

Основы ПЛК

4

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

В этой главе

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

Выполнение логики управления с помощью S7-200

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

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

Основной принцип действия S7-200 очень прост:

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

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

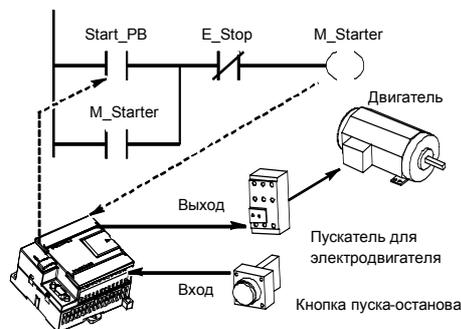


Рис. 4-1. Управление входами и выходами

S7-200 выполняет все задачи в цикле

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

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

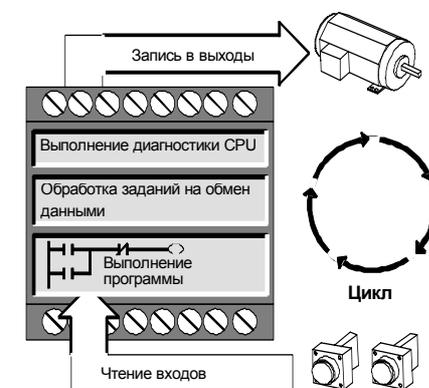


Рис. 4-2. Цикл S7-200

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

Чтение входов

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

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

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

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

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



Совет

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

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

Исполнение программы

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

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

Обработка запросов на обмен данными

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

Самодиагностика CPU

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

Запись в цифровые выходы

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

Доступ к данным S7–200

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

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

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

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

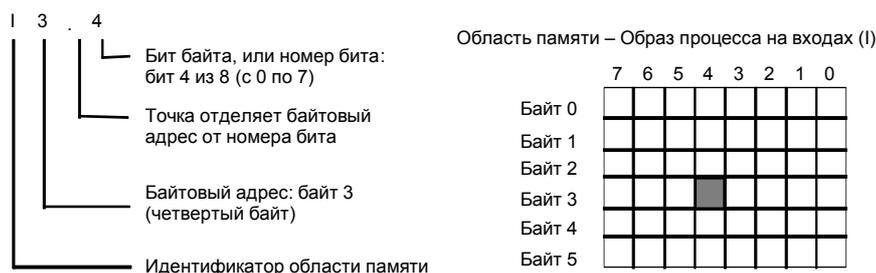


Рис. 4–3. Адресация байт.бит

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

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

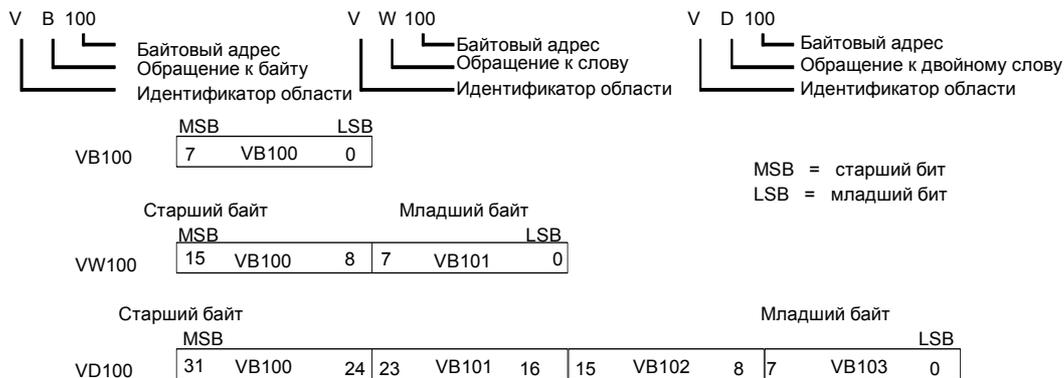


Рис. 4–4. Обращение к одному и тому же адресу в формате байта, слова и двойного слова

Обращение к данным в областях памяти

Регистр входов образа процесса: I

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

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

Регистр выходов образа процесса: Q

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

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

Область памяти переменных: V

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

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

Область битовой памяти: M

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

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

Таймеры: T

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

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

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



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

Счетчики: C

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

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

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



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

Скоростные счетчики: HC

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

Формат: HC[номер скоростного счетчика] HC1

Аккумуляторы: AC

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

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

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

Формат: AC[номер аккумулятора] AC0

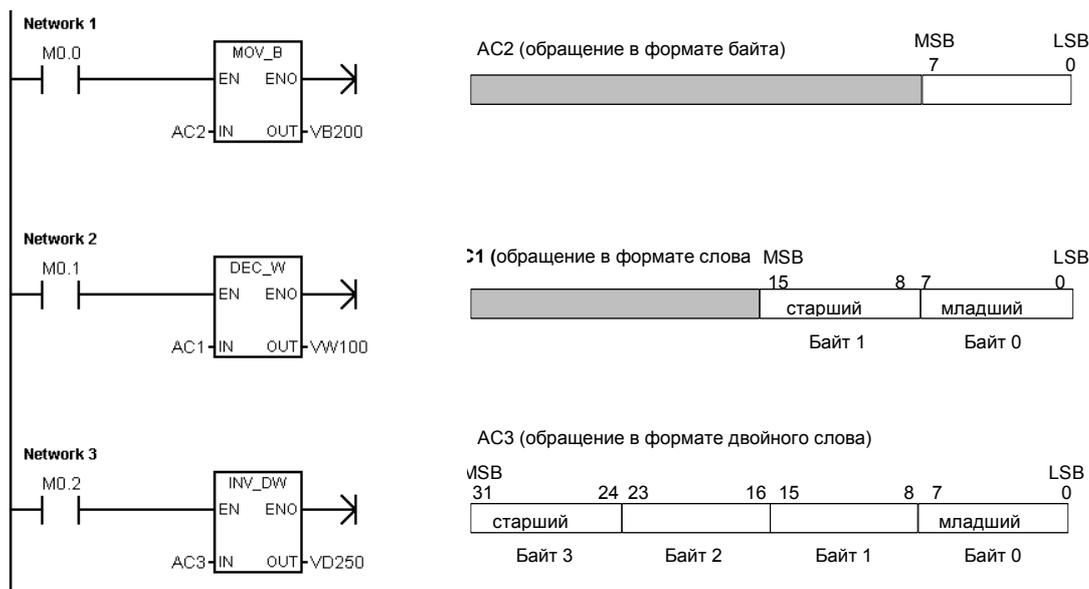


Рис. 4–7. Обращение к аккумуляторам

Специальные биты памяти: SM

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

Бит: SM[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.

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

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

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

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

Бит: L[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). Обращение к этим значениям производится через идентификатор области (AI), длину данных (W) и начальный адрес байта. Так как в случае аналоговых входов речь идет о словах, которые всегда начинаются на байтах с четными номерами (например, 0, 2, 4 и т.д.), то обращаются к этим значениям с помощью адресов четных байтов (например, AIW0, AIW2, AIW4). Аналоговые входы можно только считывать.

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

Аналоговые выходы: AQ

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

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

Реле управления очередностью (SCR): S

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

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

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.

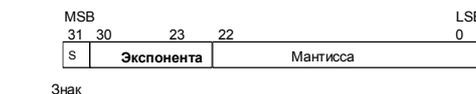


Рис. 4–8. Формат вещественного числа

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. Строковая константа ограничена 126 байтами.



Рис. 4–9. Формат строк

Задание постоянного значения для команд S7–200

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

Таблица 4–2. Представление постоянных величин

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



Совет

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

Адресация встроенных входов/выходов и входов/выходов модулей расширения

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



Совет

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

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

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

CPU 224XP	4 вх. / 4 вых.	8 вх.	4 аналог. вх. 1 аналог. вых.	8 вых.	4 аналог. вх. 1 аналог. вых.
I0.0 Q0.0 I0.1 Q0.1 I0.2 Q0.2 I0.3 Q0.3 I0.4 Q0.4 I0.5 Q0.5 I0.6 Q0.6 I0.7 Q0.7 I1.0 Q1.0 I1.1 Q1.1 I1.2 Q1.2 I1.3 Q1.3 I1.4 Q1.4 I1.5 Q1.5 I1.6 Q1.6 I1.7 Q1.7 AIW0 AQW0 AIW2 AQW2 Встроенные вх/вых	Модуль 0 I2.0 Q2.0 I2.1 Q2.1 I2.2 Q2.2 I2.3 Q2.3 I2.4 Q2.4 I2.5 Q2.5 I2.6 Q2.6 I2.7 Q2.7	Модуль 1 I3.0 I3.1 I3.2 I3.3 I3.4 I3.5 I3.6 I3.7	Модуль 2 AIW4 AQW4 AIW6 AQW6 AIW8 AIW10	Модуль 3 Q3.0 Q3.1 Q3.2 Q3.3 Q3.4 Q3.5 Q3.6 Q3.7	Модуль 4 AIW12 AQW8 AIW14 AQW10 AIW16 AIW18
Входы/выходы модулей расширения					

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

Косвенная адресация областей памяти S7–200 с помощью указателей

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

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

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

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

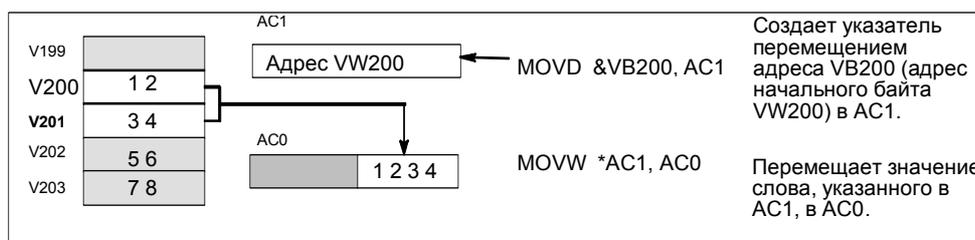
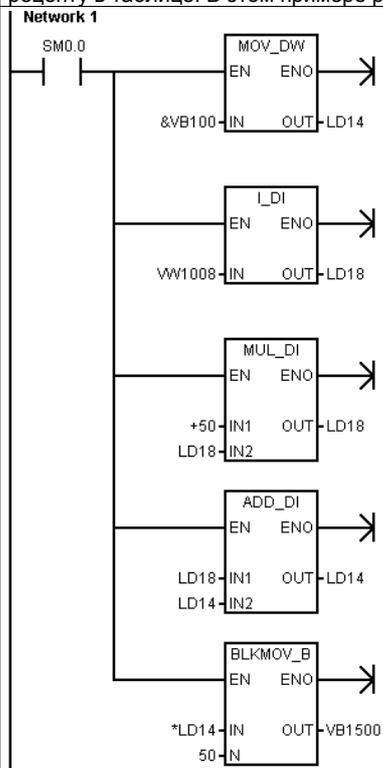


Рис. 4–11. Создание и использование указателя

Пример программы для обращения к данным в таблице с использованием указателя

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



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

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

Сохранение и извлечение данных с помощью S7–200

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

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

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

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

Осторожно

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

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

Загрузка компонентов проекта в CPU и из CPU

Ваш проект состоит из различных компонентов:

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

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

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

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

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

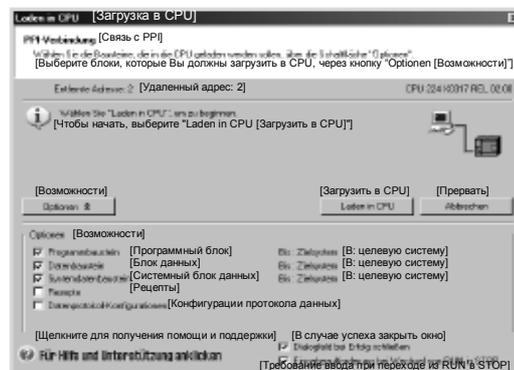


Рис. 4–13. Загрузка проекта в CPU S7–200

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

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

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

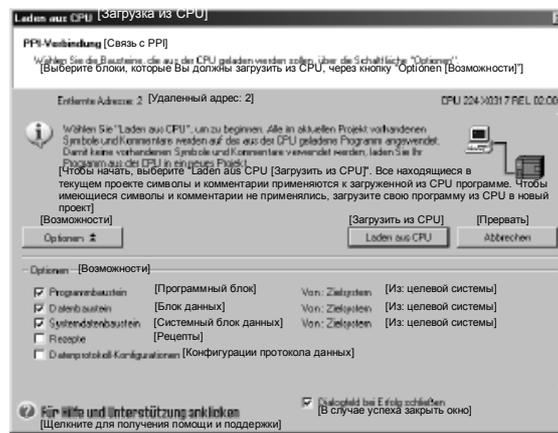


Рис. 4–14. Загрузка проекта из CPU S7–200 в компьютер

Сохранение программы в модуле памяти

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

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

Для программирования модуля памяти ПЛК должен находиться в состоянии STOP.

Для сохранения программы в модуле памяти:

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

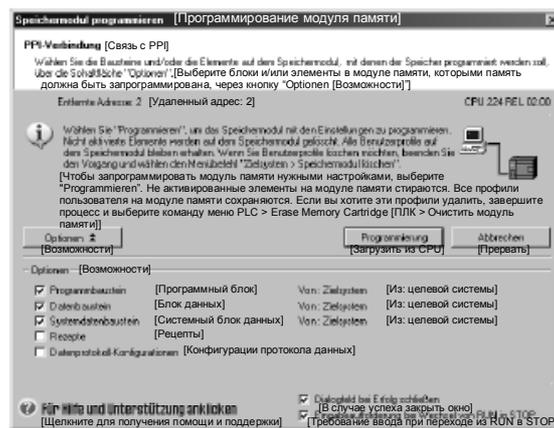


Рис. 4–15. Сохранение программы в модуле памяти

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

Извлечение программы из модуля памяти

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

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

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

Примечание

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

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

Сохранение реманентной битовой памяти M при потере питания

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

Извлечение данных после запуска

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

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

Сохранение памяти переменных в постоянной памяти с помощью программы

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

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



Совет

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

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

Копирование V-памяти в постоянную память

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

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

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

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

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

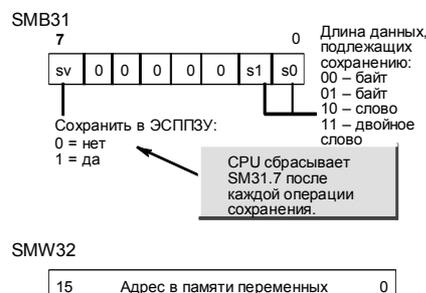
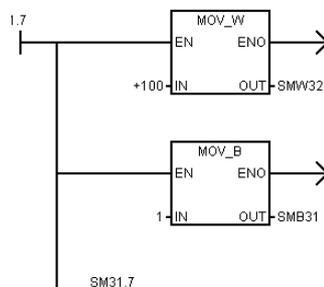


Рис. 4-16. SMB31 и SMW32

Пример программы: Копирование V-памяти в постоянную память

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

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



```

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

Установка режима работы CPU S7–200

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

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

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

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

Работа с проводником S7–200

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

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

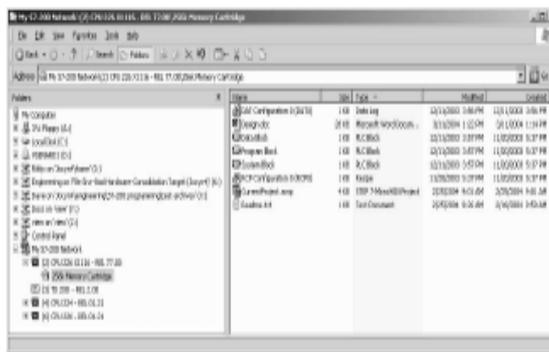


Рис. 4–17. Проводник S7–200

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

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

Функции S7–200

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

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

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

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



Совет

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

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

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

Программа S7–200 может прерывать цикл

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

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

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

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

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

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

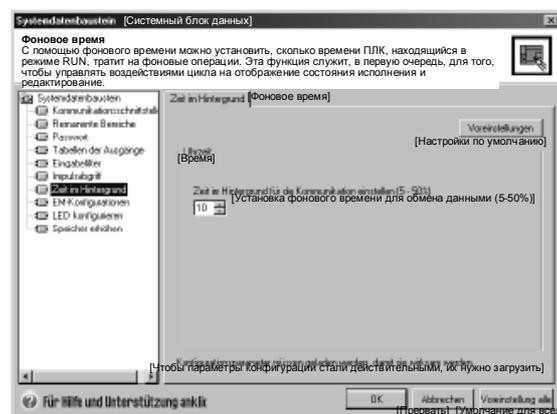


Рис. 4-18. Фоновое время для обмена данными

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

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

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

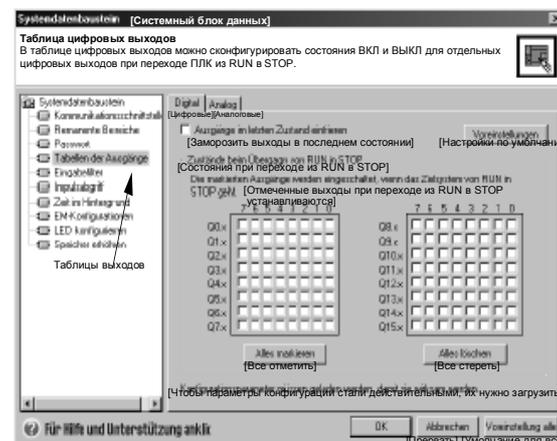


Рис. 4-19. Таблица цифровых выходов

S7-200 позволяет конфигурировать значения аналоговых выходов

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

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

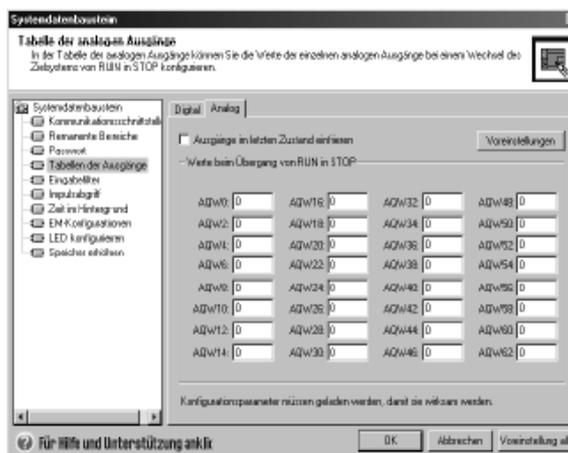


Рис. 4-20. Таблица аналоговых выходов

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

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

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



Совет

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

Для определения сохраняемой памяти:

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

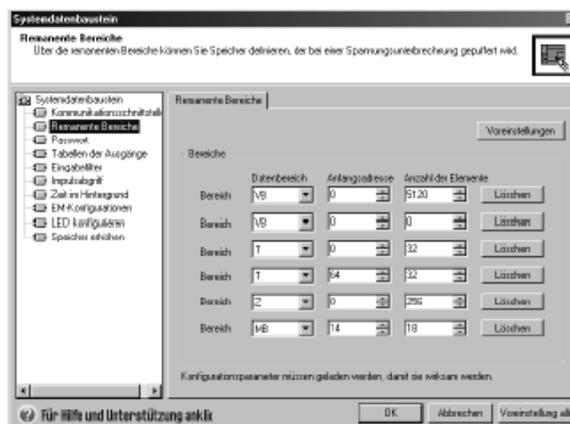


Рис. 4-21. Сохраняемая память

S7-200 дает возможность фильтровать цифровые входы

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

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

Для конфигурирования времен задержки для входного фильтра:

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

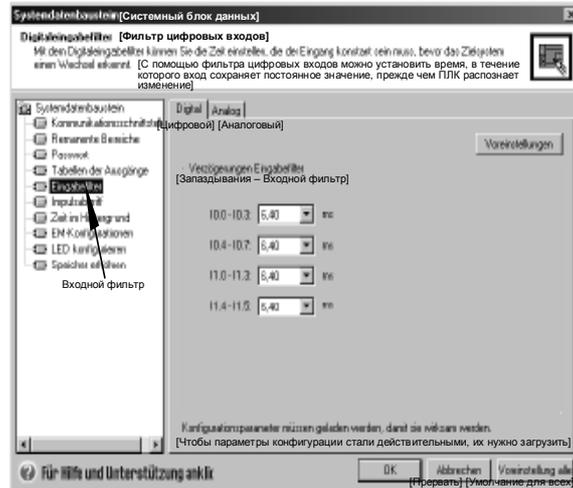


Рис. 4-22. Фильтр цифровых входов



Совет

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

S7-200 дает возможность фильтровать аналоговые входы

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

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

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

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

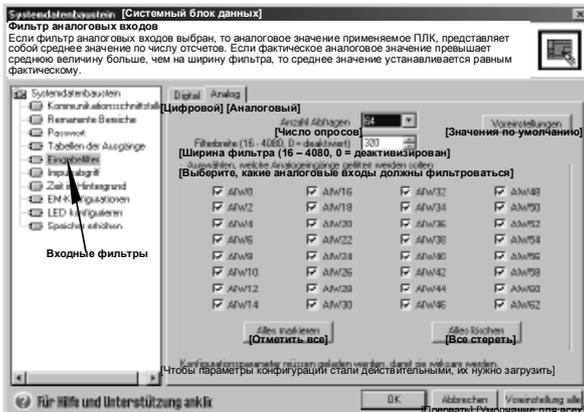


Рис. 4-23. Фильтр аналоговых входов

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

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

S7-200 дает возможность регистрировать короткие импульсы

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

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

Чтобы вызвать диалоговое окно для конфигурирования регистрации импульсов:

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

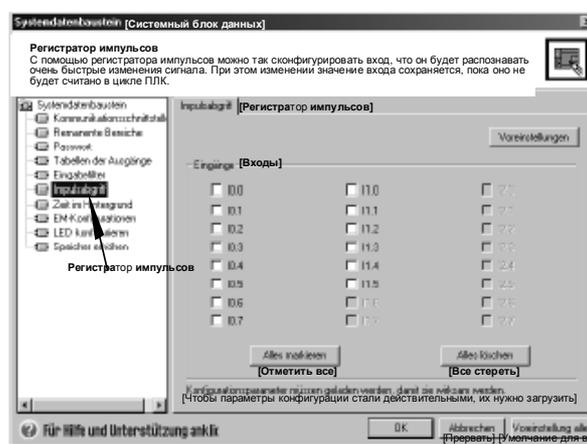


Рис. 4-24. Регистратор импульсов

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



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

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

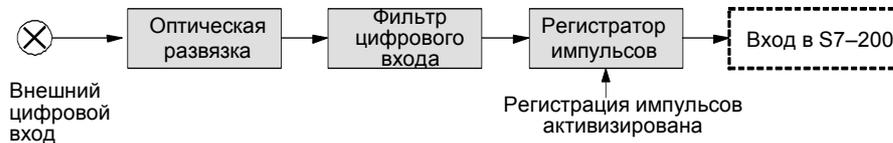


Рис. 4–26. Цепь цифрового входа

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

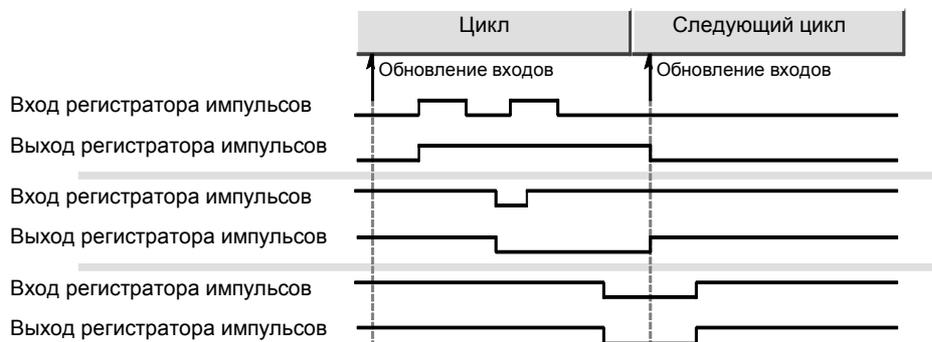


Рис. 4–27. Реакции регистратора импульсов на различные входные условия

У S7–200 имеются светодиод, которым может управлять пользователь

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

Для конфигурирования автоматических настроек диагностического светодиода действуйте следующим образом:

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

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

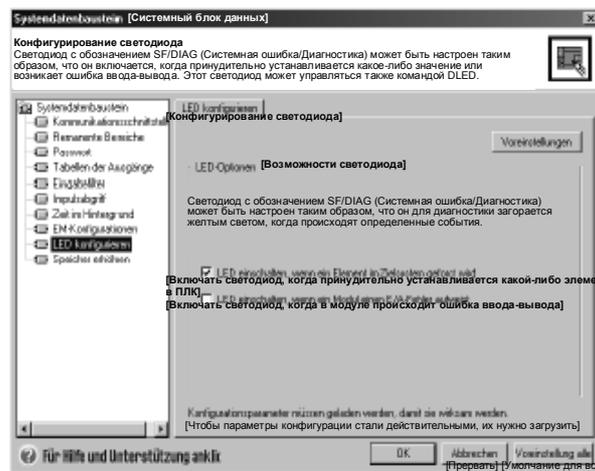


Рис. 4–28. Диагностический светодиод

S7-200 поддерживает протокол существенных событий CPU

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

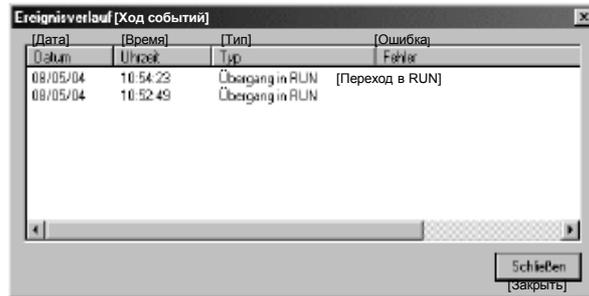


Рис. 4-29. Просмотр протокола истории событий

S7-200 позволяет увеличить доступную память для программы пользователя

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

Чтобы заблокировать функцию редактирования в режиме RUN, действуйте следующим образом

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

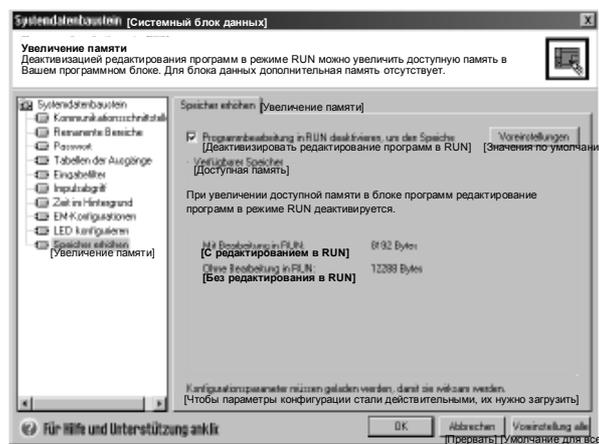


Рис. 4-30. Блокирование функции редактирования в режиме RUN

S7-200 предоставляет защиту с помощью пароля

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

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

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

Таблица 4-3. Ограничение доступа к S7-200

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

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



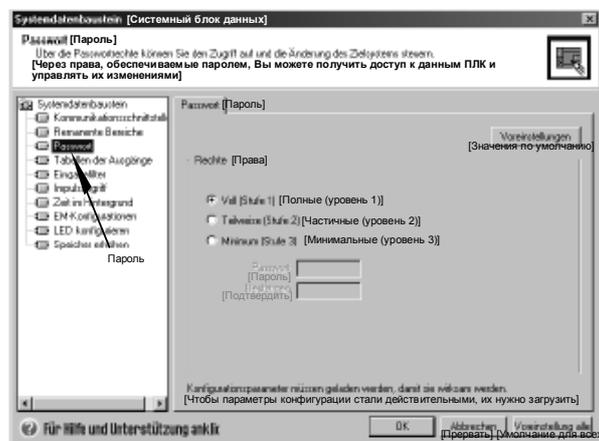
Совет

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

Установка пароля для S7-200

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

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



Последовательность действий при утере пароля

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

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

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

Предупреждение

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

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

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

S7–200 имеет в своем распоряжении аналоговые потенциометры

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

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

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

У S7–200 имеются скоростные входы и выходы

Скоростные счетчики

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

Скоростные импульсные выходы

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

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

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

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