7 1 2 3 4 5 6 7 8 Конфигурация ↑1 - Вкл ↓0 - Выкл	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**
Переключатель 8 1 2 3 4 5 6 7 8 Конфигурация ↑1 - Вкл ↓0 - Выкл	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	



Примечание: R_{L1} – сопротивление провода от клеммы a+ до RTD
R_{L2} – сопротивление провода от клеммы a- до RTD

Рис. A–19. 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–31 показывает индикаторы состояния, предоставляемые модулем EM 231 RTD.

Совет
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 Ом/300 Ом * 27648 = 20736)

Таблица А–31. 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 конфигурирования модуля. Бит состояния для неисправности питания 24 В может быть установлен или не установлен перед ошибкой конфигурирования модуля.

Данные канала обновляются каждые 405 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.

Совет
При использовании модуля RTD нужно отключить аналоговую фильтрацию в ПЛК. Аналоговая фильтрация может препятствовать своевременному обнаружению состояний ошибки.

Обнаружение обрыва провода выполняется программным обеспечением внутри модуля RTD. Объявляются входы, находящиеся вне диапазона, и данные об обрыве провода сообщаются как данные о перегорании. Обнаружение обрыва провода занимает, как минимум, три цикла сканирования модуля и может длиться больше в зависимости от того, какой провод(а) имеет обрыв. Обрыв провода “Источник+” и/или провода “Источник–“ обнаруживается в минимальный срок. 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/выхода сигнала за пределы диапазона.

Диапазоны модуля EM 231 RTD

Температурные диапазоны и точность для каждого типа модулей EM 231 RTD представлены в таблицах A–32 и A–33.

Таблица A–32. Диапазоны температур (°C) и точность для различных типов RTD

Системное слово (1 цифра = 0,1 °C)		Pt10000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu9,035	0 - 150 Ом	0 – 300 Ом	0 - 600 Ом	
Десятич.	16-ричн.								
32767	7FF.								
32766	7FFE					↑	↑	↑	
32511	7EFF					176,383 Ом	352,767 Ом	705,534 Ом	
29649	6C01					150,005 Ом	300,011 Ом	600,022 Ом	
27648	6C00					150,000 Ом	300,000 Ом	600,000 Ом	
25000	61A8								↑
18000	4650								OR
15000	3A98								
13000	32C8	↑	↑						
10000	2710	1000,0 °C	1000,0 °C						
8500	2134		850,0 °C						
6000	1770	600,0 °C			↑				
3120	0C30			↑	312,0 °C				NR
2950	0B86			295,0 °C					
2600	0A28				260,0 °C				
2500	09C4			250,0 °C					
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,005 Ом	0,011 Ом	0,022 Ом	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,000 Ом	0,000 Ом	0,000 Ом	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	(отрицательные значения невозможны)			
						↓	↓	↓	
-600	FDA8			-60,0 °C					
-1050	FBE6			-105,0 °C					
				↓					
-2000	F830	-200,0 °C	-200,0 °C		-200,0 °C				
-2400	F6A0				-240,0 °C				
-2430	F682	-243,0 °C	-243,0 °C		↓				
		↓	↓						
-5000	EC78								
-6000	E890								UR
-10500	D6FC								↓
-12000	D120								
-20000	4E20								
-32767	8001								
-32768	8000								
Точность по всему диапазону		±0,4%	±0,1%	±0,2%	±0,5%	±0,1%	±0,1%	±0,1%	
Точность (номинальный диапазон)		±4 °C	±1 °C	±0,6 °C	±2,8 °C	±0,15 Ом	±0,3 Ом	±0,6 Ом	

* OR = Overrange (перегрузка); NR = Nominal range (номинальный диапазон); UR = Underrange (отрицательная перегрузка)

↑ и ↓ показывают, что все аналоговые значения, выходящие за пределы этого диапазона, выдают установленное значение перегорания, 32767 (0x7FF.) или -32768 (0x8000).

Таблица A–33. Диапазоны температур (°F) для различных типов RTD

Системное слово (1 цифра = 0,1 °F)		PT1000	PT100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu 9.035	
Десятич.	16-ричн.					
32767	7FF.					
32766	7PHAGE					
						↑
						Перегрузка
		↑	↑			
18320	4790	1832,0 °F	1832,0 °F			
15620	3D04		1562,0 °F			
11120	2B70	1112,0 °F				
					↑	
5936	1730			↑	593,6 °F	
5630	15FE			563,0 °F		
5000	1388				500,0 °F	
4820	12D4			482,0 °F		
						Номинальный диапазон
1	0001	0,1 °F	0,1 °F	0,1 °F	0,1 °F	
0	0000	0,0 °F	0,0 °F	0,0 °F	0,0 °F	
-1	FFFF	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	
-760	FD08			-76,0 °F		
-1570	F9DE			-157,0 °F		
				↓		
-3280	F330	-328,0 °F	-328,0 °F		-328,0 °F	
-4000	F060				-400,0 °F	
-4054	F02A	-405,4 °F	-405,4 °F		↓	
		↓	↓			
-5000	EC78					
-6000	E890					Отрицательная перегрузка
-10500	D6FC					↓
-32767	8001					
-32768	8000					

↑ и ↓ показывают, что все аналоговые значения, выходящие за пределы этого диапазона, выдают установленное значение перегорания, 32767 (0x7FF) или -32768 (0x8000).

Технические данные модуля EM 277 PROFIBUS-DP

Таблица A–34. 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 A–35. 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 A–36. Технические данные модуля EM 277 PROFIBUS-DP

Общие данные	6ES7 277-0AA22-0XA0
Количество портов	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 или от блока питания датчиков 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
Пульсирующая помеха (<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 A–37.

Таблица A–37. Совместимость модуля EM 277 PROFIBUS-DP с CPU S7–200

CPU	Описание
CPU 222 версии 1.10 или выше	CPU 222 DC/DC/DC
	CPU 222 AC/DC/реле
CPU 224 версии 1.10 или выше	CPU 224 DC/DC/DC
	CPU 224 AC/DC/реле
CPU 226 версии 1.00 или выше	CPU 226 DC/DC/DC
	CPU 226 AC/DC/реле
CPU 226XM версии 1.00 или выше	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/реле

Переключатели адресов и светодиоды

Переключатели адресов и светодиоды состояния расположены на передней панели модуля, как показано на рис. A–20. Показано также распределение контактов порта для slave-устройства DP.

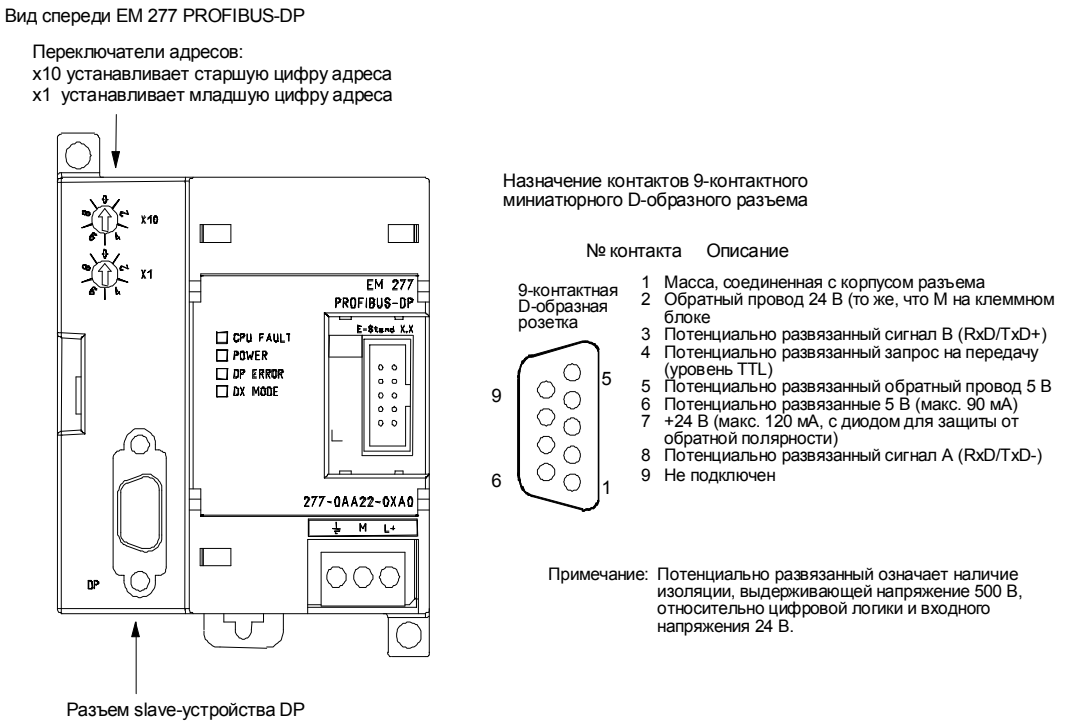


Рис. A–20. EM 277 PROFIBUS-DP

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) описывает доступ к шине и протокол передачи данных и определяет характеристики среды передачи данных.
- 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 с 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 или 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. A–21 показана сеть PROFIBUS с CPU 224 и модулем EM 277 PROFIBUS-DP.

- CPU 315-2 является master-устройством DP и был конфигурирован при помощи устройства программирования SIMATIC с программным обеспечением для программирования STEP 7.
- CPU 224 является slave-устройством DP, принадлежащим CPU 315-2. Модуль ввода-вывода ET 200 также является slave-устройством DP, принадлежащим CPU 315-2.
- CPU S7-400 присоединен к сети PROFIBUS и считывает данные из CPU 224 посредством команд XGET в программе пользователя в S7-400.

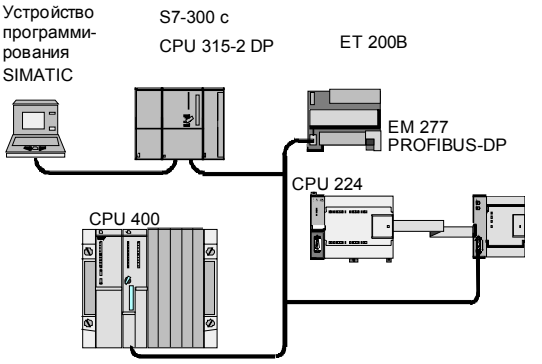


Рис. A–21. Модуль EM 277 PROFIBUS-DP и CPU 224 в сети PROFIBUS

Конфигурирование

Чтобы использовать EM 277 как DP-slave, вы должны установить для порта DP адрес станции, соответствующий адресу в конфигурации master-устройства. Адрес станции устанавливается в модуле EM 277 при помощи поворотных переключателей. Чтобы новый адрес slave-устройства вступил в силу, вы должны после изменения положения переключателей выключить, а затем включить CPU.

Master-устройство выполняет обмен данными с каждым из своих slave-устройств, передавая информацию из своей области вывода в буфер вывода slave-устройства (называемый “почтовым ящиком приема”). Slave-устройство реагирует на сообщение master-устройства тем, что возвращает буфер ввода (называемый “почтовым ящиком передачи”), который master-устройство сохраняет в области ввода.

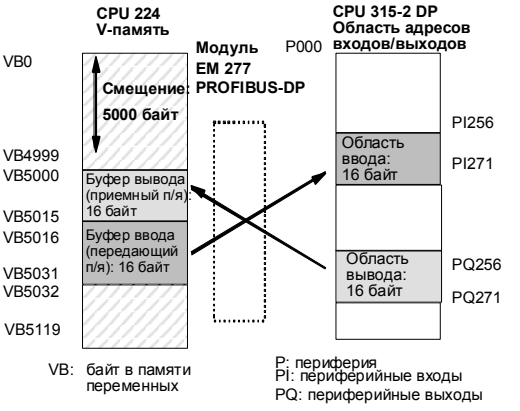


Рис. A–22. Память переменных и область адресов входов и выходов

На рис. A–22 показан пример памяти переменных и области адресов входов и выходов master-устройства PROFIBUS-DP.

DP-master может конфигурировать EM 277 так, чтобы он принимал из master-устойства выходные данные и возвращал master-устойству 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. A–22 показана модель V-памяти CPU 224 и области адресов входов-выходов CPU master-устрйства 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 A–38 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 A–38. 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 COM PROFIBUS Windows
- 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 справочникам по этим устройствам. Для получения подробной информации о сети PROFIBUS и ее компонентов обратитесь к *Системному руководству по децентрализованной периферии ET 200*.

Целостность данных

PROFIBUS поддерживает целостность данных трех типов:

- Целостность байтов гарантирует, что байты передаются как единые блоки.
- Целостность слов гарантирует, что передача слова не может прерываться другими процессами в CPU (два байта, составляющие слово, всегда пересылаются вместе и не могут быть разделены). Целостность слов следует использовать, когда передаваемые значения данных являются целыми числами.
- Целостность буферов гарантирует, что весь буфер данных передается как единый блок, не прерываемый каким-либо другим процессом в CPU. 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.

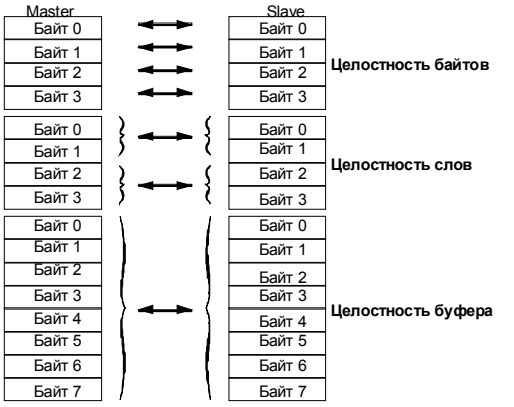


Рис. A–23. Целостность байтов, слов и буферов данных

Целостность данных устанавливается как часть конфигурации входов-выходов в master-устройстве. Выбор целостности данных записывается в slave-устройство как часть инициализации slave-устройства. И DP-master, и DP-slave используют выбор целостности 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. A–23.

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 в распоряжении имеются 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–39.

Указание
Начиная с версии 2.2, изменился способ назначения адресов специальной памяти для интеллектуальных модулей.

Если вы используете более раннюю версию CPU, чем 2.2, то для обеспечения совместимости вам следует размещать все интеллектуальные модули в слотах, соседних с CPU, перед всеми неинтеллектуальными модулями.

Таблица A–39. 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–40, 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.

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.

Таблица A–40. 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 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 инициирован 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 потребитель напряжения 24 В постоянного тока не снабжается питанием, то светодиод POWER [питание] будет выключен.

В таблице A–41 приведены сигналы состояния, отображаемые светодиодами состояния EM 277.

Таблица A–41. 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 у пользователя	--	--	Питание 24 В пост. тока у пользователя 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. A–24.

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 Freeport.

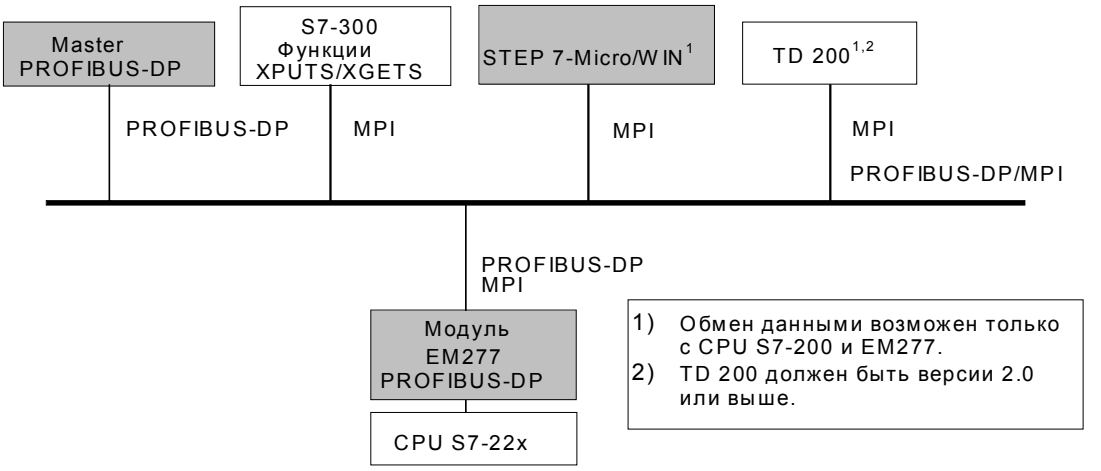


Рис. A–24. Сеть PROFIBUS-DP/MPI

Файл базы данных устройства: GSD

Различные устройства PROFIBUS имеют разные рабочие характеристики. Эти характеристики различаются в части функциональных возможностей (например, количество входных/выходных сигналов и диагностических сообщений) или параметров шины, таких как скорость передачи и контроль времени. Эти параметры изменяются в зависимости от типа устройства и поставщика и обычно документированы в техническом руководстве. Чтобы помочь вам построить простую конфигурацию PROFIBUS, рабочие характеристики конкретного устройства определяются в электронной спецификации, называемой файлом базы данных устройства или файлом GSD. Инструменты конфигурирования, основанные на файлах GSD, допускают простое объединение устройств от различных поставщиков в единую сеть.

Файл базы данных устройства дает всестороннее описание характеристик устройства в точно определенном формате. Эти файлы GSD подготавливаются поставщиком для каждого типа устройств и предоставляются в распоряжение пользователя PROFIBUS. Файл GSD позволяет системе конфигурирования считывать характеристики устройства PROFIBUS и использовать эту информацию при конфигурировании сети.

Самые последние версии программного обеспечения COM PROFIBUS или STEP 7 включают файлы конфигурации для модуля EM 277 PROFIBUS-DP. Если ваша версия программного обеспечения не включает файл конфигурации для EM 277, то вы можете обратиться за самой последней версией файла GSD (SIEM089D.GSD) на web-сайт www.profibus.com.

Если вы используете master-устройство не фирмы Siemens, то обратитесь к поставляемой изготовителем документации о том, как конфигурировать master-устройство, используя GSD-файл.

```
;==================================================
; GSD File для EM 277 PROFIBUS-DP с DPC31
; MLFB  : 6ES7 277-0AA2.-0XA0
; DATE  : 26-March-2001
;==================================================
#Profibus_DP
;General parameters [Общие параметры]
GSD_Revision        = 1 [Версия GSD]
Vendor_Name         = »Siemens» [Имя поставщика]
Model_Name          = »EM 277 PROFIBUS-DP» [Имя модели]
Revision            = »V1.02» [Версия]
Ident_Number        = 0x089D
Protocol_Ident      = 0
Station_Type        = 0
FMS_supp            = 0
Hardware_Release    = »1.00» [Версия аппаратуры]
Software_Release    = »1.02» [Версия программ.
обеспечения]
9.6_supp            = 1
19.2_supp           = 1
45.45_supp          = 1
93.75_supp          = 1
187.5_supp          = 1
500_supp            = 1
1.5M_supp           = 1
3M_supp             = 1
6M_supp             = 1
12M_supp            = 1
MaxTsdr_9.6         = 60
MaxTsdr_19.2        = 60
MaxTsdr_45.45       = 250
MaxTsdr_93.75       = 60
MaxTsdr_187.5       = 60
MaxTsdr_500         = 100
MaxTsdr_1.5M        = 150
MaxTsdr_3M          = 250
MaxTsdr_6M          = 450
MaxTsdr_12M         = 800
Redundancy          = 0
Repeater_Ctrl_Sig   = 2
24V_Pins            = 2
; Slave-Specification [Данные slave-устройства]:
OrderNumber=»6ES7 277-0AA2.-0XA0» [Номер для заказа]
Periphery=»SIMATIC S5» [Периферия]
Slave_Family=10@TdF@SIMATIC [Семейство slave-
устройств]
Freeze_Mode_supp    = 1
Sync_Mode_supp      = 1
Set_Slave_Add_Supp  = 0
Auto_Baud_supp      = 1
Min_Slave_Intervall = 1
Fail_Safe           = 0
Max_Diag_Data_Len   = 6
Modul_Offset        = 0
Modular_Station     = 1
Max_Module          = 1
Max_Input_len       = 128
Max_Output_len      = 128
Max_Data_len        = 256
; UserPrmData-Definition
ExtUserPrmData=1 »I/O Offset in the V-memory»
Unsigned16 0 0-10239
EndExtUserPrmData
; UserPrmData: Length and Preset:
User_Prm_Data_Len=3
User_Prm_Data= 0,0,0
Max_User_Prm_Data_Len=3
Ext_User_Prm_Data_Const(0)=0x00,0x00,0x00
Ext_User_Prm_Data_Ref(1)=1
```

```
;==================================================
; Продолжение GSD-файла
;==================================================
; Module Definition List [Список определений модуля]
Module = »2 Bytes Out/ 2 Bytes In     -» 0x31
EndModule
Module = »8 Bytes Out/ 8 Bytes In     -» 0x37
EndModule
Module = »32 Bytes Out/ 32 Bytes In   -» 0xC0,0x1F,0x1F
EndModule
Module = »64 Bytes Out/ 64 Bytes In   -» 0xC0,0x3F,0x3F
EndModule
Module = »1 Word Out/ 1 Word In       -» 0x70
EndModule
Module = »2 Word Out/ 2 Word In       -» 0x71
EndModule
Module = »4 Word Out/ 4 Word In       -» 0x73
EndModule
Module = »8 Word Out/ 8 Word In       -» 0x77
EndModule
Module = »16 Word Out/ 16 Word In     -» 0x7F
EndModule
Module = »32 Word Out/ 32 Word In     -» 0xC0,0x5F,0x5F
EndModule
Module = »2 Word Out/ 8 Word In       -» 0xC0,0x41,0x47
EndModule
Module = »4 Word Out/ 16 Word In      -» 0xC0,0x43,0x4F
EndModule
Module = »8 Word Out/ 32 Word In      -» 0xC0,0x47,0x5F
EndModule
Module = »8 Word Out/ 2 Word In       -» 0xC0,0x47,0x41
EndModule
Module = »16 Word Out/ 4 Word In      -» 0xC0,0x4F,0x43
EndModule
Module = »32 Word Out/ 8 Word In      -» 0xC0,0x5F,0x47
EndModule
Module = »4 Byte buffer I/O           -» 0xB3
EndModule
Module = »8 Byte buffer I/O           -» 0xB7
EndModule
Module = »12 Byte buffer I/O          -» 0xBB
EndModule
Module = »16 Byte buffer I/O          -» 0xBF
EndModule
```

Рис. А–25. Распечатка GSD-файла для модуля EM 277 PROFIBUS

Пример программы для обмена данными с CPU на основе DP

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 из 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.

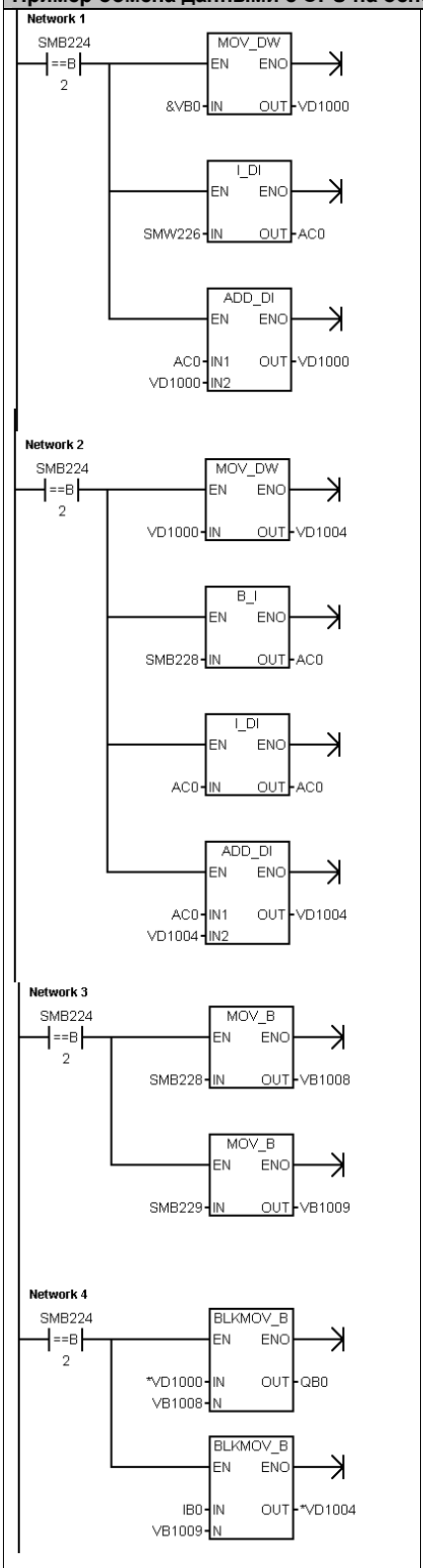
Указание
Начиная с версии 2.2, изменился способ назначения адресов специальной памяти для интеллектуальных модулей.

Если вы используете более раннюю версию CPU, чем 2.2, то для обеспечения совместимости вам следует размещать все интеллектуальные модули в слотах, соседних с CPU, перед всеми неинтеллектуальными модулями.

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 с CPU на основе DP




```
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 A–42. 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 A–43. Общие технические данные модемного модуля EM 241

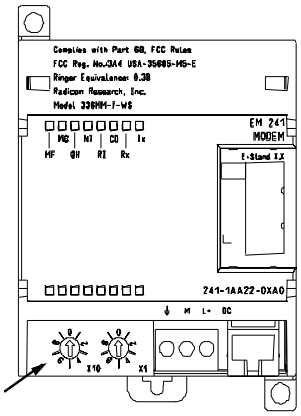
Номер для заказа	Наименование и описание модуля	Размеры (мм) (Ш x В x Г)	Вес	Мощность потерь	Потребность в напряжении пост. тока	
					+5 В	+24 В
6ES7 241-1AA22-0XA0	Модемный модуль EM 241	71,2 x 80 x 62	190 г	2,1 Вт	80 мА	70 мА

Таблица A–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, B.21, B.22, B.22 bis, B.23c, B.32, B.32 bis, B.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 Электрическая развязка (полевое напряжение с логикой)	от 20,4 до 28,8 В пост. тока 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 конфигурировании 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".



Переключатель кода страны

Рис. A–26. 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 A–45.

Таблица A–45. Совместимость модемного модуля EM 241 с CPU S7–200

CPU	Описание
CPU 222 версии 1.10 или выше	CPU 222 DC/DC/DC и CPU 222 AC/DC/реле
CPU 224 версии 1.10 или выше	CPU 224 DC/DC/DC и CPU 224 AC/DC/реле
CPU 226 версии 1.00 или выше	CPU 226 DC/DC/DC и CPU 226 AC/DC/реле
CPU 226XM версии 1.00 или выше	CPU 226XM DC/DC/DC и CPU 226XM AC/DC/реле

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 A–46. 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]).

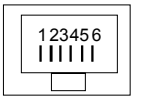
Ваш EM 241 теперь готов к обмену данными.

Таблица A–46. 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. A–27 показан штекер RJ11. Вы можете использовать адаптеры для других стандартных телефонных присоединений. Подробную информацию вы найдете в документации к своему адаптерному соединению.



Контакт	Описание	
3	Звонок	Допустимо обратное присоединение
4	Контакт	

Рис. A–27. 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 A–47. 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 функциями перемещения, а не для непосредственного управления внешними сигналами.

Таблица A–48. 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 A–49. 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 (Bero) 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 °C	 5

Таблица A–49. Технические данные модуля позиционирования EM 253, продолжение

Общие данные	6ES7 253-1AA22-0XA0	
Выходные характеристики		
Количество встроенных выходов	6 выходов (4 сигнала)	
Тип выхода		
P0+, P0-, P1+, P1-	Драйвер RS422/485	
P0, P1, DIS, CLR	транзисторные выходы с открытым стоком	
Выходное напряжение		
Драйверы P0, P1, RS-422, дифференциальное выходное напряжение		
открытая цепь тока	Тип. 3,5 В	
В диоде оптрона с последовательным сопротивлением 200 Ом	мин. 2,8 В	
Нагрузка 100 Ом	мин. 1,5 В	
Нагрузка 54 Ома	мин. 1,0 В	
P0, P1, DIS, CLR, с открытым стоком		
рекомендуемое напряжение, открытая цепь тока	5 В пост. тока, из модуля	
допустимое напряжение, открытая цепь тока	30 В пост. тока [1]	
Выход тока	50 мА максимум	
Сопротивление во включенном состоянии	макс. 15 Ом	
Ток утечки в выключенном состоянии, 30 В пост. тока	макс. 10 мкА	
Внутренний корректирующий резистор, выходная утечка в T1	3,3 кОм[2]	
Выходной ток		
Количество групп выходов	1	
Макс. количество включенных выходов	6	
Ток утечки на выход		
P0, P1, DIS, CLR	макс. 10 мкА	
Защита от перегрузки	Нет	
Электрическая развязка (полевые устройства с логикой)		
Электрическая развязка (гальваническая)	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	макс. 75 нс	
Выходы P0, P1, транзисторные с открытым стоком, 5 В / внешняя нагрузка 470 Ом	макс. 300 нс	
Частота переключения		
P0+, P0-, P1+, P1-, P0 и P1	200 кГц	
Длина кабеля		
неэкранированный	не рекомендуется	
экранированный	10 метров	
Блок питания		
Питающее напряжение L+	от 11 до 30 В пост. тока	
Питание логики	макс. +5 В пост. тока +/- 10%, 200 мА	
Питающий ток L+ при нагрузке 5 В пост. тока		
Ток нагрузки	Вход 12 В пост. тока	Вход 24 В пост. тока
0 мА (без нагрузки)	120 мА	70 мА
200 мА (при номинальной нагрузке)	300 мА	130 мА
Электрическая развязка		
напряжение на L+ с логикой	500 В перем. тока в течение 1 минуты	
напряжение на L+ с входами	500 В перем. тока в течение 1 минуты	
напряжение на L+ с выходами	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 A–50.

Таблица A–50. Совместимость модуля позиционирования EM 253 с CPU S7–200

CPU	Описание
CPU 222 версии 1.10 или выше	CPU 222 DC/DC/DC и CPU 222 AC/DC/реле
CPU 224 версии 1.10 или выше	CPU 224 DC/DC/DC и CPU 224 AC/DC/реле
CPU 226 версии 1.00 или выше	CPU 226 DC/DC/DC и CPU 226 AC/DC/реле
CPU 226XM версии 1.00 или выше	CPU 226XM DC/DC/DC и CPU 226XM AC/DC/реле

Светодиоды состояния модуля позиционирования EM 253

Светодиоды состояния модуля позиционирования показаны в таблице A–51.

Таблица A–51. 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

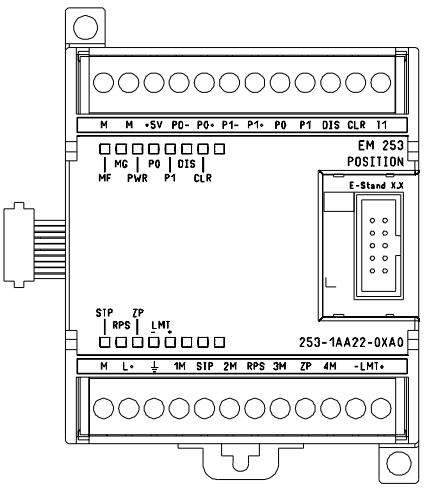


Рис. A–28. Модуль позиционирования EM 253

Коммутационные схемы

На следующих рисунках клеммы расположены не по порядку. Расположение клемм см. на рис. A–28.

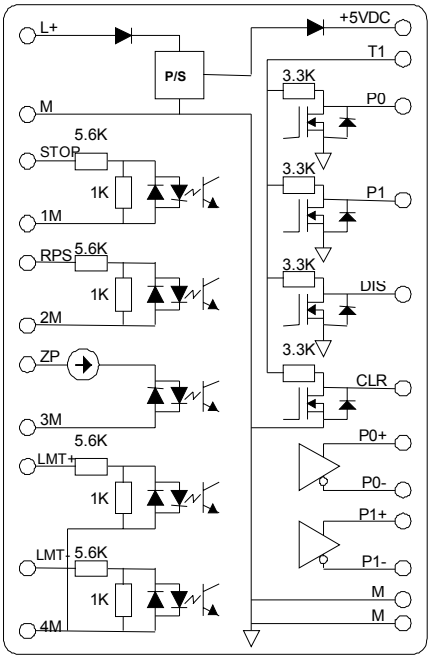
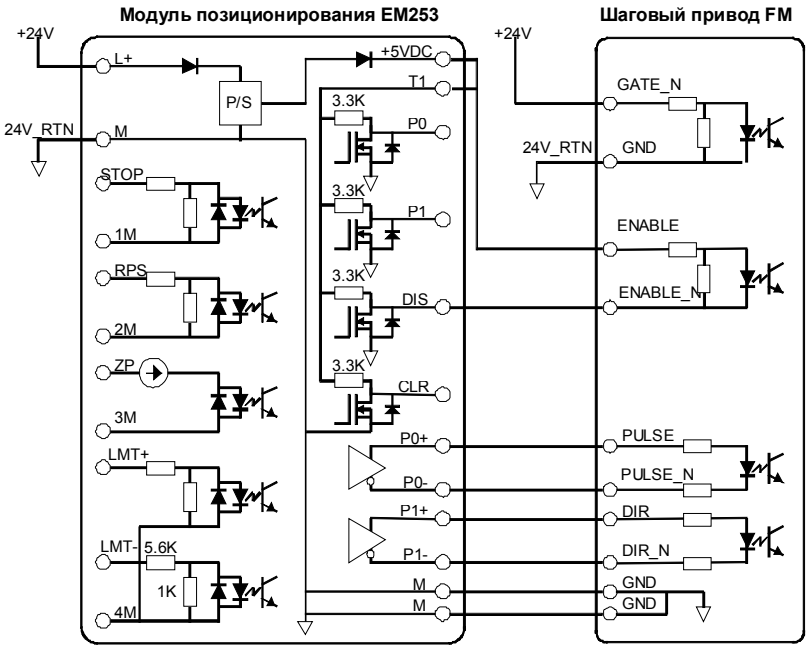


Рис. A–29. Внутренняя схема входов и выходов модуля позиционирования EM 253



Клеммы расположены не по порядку. Расположение клемм см. на рис. A-28.

Рис. A–30. Подключение модуля позиционирования EM 253 к шаговому приводу SIMATIC FM

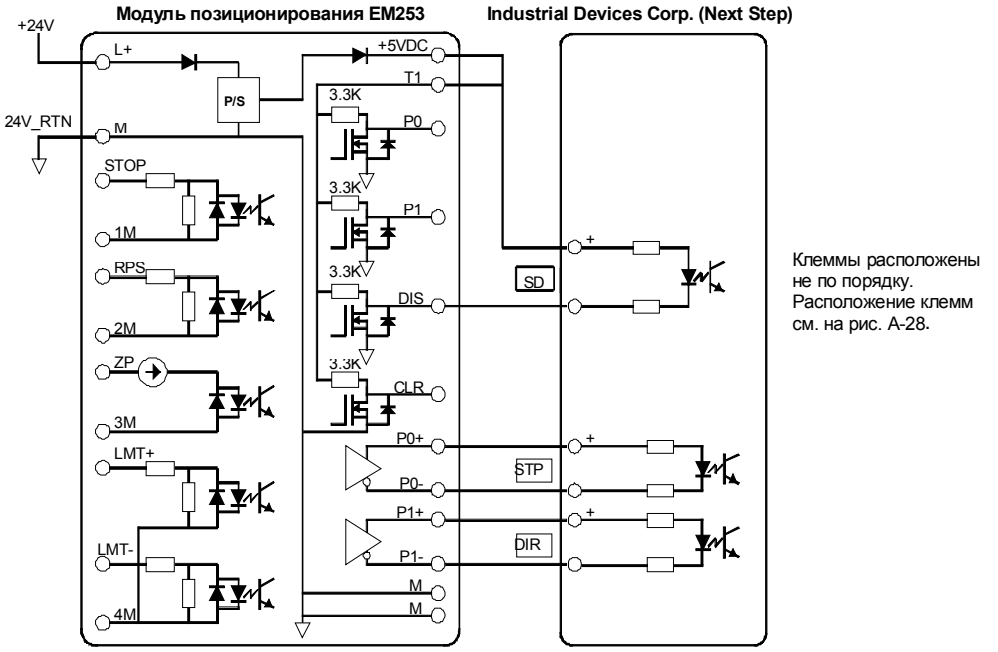


Рис. A–31. Подключение модуля позиционирования EM 253 к Industrial Devices Corp. (Next Step)

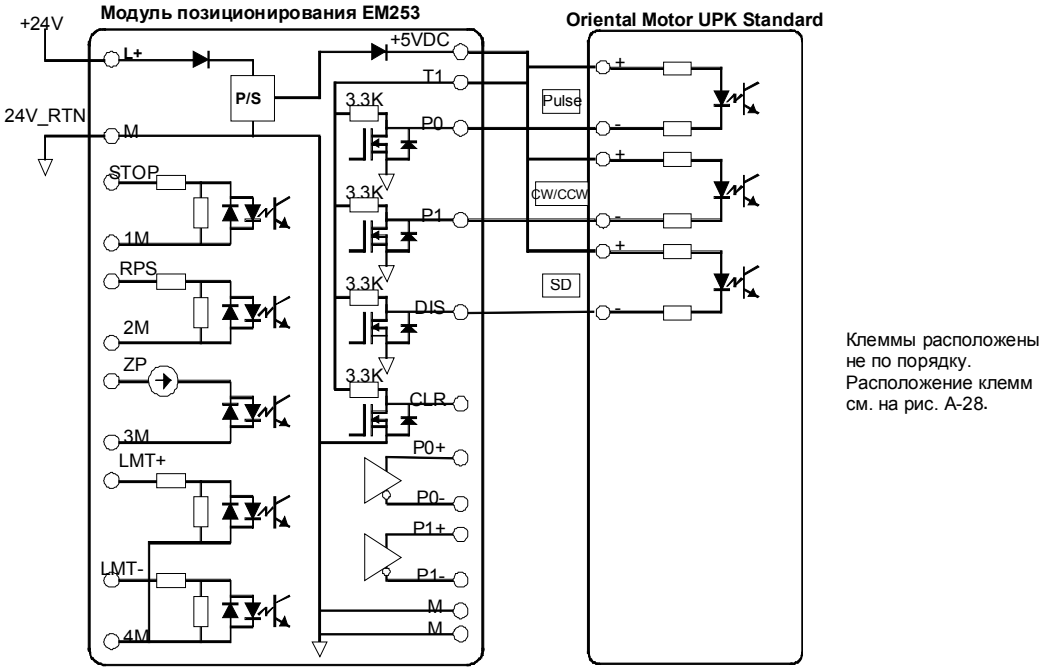


Рис. A–32. Подключение модуля позиционирования EM 253 к Oriental Motor UPK Standard

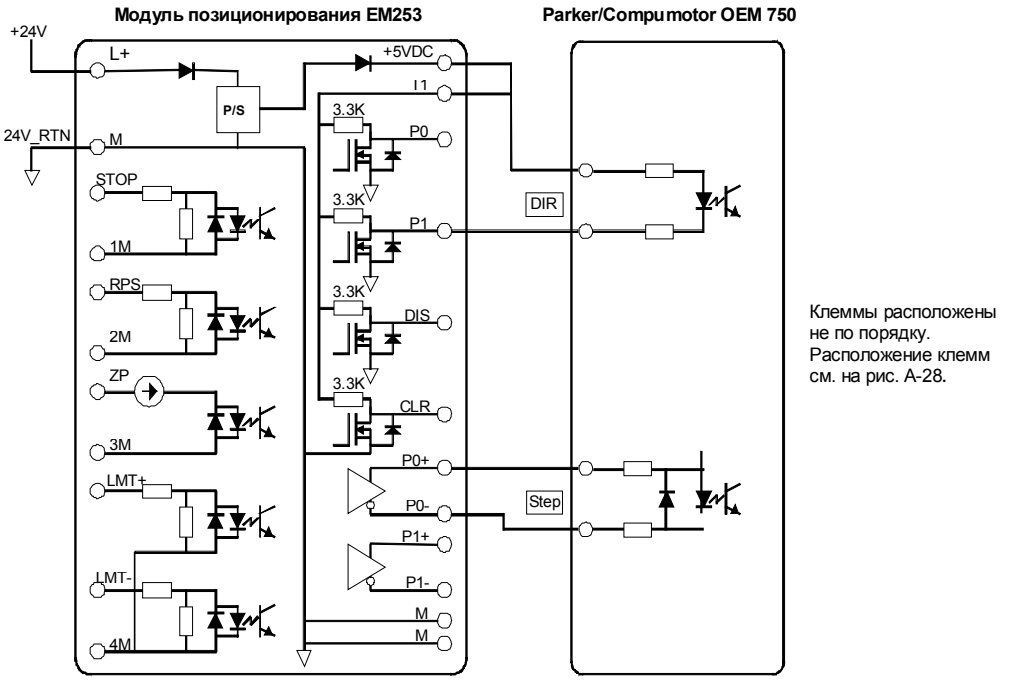


Рис. A–33. Подключение модуля позиционирования EM 253 к Parker/Compumotor OEM 750

Технические данные модуля интерфейса с исполнительными устройствами и датчиками (AS-Interface) (CP 243-2)

Таблица A–52. 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 A–53. 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 A–54. 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
Конфигурация	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 Mast*er)
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).

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 сконфигурированы простым нажатием на кнопку. Дальнейшее конфигурирование 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

Соединительные элементы

Модуль AS-Interface имеет следующие соединительные элементы:

- Две пары клемм для кабеля модуля AS-Interface (с внутренней перемычкой)
- Одна клемма для функциональной земли

Клеммы находятся под крышкой передней панели, как показано на рис. A–34.

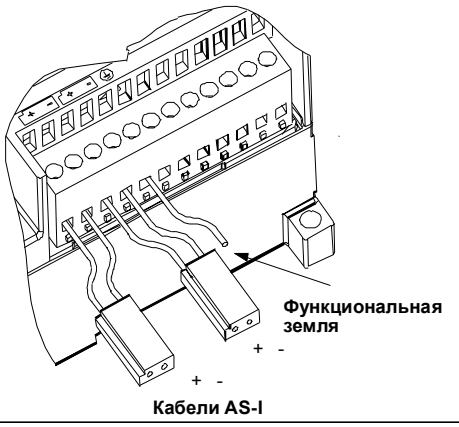


Рис. A–34. Присоединение кабеля к модулю AS-Interface

Осторожно
Нагрузка на контакты модуля AS-Interface не должна превышать 3 A. 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 заземлительную 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 съемные модули

Съемный модуль	Описание	Номер для заказа
Модуль памяти	Накопитель на базе модуля памяти: программа, данные и конфигурация	6ES7 291-8GE20-0XA0
Часы реального времени с батарейкой	Точность модуля часов: 2 минуты в месяц при 25°C, 7 минут в месяц в диапазоне от 0°C до 55°C	6ES7 297-1AA20-0XA0
Батарейный модуль	Батарейный модуль (время хранения данных): 200 дней, как правило	6ES7 291-8BA20-0XA0

Общие технические характеристики		Размеры
Батарейка	3 В, 30 мА-час., Renata CR 1025	18 мм; 10 мм; 18 мм
Размеры	9,9 x 2,5 мм	
Тип	Литиевая < 0,6 г	

Модуль памяти хранит всю программу и блок данных для любых CPU (CPU 221, CPU 222, CPU 224, CPU 226 и CPU 226XM).

Кабель расширения входов-выходов

Общие технические характеристики (6ES7 290-6AA20-0XA0)	
Длина кабеля	0,8 м
Вес	25 г
Тип соединителя	10-контактный ленточный

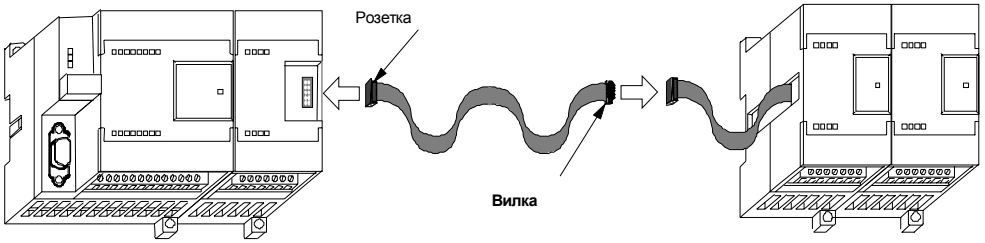


Рис. A–35. Типичный монтаж кабеля расширения входов-выходов

Совет
В цепи CPU/модуль расширения допустим только один кабель расширения.

Кабель PC/PPI

Кабель PC/PPI (6ES7 901-3BF21-0XA0) Общие характеристики	
Напряжение питания	от 14,4 до 28,8 В пост. тока
Ток источника питания при номинальном напряжении источника 24 В	50 мА макс. эффективное значение
Задержка изменения направления: от фронта принятого стартового бита RS-232 до фронта переданного стартового бита RS-485	1,2 мкс, макс.
Задержка изменения направления: от фронта принятого стоп-бита RS-232 до блокировки передачи RS-485	1,4 времени передачи символа, макс. (1,4 x 11/скорость передачи в бодах) = 1,6 мс при 9600 Бод
Задержка при распространении	4 мкс, максимум, от RS-485 к RS-232, 1,2 мкс, максимум, от RS-232 к RS-485
Электрическая развязка (RS-485 с RS-232)	500 В пост. тока
Электрические характеристики на стороне RS-485	
Диапазон напряжений синфазной помехи	от -7 В до +12 В, в течение 1 секунды, 3 $В_{эфф}$ постоянно
Полное входное сопротивление приемника	5,4 кОм, минимум, включая оконечную нагрузку
Оконечная нагрузка/смещение	10 кОм относительно +5 В на В, PROFIBUS, контакт 3 10 кОм относительно GND [заземление] на А, PROFIBUS, контакт 8
Порог срабатывания/чувствительность приемника	+/- 0,2 В, типичный гистерезис 60 мВ
Дифференциальное выходное напряжение передатчика	мин. 2 В при R_L = 100 Ом, мин. 1,5 В при R_L = 54 Ом
Электрические характеристики на стороне RS-232	
Полное входное сопротивление приемника	мин. 3 кОм
Порог срабатывания/чувствительность приемника	мин. 0,8 В нижний, макс. 2,4 В верхний, типичный гистерезис 0,5 В
Выходное напряжение передатчика	мин. +/- 5 В при R_L = 3 кОм

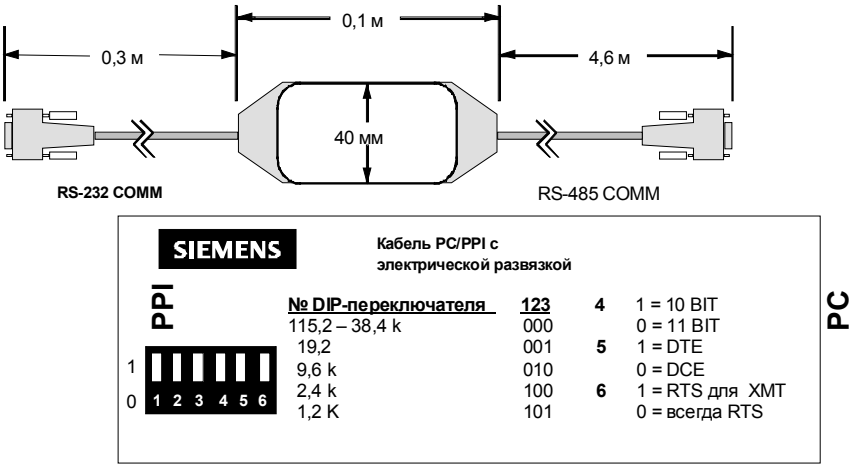


Рис. A–36. Размеры кабеля PC/PPI

Таблица A–55. Положения переключателей на кабеле PC/PPI

Переключатели скорости передачи 1,2,3*		Переключатель режима модема 4*		Переключатель DCE/DTE 5*		Переключатель готовности к передаче (RTS) для DTE 6*	
115200 - 38400	000	11–битовый модем	0	DCE (аппаратура передачи данных)	0	Режим RTS всегда активен	0
19200	001	10–битовый модем	1	DTE (терминальное оборудование)	1	Режим RTS активен, когда ПЛК ведет передачу	1
9600	010						
4800	011						
2400	100						
1200	101						
600	110						
* Переключатель: 1 = вверх; 0 = вниз							

Таблица A–56. Назначение контактов разъемов RS–485 и RS–232 DCE

Назначение контактов разъема RS–485		Назначение контактов разъема RS–232 DCE	
Номер контакта	Описание сигнала	Номер контакта	Описание сигнала
1	Земля (логическая земля RS-485)	1	Data Carrier Detect (DCD) [обнаружение несущей сигнала данных] (не используется)
2	Обратный провод 24 В (логическая земля RS-485)	2	Receive Data (RD) [Прием данных] (выход кабеля PC/PPI)
3	Сигнал B (RxD/TxD+)	3	Transmit Data [Передача данных] (TD) (вход кабеля PC/PPI)
4	RTS (уровень TTL)	4	Data Terminal Ready [Готовность терминала данных] (DTR) (не используется)
5	Земля (логическая земля RS–485)	5	Земля (логическая земля RS–232)
6	Не соединен	6	Data Set Ready [Набор данных готов] (DSR) (не используется)
7	Питание 24 В	7	Request To Send [Запрос на передачу] (RTS) (не используется)
8	Сигнал A (RxD/TxD-)	8	Clear To Send [Готовность к приему] (CTS) (не используется)
9	Выбор протокола	9	Ring Indicator [Индикатор вызова] (RI) (не используется)

Таблица A–57. Назначение контактов разъемов RS-485 и RS-232 DTE

Назначение контактов разъема RS–485		Назначение контактов разъема RS–232 DTE [1]	
Номер контакта	Описание сигнала	Номер контакта	Описание сигнала
1	Земля (логическая земля RS-485)	1	Data Carrier Detect (DCD) [обнаружение несущей сигнала данных] (не используется)
2	Обратный провод 24 В (логическая земля RS-485)	2	Receive Data (RD) [Прием данных] (вход кабеля PC/PPI)
3	Сигнал B (RxD/TxD+)	3	Transmit Data [Передача данных] (TD) (выход кабеля PC/PPI)
4	RTS (уровень TTL)	4	Data Terminal Ready [Готовность терминала данных] (DTR) (не используется)
5	Земля (логическая земля RS–485)	5	Земля (логическая земля RS–232)
6	Не соединен	6	Data Set Ready [Набор данных готов] (DSR) (не используется)
7	Питание 24 В	7	Request To Send (RTS) [Запрос на передачу] (выход кабеля PC/PPI) (выбирается с помощью переключателя)
8	Сигнал A (RxD/TxD-)	8	Clear To Send [Готовность к приему] (CTS) (не используется)
9	Выбор протокола	9	Ring Indicator [Индикатор вызова] (RI) (не используется)

[1] Для модемов требуется переходное устройство от розетки к вилке и от 9-контактного разъема к 25-контактному.

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 36 x 22 мм	91 x 36 x 22 мм	147 x 36 x 25 мм
Вес	0,02 кг	0,03 кг	0,04 кг
Количество входов и выходов	8	14	24

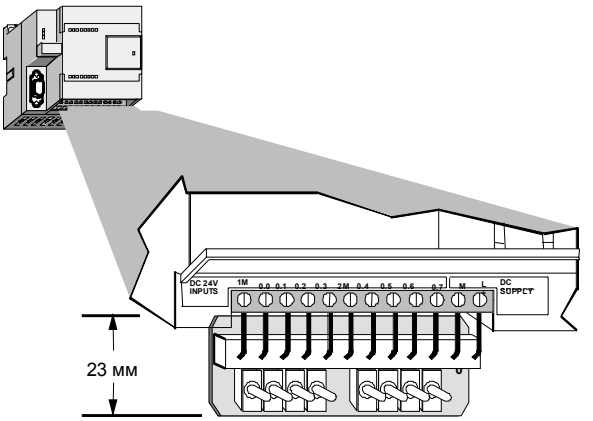


Рис. A–37. Установка имитатора входов

Предостережение

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 A дают информацию о наличных мощностях отдельных CPU и потребностях в мощности модулей расширения.

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 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 B–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.

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 B–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 A.

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 404

Ошибки этапа выполнения 405

Нарушение правил компиляции 406

Коды фатальных ошибок и сообщения

Фатальные (неустранимые) 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 светодиод системной ошибки (SF) и светодиод 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 C–1 приведен список с описаниями кодов ошибок, которые могут быть считаны из S7–200.

Таблица C–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	Ошибка времени при контроле времени цикла
0004	Неисправность внутреннего ЭСППЗУ
0005	Ошибка контрольной суммы внутреннего ЭСППЗУ в программе пользователя
0006	Ошибка контрольной суммы внутреннего ЭСППЗУ в параметрах конфигурации (SDB0)
0007	Ошибка контрольной суммы внутреннего ЭСППЗУ в принудительно задаваемых данных
0008	Ошибка контрольной суммы внутреннего ЭСППЗУ в устанавливаемых по умолчанию значениях таблицы выходов
0009	Ошибка контрольной суммы внутреннего ЭСППЗУ 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 C–2 приведены описания кодов нефатальных ошибок.

Таблица C–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 ENI, DISI, SPA или HDEF 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
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)

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 C–3 дает описания кодов ошибок, которые генерируются при нарушениях правил компиляции.

Таблица C–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 команда блокирования перед командой SCRE
0097	Программа пользователя содержит как ненумерованные, так и нумерованные команды EV/ED
0098	Недопустимое редактирование в режиме RUN (предпринята попытка редактирования в программе с ненумерованными 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

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 408
SMB1: биты состояния 408
SMB2: принимаемые символы при свободно программируемом обмене данными 409
SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными 409
SMB4: переполнение очереди ожидания 409
SMB5: состояние входов/выходов 410
SMB6: идентификационный регистр CPU 410
SMB7: резерв 410
SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода 411
SMW22 – SMW26: времена цикла 412
SMB28 и SMB29: аналоговый потенциометр 412
SMB30 и SMB130: регистры управления свободно программируемым обменом данными 412
SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ) 413
SMB34 и SMB35: регистры интервалов времени для прерываний, управляемых временем 413
SMB36 – SMB65: регистр HSC0, HSC1 и HSC2 413
SMB66 – SMB85: регистры PTO/PWM 415
SMB86 – SMB94 и SMB186 – SMB194: управление приемом сообщений 416
SMW98: ошибки в шине расширения 417
SMB130: регистр управления свободно программируемым обменом данными (см. SMB30) 417
SMB131 – SMB165: регистры HSC3, HSC4 и HSC5 417
SMB166 – SMB185: таблица определения профилей PTO0, PTO1 418
SMB186 – SMB194: управление приемом сообщений (см. SMB86 – SMB94) 418
SMB200 – SMB549: состояние интеллектуальных модулей 419

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.

Совет
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.

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	MSB 7 … LSB 0 x x x x r r r r 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.

SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода

Байты SMB8 – SMB21 разбиты на пары для модулей расширения с 0 по 6. Как описано в таблице D–8, байт каждой пары, имеющий четный номер, является идентификационным регистром модуля. Эти байты идентифицируют тип модуля, а также вид и количество входов и выходов. Байт каждой пары, имеющий нечетный номер, является регистром ошибок модуля. Эти байты предоставляют в распоряжение индикацию любых ошибок во вводе/выводе для этого модуля.

Таблица D–8. Байты специальной памяти от SMB8 до SMB21

SM-байт	Описание (только чтение)
Формат	Байт с четным номером: идентификационный регистр модуля MSB 7 … LSB 0 m t t a i i q q m: Наличие модуля 0 = имеется; 1 = отсутствует tt: Тип модуля 00 неинтеллектуальный модуль ввода/вывода 01 интеллектуальный модуль ввода/вывода 10 резерв 11 резерв a: Вид входов/выходов 0 = цифровой; 1 = аналоговый ii: Входы 00 нет входов 01 2 аналоговых или 8 цифровых 10 4 аналоговых или 16 цифровых 11 8 аналоговых или 32 цифровых qq: Выходы 00 нет выходов 01 2 аналоговых или 8 цифровых 10 4 аналоговых или 16 цифровых 11 8 аналоговых или 32 цифровых Байт с нечетным номером: регистр ошибок модуля MSB 7 … LSB 0 c 0 0 b r p f t c: Ошибка конфигурации 0 = нет ошибки; 1 = ошибка b: Неисправность шины или ошибка при проверке четности r: Выход за пределы диапазона p: Нет питания у пользователя f: Перегорел предохранитель t: Клеммный блок не закреплен
SMB8 SMB9	Модуль 0: регистр идентификации Модуль 0: регистр ошибок
SMB10 SMB11	Модуль 1: регистр идентификации Модуль 1: регистр ошибок
SMB12 SMB13	Модуль 2: регистр идентификации Модуль 2: регистр ошибок
SMB14 SMB15	Модуль 3: регистр идентификации Модуль 3: регистр ошибок
SMB16 SMB17	Модуль 4: регистр идентификации Модуль 4: регистр ошибок
SMB18 SMB19	Модуль 5: регистр идентификации Модуль 5: регистр ошибок
SMB20 SMB21	Модуль 6: регистр идентификации Модуль 6: регистр ошибок

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 конфигурируют соответствующий коммуникационный порт для работы в режиме свободно программируемого обмена данными и предоставляют выбор 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 MSB 7 … 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 исполнение 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	SMB31: команда MSB 7 … LSB 0 c 0 0 0 0 0 s s SMW32: адрес в V-памяти MSB 15 … LSB 0 Адрес в V-памяти
SM31.0 и SM31.1	ss: размер данных 00 = байт 10 = слово 01 = байт 11 = двойное слово
SM31.7	c: Сохранить в ЭСППЗУ 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
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 PTO/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 PTO0 прерван: 0 = нет ошибки, 1 = прерван из-за ошибки в расчете приращения
SM66.5	Профиль PTO0 прерван: 0 = прерван не по команде пользователя, 1 = прерван по команде пользователя
SM66.6	Переполнение конвейера PTO0 (очищается системой при использовании внешних профилей, в противном случае должен быть сброшен пользователем): 0 = нет переполнения, 1 = конвейер переполнен
SM66.7	Бит холостого хода PTO0: 0 = PTO активен, 1 = PTO не активен
SM67.0	Актуализация значения времени цикла PTO0/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 PTO0: 1 = записать новое количество импульсов
SM67.3	База времени PTO0/PWM0: 0 = 1 мкс/такт; 1 = 1 мс/такт
SM67.4	Обновлять PWM0 синхронно: 0 = асинхронное обновление, 1 = синхронное обновление
SM67.5	Режим работы PTO0: 0 = односегментный режим (время цикла и количество импульсов хранятся в SM-памяти), 1 = многосегментный режим (таблица профиля хранится в памяти переменных)
SM67.6	Выбор PTO0 или PWM0: 0 = PTO, 1 = PWM
SM67.7	Бит разблокировки PTO0/PWM0: 1 = разблокировать
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	Байт состояния приема сообщения MSB 7 … LSB 0 n r e 0 0 t c 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
SMB87	SMB187	Байт управления приемом сообщений MSB 7 … LSB 0 en sc ec il c/m tmr bk 0 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.
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	Максимальное количество символов, которое должно быть принято (от 1 до 255 байтов). Примечание: Этот диапазон должен быть установлен на ожидаемый максимальный размер буфера, даже если завершение сообщения с помощью подсчета символов не используется.

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 контроля 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
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

Таблица D–18. Байты специальной памяти SMB131 – SMB165, продолжение

SM-байт	Описание
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 – SMB175	Резерв
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 – SMB185	Резерв

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.

Способ назначения адресов SM-памяти для интеллектуальных модулей изменился, начиная с версии 2.2.

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							
Интеллекту-альный модуль в слоте 0	**Интеллекту-альный модуль в слоте 1**	**Интеллекту-альный модуль в слоте 2**	**Интеллекту-альный модуль в слоте 3**	**Интеллекту-альный модуль в слоте 4**	**Интеллекту-альный модуль в слоте 5**	**Интеллекту-альный 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

E

CPU	Номер для заказа
CPU 221 DC/DC/DC 6 входов/4 выхода	6ES7 211-0AA22-0XB0
CPU 221 AC/DC/Relay 6 входов/4 реле	6ES7 211-0BA22-0XB0
CPU 222 DC/DC/DC 8 входов/6 выходов	6ES7 212-1AB22-0XB0
CPU 222 AC/DC/Relay 8 входов/6 реле	6ES7 212-1BB22-0XB0
CPU 224 DC/DC/DC 14 входов/10 выходов	6ES7 214-1AD22-0XB0
CPU 224 AC/DC/Relay 14 входов/10 реле	6ES7 214-1BD22-0XB0
CPU 226 DC/DC/DC 24 входа/16 выходов	6ES7 216-2AD22-0XB0
CPU 226 AC/DC/Relay 24 входа/16 реле	6ES7 216-2BD22-0XB0
CPU 226XM DC/DC/DC 24 входа/16 реле	6ES7 216-2AF22-0XB0
CPU 226XM AC/DC/Relay 24 входа/16 реле	6ES7 216-2BF22-0XB0
Модули расширения	**Номер для заказа**
EM 221, 24 В пост. тока, цифровой, 8 входов	6ES7 221-1BF22-0XA0
EM 221, цифровой, 8 входов перемен. тока (8 x 120/230 В перем. тока)	6ES7 221-1EF22-0XA0
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 223, 24 В пост. тока, цифровой, комбинированный, 4 входа/4 выхода	6ES7 223-1BF22-0XA0
EM 223, 24 В пост. тока цифровой, комбинированный, 4 входа/4 релейных выхода	6ES7 223-1HF22-0XA0
EM 223, 24 В пост. тока цифровой, комбинированный, 8 входов/8 выходов	6ES7 223-1BH22-0XA0
EM 223, 24 В пост. тока цифровой, комбинированный, 8 входов/8 релейных выходов	6ES7 223-1PH22-0XA0
EM 223, 24 В пост. тока цифровой, комбинированный, 16 входов/16 выходов	6ES7 223-1BL22-0XA0
EM 223, 24 В пост. тока цифровой, комбинированный, 16 входов/16 релейных выходов	6ES7 223-1PL22-0XA0
EM 231, аналоговый ввод, 4 входа	6ES7 231-0HC22-0XA0
EM 231, аналоговый ввод, термометр сопротивления, 2 входа	6ES7 231-7PB22-0XA0
EM 231, аналоговый ввод, термопара, 4 входа	6ES7 231-7PD22-0XA0
EM 232, аналоговый вывод, 2 выхода	6ES7 232-0HB22-0XA0
EM 235, аналоговый, комбинированный, 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
CP 243-2 AS Interface, коммуникационный процессор	6GK7 243-2AX01-0XA0

Сменные модули и кабели	Номер для заказа
MC 291, 32K x 8 EEPROM, модуль памяти	6ES7 291-8GE20-0XA0
CC 292, часы реального времени с батарейным модулем для CPU 22x	6ES7 297-1AA20-0XA0
BC 293, батарейный модуль для CPU 22x	6ES7 291-8BA20-0XA0
Кабель, расширение ввода/вывода, .8 метров, CPU 22x/EM	6ES7 290-6AA20-0XA0
Кабель, PC/PPI, с гальванической развязкой, штекер под 90 град., переключатель RTS	6ES7 901-3BF21-0XA0
Программное обеспечение для программирования	**Номер для заказа**
STEP 7-Micro/WIN 32 (V3.2), отдельная лицензия (CD–ROM)	6ES7 810-2BC02-0YX0
STEP 7-Micro/WIN 32 (V3.2), лицензия на модернизацию (CD–ROM)	6ES7 810-2BC02-0YX3
S7–200 Toolbox: дизайнер TP для TP070, версия 1.0 (CD–ROM)	6ES7 850-2BC00-0YX0
STEP 7-Micro/WIN Add–on [дополнительный пакет]: STEP 7-Micro/WIN 32, библиотека команд, V1.1 (CD–ROM)	6ES7 830-2BC00-0YX0

Коммуникационные платы	Номер для заказа
CP 5411: укороченная AT ISA	6GK 1 541-1AA00
CP 5511: PCMCIA, тип II	6GK 1 551-1AA00
CP 5611: плата PCI (версия 3.0 или выше)	6GK 1 561-1AA00
Руководства	**Номер для заказа**
Руководство пользователя интерфейса оператора TD 200	6ES7 272-0AA20-8BA0
Руководство пользователя сенсорной панели TP 070 (английский)	6AV6 591-1DC01-0AB0
Руководство S7–200 Point–to–Point Interface Communication [Обмен данными через двухточечный интерфейс S7-200] (английский/немецкий)	6ES7 298-8GA00-8XH0
Руководство CP 243-2 SIMATIC NET AS–Interface Master [Master-устройство интерфейса с датчиками и исполнительными устройствами CP 243-2 SIMATIC NET] (английский)	6GK7 243-2AX00-8BA0
Программируемый контроллер S7–200. Системное руководство (немецкий)	6ES7 298-8FA22-8AH0
Программируемый контроллер S7–200. Системное руководство (английский)	6ES7 298-8FA22-8BH0
Программируемый контроллер S7–200. Системное руководство (французский)	6ES7 298-8FA22-8CH0
Программируемый контроллер S7–200. Системное руководство (испанский)	6ES7 298-8FA22-8DH0
Программируемый контроллер S7–200. Системное руководство (итальянский)	6ES7 298-8FA22-8EH0
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
Соединительный блок CPU 22x/EM, 7 клемм, съемный	6ES7 292-1AD20-0AA0
Соединительный блок CPU 22x/EM, 12 клемм, съемный	6ES7 292-1AE20-0AA0
Соединительный блок CPU 22x/EM, 14 клемм, съемный	6ES7 292-1AF20-0AA0
Соединительный блок CPU 22x/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
Интерфейс оператора TD 200	6ES7 272-0AA20-0YA0
Интерфейс оператора OP3	6AV3 503-1DB10T
Интерфейс оператора OP7	6AV3 607-1JC20-0AX1
Интерфейс оператора OP17	6AV3 617-1JC20-0AX1
Сенсорная панель TP070	6AV6 545-0AA15-2AX0
Сенсорная панель TP170A	6AV6 545-0BA15-2AX0
Разное	**Номер для заказа**
Упоры для профильной шины	6ES5 728-8MAll
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, 2x12, 2x14 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

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–3.

Совет
При использовании времен исполнения из таблицы F–3 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–3 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 3 мкс.

Влияние косвенной адресации

В таблице F–3 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 22 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
Аналоговые входы (AI) С заблокированной аналоговой фильтрацией: С разблокированной аналоговой фильтрацией:	 149 мкс 0 мкс
Аналоговые выходы (AQ)	73 мкс
Локальная память (L)	5,4 мкс
Аккумуляторы (AC)	4,4 мкс

Влияние использования определенных команд CPU 226XM

Выполнение определенных команд, вызывающих передачу управления, на CPU 226XM требует дополнительного времени на выполнение команды.

В таблице F–2 представлено время, которое должно быть добавлено к основному времени выполнения для каждой из перечисленных команд.

Таблица F–2. Добавка к времени выполнения для команд CPU 226XM

Команда	Добавка к времени выполнения
ATCH	1,0 мкс
CALL	4,3 мкс
CSCRE	3,1 мкс
FOR (добавка к базе) (добавка к множителю цикла)	3,1 мкс 3,1 мкс
INT	1,7 мкс
JMP	3,1 мкс
RET	2,8 мкс

Таблица F–3. Времена выполнения команд

Команда	мкс
= Для: I SM, T, C, V, S, Q, M L	0,37 1,8 19,2
+D	55
-D	55
*D	92
/D	376
+I	46
-I	47
*I	71
/I	115
=I Для: встроенных выходов выходов расширения	29 39
+R	110 макс. 163
-R	113 макс. 166
*R	100 макс. 130
/R	300 макс. 360
A Для: I SM, T, C, V, S, Q, M L	0,37 1,1 10,8
AB < =, =, >=, >, <, <>	35
AD < =, =, >=, >, <, <>	53
AENO	.6
AI Для: встроенных выходов выходов расширения	27 35
ALD	0,37
AN При использовании: I SM, T, C, V, S, Q, M L	0,37 1,1 10,8
ANDB	37
ANDD	55
ANDW	48
ANI Для: встроенных входов входов расширения	27 35
AR <=, =, >=, >, <, <>	54
AS=, <> Время = База + (LM * N) База Множитель длины (LM) N – количество сравниваемых символов	 51 9,2
ATCH	20
ATH Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 41 55 20

Команда	мкс
ATT	70
AW < =, =, >=, >, <, <>	45
BCDI	66
BIR Для: встроенных входов входов расширения	45 53
BIW Для: встроенных выходов выходов расширения	46 56
BMB Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 21 51 11
BMD Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 21 51 20
BMW Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 21 51 16
BTI	27
CALL Без параметров: С параметрами: Время = База + Σ (время операнда) База Время операнда бит (вход, выход) байт (вход, выход) слово (вход, выход) двойное слово (вход, выход) Примечание: обработка выходных операндов происходит при выходе из подпрограммы	15 32 23, 21 21, 14 24, 18 27, 20
CFND Макс. время = База + N1 * ((LM1 * N2) + LM2) База Множитель длины 1 (LM1) Множитель длины 2 (LM2) N1 – длина исходной строки N2 – длина набора символов, строка	79 79 9,2 4,4
COS	1525 макс. 1800
CRET	13
CRETI	23
CSCRE	0,9
CTD При фронте на счетном входе Иначе	48 36
CTU При фронте на счетном входе Иначе	53 35
CTUD При фронте на счетном входе Иначе	64 45
DECB	30
DECD	42
DECO	36

Команда		мкс
DECW		37
DISI		18
DIV		119
DTA		540
DTI		36
DTCH		18
DTR		60 макс. 70
DTS		540
ED		15
ENCO		39 макс. 43
END		0,9
ENI		53
EU		15
EXP		1170 макс. 1375
FIFO	Время = База + (длина * LM) База Множитель длины (LM)	 70 14
FILL	Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 29 50 7
FND <, =, >, <>	Время = База + (длина * LM) База Множитель длины (LM)	 85 12
FOR	Время = База + (число циклов * LM) База Множитель цикла (LM)	 64 50
GPA		31
HDEF		35
HSC		37
HTA	Время = База + (длина * LM) База (постоянная длина) База (переменная длина) Множитель длины (LM)	 38 48 11
IBCD		114
INCB		29
INCD		42
INCW		37
INT	Обычно с одним прерыванием	47
INVB		31
INVD		42
INVW		38
ITA		260
ITB		27
ITD		36

Команда			мкс
ITS			260
JMP			0,9
LBL			0,37
LD	Для:	I, SM0.0 SM, T, C, V, S, Q, M L	0,37 1,1 10,9
LDB <=, =, >=, >, <, <>			35
LDD <=, =, >=, >, <, <>			52
LDI	Для:	встроенных входов входов расширения	26 34
LDN	Для:	I SM, T, C, V, S, Q, M L	0,37 1,1 10,9
LDNI	Для:	встроенных входов входов расширения	26 34
LDR<=, =, >=, >, <, <>			55
LDS			0,37
LDS=, <>	Время = База + (LM * N) База Множитель длины (LM) N – это количество сравниваемых символов		 51 9,2
LDW <=, =, >=, >, <, <>			42
LIFO			70
LN			1130 макс. 1275
LPP			0,37
LPS			0,37
LRD			0,37
LSCR			12
MEND			0,5
MOVB			29
MOVD			38
MOVR			38
MOVW			34
MUL			70
NEXT			0
NETR			179
NETW	Время = База + (длина * LM) База Множитель длины (LM)		 175 8
NOP			0,37
NOT			0,37
O	Для:	I SM, T, C, V, S, Q, M L	0,37 1,1 10,8
OB < =, =, >=, >, <, <>			35
OD < =, =, >=, >, <, <>			53

Команда			мкс
OI	Для:	встроенных входов входов расширения	27 35
OLD			0,37
ON	Для:	I SM, T, C, V, S, Q, M L	0,37 1,1 10,8
ONI	Для:	встроенных входов входов расширения	27 35
OR<=, =, >=, >, <, <>			55
ORB			37
ORD			55
ORW			48
OS=, < >		Время + База + (LM * N) База Множитель длины (LM) N – это количество сравниваемых символов	 51 9,2
OW < =, =, >=, >, <, <>			45
PID		База Добавка для перерасчета постоянных в интегральной и пропорционально-дифференциальной составляющей	750 1000
PLS:	Для:	PWM PTO, односегментный режим PTO, многосегментный режим	57 67 92
R		Длина =1 и указана как константа С операндами = C, T Со всеми остальными операндами Иначе, Время =База + (длина<LM) База для операнда = C, T База для всех остальных операндов LM для операнда = C, T LM для всех остальных операндов Если длина хранится как переменная, добавьте к базе	 17, 24 5 19, 19 28 8,6, 16,5 0,9 29
RCV			80
RET			13
RETI			23
RI		Всего = База + (длина * LM) База LM для встроенных выходов LM для выходов расширения Если длина хранится как переменная, добавьте к базе	 18 22 32 30
RLB		Всего = База + (длина + LM) База Множитель длины (LM)	 42 0,6
RLD		Всего = База + (длина * LM) База Множитель длины (LM)	 52 2,5
RLW		Всего = База + (длина * LM) База Множитель длины (LM)	 49 1,7

Команда		мкс
ROUND		108 макс. 183
RRB	Всего = База + (длина * LM) База Множитель длины (LM)	 42 0,6
RRD	Всего = База + (длина * LM) База Множитель длины (LM)	 52 2,5
RRW	Всего = База + (длина * LM) База Множитель длины (LM)	 49 1,7
RTA	Время = База + (LM * N) База (для первой цифры результата) Множитель длины (LM) N - количество дополнительных цифр в результате	 1000 240
RTS	Время = База + (LM * N) База (для первой цифры результата) Множитель длины (LM) N - количество дополнительных цифр в результате	 1000 240
S	Для длины = 1 и указанной как константа Иначе: Время = База + (длина * LM) База Множитель длины (LM) Если длина хранится как переменная, добавьте к базе	5 27 0,9 29
SBR		0
SCAT	Время = База + (LM * N) База Множитель длины (LM) N – это число присоединенных символов	 55 8,8
SCPY	Время = База + (LM * N) База Множитель длины (LM) N – это число копируемых символов	 43 8,8
SCRE		0,37
SCRT		17
SEG		30
SFND	Макс. время = База + (N1-N2) * ((LM1*N2)+LM2) База Множитель длины 1 (LM1) Множитель длины 2 (LM2) N1 – это длина исходной строки N2 – это длина строки поиска	 79 11,5 17,8
SHRB	Всего = База + (длина * LM1) + ((длина /8) * LM2) База (постоянная длина) База (переменная длина) Множитель длины 1 (LM1) Множитель длины 2 (LM2)	 76 84 1,6 4

Команда		мкс
SI	Всего = База + (длина * LM) База LM для встроенного выхода LM для выхода расширения Если длина хранится как переменная, добавьте к базе База	 18 22 32 30
SIN		1525 макс. 1800
SLB	Всего = База + (длина * LM) База Множитель длины (LM)	 43 0,7
SLD	Всего = База + (длина * LM) База Множитель длины (LM)	 53 2,6
SLEN		46
SLW	Всего = База + (длина * LM) База Множитель длины (LM)	 51 1,3
SPA		243
SQRT		725 макс. 830
SRB	Всего = База + (длина * LM) База Множитель длины (LM)	 43 0,7
SRD	Всего = База + (длина * LM) База Множитель длины (LM)	 53 2,6
SRW	Всего = База + (длина * LM) База Множитель длины (LM)	 51 1,3
SSCPY	Время = База + (LM * N) База Множитель длины (LM) N – число копируемых символов	 82 8,8

Команда		мкс
STD	Время = База + (LM*N) База (для 1-го символа источника) Множитель длины (LM) N – число дополнительных символов источника	 84 59
STI	Время = База + (LM*N) База (для 1-го символа источника) Множитель длины (LM) N – число дополнительных символов источника	 84 59
STOP		16
STR	Время = База + (LM*N) База (для 1-го символа источника) Множитель длины (LM) N – число дополнительных символов источника	 100 120
SWAP		32
TAN		1825 макс. 2100
TODR		2400
TODW		1600
TOF		64
TON		64
TONR		56
TRUNC		103 макс. 178
WDR		16
XMT		78
XORB		37
XORD		55
XORW		48

Краткая справочная информация об S7–200

G

Чтобы облегчить вам поиск информации, в этом разделе собрана следующая информация:

- 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 символа		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 символа		1
19	Прерывание при завершении PTO 0	Прерывания по вводу/выводу (средняя)	0
20	Прерывание при завершении PTO 1		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: 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 CT=PT		2
22	Таймер T96, прерывание по CT=PT		3

Таблица G–3. Обзор областей памяти и характеристик CPU S7–200

Описание	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Размер программы пользователя	2 Кслова	2 Кслова	4 Кслова	4 Кслова	8 Кслов
Размер данных пользователя	1 Кслово	1 Кслово	2,5 Кслова	2,5 Кслова	5 Кслов
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 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	--	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB5119	VB0 – VB5119	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	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 включения/выключения 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, HC3, HC4 и HC5	HC0, HC3, HC4 и 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
	I0.0	I0.1	I0.2	I0.1	I0.3	I0.4	I0.5	I0.4
0	Датчик такт. имп.			Датчик такт. имп.	Датчик такт. имп.			Датчик такт. имп.
1	Датчик такт. имп.		Сброс		Датчик такт. имп.		Сброс	
2								
3	Датчик такт. имп.	Направле-ние			Датчик такт. имп.	Направле-ние		
4	Датчик такт. имп.	Направле-ние	Сброс		Датчик такт. имп.	Направле-ние	Сброс	
5								
6	Датчик такт. имп. вперед	Датчик такт. имп. назад			Датчик такт. имп. вперед	Датчик такт. имп. назад		
7	Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс		Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс	
8								
9	Фаза A	Фаза B			Фаза A	Фаза B		
10	Фаза A	Фаза B	Сброс		Фаза A	Фаза B	Сброс	
11								

Таблица G–5. Скоростные счетчики HSC1 и HSC2

Режим	HSC1				HSC2			
	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
0	Датчик такт. имп.				Датчик такт. имп.			
1	Датчик такт. имп.		Сброс		Датчик такт. имп.		Сброс	
2	Датчик такт. имп.		Сброс	Пуск	Датчик такт. имп.		Сброс	Пуск
3	Датчик такт. имп.	Направле-ние			Датчик такт. имп.	Направле-ние		
4	Датчик такт. имп.	Направле-ние	Сброс		Датчик такт. имп.	Направле-ние	Сброс	
5	Датчик такт. имп.	Направле-ние	Сброс	Пуск	Датчик такт. имп.	Направле-ние	Сброс	Пуск
6	Датчик такт. имп. вперед	Датчик такт. имп. назад			Датчик такт. имп. вперед	Датчик такт. имп. назад		
7	Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс		Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс	
8	Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс	Пуск	Датчик такт. имп. вперед	Датчик такт. имп. назад	Сброс	Пуск
9	Фаза A	Фаза B			Фаза A	Фаза B		
10	Фаза A	Фаза B	Сброс		Фаза A	Фаза B	Сброс	
11	Фаза A	Фаза B	Сброс	Пуск	Фаза A	Фаза B	Сброс	Пуск

Булевы команды		
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:<, <=,=, >=, >, <>I) 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 IN1 (x:<, <=,=, >=, >, <>) IN2 И вершина стека
ORx	IN1, IN2	Результат сравнения вещественных чисел IN1 (x:<, <=,=, >=, >, <>) IN2 ИЛИ вершина стека
NOT		Инверсия стека
EU		Обнаружение нарастающ. фронта
ED		Обнаружение убывающ. 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 IN1+OUT=OUT
+D	IN1, OUT	
+R	IN1, OUT	
-I	IN1, OUT	Вычесть целые, двойные целые или вещественные числа OUT-IN1=OUT
-D	IN1, OUT	
-R	IN1, OUT	
MUL	IN1, OUT	Перемножить целые числа (16*16->32)
*I	IN1, OUT	Перемножить целые, двойные целые или вещественные числа IN1 * OUT = OUT
*D	IN1, OUT	
*R	IN1, IN2	
DIV	IN1, OUT	Разделить целые числа (16/16->32)
/I	IN1, OUT	Разделить целые, двойные целые или вещественные числа OUT / IN1 = OUT
/D,	IN1, OUT	
/R	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	
INCD	OUT	
DECB	OUT	Уменьшить на 1 байт, слово или двойное слово
DECW	OUT	
DECD	OUT	
PID	TBL, LOOP	PID-регулятор
Таймеры и счетчики		
TON	Txxx, PT	Таймер с задержкой включения
TOF	Txxx, PT	Таймер с задержкой выключения
TONR	Txxx, PT	Таймер с задержкой включения с запоминанием
CTU	Cxxx, PV	Прямой счет
CTD	Cxxx, PV	Обратный счет
CTUD	Cxxx, PV	Реверсивный счетчик
Часы реального времени		
TODR	T	Прочитать реальное время
TODW	T	Записать реальное время
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 и конец сегмента реле управления последовательностью
SCRT	N	
CSCRE		
SCRE		

Команды передачи, сдвига и циклического сдвига		
MOVB	IN, OUT	Передать байт, слово, двойное слово, вещественное число
MOVW	IN, OUT	
MOVD	IN, OUT	
MOVR	IN, OUT	
BIR	IN, OUT	Передача непосредственно считанного байта
BIW	IN, OUT	Передача байта для непосредственной записи
BMB	IN, OUT, N	Групповая передача байтов, слов, двойных слов
BMW	IN, OUT, N	
BMD	IN, OUT, N	
SWAP	IN	Переставить местами байты
SHRB	DATA, S_BIT, N	Вдвинуть бит в регистр сдвига
SRB	OUT, N	Сдвинуть вправо байт, слово, двойное слово
SRW	OUT, N	
SRD	OUT, N	
SLB	OUT, N	Сдвинуть влево байт, слово, двойное слово
SLW	OUT, N	
SLD	OUT, N	
RRB	OUT, N	Циклически сдвинуть вправо байт, слово, двойное слово
RRW	OUT, N	
RRD	OUT, N	
RLB	OUT, N	Циклически сдвинуть влево байт, слово, двойное слово
RLW	OUT, N	
RLD	OUT, N	
Логические операции		
ANDB	IN1, OUT	Логическое И с байтами, словами и двойными словами
ANDW	IN1, OUT	
ANDD	IN1, OUT	
ORB	IN1, OUT	Логическое ИЛИ с байтами, словами и двойными словами
ORW	IN1, OUT	
ORD	IN1, OUT	
XORB	IN1, OUT	Логическое исключающее ИЛИ с байтами, словами и двойными словами
XORW	IN1, OUT	
XORD	IN1, OUT	
INVB	OUT	Инвертирование байта, слова или двойного слова (дополнение до 1)
INVW	OUT	
INVD	OUT	
Строковые команды		
SLEN	IN, OUT	Длина строки
SCAT	IN, OUT	Слияние строк
SCPY	IN, OUT	Копирование строки
SSCPY	IN, INDX, N, OUT	Копирование подстроки из строки
CFND	IN1, IN2, OUT	Поиск первого вхождения символа в строку
SFND	IN1, IN2, OUT	Поиск строки в строке

Табличные команды, поиск и преобразование		
ATT	DATA, TBL	Добавить данные к таблице
LIFO	TBL, DATA	Получить данные из таблицы
FIFO	TBL, DATA	
FND=	TBL, PTN, INDX	Найти в таблице значение данных, удовлетворяющее заданному критерию
FND<>	TBL, PTN, INDX	
FND<	TBL, PTN, INDX	
FND>	TBL, PTN, INDX	
FILL	IN, OUT, N	Заполнить диапазон памяти по образцу
BCDI	OUT	Преобразовать BCD в целое
IBCD	OUT	Преобразовать целое в BCD
BTI	IN, OUT	Преобразовать байт в целое
ITB	IN, OUT	Преобразовать целое в байт
ITD	IN, OUT	Преобразовать целое в двойное целое
DTI	IN, OUT	Преобразовать двойное целое в 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	Преобразовать ASCII в 16-ричн.
HTA	IN, OUT, LEN	Преобразовать 16-ричн. в ASCII
ITA	IN, OUT, FMT	Преобразовать целое в ASCII
DTA	IN, OUT, FM	Преобразовать двойное целое в ASCII
RTA	IN, OUT, FM	Преобразовать вещественное в ASCII
DECO	IN, OUT	Декодировать
ENCO	IN, OUT	Закодировать
SEG	IN, OUT	Генерировать 7-сегментное изображение
ITS	IN, FMT, OUT	Преобразовать целое в строку
DTS	IN, FMT, OUT	Преобразовать двойное целое в строку
RTS	IN, FMT, OUT	Преобразовать вещественное в строку
STI	STR, INDX, OUT	Преобразовать подстроку в целое
STD	STR, INDX, OUT	Преобразовать подстроку в двойное целое
STR	STR, INDX, OUT	Преобразовать подстроку в вещественное
Команды прерывания		
CRETI		Условный возврат из прерывания
ENI		Разблокировать прерывания
DISI		Заблокировать прерывания
ATCH	INT, EVNT	Назначить программу обработки прерывания событию
DTCH	EVNT	Отсоединить событие
Команды обмена данными		
XMT	TBL, PORT	Передача в режиме свободно программируемого обмена данными
RCV	TBL, PORT	Прием сообщения в режиме свободно программируемого обмена данными
NETR	TBL, PORT	Читать из сети
NETW	TBL, PORT	Записать через сеть
GPA	ADDR, PORT	Получить адрес порта
SPA	ADDR, PORT	Установить адрес порта
Скоростные команды		
HDEF	HSC, MODE	Определить режим скоростного счетчика
HSC	N	Активизировать скоростной счетчик
PLS	Q	Импульсный выход

Кому
SIEMENS ENERGY & AUTOMATION INC
ATTN: TECHNICAL COMMUNICATIONS M/S 519
3000 BILL GARLAND ROAD
PO BOX 1255
JOHNSON CITY TN USA 37605-1255

От
Имя:_ _
Название должности_ _
Название компании: _
Улица: _
Город и область:_ _
Страна:__ _
Телефон:_ _

Отметьте, пожалуйста относящуюся к Вам отрасль промышленности:

- ❒ Автомобильная
- ❒ Химическая
- ❒ Электромашиностроение
- ❒ Пищевая
- ❒ Приборостроение и средства управления
- ❒ Неэлектрическое машиностроение
- ❒ Нефтехимическая
- ❒ Фармацевтическая
- ❒ Пластмассовое производство
- ❒ Целлюлозно-бумажная
- ❒ Текстильная
- ❒ Транспорт
- ❒ Другое ___________________________

Форма для замечаний

Ваши комментарии и рекомендации помогут нам улучшить качество и полезность наших публикаций. При первой возможности заполните, пожалуйста, этот вопросник и верните его фирме Siemens.

Дайте, пожалуйста, каждому из следующих вопросов свою оценку в диапазоне от 1 (очень хорошо) до 5 (очень плохо).

1. Удовлетворяет ли содержание Вашим требованиям? ☐
2. Легко ли найти нужную Вам информацию? ☐
3. Понятен ли текст? ☐
4. Соответствует ли уровень технических подробностей Вашим потребностям? ☐
5. Оцените, пожалуйста, качество графики и таблиц. ☐

Дополнительные комментарии:

Области памяти S7–200 и их характеристики

Описание	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Размер программы пользователя	2 Кслова	2 Кслова	4 Кслова	4 Кслова	8 Кслов
Размер данных пользователя	1 Кслово	1 Кслово	2,5 Кслова	2,5 Кслова	5 Кслов
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 – AIW62	AIW0 – AIW62	AIW0 – AIW62
Аналоговые выходы (только запись)	--	AQW0 – AQW30	AQW0 – AQW62	AQW0 – AQW62	AQW0 – AQW62
Память переменных (V)	VB0 – VB2047	VB0 – VB2047	VB0 – VB5119	VB0 – VB5119	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)	SM0.0 – SM179.7	SM0.0 – SM299.7	SM0.0 – SM549.7	SM0.0 – SM549.7	SM0.0 – SM549.7
Только чтение	SM0.0 – SM29.7	SM0.0 – SM29.7	SM0.0 – SM29.7	SM0.0 – SM29.7	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	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 включения/ выключения 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, HC3, HC4 и HC5	HC0, HC3, HC4 и 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
Порт	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	Стр.
=	68
+D	140
-D	140
* D	140
/ D	140
+I	140
-I	140
=I	68
* I	140
/ I	140
+R	140
-R	140
*R	140
/R	140
A	66
AB < =	89
AB =	89
AB >	89
AB<	89
AB > =	89
AB <>	89
AD <	89
AD < =	89
AD =	89
AD >	89
AD > =	89
AD <>	89
AENO	70
AI	66
ALD	70
AN	66
ANDB	163
ANDD	163
ANDW	163
ANI	66
AR=	89
AR <	89
AR<=	89
AR >	89
AR>=	89
AR <>	89
AS=	91
AS<>	91
ATCH	155
ATH	96
ATT	189
AW <	89
AW < =	89

STL	Стр.
AW=	89
AW >	89
AW > =	89
AW <>	89
BCDI	92
BIR	166
BIW	166
BMB	167
BMD	167
BMW	167
BTI	92
CALL	203
CFND	187
COS	143
CRET	203
CRETI	155
CSCRE	173
CTD	106
CTU	106
CTUD	106
DECB	144
DECD	144
DECO	105
DECW	144
DISI	155
DIV	142
DTA	96
DTCH	155
DTI	92
DTR	92
DTS	100
ED	66
ENCO	105
END	168
ENI	155
EU	66
EXP	143
FIFO	190
FILL	192
FND <	193
FND <>	193
FND =	193
FND >	193
FOR	170
GPA	88
HDEF	111
HSC	111
HTA	96

STL	Стр.
IBCD	92
INCB	144
INCD	144
INCW	144
INVB	162
INVD	162
INVW	162
ITA	96
ITB	92
ITD	92
ITS	100
JMP	172
LBL	172
LD	66
LDB <=	89
LDB =	89
LDB >=	89
LDB >	89
LDB <	89
LDB <>	89
LDD >=	89
LDD <	89
LDD <=	89
LDD =	89
LDD >	89
LDD <>	89
LDI	66
LDN	66
LDNI	66
LDR=	89
LDR <	89
LDR<=	89
LDR >	89
LDR>=	89
LDR <>	89
LDS	70
LDS=	91
LDS<>	91
LDW <=	89
LDW <	89
LDW =	89
LDW >	89
LDW >=	89
LDW <>	89
LIFO	190
LN	143
LPP	70
LPS	70

STL	Стр.
LRD	70
LSCR	173
MOVB	165
MOVD	165
MOVR	165
MOVW	165
MUL	142
NEXT	170
NETR	74
NETW	74
NOT	66
O	66
OB =	89
OB > =	89
OB >	89
OB <	89
OB < =	89
OB <>	89
OD <	89
OD < =	89
OD =	89
OD >	89
OD > =	89
OD <>	89
OI	66
OLD	70
ON	66
ONI	66
OR=	89
OR <	89
OR<=	89
OR >	89
OR >=	89
OR <>	89
ORB	163
ORD	163
ORW	163
OS=	91
OS<>	91
OW <	89
OW < =	89
OW =	89
OW >	89
OW > =	89
OW <>	89
PID	145
PLS	125
R	68

STL	Стр.
RCV	79
RI	68
RLB	179
RLD	179
RLW	179
ROUND	92
RRB	179
RRD	179
RRW	179
RTA	96
RTS	100
S	68
SCAT	184
SCPY	184
SCRE	173
SCRT	173
SEG	92
SFND	187
SHRB	181
SI	68
SIN	143
SLB	179
SLD	179
SLEN	184
SLW	179
SPA	88
SQRT	143
SRB	179
SRD	179
SRW	179
SSCPY	186
STD	100
STI	100
STOP	168
STR	100
SWAP	183
TAN	143
TODR	73
TODW	73
TOF	196
TON	196
TONR	196
TRUNC	92
WDR	168
XMT	79
XORB	163
XORD	163
XORW	163

Предметный указатель

Символы

&, 32
*, 32

А

Аварийный контроль, PID-регулятор, 151
Автоматический режим, PID-регулятор, 151
Адаптер безмодемного режима, 226
Адрес
  назначение, 58
  сеть, 208
  символический, 58
  скоростные счетчики, 119
  установка удаленного адреса, 210
  Modbus, 331
Адреса специальной памяти, модуль позиционирования EM 253, 281
Адресация
  аккумуляторы, 27
  аналоговые входы, 29
  аналоговые выходы, 29
  байт:бит, 24
  битовая память, 25
  биты специальной памяти (SM), 28
  встроенные входы и выходы, 31
  косвенная (указатели), 32
  локальная память, 28
  непосредственная, 24
  области памяти, 25–28
  память переменных, 25
  память реле управления очередностью (SCR), 29
  память S7-200, 24
  расширение входов/выходов, 31
  регистр входов образа процесса, 25
  регистр выходов образа процесса, 25
  скоростные счетчики, 27
  счетчики, 26
  таймеры, 26
Адресация входов и выходов, 31
Аккумуляторы, 27
Аналоговый вход (AI)
  адресация, 29
  фильтрация, 42
Аналоговый выход (AQ), адресация, 29
Аналоговые модули, 3
  EM 231, аналоговый ввод, 354
  EM 231, термометр сопротивления (RTD), 361
  EM 231, термопара, 361
  EM 232, аналоговый вывод, 358
  EM 235 аналоговый ввод/вывод, 355
Аналоговый модуль ввода EM 231
  калибровка, 353
  конфигурирование, 354
  монтаж, 358
  схема соединений для входов, 356
  технические данные, 360
  точность и повторяемость, 359
  формат слова данных для входов, 356
Аналоговый модуль ввода/вывода EM 235
  калибровка, 353
  конфигурирование, 355
  монтаж, 358
  схема соединений для входов , 357
  схема соединений для выходов , 358
  технические данные, 360
  точность и повторяемость, 359
  формат слова данных для входов, 356
  формат слова данных для выходов, 357
Аналоговый модуль вывода EM 232
  схема соединений для выходов , 358
  формат слова данных для выходов, 357
Аналоговый потенциометр
  потенциометры, 45
  SMB28 и SMB29, 412
Аппаратура, устранение неисправностей, 241
Арифметические операции
  вычитание, 140
  деление, 140
  деление целых чисел с остатком (DIV), 142
  сложение, 140
  уменьшение на 1, 144
  умножение, 140
  умножение целого числа на двойное целое (MUL), 142
Арифметические операции с вещественными числами, пример, 141
Арифметические операции с целыми числами, пример, 141
Асинхронные обновления, команда PWM, 127

Б

Байт состояния, скоростной счетчик, 119
Байты специальной памяти, EM 277 PROFIBUS-DP, 379
Батарейный модуль, 34, 395
  номер для заказа, 422
Безопасность, пароли, 44
Безусловные входы, 55
Библиотека протокола Modbus, 329
Библиотека протокола USS, управление приводами MicroMaster, 311
Библиотеки команд, 60
Библиотеки команд STEP 7-Micro/WIN 32, 422
Битовая память, 25
Битовые логические операции
  катушки, 68
  контакты, 66
  стековые операции, 70
  функциональный блок с двумя устойчивыми состояниями, 72
  функциональный блок с двумя устойчивыми состояниями и преимуществом сброса, 72
Биты специальной памяти, 408–416
  обзор, 431
Блок данных, 50
Блокировка, скоростные счетчики, 123
Блок питания, 6, 15

Булевы операции
блок с двумя устойчивыми состояниями, 72
катушки, 68
контакты, 66
логический стек, 70

В

Варианты режимов поиска RP, 253–257
модуль позиционирования EM 253, 253–257
Вид измерения, модуль позиционирования EM 253, 246
Включение питания, сохранение памяти, 35–37
Вложенность, подпрограммы, 203
Восстановление
данных из ЭСППЗУ, 35
программы из модуля памяти, 37
Восстановление, потерянный пароль, 45
Времена входного фильтра, модуль позиционирования EM 253, 247
Времена исполнения, команды STL, 425
Времена цикла: SMW22 – SMW26, 412
Время компенсации толчков, модуль позиционирования EM 253, 250
Время оборота маркера, 228
сравнение, 230
Время, установка, 73
Время цикла (функция PTO), 125
Вставка команды, 9
Вставленные переменные, в текст и сообщения SMS, 291
Встроенные входы/выходы, 31
Входы, 22, 23
запуск и сброс (HSC), 116
калибровка, 353
модуль CPU, 343
модуль позиционирования EM 253, 244
скоростные счетчики, 113
условные и безусловные, 55
фильтрация аналоговых входов, 42
фильтрация цифровых входов, 41
цифровой модуль расширения, 348
Входы/выходы модулей расширения, 31
ошибки в шине расширения (SMW98), 417
Входы/выходы, чтение и запись, 39
Входы контура регулирования
нормализация, 149
преобразование, 149
Высоковольтное испытание изоляции, 339
Выбор
двухпозиционные переключатели термопары, 364
двухпозиционные переключатели RTD, 368–369
кабеля PC/PPI, 220
набора команд, 53
платы CP, 220
протокола обмена данными, 211
редактора программ, 51
режима работы S7-200, 37
Выходы, 22
команды без выходов, 55
модуль CPU, 343
модуль позиционирования EM 253, 244
установка состояний, 40
цифровой модуль расширения, 348
Выходы и реле, 20
Выходы и реле переменного тока, 20
Выходы и реле постоянного тока, 20
Выходы контура регулирования, преобразование в масштабированную целую величину, 150
Вычитающий счетчик, 109

Г

Генератор тактовых импульсов
биты состояния, 408

Д

Данные
передача, 80
прием, 80, 85
сохранение и восстановление, 34
Данные о мощности, модули CPU, 341
Дата, установка, 73
Двигатели, нагрузочная характеристика, типовая, 248
Двухпозиционные переключатели
термометр сопротивления (RTD), 368–369
термопара, 364
Дерево команд, 9, 51
Диагностика, самотестирование CPU, 23
Диагностическая информация, модуль позиционирования EM 253, 275
Диалоговое окно с информацией о ПЛК, 56
Диапазоны байтов и целых чисел, 24
Диапазоны операндов, 65
Диапазоны температур
EM 231 с термометром сопротивления (RTD), 371–372
EM 231 с термопарой, 366–367
Диапазоны, PID-регулятор, 150
Дизайнер TP для TP070, версия 1.0, 422
Дифференциальная составляющая, PID-алгоритм, 148
Доступ
данные S7-200, 24, 32
прямая адресация, 24
таблица состояний, 59
Доступ к слову, 24

З

Завершение по инициативе пользователя, 85
Загрузка
нового предустановленного значения в HSC, 123
нового текущего значения в HSC, 123
Загрузка программы, 11, 34
Загрузка программы из CPU, 34
Загрузка реле управления очередностью, 173
Заземление, 18, 19
Закладки, 236
Запросы на обмен данными, обработка, 23
Запуск, STEP 7-Micro/WIN, 7
Захват импульса, 41
Защита с помощью пароля, модемный модуль EM 241, 291
Защитная схема с диодом, 20
Защитные схемы, 20
Защитные схемы, выходы постоянного тока, 20

Защитный контур, 338
Значение для наибольшего числа символов, 85
Значения вещественных чисел, 24, 29
Значения таблицы профиля, генераторы PTO/PWM, 130

И

Изменение
времени цикла PTO, 134
времени цикла и количества импульсов PTO, 135
количества импульсов PTO, 135
направления скоростного счетчика, 123
предустановленного значения в HSC, 123
текущего значения в HSC, 123
ширины импульсов, 132
Изменение указателей, 32
Имитаторы входов, 398
Имитаторы, номера для заказа, 423
Импульсные выходы
команда вывода импульсов (PLS), 125
команда вывода последовательности импульсов (PTO), 125
скоростные, 46
широтно-импульсная модуляция (PWM), 125
Импульсные выходы и выходы направления, модуль позиционирования EM 253, 247
Импульсный таймер (TP), 201
Индикаторные панели
сенсорная панель TP070, 4
текстовый дисплей TD 200, 4
Инициализация
вывода PWM, 132
протокола Modbus, 330
скоростных счетчиков, 120
PTO для многосегментного режима, 135
PTO для односегментного режима, 134
Инкрементирование указателей, 32
Интегральная составляющая, PID-алгоритм, 147
Интерфейс с телефонной линией, международной, модемный модуль EM 241, 288
Интеллектуальные модули, 3
поддерживающие CPU, 303
состояние (SMB200 – SMB549), 419
Интерфейсы оператора, номер для заказа, 423
Исполнение
команды, 23
логика управления, 22
Исполнение программы, 11

К

Кабели
номера для заказа, 422
оконечная нагрузка, 220
расширение ввода/вывода, 395
сеть, 218, 219
смещение, 220
PC/PPI, 396
Кабель для присоединения компьютера, 4
Кабель расширения, 395
Кабель расширения ввода/вывода, монтаж, 395
Кабель MPI, 4
Кабель PC/PPI, 4, 6
выбор, 220
номер для заказа, 396
положения переключателей скорости передачи, 226, 397
режим свободно программируемого обмена данными (Freeport), 223
с модемом, 226
стандарт RS-232, 223
технические данные, 396
Калибровка
входов, 353
EM 231, 353
EM 235, 353
Катушки
выход, 68
непосредственная установка, 68
непосредственно управляемый выход, 68
непосредственный сброс, 68
сброс, 68
установка, 68
Клеммный блок
повторная установка, 17
снятие, 17
Коды ошибок, 404
команды для модуля позиционирования EM 253, 276
команды протокола USS, 323
модуль позиционирования EM 253, 276
нарушения правил компиляции, 406
ошибки модуля позиционирования EM 253, 277
ошибки этапа выполнения, 405
фатальные ошибки, 404
Коды ошибок исполнения, команды протокола USS, 323
Коды ошибок команд, модуль позиционирования EM 253, 276
Коды ошибок модуля, модуль позиционирования EM 253, 277
Коды стран, поддерживаемые EM 241, 288
Команда блокировки прерываний, 155
Команда, ввод, 9
Команда ввода значения в таблицу, 189
пример, 189
Команда возврата из подпрограммы, 203
пример, 205
Команда возврата из программы обработки прерывания, 155
Команда вывода импульсов (PLS), 125
Команда вывода последовательности импульсов (PTO), 46, 125
время цикла, 126
значения таблицы профилей, 130
изменение времени цикла, 134
изменение времени цикла и количества импульсов, 135
изменение количества импульсов, 135
инициализация для многосегментного режима, 135
инициализация для односегментного режима, 134
конфигурирование с помощью SM-памяти, 128
мастер управления позиционированием , 125
многосегментный конвейерный режим, 126
объяснение, 126
односегментный конвейерный режим, 126
примеры, 136, 138

управление шаговым двигателем, 130
Команда вызова подпрограммы, 203
пример, 204
Команда вычисления квадратного корня, 143
Команда вычисления косинуса, 143
Команда вычисления натурального логарифма, 143
Команда вычисления натуральной экспоненты, 143
Команда вычисления синуса, 143
Команда вычисления тангенса, 143
Команда вычитания, 140
Команда генерирования комбинации двоичных разрядов для семисегментного индикатора, 95
пример, 95
Команда групповой пересылки байтов, 167
Команда групповой пересылки двойных слов, 167
Команда групповой пересылки слов, 167
Команда декодирования, 105
пример, 105
Команда декрементирования, 144
пример, 144
Команда деления, 140
Команда деления целых чисел с остатком (DIV), 142
пример, 142
Команда дублирования вершины логического стека, 70
Команда завершения, 168
Команда завершения реле управления очередностью, 173
Команда загрузки стека, 70
Команда записи через сеть, 74, 75
коды ошибок, 75
пример, 77
Команда заполнения памяти, 192
пример, 192
Команда извлечения вершины стека, 70
Команда инвертирования байта, 162
Команда инвертирования двойного слова, 162
Команда инвертирования слова, 162
Команда инкрементирования, 144
пример, 144
Команда кодирования, 105
пример, 105
Команда копирования второго уровня стека, 70
Команда копирования подстроки из строки, 186
Команда копирования строки, 184
Команда логического сопряжения 1-го и 2-го уровня стека по ИЛИ, 70
Команда назначения прерывания, 155
Команда непосредственного присваивания, 68
Команда непосредственного сброса, 68
Команда непосредственной установки, 68
Команда обнаружения отрицательного фронта, 66
Команда обнаружения положительного фронта, 66
Команда округления, 94
Команда округления отбрасыванием, 94
Команда определения длины строки, 184
Команда определения скоростного счетчика (HDEF), 111
Команда останова, 168
пример, 169
Команда отсоединения прерывания, 155
Команда передачи, 79
передача данных, 80
пример, 86
режим свободно программируемого обмена данными (Freeport), 79
Команда перемещения значения бита в регистр сдвига, 181
пример, 182
Команда перестановки байтов, 183
пример, 183
Команда пересылки байта, 165
Команда пересылки байта для непосредственной записи, 166
Команда пересылки вещественного числа, 165
Команда пересылки двойного слова, 165
Команда пересылки непосредственно считанного байта, 166
Команда пересылки слова, 165
Команда перехода к следующему реле управления очередностью, 174
Команда перехода на метку, 172
пример, 172
Команда поиска в таблице, 193
пример, 195
Команда поиска первого появления символа внутри строки, 187
Команда поиска строки внутри строки, 187
Команда получения адреса порта, 88
Команда преобразования 16-ричного числа в ASCII, 96
Команда преобразования байта в целое, 93
Команда преобразования вещественного числа в строку, 100, 103
Команда преобразования вещественного числа в ASCII, 98
пример, 99
Команда преобразования двойного целого числа в вещественное, 93
Команда преобразования двойного целого числа в целое, 93
Команда преобразования двойного целого в строку, 100, 103
Команда преобразования двойного целого числа в ASCII, 97
Команда преобразования подстроки в вещественное число, 100, 103
Команда преобразования подстроки в двойное целое число, 100, 103
Команда преобразования подстроки в целое число, 100, 103
Команда преобразования строки символов ASCII в 16-ричное число, 96
пример, 99
Команда преобразования целого числа в строку, 100, 103
Команда преобразования целого числа в байт, 93
Команда преобразования целого числа в двойное целое, 93
Команда преобразования целого числа в ASCII
команда, 96
пример, 99
Команда преобразования целого числа в BCD, 93
Команда преобразования BCD в целое число, 93
Команда приема, 79
завершение по инициативе пользователя, 85
максимальное количество символов, 85
межсимвольный таймер, 84

обнаружение конечного символа, 84
обнаружение начального символа, 82
обнаружение паузы, 83
обнаружение простоя линии, 82
ошибки, выявленные проверкой на четность, 85
прием данных, 80
пример, 86
режим свободно программируемого обмена данными (Freeport), 79
таймер сообщений, 84
условия завершения, 82
условия начала сообщения, 82
SMB86 – SMB94, SMB186 – SMB194, 416
Команда присваивания, 68
Команда разблокирования прерываний, 155
Команда сброса, 68
пример, 68
Команда сброса контроля времени, 168
пример, 169
Команда сдвига байта влево, 179
Команда сдвига байта вправо, 179
Команда сдвига влево двойного слова, 179
Команда сдвига вправо двойного слова, 179
Команда сдвига вправо слова, 179
Команда слияния строк, 184
Команда сложения, 140
Команда сравнения байтов, 89
Команда сравнения вещественных чисел, 89
Команда сравнения двойных слов, 89
Команда сравнения строк, 91
Команда сравнения целых чисел, 89
Команда удаления первой записи из таблицы, 190
пример, 190
Команда удаления последней записи из таблицы, 190
пример, 191
Команда умножения, 140
Команда умножения целого числа на двойное целое (MUL), 142
пример, 142
Команда условного возврата из подпрограммы, 203
Команда условного возврата из прерывания, 155
Команда условного завершения, 168
пример, 169
Команда условного завершения реле управления очередностью, 173
Команда установки, 68
пример, 68
Команда установки адреса порта, 88
Команда установки часов реального времени, 73
Команда циклического сдвига байта влево, 179
Команда циклического сдвига байта вправо, 179
Команда циклического сдвига влево двойного слова, 179
Команда циклического сдвига влево слова, 179
Команда циклического сдвига вправо двойного слова, 179
Команда циклического сдвига вправо слова, 179
Команда чтения из сети, 74, 75
коды ошибок, 75
пример, 77
Команда чтения часов реального времени, 73
Команда AENO, 70
Команда MBUS_INIT, 333
Команда MBUS_SLAVE, 335
Команда MODx_CTRL, модемный модуль EM 241, 299
Команда MODx_XFR, модемный модуль EM 241, 299
Команда MODx_MSG, модемный модуль EM 241, 300
Команда Next, 170
пример, 171
Команда NOT, 66
Команда USS4_DRV_CTRL, 316
Команда USS4_INIT, 315
Команда USS4_RPM_x, 319, 320
Команда USS4_WPM_x, 319, 320
Командный байт, модуль позиционирования EM 253, 283
Команды
без выходов, 55
битовая логика, 66
ввод значения в таблицу, 189
блокирование прерываний, 155
возврат из подпрограммы, 203
возврат из программы обработки прерываний, 155
вывод импульсов (PLS), 125
вывод последовательности импульсов (PTO), 125
вызов подпрограммы, 203
вычисление квадратного корня, 143
вычисление косинуса, 143
вычисление синуса, 143
вычисление натурального логарифма, 143
вычисление натуральной экспоненты, 143
вычисление тангенса, 143
вычитание, 140
вычитающий счетчик, 109
групповая пересылка байтов, 167
групповая пересылка двойных слов, 167
групповая пересылка слов, 167
декодирование, 105
деление, 140
деление целых чисел с остатком (DIV), 142
дублирование вершины логического стека, 70
завершение, 168
завершение реле управления очередностью, 173
загрузка реле управления очередностью, 173
загрузка стека, 70
запись через сеть, 74
заполнение памяти, 192
извлечение вершины стека, 70
импульсный таймер (TP), 201
инвертирование байта, 162
инвертирование двойного слова, 162
инвертирование слова, 162
исполнение, 23
кодирование, 105
копирование второго уровня стека, 70
копирование подстроки из строки, 186
копирование строки, 184
логическое сопряжение 1-го и 2-го уровня стека по И, 70
логическое сопряжение 1-го и 2-го уровня стека по ИЛИ, 70
метка, 172

Команды (продолжение)
модемный модуль EM 241, 298
модуль позиционирования EM 253, 257
назначение прерывания, 155
непосредственная установка, 68
непосредственное присваивание, 68
непосредственно управляемые контакты, 66
непосредственный сброс, 68
обратный счет, 106
округление, 94
округление отбрасыванием, 94
определение длины строки, 184
определение скоростного счетчика (HDEF), 111
отрицательный фронт, 66
отсоединение прерывания, 155
передача, 79
перемещение значения бита в регистр сдвига, 181
перестановка байтов, 183
пересылка байта, 165
пересылка байта для непосредственной записи, 166
пересылка вещественного числа, 165
пересылка двойного слова, 165
пересылка непосредственно считанного байта, 166
пересылка слова, 165
переход на метку, 172
поиск в таблице, 193
поиск первого появления символа в строке, 187
поиск строки в строке, 187
положительный фронт, 66
получение адреса порта, 88
поразрядное исключающее ИЛИ, 163
поразрядное логическое И, 163
поразрядное логическое ИЛИ, 163
преобразование 16-ричного числа в ASCII, 96
преобразование байта в целое число, 93
преобразование вещественного числа в строку, 100, 103
преобразование вещественного числа в ASCII, 98
преобразование двойного целого числа в вещественное, 93
преобразование двойного целого числа в строку ASCII, 97
преобразование двойного целого числа в строку, 100, 103
преобразование двойного целого числа в целое, 93
преобразование подстроки в вещественное число, 100, 103
преобразование подстроки в двойное целое число, 100, 103
преобразование подстроки в целое число, 100, 103
преобразование строки символов ASCII в 16-ричное число, 96
преобразование целого числа в байт, 93
преобразование целого числа в двойное целое, 93
преобразование целого числа в строку, 100, 103
преобразование целого числа в ASCII, 96
преобразование целого числа в BCD, 93

Команды (продолжение)
преобразование BCD в целое число, 93
прерывание, 155–161
прием, 79
присваивание, 68
протокол Modbus Slave, 332
протокол USS, 314
прямой счет, 106
разблокирование прерываний, 155
реверсивный счетчик, 106, 109
сброс, 68
сброс контроля времени, 168
сдвиг байта влево, 179
сдвиг байта вправо, 179
сдвиг влево двойного слова, 179
сдвиг влево слова, 179
сдвиг вправо двойного слова, 179
сдвиг вправо слова, 179
семисегментный индикатор, 95
скоростной импульсный вывод (PLS), 125
скоростной счетчик (HSC), 111
сложение, 140
соединение строк, 184
создание команд позиционирования, 286
сравнение, 55
стандартный контакт, 66
суммирующий счетчик, 109
таблица, 190–195
таймер с задержкой включения (TON), 196, 201
таймер с задержкой выключения (TOF), 196, 201
таймер с задержкой включения с запоминанием (TONR), 196
триггер с преимуществом сброса, 72
триггер с преимуществом установки, 72
увеличение на 1, 144
удаление первой записи из таблицы, 190
удаление последней записи из таблицы, 190
уменьшение на 1, 144
умножение, 140
умножение целого числа на двойное целое (MUL), 142
управление с обратной связью (PID), 145
условное завершение, 168
условное завершение реле управления очередностью, 173
условный возврат из подпрограммы, 203
условный возврат из программы обработки прерываний, 155
установка, 68
установка адреса порта, 88
установка часов реального времени, 73
циклический сдвиг байта влево, 179
циклический сдвиг байта вправо, 179
циклический сдвиг влево двойного слова, 179
циклический сдвиг влево слова, 179
циклический сдвиг вправо двойного слова, 179
циклический сдвиг вправо слова, 179
часы реального времени, 73
чтение из сети, 74
чтение часов реального времени, 73
широтно-импульсная модуляция (PWM), 125
AENO, 70
for, 170
MBUS_INIT, 333

Команды (продолжение)
MBUS_SLAVE, 335
MODx_CTRL, 299
MODx_MSG, 300
MODx_XFR, 299
next, 170
NOT, 66
PID-регулятор, 145
POSx_CFG, 268
POSx_CLR, 267
POSx_CTRL, 258
POSx_DIS, 266
POSx_GOTO, 260
POSx_LDOFF, 263
POSx_LDPOS, 264
POSx_MAN, 259
POSx_RSEEK, 262
POSx_RUN, 261
POSx_SRATE, 265
STOP, 168
Команды групповой пересылки, пример, 167
Команды для подпрограмм
вызов подпрограммы, 203
пример, 205
условный возврат из подпрограммы, 203
Команды для часов реального времени (TOD), 73
Команды инвертирования, пример, 162
Команды, обзор, 435
Команды обнаружения фронта, 66
Команды пересылки, пример, 183
Команды преобразования
16-ричное число в ASCII, 96
7-сегментный индикатор, 95
декодирование, 105
кодирование, 105
округление, 94
округление отбрасыванием, 94
пример, 94
стандартные, 92
строка, 100, 103
ASCII, 96
ASCII в 16-ричное число, 96
Команды, относящиеся к часам реального времени, 73
Команды преобразования символов ASCII, 96
Команды прерывания
блокирование прерываний, 155
назначение прерывания, 155
отсоединение прерывания, 155
пример, 161
разблокирование прерываний, 155
условный возврат из программы обработки прерываний, 155
Команды перехода
метка, 172
переход на метку, 172
Команды позиционирования, модуль позиционирования EM 253, 284
Команды протокола USS
коды ошибок исполнения, 323
пример программы, 322
указания по использованию, 314
USS4_DRV_CTRL, 316
USS4_INIT, 315
USS4_RPM_x и USS4_WPM_x, 319, 320
Команды работы с часами
установка часов реального времени, 73
чтение часов реального времени, 73
Команды реле управления очередностью
завершение реле управления очередностью, 173
загрузка реле управления очередностью, 173
ограничения, 173
переход к следующему реле управления очередностью, 174
пример, 173
разделение потоков управления, 176
слияние потоков управления, 176
условное завершение реле управления очередностью, 173
Команды сдвига
виды, 179
пример, 180
Команды сравнения, 55
пример, 89
сравнение байтов, 89
сравнение вещественных чисел, 89
сравнение двойных слов, 89
сравнение строк, 91
сравнение целых чисел, 89
Команды счетчиков IEC
вычитающий счетчик, 109
пример, 109
реверсивный счетчик, 109
суммирующий счетчик, 109
Команды счетчиков SIMATIC
обратный счет, 106
примеры, 108
прямой и обратный счет, 106
прямой счет, 106
Команды таймеров IEC, 201
пример, 201
Команды управления по замкнутому контуру (PID), 145–156
выбор типа регулятора, 148
диапазоны и переменные, 150
корректировка смещения, 150
ошибочные состояния, 151
положительная и отрицательная обратная связь, 150
преобразование входов, 149
преобразование выходов, 150
пример программы, 152–154
режимы, 151
Команды управления программой
команды перехода, 172
основы управления программой, 168
останов, 168
реле управления очередностью(SCR), 173
сброс контроля времени, 168
условное завершение, 168
цикл for-next, 170
Команды цикла For-next
пример, 171
for, 170
next, 170
Команды циклического сдвига, 179
виды, 179
пример, 180

Команды STL
времена исполнения, 425
обзор, 435
Коммуникационные команды
запись через сеть, 74
передача, 79
получение адреса порта, 88
прием, 79
установка адреса порта, 88
чтение из сети, 74
Коммуникационные платы, номера для заказа, 422
Коммуникационные процессоры. *См.* Платы CP
Коммуникационные процессоры, CP 243-2, 393
Коммуникационный порт
назначение контактов разъема, 219
прерывания, 158
режим свободно программируемого обмена данными, 222
Коммуникационный процессор CP 243-2
номер для заказа, 393
режим, 394
соединительные элементы, 394
функции, 394
характеристики, 393
Коммутационные схемы
аналоговые модули расширения, 352
входы и выходы CPU, 343
модули CPU, 343–345
модуль позиционирования EM 253, 391–392
цифровые модули расширения, 349–350
CPU 221 AC/DC/Relay, 343
CPU 221 DC/DC/DC, 343
CPU 222 AC/DC/Relay, 344
CPU 222 DC/DC/DC, 344
CPU 224 AC/DC/Relay, 344
CPU 224 DC/DC/DC, 344
CPU 226 AC/DC/Relay, 345
CPU 226 DC/DC/DC, 345
CPU 226XM AC/DC/Relay, 345
CPU 226XM DC/DC/DC, 345
EM 221 DI 8 x AC, 349
EM 221 DI 8x24 VDC, 349
EM 222 DO 8 x Relay, 349
EM 223 DI 16/DO 16 x 24 VDC Relay, 350
EM 222 DO 8x24 VDC, 349
EM 223 4x24 VDC In/4x24 VDC Out, 349
EM 223 DI 16/DO 16 x DC 24V, 350
EM 223 DI 4/DO 4 x DC 24V/Relay, 349
EM 223 DI 8/ DO 8 x 24 VDC/Relay, 350
EM 223 DI 8/DO 8 x 24 VDC, 350
EM 231 AI 4 x 12 Bit, 352
EM 231 с термометром сопротивления (RTD), 362
EM 231 с термопарой (thermocouple), 362
EM 232 AQ 2 x 12 Bit, 352
EM 235 AI 4/AQ 1 x 12 Bit, 352
Конвейерная обработка, импульсы PTO, 126
Контакты, 66
пример, 67
Контакты разъема, назначения для коммуникационного порта, 219
Контур регулирования с отрицательной обратной связью, 150
Контур регулирования с положительной обратной связью, 150
Конфигурации входов, редактирование конфигураций, установленных по умолчанию в модуле позиционирования EM 253, 246
Конфигурации выходов, редактирование конфигураций, установленных по умолчанию в модуле позиционирования, 247
Конфигурационная таблица, модемный модуль EM 241, 293, 306
Конфигурирование
модуля позиционирования EM 253, 275
отображения состояния FBD и LAD, 238
отображения состояния STL, 239
пароля, 44
сети, 218
состояния выходов, 40
сохраняемые области памяти, 41
таблица символов для Modbus, 331
функций PTO/PWM, 128
EM 231, 354
EM 231 с термометром сопротивления (RTD), 368
EM 231 с термопарой (thermocouple), 363
EM 235, 355
EM 277 PROFIBUS-DP, 376–377
Конфликты, устранение в сети, 231
Копирование программы в модуль памяти, 36
Косвенная адресация, 32
& и *, 32
изменение указателей, 32
Коэффициент актуализации пропусков (GUF), 228

Л

Логика, управление, 22
Логика управления, 22
Логические связи
MPI, 212
PPI, 211
Логические операции
инвертирование, 162
поразрядные логические операции И, ИЛИ, исключающее ИЛИ, 163
Логическое сопряжение 1-го и 2-го уровня стека по И, 70
Локальная память, 28
Люфт
выбор рабочей области, 256
компенсация, 251

М

Мастера
конфигурирования модемного модуля, 294
управления перемещением, 246
PID, 145
Мастер конфигурирования модемного модуля, модемный модуль EM 241, 294
Мастер управления позиционированием , 246
PTO/PWM, 125
Масштабированная целая величина, преобразование выходов контура управления, 150
Международные стандарты, 338
Межсимвольный таймер, 84
Метка, 172

Многосегментный режим, инициализация PTO, 135
Модемный модуль, 385
защита паролем, 291
интерфейс с международными телефонными линиями, 288
команда MODx_CTRL, 299
команда MODx_MSG, 300
команда MODx_XFR, 299
команды, 298
конфигурационная таблица, 293
мастер конфигурирования модемного модуля, 294
обратный вызов для обеспечения безопасности, 292
ошибки, возвращаемые командами, 301
пейджинговая связь, 290
передача данных, 291
пример, 303
передача коротких сообщений (SMS), 290
светодиоды состояния, 294
служба коротких сообщений, 290
текстовая пейджинговая связь, 290
телефонный номер для передачи сообщений , 308
формат сообщения для передачи данных CPU, 310
формат текстовых сообщений, 309
характеристики, 288
цифровая пейджинговая связь, 290
штекер RJ11, 288
Модемный модуль EM 241, 385
адреса специальной памяти, 304
адреса Modbus, 290
защита паролем, 291
интерфейс с международными телефонными линиями, 288
интерфейс со STEP 7-Micro/WIN, 289
команда MOD_XFR, 299
команда MODx_CTRL, 299
команда MODx_MSG, 300
команды, 298
конфигурационная таблица, 293, 306
мастер конфигурирования модемного модуля, 294
монтаж, 386
обратный вызов для обеспечения безопасности, 292
ошибки, возвращаемые командами, 301
пейджинговая связь, 290
передача данных, 291
поддерживаемые страны, 288
пример, 303
протокол Modbus RTU, 289
светодиоды состояния, 294
Служба коротких сообщений (SMS), 290
текстовая пейджинговая связь, 290
телефонный номер для передачи сообщений , 308
технические данные, 385
формат сообщения для передачи данных CPU, 310
характеристики, 288
цифровая пейджинговая связь, 290
штекер RJ11, 288
Модем, с кабелем PC/PPI, 226
Модули интерфейса с исполнительными устройствами и датчиками (AS-interface), номера для заказа руководств, 422
Модули расширения, 3
адресация входов и выходов, 31
монтаж, 16
номера для заказа, 421
потребности в мощности, 15, 399
размеры, 16
регистры идентификации и ошибок, 411
снятие, 17
Модули расширения (EM)
аналоговые
номера для заказа, 351, 361
схемы подключения, 352
технические данные, 351
технические данные входов, 351
технические данные выходов, 352
цифровые
номера для заказа, 346
общие технические данные, 346
схемы подключения, 349–350
технические данные входов, 346
технические данные выходов, 347
Модули CP242, CP243, номера для заказа руководств, 422
Модули CPU
данные о мощности, 341
коммутационные схемы, 343–345
номера для заказа, 340, 421
области памяти, 64, 433, 461
самотестирование, 23
снятие, 17
технические данные, 340
технические данные входов, 341
технические данные выходов, 342
установка, 16
характеристики, 64, 461
Модуль памяти, 34, 36, 395
восстановление программы, 37
коды ошибок, 404
копирование программы, 36
номер для заказа, 422
установка, 36
Модуль позиционирования
вид измерения, выбор, 246
времена входного фильтра, 247
время компенсации толчков, 250
входы и выходы, 244
диагностическая информация, 275
импульсные выходы и выходы направления, 247
коды ошибок, 276
коды ошибок команд, 276
коды ошибок модуля, 277
команды, 257
команды позиционирования, 284
конфигурации входов и выходов , 247
конфигурация, 275
конфигурирование, 246
мастер управления позиционированием , 246
опорная точка, 250
отображение и управление работой, 274
отрицательная полярность, 247
панель управления EM 253, 274–276

Модуль позиционирования (продолжение)
параметры стартстопного режима, 249
поиск опорной точки, 250
положительная полярность, 247
последовательность поиска опорной точки, 251
пример программы для управления, 286
программирование, 245
профили, 252
профиль перемещения, определение, 252
режим работы для профиля, 252
режимы поиска опорной точки, 253–257
создание команд, 286
специальная память, 281
таблица конфигурации и профилей, 278
указания по использованию команд , 257
уровни активизации входов, 247
устранение люфта, 256
характеристики, 244
шаги профиля, 253
ACCEL_TIME, 249
DECEL_TIME, 249
MAX_SPEED, 248
POSx_CFG, 268
POSx_CLR, 267
POSx_CTRL, 258
POSx_DIS, 266
POSx_GOTO, 260
POSx_LDOFF, 263
POSx_LDPOS, 264
POSx_MAN, 259
POSx_RSEEK, 262
POSx_RUN, 261
POSx_SRATE, 265
SS_SPEED, 248
Модуль позиционирования EM 253
вид измерения, выбор, 246
времена входного фильтра, 247
время компенсации толчков, 250
входы и выходы, 244
диагностическая информация, 275
импульсные выходы и выходы направления, 247
коды ошибок, 276
коды ошибок команд, 276
коды ошибок модуля, 277
командный байт, 283
команды, 257
команды позиционирования, 284
конфигурация, 275
конфигурирование, 246
максимальная скорость и скорость при пуске и останове, 248
мастер управления позиционированием , 246
опорная точка (RP), 250
отображение и управление работой, 274
отрицательная полярность, 247
панель управления EM 253, 274–276
параметры стартстопного режима, 249
поиск опорной точки (поиск RS), 250
положительная полярность, 247
последовательность поиска опорной точки, 251
пример программы управления модулем, 286
примеры, 269–273
программирование, 245
профили, 252
Модуль позиционирования EM 253 (продолжение)
профиль перемещения, определение, 252
реакция на физические входы, 248
режим работы для профиля, 252
режимы поиска опорной точки, 253–257
создание команд, 286
специальная память, 281
схема подключения, 391–392
таблица конфигурации и профилей, 278
указания по использованию команд , 257
уровни активизации входов, 246
устранение люфта, 256
характеристики, 244
шаги профиля, 253
ACCEL_TIME, 249
DECEL_TIME, 249
POSx_CFG, 268
POSx_CLR, 267
POSx_CTRL, 258
POSx_DIS, 266
POSx_GOTO, 260
POSx_LDOFF, 263
POSx_LDPOS, 264
POSx_MAN, 259
POSx_RSEEK, 262
POSx_RUN, 261
POSx_SRATE, 265
SS_SPEED, 248
Модуль с термометром сопротивления (RTD) (EM 231), 368
Модуль с термопарой (EM 231)
диапазоны температур, 366–367
индикаторы состояния, 365
конфигурирование, 363
основы, 363
Модуль EM 231 для термометра сопротивления (RTD)
диапазоны температуры и точность, 371–372
индикаторы состояния, 370
клеммы, 362
конфигурирование, 368
совместимость с CPU, 362
технические данные, 361
установка двухпозиционных переключателей, 368–369
Модуль EM 231 для термопары
диапазоны температуры и точность, 366–367
индикаторы состояния, 365
клеммы, 362
конфигурирование, 363
основы, 363
совместимость с CPU, 362
технические данные, 361
установка двухпозиционных переключателей, 364
Модуль EM 277 PROFIBUS-DP
байты специальной памяти, 379
в сети PROFIBUS, 375
дополнительные характеристики, 380
как slave-устройство DP, 375
конфигурирование, 376–377
конфигурационные возможности, 377
переключатели адресов, 374
протокол DP, 375
режим обмена данными, 378

светодиодные индикаторы состояния, 380
светодиоды состояния, 374
совместимость с CPU, 374
технические данные, 373
файл базы данных устройства, 381–382
Монтаж
блок питания, 15
высоковольтные установки, 14
кабель расширения ввода/вывода, 395
модуль памяти, 36
модуль расширения, 16
модуль CPU, 16
нагревательные установки, 14
панель, 16
стандартная профильная шина, 16
требования к зазорам, 14, 16
требования к монтажу, 16
указания, 14
электрические помехи, 14
EM 231, 358
EM 235, 358
S7-200, 15
Монтаж на панели, 16
Монтаж на стандартной профильной шине, 16
Морское агентство Германский Ллойд (GL), 338
Морское агентство American Bureau of Shipping (ABS), 338
Морское агентство Bureau Veritas (BV), 338
Морское агентство Det Norske Veritas (DNV), 338
Морское агентство Lloyds Register of Shipping (LRS), 338
Морское агентство Nippon Kaiji Kyokai (NK), 338

Н

Набор команд IEC 1131-3, 53
Набор команд SIMATIC, 53
Наборы команд
выбор, 53
IEC 1131-3, 53
SIMATIC, 53
Наблюдение, 11
переменных процесса, 59
переменных с помощью таблицы состояний, 239
состояния программы, 238
Навигационная панель, 51
Нагрузочная характеристика, типовая для двигателей, 248
Назначение
адресов, 58
начальных значений, 58
прерываний для HSC, 119
сетевых адресов, 208
Назначение контактов, коммуникационный порт, 219
Наибольший адрес станции (HSA), 228
Направление, изменение в HSC, 123
Нарушение правил компиляции, 406
Настройка
привода MM3, 325
привода MM4, 328
Настройка обмена данными, STEP 7-Micro/WIN, 7
Национальные стандарты, 338
Начальные значения, присваивание, 58
Непосредственно управляемые контакты, 66
Нефатальные ошибки, 56
Номера для заказа, 421
аналоговый модуль расширения, 351, 361
имитаторы входов, 398
кабель PC/PPI, 396
модули CPU, 340
цифровой модуль расширения, 346
CP 243-2, 393
Нормализация, входы контура регулирования, 149

О

Обзорная информация, 431
Области памяти
диапазоны операндов, 65
модули CPU, 64, 461
обзор, 433
очистка, 45
Область битовой памяти (M), 25
сохранение при потере питания, 35
Область памяти для входов (I), 25
Область памяти локальных данных (L), 28
Область памяти переменных (V), 25
Область памяти реле управления очередностью (S), 29
Область памяти счетчиков (C), 26
Область памяти таймеров (T), 26
Область специальной памяти (SM), 28
аналоговые потенциометры, 45
Обмен данными, 208
конфликты, 231
модем, 226–227
поддерживаемые протоколы, 211
протокол Modbus Slave, 330
скорость передачи, 226
с приводами MicroMaster, 312
S7-200, 7
Обмен данными при двухточечном соединении, 213–214
Обмен данными с приводом, расчет необходимого времени, 313
Обмен данными через PPI, переход к режиму свободно программируемого обмена данными (Freeport), 80
Обнаружение конечного символа, 84
Обнаружение начального символа, 82
Обнаружение паузы, 83
Обновление характеристик импульсов PWM, 127
Обработка, запросы на обмен данными, 23
Обработка коммуникационных задач, 40
Обработка ошибок, 56
Обратная установка клеммного блока, 17
Обратный вызов для обеспечения безопасности, модемный модуль EM 241, 292
Ограничение доступа, 44
Ограничение пикового напряжения, 20
Односегментный режим
изменение времени цикла PTO, 134
изменение времени цикла и количества импульсов PTO, 135
изменение количества импульсов PTO, 135
инициализация PTO, 134
Оконечное сопротивление, сетевой кабель, 220

Опорная точка (RP), 250
поиск RP
направление поиска, 250
RP_APPR_DIR, 250
RP_FAST, 250
RP_SEEK_DIR, 250
RP_SLOW, 250
последовательность поиска RP, режимы, 251
RP_OFFSET, 251
Определение профиля перемещения, модуль позиционирования EM 253, 252
Оптимизация, производительность сети, 228
Отладка
исполнение программы в течение определенного количества циклов, 240
принудительно присваиваемые значения, 240
редактирование в режиме RUN, 236
характеристика, 236
Отладка программного обеспечения, 235
Отображение
состояния программы, 238
элементов программы, 51
Отображение адресов в S7-200, Modbus, 331
Отрицательная полярность, модуль позиционирования EM 253, 247
Очереди, программы обработки прерываний, 159
Очередь коммуникационных прерываний, 159
Очередь прерываний по вводу/выводу, 159
Очередь прерываний, управляемых временем, 159
Ошибки
ввода/вывода, 57
исполнения программы, 57
исполнения протокола Modbus Slave, 335
команд модемного модуля EM 241, 301
команд чтения из сети и записи через сеть, 75
компиляции программы, 56
конфигурирования модемного модуля EM 241, 293
нефатальные, 56
обработка, 56
просмотр, 56
фатальные, 57
этапа выполнения, 57
PID-регулятор, 151
SMB1, ошибки исполнения, 408
Ошибки ввода/вывода, 57
Ошибки, выявленные проверкой на четность
SMB3, 409
SMB30 и SMB130, 85
Ошибки исполнения, 57
Ошибки компиляции, 56
Ошибки этапа исполнения, 57

П

Память
восстановление из ЭСППЗУ, 35
доступ, 24
сохраняемая, 41
CPU, 2
Память переменных (V), 25
восстановление из ЭСППЗУ, 35
назначение адресов, 58
сохранение в ЭСППЗУ, 38
Память реле управления очередностью (S), 29
Память таймеров (T), 26
Панели инструментов, 51
Панель управления EM 253, 274–276
Параметры
виды для подпрограмм, 204
в подпрограммах, 203
Параметры поиска, модуль позиционирования EM 253, 250
Параметры стартстопного режима
модуль позиционирования EM 253, 249
стартстопный режим, 249
Пароль
восстановление потерянного, 45
ограничение доступа, 45
очистка, 45
установка, 44
функции CPU, 44
Пейджинговая связь, модемный модуль, 290
Передача данных, модемный модуль EM 241, 291
Перезапуск, после фатальной ошибки, 57
Переключатели адресов, EM 277 PROFIBUS-DP, 374
Переключатель режимов работы, 37
Переменные
наблюдение, 59
наблюдение с помощью таблицы состояний, 239
символическая адресация, 58
PID-регулятор, 150
Переполнение очереди (SMB4), 409
Платы CP
выбор, 220
номера для заказа, 422
Повторители
номера для заказа, 422
сеть, 218
Подключение, 18, 19
Подключение
кабеля PC/PPI, 6
модема к S7-200, 226
привода MM3, 324
привода MM4, 327
сетевых устройств, 230
Подпрограммы
вложенность, 203
вызов из программы обработки прерываний, 158
поток сигнала, 204
пример, 49
рекомендации, 50
с параметрами, 203
типы данных, 204
типы параметров, 204
Положительная полярность, модуль позиционирования EM 253, 247
Пользовательские библиотеки, 60
Поразрядное логическое И, 163
пример, 164
Поразрядное логическое ИЛИ, 163
пример, 164
Поразрядное логическое исключающее ИЛИ, 163
пример, 164
Порядок прерывающих событий, 160
Постоянная память для хранения программ, 38
Постоянные значения, 30

Построение сети, 218
Потенциометры, аналоговый потенциометр, 45
Потеря пароля, 45
Потеря питания, сохраняемая память, 35, 41
Поток сигнала, параметр подпрограммы, 204
Потребности в мощности, 15
  модуль расширения, 399
  пример, 400
  расчет, 399, 401
  таблица для расчета, 401
  CPU, 399
Предустановленное значение
  изменение в HSC, 123
  установка для HSC, 118
Преобразование
  входов контура регулирования, 149
  входов контура регулирования в масштабированную целую величину, 150
Преобразования
  программирование S7-200, 55
  редакторы программ, 54
Прерывания
  назначение HSC, 119
  скоростные счетчики, 113
Прерывания по вводу/выводу, 158
Прерывания, управляемые временем, 158
Прерывания, управляемые временем, регистры интервалов времени (SMB34, SMB35), 413
Прерывающие события
  виды, 156
  обзор, 432
  приоритет, 160
Приводы. *См.* Приводы MicroMaster
Привод MicroMaster
  обмен данными, 312
  подключение, 324
  управление, 311
  чтение и запись, 319, 320
Привод MM3
  настройка, 325
  подключение, 324
Привод MM4
  настройка, 328
  подключение, 327
Прием данных, 85
Прием данных, управляемый прерываниями, 85
Пример программы, 8
Примеры
  арифметические команды с целыми числами, 141
  ввод значения в таблицу, 189
  вызов подпрограммы, 204
  генерирование комбинации двоичных разрядов для семисегментного индикатора, 95
  команда групповой пересылки, 167
  команда декрементирования, 144
  команда заполнения памяти, 192
  команда инкрементирования, 144
  команда обратного счета SIMATIC, 107
  команда останова, 169
  команда перемещения, 183
  команда перемещения значения бита в регистр сдвига, 182
  команда перестановки байтов, 183
  команда перехода на метку, 172
Примеры (продолжение)
  команда поиска в таблице, 195
  команда преобразования вещественного числа в ASCII, 99
  команда преобразования целого числа в ASCII, 99
  команда сброса, 68
  команда сброса контроля времени, 169
  команда удаления первой записи из таблицы, 190
  команда удаления последней записи, 191
  команда установки, 68
  команда DIV, 142
  команда MUL, 142
  команды возврата из подпрограммы, 205
  команды декодирования, 105
  команды инвертирования, 162
  команды кодирования, 105
  команды передачи, 86
  команды подпрограммы, 205
  команды прерываний, 161
  команды приема, 86
  команды реле управления очередностью, 173
  команды сдвига, 180
  команды скоростного счетчика, 124
  команды сравнения, 89
  команды счетчиков IEC, 109
  команды цикла for-next, 171
  команды циклического сдвига, 180
  команды чтения из сети и записи через сеть, 77
  контакты, 67
  многосегментная последовательность импульсов PTO, 138
  модемный модуль EM 241, 303
  модуль позиционирования, 269–273
  односегментная последовательность импульсов PTO, 136
  подпрограмма, 49
  поразрядное логическое И, 164
  поразрядное логическое ИЛИ, 164
  поразрядное логическое исключающее ИЛИ, 164
  преобразование ASCII в 16-ричное число, 99
  программа обработки прерывания, управляемого временем, 161
  программа с командами протокола USS, 322
  программа с PID-регулятором, 152
  программы обработки прерываний, 49
  протокол Modbus Slave, программирование, 336
  разделение потоков управления, 176
  расчет потребностей в мощности, 399
  реверсивный счетчик SIMATIC, 108
  режим свободно программируемого обмена данными, 222
  режимы скоростного счетчика, 114
  сеть с передачей маркера, 229
  слияние потоков управления, 177
  стандартные команды преобразования, 94
  стековые операции, 71
  счетчики SIMATIC, 108
  таймер с задержкой включения, 198
  таймер с задержкой включения с запоминанием, 200
  таймер с задержкой выключения, 199

Примеры (продолжение)
таймеры IEC, 201
таймеры SIMATIC, 198, 199, 200
условное завершение, 169
условные переходы, 178
широтно-импульсная модуляция (PWM), 132
PID-регулятор, 153
Принудительно присваиваемые значения, 239, 240
Приоритет
прерывающих событий, 160
программ обработки прерываний, 159
Программа
аналоговые входы, 22
восстановление из модуля памяти, 37
загрузка в CPU, 11, 34
загрузка из CPU, 34
задание количества циклов, 240
исполнение, 11
копирование в модуль памяти, 36
наблюдение, 11
основные элементы, 49
подпрограммы, 50
отображение состояния, 238
ошибки исполнения, 57
ошибки компиляции, 56
пример PID, 152
редактирование в режиме RUN, 236
совместное использование данных с программами обработки прерываний, 157
создание, 8
создание с помощью STEP 7-Micro/WIN, 51
сохранение, 11
сохранение в ЭСППЗУ, 34–38
структурирование, 49
таблица состояний, 59
характеристики отладки, 236
Программа обработки прерываний, управляемых временем, пример, 161
Программа, пример, управление модулем позиционирования, 286
Программирование
модуля позиционирования EM 253, 245
скоростных счетчиков, 112
Программное обеспечение для программирования, номера для заказа, 422
Программы обработки прерываний, 23, 39
виды, поддерживаемые S7-200, 158
вызов подпрограммы из, 158
коммуникационный порт, 158
нарастающий/падающий фронт, 158
объяснение, 157
очереди, 159
по вводу/выводу, 158
пример, 49
приоритет, 159
рекомендации, 50, 157
системная поддержка, 157
совместное использование данных с основной программой, 157
управляемые временем, 158
Проектирование
системы с микроконтроллером, 48
схем защиты, 48
Проектирование системы, микроконтроллер, 48
Производительность, оптимизация сети, 228
Пропорциональная составляющая, PID-алгоритм, 147
Просмотр ошибок, 56
Протокол, обмен данными
выбор, 211
определяемый пользователем, 222
Протокол обмена данными
выбор, 211
двухточечный интерфейс (PPI), 211, 230
многоточечный интерфейс (MPI), 212, 230
определяемый пользователем, 222
PROFIBUS, 212, 230
Протокол, определяемый пользователем, режим свободно программируемого обмена данными (Freeport), 222
Протоколы, PROFIBUS-DP, 374
Протокол Modbus RTU, 332
модемный модуль EM 241, 289
отображение адресов, 290
функции, поддерживаемые модемным модулем, 289
Протокол Modbus Slave
адреса, 331
время исполнения, 330
инициализация, 330
используемые ресурсы, 330
коды ошибок исполнения, 335
команды, 332
конфигурирование таблицы символов, 331
пример программирования, 336
поддерживаемые функции, 332
специальная память, 330
отображение адресов в S7-200, 331
таблица CRC, 330
MBUS_INIT, 333
MBUS_SLAVE, 335
Протокол MPI, 212, 230
Протокол PPI, 211, 230
сеть с несколькими master-устройствами, 213
сеть с одним master-устройством, 213
сложная сеть, 213
Протокол PROFIBUS, 212, 230
Протокол USS, требования, 312
Профильная шина
монтаж, 16
размеры, 16
Профиль перемещения, определение для модуля позиционирования EM 253, 252
Пуск, скоростной счетчик, 116

Р

Работа со сложными сетями, 231
Работа с редактором программ, 9
Развязка
сеть, 218
указания по подключению, 18
Разделение потока управления
пример, 176
команды реле управления очередностью, 176
Размеры, CPU, 2
Разрешающая способность таймера, 197, 198
Разрешающий вход (EN), 55
Разрешающий выход (ENO), 55
Распознавание бездействия линии, 82

Расположение рабочей области, модуль позиционирования EM 253, 256
Расчет
времени оборота маркера, 228
потребностей в мощности, 399–401
Реверсивный счетчик, 106, 109
пример, 108
Регистр входов образа процесса, 23, 25
Регистр выходов образа процесса, 22, 25
Регистр образа процесса, 39
Регистры идентификации и ошибок модулей ввода/вывода, 411
Регистры управления свободно программируемым обменом данными (SMB30 и SMB130), 412
Регистры HSC0, HSC1, HSC2 (SMB36 – SMB65), 413
Регистры HSC3, HSC4, HSC5 (SMB131 – SMB165), 417
Редактирование, в режиме RUN, 236
Редактор блоков данных
назначение адресов, 58
присваивание начальных значений, 58
Редакторы
список команд (STL), 51
функциональный план (FBD), 52
цепная логическая схема (LAD), 52
Редакторы программ, 51
выбор, 51
открытие, 9
соглашения, 54
список команд (STL), 51
функциональный план (FBD), 51
цепная логика (контактный план) (LAD), 51
Редактор FBD
описание, 52
соглашения, 54
характеристики, 52
Редактор LAD
описание, 52
соглашения, 54
характеристики, 52
Редактор STL
описание, 51
характеристики, 51
Режим обмена данными, EM 277 PROFIBUS-DP, 378
Режим работы для профиля, модуль позиционирования EM 253, 252
Режим свободно программируемого обмена данными
активизация, 79
команды передачи и приема, 79
определение, 158
ошибка, выявленная проверкой на четность (SMB3), 409
переход из PPI, 80
прием данных, управляемый прерываниями, 85
принимаемый символ (SMB2), 409
пример, 222
протоколы, определяемые пользователем, 222
стандарт RS-232, 223
Режимы
скоростных счетчиков, 113
PID-регулятора, 151
Режимы работы, CPU
биты состояния, 408
изменение, 37
протокол свободно программируемого обмена данными (Freeport), 222
Режим RUN, 11, 37
редактирование программы, 236
Рекомендации
по проектированию системы с микроконтроллером, 48
Реле, 20
срок службы, 338
Руководства, номера для заказа, 422
Ручной режим, PID-регулятор, 151

С

Сброс, скоростной счетчик, 116
Светодиоды, модемный модуль EM 241, 294
Светодиоды состояний
модуль позиционирования EM 253, 389
EM 231 с термометром сопротивления (RTD), 370
EM 231 с термопарой, 365
EM 277 PROFIBUS-DP, 374, 380
Свободно программируемый обмен данными, 409
SMB30 и SMB130, 412
Свойства функции захвата импульса, 42
Семисегментный индикатор, 95
Сенсорная панель TP070, 4
номера для заказа, 423
номер для заказа руководства, 422
Сертификаты морских агентств, 338
Сертификаты агентств, 338
Сетевые соединители, номера для заказа, 422
Сеть
адреса, 208
адреса устройств, 211
время оборота маркера, 228
кабель, 218, 219
кабель PC/PCI, 220
коммуникационный порт, 219
компоненты, 218–221
коэффициент актуализации пропусков (GUF), 228
модем, 226
наибольший адрес станции (HSA), 228
настройка обмена данными, 208–219
оконечная нагрузка кабеля, 220
оптимизация производительности, 228
плата CP, 220
повторители, 218
построение, 218
примеры конфигураций, 213, 214, 215
развязка, 218
расчет расстояний, 218
скорость передачи, 218
сложная, 231
сложная, PPI, 213
смещение кабеля, 220
указания по конфигурированию, 218
устройства ЧМ-интерфейса, 221
master-устройства, 208
MPI, больше 187,5 кБод, 214
MPI, меньше 187,5 кБод, 214

Сеть (продолжение)
PPI с несколькими master-устройствами, 213
PROFIBUS-DP, 215
PPI с одним master-устройством, 213
slave-устройства, 208
Сеть с передачей маркера, пример, 229
Сеть MPI
больше 187,5 кБод, 214
меньше 187,5 кБод, 214
Сеть PPI с несколькими master-устройствами, 213
Сеть PPI с одним master-устройством, 213
Сеть PROFIBUS-DP
назначение контактов, 219
повторители, 218
технические данные кабеля, 218
S7-315-2 и EM 277, 215
STEP 7-Micro/WIN и ЧМ-интерфейс, 215
Символическая адресация, 58
Символические имена, создание списка, 49
Синхронное обновление, команда PWM, 127
Система с микроконтроллером, проектирование, 48
Системная поддержка, для программы обработки прерываний, 157
Системное руководство по S7-200, номера для заказа, 422
Системный блок, 50
Скорость передачи, сеть, 218
Скорости передачи
в сети, 218
оптимальная, 228
установка, 208
установки переключателей, кабель PC/PPI, 6, 210, 221, 223, 226, 397
Скоростной вывод импульсов
изменение ширины импульсов, 132
функционирование, 125
SMB66-SMB85, 415
Скоростной счетчик (HSC), 111
пример, 124
Скоростные импульсные выходы, 46
Скоростные счетчики, 46
адресация, 119
байт состояния, 119
блокировка, 123
временные диаграммы, 114–116
выбор активного уровня, 117
изменение направления, 123
изменение предустановленного значения, 123
изменение текущего значения, 123
назначение прерываний, 119
область памяти, адресация, 27
объяснение, 112
операции сброса и запуска, 116
определение режимов и входов, 113
последовательность инициализации, 120
прерывания, 113
программирование, 112
режимы, 114, 434
управляющий байт, 112
установка текущего и предустановленного значений, 118
установка управляющего байта, 118
SMB36-SMB65, 413
Слияние потоков управления
пример, 177
команды реле управления очередностью, 176
Сложная сеть PPI, 213
Служба коротких сообщений, модемный модуль EM 241, 290
Смещение
сетевой кабель, 220
PID-регулятор, 145, 147
Снятие
клеммного блока, 17
модуля памяти, 36
модуля расширения, 17
модуля CPU, 17
Совместимость
EM 231 с термометром сопротивления (RTD), 362
EM 231 с термопарой (thermocouple), 362
EM 277 PROFIBUS-DP, 374
Соединение
с S7-200, 7
Соединительные элементы, CP 243-2, 394
Создание
программы, 8
программы с помощью Micro/WIN, 51
протоколов, определяемых пользователем, 222
списка символических имен, 49
чертежей конфигурации, 49
Сообщения, сеть с передачей маркера, 229
Состояние
в конце цикла, 238
во время исполнения, 238
наблюдение состояния программы, 238
отображение в LAD и FBD, 238
отображение в STL, 239
Состояние входов/выходов (SMB5), 410
Состояние STOP, 11, 37
Сохранение
битовой памяти (M) в ЭСППЗУ, 35
данных программы S7-200, 34
значения в ЭСППЗУ, 413
памяти переменных (V) в ЭСППЗУ, 38
программы, 11
Сохранение, данные программы S7-200, 34
Сохранение памяти, 34–37
батарейный модуль (необязательный), 34
включение питания, 35–37
области, 41
ЭСППЗУ, 34–36
Сохраняемая битовая память, 35
Сохраняемая память, 41
Специальная память
модемный модуль EM 241, 304
протокол Modbus Slave, 330
Список команд. *См.* Редактор STL
Сравнение, времена оборота маркера, 230
Срок службы электрической части, 338
Стандартная профильная шина, 15
Стандартные команды преобразования, 92
Стандартный контакт, 66
Стандарты, национальные и международные, 338
Стандарт RS-232
кабель PC/PPI, 223
режим свободно программируемого обмена данными (Freeport), 223

Стандарт RS-485, 219
Станции оператора, определение, 48
Стековые команды
дублирование вершины логического стека, 70
загрузка стека, 70
извлечение вершины стека, 70
копирование второго уровня стека, 70
логическое сопряжение первого и второго уровня стека по И, 70
логическое сопряжение первого и второго уровня стека по ИЛИ, 70
пример, 71
AENO, 70
Строковые команды
копирование подстроки из строки, 186
копирование строки, 184
определение длины строки, 184
поиск первого появления символа в строке, 187
поиск строки в строке, 187
соединение строк, 184
Структурирование программы, 49
Суммирующий счетчик, 109
Схема соединений для входов
EM 231, 356
EM 235, 357
Схема соединений для выходов
EM 232, 358
EM 235, 358
Схемы защиты, проектирование, 48
Счетчики, 26
определение скоростного счетчика (HDEF), 111
скоростной счетчик (HSC), 111
IEC
вычитающий счетчик, 109
суммирующий счетчик, 109
реверсивный счетчик, 109
SIMATIC
обратный счет, 106
прямой счет, 106
реверсивный счетчик, 106
Счетчик, обратный счет, 106
пример, 108
Счетчики, скоростные, 46
Счетчик, прямой счет, 106
Съемные модули
номера для заказа, 422
память, 395

Т

Таблица глобальных переменных, 58
Таблица конфигурации и профилей, модуль позиционирования EM 253, 278
Таблица контура регулирования, 152
Таблица локальных переменных, 51, 59
Таблица определения профиля PTO0, PTO1 (SMB166 – SMB185), 418
Таблица перекрестных ссылок, 236
Таблица профилей, модуль позиционирования EM 253, 278
Таблица символов, 58
адресация, 58
конфигурирование для Modbus, 331
Таблица состояний, 59
наблюдение за значениями, 239
принудительное присваивание значений, 240
Таблица CRC, протокол Modbus Slave, 330
Табличные команды
ввод значений в таблицу, 189
заполнение памяти, 192
поиск в таблице, 193
удаление первой записи из таблицы, 190
удаление последней записи из таблицы, 190
Таймерные команды
прерывания, 158
IEC
импульсный таймер (TP), 201
таймер с задержкой включения (TON), 201
таймер с задержкой выключения (TOF), 201
SIMATIC
таймер с задержкой включения (TON), 196
таймер с задержкой включения с запоминанием (TONR), 196
таймер с задержкой выключения (TOF), 196
Таймерные команды SIMATIC, 196
пример, 198, 199, 200
Таймер с задержкой включения (TON), 196, 201
пример, 198
Таймер с задержкой включения с запоминанием (TONR), 196
пример, 200
Таймер с задержкой выключения (TOF), 196, 201
пример, 199
Таймер сообщения, 84
Текстовая пейджинговая связь, модемный модуль EM 241, 290
Текстовый дисплей TD 200
номер для заказа, 423
номер для заказа руководства, 422
Текущее значение
изменение в HSC, 123
установка для HSC, 118
Технические данные
аналогового модуля расширения, 351
кабеля PC/PPI, 396
модуля CPU, 340
системы с микроконтроллером, 48
цифрового модуля расширения, 346
EM 231 с термометром сопротивления (RTD), 361
EM 231 с термопарой, 361
EM 241 с модемом, 385
EM 277 PROFIBUS-DP, 373
Технические данные входов
аналоговый модуль расширения, 351
модули CPU, 341
цифровой модуль расширения, 346
Технические данные выходов
аналоговый модуль расширения, 352
модули CPU, 342
цифровой модуль расширения, 347
Типы данных, параметры подпрограммы, 204
Требования к зазорам, 14
Требования к оборудованию, 3
Требования, протокол Modbus Slave, 330
Триггер с преимуществом сброса, 72
Триггер с преимуществом установки, 72

У

Удаленный адрес, установка для S7-200, 210
Указания
  об изменении указателей для косвенной адресации, 32
  о вертикальном монтаже, 17
  о гальванической развязке, 18
  о защитных схемах, 20
  о конфигурировании сетей, 218
  о подключении, 18
  о подпрограммах, 50
  относительно команд для модуля позиционирования EM 253, 257
  относительно программ обработки прерываний, 50, 157
  относительно сред, в которых возникают сильные вибрации, 16
  по заземлению и подключению, 19
  по монтажу, 14
Указания по подключению устройств переменного тока, 19
Указания по подключению устройств постоянного тока, 19
Указатели, косвенная адресация, 32
Управление шаговым двигателем, генераторы PTO/PWM, 130
Управляющий байт, настройка для HSC, 118
Уровни активизации входов, модуль позиционирования EM 253, 247
Условия окружающей среды
  транспортировка и хранение, 339
  эксплуатация, 339
Условные входы, 55
Условные переходы, пример, 178
Установка
  времени, 73
  даты, 73
  скорости передачи, 208
  текущего и предустановленного значений для HSC, 118
  удаленного адреса для S7-200, 210
  управляющего байта (HSC), 118
Установка
  STEP 7-Micro/WIN, 4
Установка двухпозиционных переключателей, кабель PC/PPI, 6, 223
Устранение конфликтов в сети, 231
Устранение неисправностей
  аппаратные средства S7-200, 241
  коды ошибок, 404
  нефатальные ошибки, 56
  указания, 241
  фатальные ошибки, 57
Устройства человеко-машинного (ЧМ) интерфейса, 221

Ф

Файл базы данных устройства (GSD), EM 277 PROFIBUS-DP, 381–382
Фатальные ошибки, 57
  просмотр, 404
Физические входы, реакция модуля позиционирования EM 253, 248
Фильтрация
  аналоговые входы, 42
  цифровые входы, 41
Фильтрация помех, входной фильтр, 41
Формат байтового адреса, 24
Формат слова данных для выходов
  EM 232, 357
  EM 235, 357
Формат слова данных для входов
  EM 231, 356
  EM 235, 356
Формат сообщений для передачи данных CPU, модемный модуль EM 241, 310
Формат текстовых сообщений, модемный модуль EM 241, 309
Формат телефонного номера для передачи сообщений, модемный модуль EM 241, 308
Функции памяти
  команда перемещения значения бита в регистр сдвига, 181
  команды групповой пересылки, 167
  команды обмена, 183
  команды пересылки, 165
  команды сдвига, 179
  команды циклического сдвига, 179
Функции, Modbus, 332
Функции PTO/PWM, регистры (WMB66 – SMB85), 415
Функциональный план. *См.* Редактор FBD

Х

Характеристики
  модемного модуля EM 241, 288
  модулей CPU, 64, 461
  модуля позиционирования EM 253, 244

Ц

Целостность байтов, PROFIBUS, 378
Целостность буферов, PROFIBUS, 378
Целостность данных, PROFIBUS, 378
Целостность слов, PROFIBUS, 378
Цепная логическая схема. *См.* Редактор LAD
Цикл сканирования, 22
  определение количества, 240
  таймеры, 198
Цифровая пейджинговая связь, модемный модуль EM 241, 290
Цифровой модуль расширения, адресация, 31
Цифровые входы
  захват импульса, 41
  фильтрация, 41
  чтение, 39
Цифровые выходы, конфигурирование состояний, 40
Цифровые модули, 3

Ч

Часы
  съемный модуль, 395
Часы реального времени (TOD), 73
Чертежи конфигурации, 49
Чертежи, создание конфигурации, 49
Числа, представление, 24, 30

Числа с плавающей точкой, 29, 149
Числовые команды
квадратный корень, 143
косинус, 143
натуральная экспонента, 143
натуральный логарифм, 143
синус, 143
тангенс, 143
Чтение данных из входов, 22, 23

Ш

Шаги профиля, модуль позиционирования EM 253, 253
Широтно-импульсная модуляция (PWM), 46, 125
время цикла, 127
значения таблицы профиля, 130
изменение ширины импульсов, 132
инициализация, 132
конфигурирование с помощью SM-памяти, 128
мастер управления позиционированием , 125
объяснение, 127
пример, 132
способы обновления, 127
управление шаговым двигателем, 130
Штекер RJ11, модемный модуль EM 241, 288, 386

Э

Электромагнитная совместимость
S7-200, 339
стандарты излучения, 339
стандарты помехоустойчивости, 339
ЭСППЗУ, 34
коды ошибок, 404
сохранение битовой памяти (M), 35
сохранение памяти переменных (V), 38, 413

A

ACCEL_TIME (время ускорения), модуль позиционирования EM 253, 249
AI. *См.* Аналоговый вход (AI)
AQ. *См.* Аналоговый вывод (AQ)

C

CP5411, 422
CP5511, 422
CP5611, 422
CPU
буферизация, 2
защита паролем, 44
идентификационный регистр (SMB6), 410
имитаторы входов, 398
коммуникационные порты, 2
модули расширения, 2
обработка ошибок, 56
память, 2
потребности в мощности, 15, 399
размеры, 2
размеры винтов для монтажа, 16
скорость выполнения, 2
цикл сканирования, 22
цифровые входы/выходы, 2
часы реального времени, 2
CPU 224, пример программы DP, 383
CPU, поддерживающие интеллектуальные модули, 303

D

DECEL_TIME (время замедления), модуль позиционирования EM 253, 249

E

EN, 55, 63
ENO, 55, 63

G

GSD-файл, EM 277 PROFIBUS-DP, 381–382

H

HSC, 46

J

JOG_INCREMENT, модуль позиционирования EM 253, 249
JOG_SPEED, модуль позиционирования EM 253, 249

M

Master-устройства, 208
MAX_SPEED, модуль позиционирования EM 253, 248
Micro/WIN. *См.* STEP 7-Micro/WIN

O

OP3, OP7, OP17, номера для заказа, 423

P

PID-регулятор, 145
аварийный контроль, 151
автоматический режим, 151
виды управления по замкнутому контуру, 148
диапазоны, 150
дифференциальная составляющая, 148
интегральная составляющая, 147
мастер, 145
нормализация входов контура регулирования, 149
объяснение, 146
переменные, 150
преобразование входов контура регулирования, 149
преобразование выходов контура регулирования в масштабированное целое число, 150
пример, 153
пример программы, 152
пропорциональная составляющая, 147
режимы, 151

ручной режим, 151
сбойные ситуации, 151
с отрицательной обратной связью, 150
с положительной обратной связью, 150
таблица контура регулирования, 152
управление по замкнутому контуру, 148
POSx_CFG, 268
POSx_CLR, 267
POSx_CTRL, 258
POSx_DIS, 266
POSx_GOTO, 260
POSx_LDOFF, 263
POSx_LDPOS, 264
POSx_MAN, 259
POSx_RSEEK, 262
POSx_RUN, 261
POSx_SRATE, 265
PROFIBUS-DP
модуль (EM 277), 375
пример программы, 383
стандартный обмен данными, 374
целостность данных, 378

S

S7-200
адресация, 24
аккумуляторы, 27
аналоговые входы (AI), 29
аналоговые выходы (AQ), 29
битовая память (M), 25
блок питания, 6
выполнение логики управления, 22
доступ к данным, 24
загрузка в CPU, 34
загрузка из CPU, 34
захват импульсов, 42
защита паролем, 44
как slave-устройство, 208, 375
коды ошибок, 404
компоненты системы, 2
модем, 226
модули расширения, 3
модули CPU, 2
модуль памяти, 36
монтаж, 15
области памяти, 64, 461
область битовой памяти (M), 25
область памяти выходов (Q), 25
область памяти локальных данных (L), 28
область памяти переменных (V), 25
область памяти счетчиков (C), 26
область памяти таймеров (T), 26
область специальной памяти (SM), 28
обмен данными в сети, 208
обработка коммуникационных задач, 40
обработка ошибок, 56
память, 24
память локальных данных (L), 28
память переменных (V), 25
память реле управления очередностью (S), 29
память таймеров (T), 26
поддерживаемые программы обработки прерываний, 158
постоянные значения, 30
S7-200 (продолжение)
пример сетевой конфигурации, 213, 214, 215
программы обработки прерываний, 157
размеры, 2
реакция на фатальную ошибку, 404
регистр входов образа процесса (I), 25
регистр выходов образа процесса (Q), 25
регистр образа процесса, 39
режим RUN, 11, 37
реле управления очередностью, область памяти (S), 29
сетевой адрес, 209–211
скорости передачи, 208, 209–211
скоростные счетчики, 27
соглашения о программировании, 55
соединение со STEP 7-Micro/WIN, 7
состояние STOP, 11, 37
сохранение данных, 34
сохраняемая память, 41
специальная битовая память (SM), 28
специальные характеристики, 39
стандарт RS-232, 223
схемы подключения, 343–345
счетчики (C), 26
таблица состояний, 239
технические данные, 339
указания по заземлению, 19
указания по монтажу, 14
указания по подключению, 19
условия окружающей среды, 339
устранение неисправностей, 241
устранение неисправностей аппаратуры, 241
цикл сканирования, 22, 39
чтение и запись данных, 22
электромагнитная совместимость, 339
S7-300, пример сетевой конфигурации, 214
S7-400, пример сетевой конфигурации, 214
Slave-устройство, 208
EM 277 PROFIBUS-DP, 375
S7-200, 375
SM-память, 28
протокол Modbus Slave, 330
работа с PTO/PWM, 128
SM0.2: бит потери сохраняемых данных, 35
SMB0: биты состояния, 408
SMB1: биты состояния, 408
SMB130: регистры управления режимом свободно программируемого обмена данными (freeport), 412
SMB131 – SMB165: регистры HSC3, HSC4, HSC5, 417
SMB166 – SMB185: таблица определения профилей PTO0, PTO1, 418
SMB186 – SMB194: управление приемом сообщений, 416
SMB2: буфер для приема символов в режиме свободно программируемого обмена данными, 409
SMB200 – SMB549: состояние интеллектуальных модулей, 419
SMB28, SMB29: аналоговый потенциометр, 45, 412
SMB3: ошибка, выявленная контролем четности, при свободно программируемом обмене данными, 409

SMB30 и SMB130: регистры управления свободно программируемым обменом данными, 412
SMB31 и SMW32: управление записью в энергонезависимую память (ЭСППЗУ), 413
SMB34 и SMB35: регистры прерываний, управляемых временем, 413
SMB36 – SMB65: регистры HSC0, HSC1, HSC2, 413
SMB4: переполнение очереди, 409
SMB5: состояние входов/выходов, 410
SMB6: идентификационный регистр CPU, 410
SMB66 – SMB85: регистры PTO/PWM, 415
SMB7: резерв, 410
SMB8 – SMB21: регистры идентификации и ошибок модулей ввода/вывода, 411
SMB86 – SMB94, SMB186 – SMB194: управление приемом сообщений, 416
SMS, модемный модуль, 290
SMW22 – SMW26: времена цикла, 412
SMW98: ошибки в шине расширения ввода/вывода, 417
SS_SPEED, модуль позиционирования EM 253, 248
STEP 7-Micro/WIN
  запуск, 7
  инструментальные средства отладки, 235
  интерфейс для EM 241, 289
  кабель PC/PPI, 221
  как master-устройство, 208
  набор команд
    выбор, 53
    IEC 1131-3, 53
    SIMATIC, 53
  настройки связи, 7
  номера для заказа, 422
  открытие, 51
  пакет программирования, 3
  плата CP, 221
  пример сетевой конфигурации, 213–216
  редакторы программ, 51
  сетевой адрес, 209–211
  скорость передачи, 209–211
  соединение с S7-200, 7
  создание программ, 51
  требования к компьютеру, 3
  требования к оборудованию, 3
  установка, 4

T

Teleservice, 289