

# Таблица расчета баланса токов

## B

Каждый CPU S7-200 (центральное устройство) обеспечивает постоянным током 5 В и 24 В модули расширения.

- Модули расширения автоматически снабжаются постоянным током 5 В через шину расширения.
- Каждый CPU имеет в своем распоряжении источник питания 24 В постоянного тока для датчиков, который снабжает током входы, а также обмотки реле модулей расширения. Вы должны подключить источник питания 24 В постоянного тока ко входам и обмоткам реле вручную.

С помощью следующей таблицы Вы можете рассчитать, какую мощность или какой ток может обеспечивать CPU Вашей конфигурации. Таблицы данных в приложении А предоставляют Вам информацию о наличной мощности отдельных CPU и потребляемой мощности модулей расширения. В разделе 2.5 показан пример расчета баланса токов.

Мощность CPU	5 В пост. тока	24 В пост. тока

*минус*

Потребляемая мощность системы	5 В пост. тока	24 В пост. тока
	Центральное устройство	
Общая потребность		

*равно*

Баланс токов	5 В пост. тока	24 В пост. тока
Общий баланс токов		



# Сообщения об ошибках

# C

Следующая информация о возможных ошибках должна помочь Вам при устранении ошибок с помощью Вашего CPU S7-200.

## Обзор главы

Раздел	Описание	Страница
C.1	Неисправимые ошибки	C-2
C.2	Ошибки во время выполнения	C-3
C.3	Нарушения правил компиляции	C-4

## С.1 Неисправимые ошибки

Если появляется неисправимая (фатальная) ошибка, то CPU прекращает обработку программы. В зависимости от степени тяжести ошибки CPU больше не может выполнять некоторые или даже любые функции. Целью устранения фатальных ошибок является перевод CPU в безопасное состояние, чтобы можно было запросить информацию о сбойной ситуации в CPU.

При обнаружении неисправимой ошибки CPU выполняет следующие функции:

- CPU переходит в режим STOP.
- На CPU загораются светодиодные индикаторы системной ошибки (SF) и режима работы STOP.
- Выходы выключаются.

CPU остается в этом режиме до тех пор, пока не будет устранена фатальная ошибка. В табл. С–1 приведено описание всех кодов фатальных ошибок, выводимых CPU.

Таблица С–1. Коды ошибок и сообщения о фатальных ошибках

Код ошибки	Описание
0000	Фатальная ошибка не встретилась.
0001	Ошибка контрольной суммы в программе пользователя.
0002	Ошибка контрольной суммы в скомпилированной КОР–программе.
0003	Ошибка времени при контроле времени цикла.
0004	Отказ внутреннего EEPROM.
0005	Ошибка контрольной суммы внутреннего EEPROM в программе пользователя.
0006	Ошибка контрольной суммы внутреннего EEPROM в параметрах конфигурации.
0007	Ошибка контрольной суммы внутреннего EEPROM в принудительно установленных данных.
0008	Ошибка контрольной суммы внутреннего EEPROM в значениях по умолчанию в таблице выходов.
0009	Ошибка контрольной суммы внутреннего EEPROM в данных пользователя, DB1.
000A	Отказ модуля памяти.
000B	Ошибка контрольной суммы модуля памяти в программе пользователя. Сообщение об этой ошибке появляется, когда модуль памяти пуст <input type="checkbox"/> не содержит программу).
000C	Ошибка контрольной суммы модуля памяти в параметрах конфигурации.
000D	Ошибка контрольной суммы модуля памяти в принудительно установленных данных.
000E	Ошибка контрольной суммы модуля памяти в значениях по умолчанию в таблице выходов.
000F	Ошибка контрольной суммы модуля памяти в данных пользователя, DB1.
0010	Внутренняя ошибка программного обеспечения.
0011	Ошибка косвенной адресации сравнивающего контакта.
0012	Недопустимое значение в сравнивающем контакте.
0013	Неподходящий модуль памяти для данного CPU.

## С.2 Ошибки во время выполнения

Во время исполнения Вашей программы могут появиться незначительные (не фатальные) ошибочные ситуации (например, ошибки адресации). В этом случае CPU генерирует код ошибки во время выполнения. Таблица С–2 дает перечень описаний кодов нефатальных ошибок.

Таблица С–2. Коды ошибок и сообщения об ошибках во время выполнения

Код ошибки	Ошибки во время выполнения (не фатальные ошибки)
0000	Ошибок нет.
0001	Блок HSC разблокирован перед обработкой блока HDEF.
0002	Назначение прерывания входу, который уже был поставлен в соответствие HSC.
0003	Назначение HSC входам, которые уже были поставлены в соответствие входному прерыванию.
0004	Попытка выполнения одной из операций ENI, DISI или HDEF в программе обработки прерываний.
0005	Попытка выполнения второго HSC с тем же самым номером до завершения первого HSC (HSC в программе обработки прерываний конфликтует с HSC в главной программе).
0006	Ошибка косвенной адресации.
0007	Ошибка в данных для операции TODW (запись в часы реального времени).
0008	Превышение максимальной глубины вложения подпрограмм.
0009	Обработка другой операции XMT в то время, когда отправитель активен.
000A	Попытка нового определения HSC с помощью второй операции HDEF для того же самого HSC.
0091	Ошибка области (вместе с адресной информацией): Проверьте области операндов.
0092	Ошибка в счетном поле операции (вместе со счетной информацией): Проверьте максимальное счетное значение.
0094	Ошибка области при записи в энергонезависимую память (вместе с адресной информацией).

### С.3 Нарушения правил компиляции

При загрузке программы в CPU, последний компилирует эту программу. Если CPU обнаруживает, что программа нарушает какое-либо правило компиляции (например, операция является недопустимой), то CPU прерывает процесс загрузки и выводит сообщение об ошибке компиляции. В табл. С–3 описаны коды ошибок при нарушении правил компиляции.

Таблица С–3. Коды ошибок и сообщения при нарушении правил компиляции

Код ошибки	Ошибки компиляции (не фатальные ошибки)
0080	Транслируемая программа слишком велика: Сократите программу.
0081	Выход за нижнюю границу стека: Разделите сегмент на несколько сегментов.
0082	Недопустимая операция: Проверьте мнемонику операции.
0083	Отсутствует MEND или недопустимая операция в главной программе: Введите операцию MEND или удалите недопустимую операцию.
0084	Резервный.
0085	Отсутствует FOR: Добавьте операцию FOR или удалите операцию NEXT.
0086	Отсутствует NEXT: Добавьте операцию NEXT или удалите операцию FOR.
0087	Отсутствует метка перехода (LBL, INT, SBR): Добавьте соответствующую метку перехода.
0088	Отсутствует RET или недопустимая операция в подпрограмме: Введите операцию RET в конце подпрограммы или удалите недопустимую операцию.
0089	Отсутствует RETI или недопустимая операция в программе обработки прерываний: Введите операцию RETI в конце программы обработки прерываний или удалите недопустимую операцию.
008A	Резервный.
008B	Дублированная область SCR.
008C	Дублированная метка перехода (LBL, INT, SBR): Переименуйте одну из меток перехода.
008D	Недопустимая метка перехода (LBL, INT, SBR): Обеспечьте, чтобы не превышалось допустимое количество меток перехода.
0090	Недопустимые параметры: Проверьте, являются ли допустимыми параметры операции.
0091	Ошибка области (вместе с адресной информацией): Проверьте области операндов.
0092	Ошибка в счетном поле операции (вместе со счетной информацией): Проверьте максимальное счетное значение.
0093	Превышение глубины вложения FOR/NEXT.
0095	Отсутствует операция LSCR (загрузка реле шагового управления).
0096	Отсутствует операция SCRE (завершение реле шагового управления) или недопустимая операция перед SCRE.

# Специальные меркеры

## D

Специальные меркеры предоставляют ряд функций состояния и управления, а также служат для того, чтобы производить обмен информацией между контроллером и программой. Специальные меркеры можно использовать как биты, байты, слова и двойные слова.

### SMB0: Биты состояния

Как описано в таблице D-1, байт SMB0 содержит восемь битов состояния, которые актуализируются CPU S7-200 в конце каждого цикла.

Таблица D-1. Специальные меркеры SMB0 (SM0.0 - SM0.7)

Специальные меркеры	Описание
SM0.0	Этот бит включен всегда.
SM0.1	Этот бит включен в первом цикле. Он используется, например, для вызова подпрограммы инициализации.
SM0.2	Этот бит включается на время одного цикла, если потеряны реманентные данные. Он может использоваться либо как меркер ошибки, либо как механизм вызова особых пусковых последовательностей.
SM0.3	Этот бит включается на время одного цикла, если режим работы RUN устанавливается при включении питания. Этот бит может быть использован, чтобы предоставить время на разогрев установки.
SM0.4	Этот бит обеспечивает тактовый импульс, который 30 секунд включен и 30 секунд выключен, то есть время цикла, равное 1 минуте. Благодаря этому, Вы имеете в своем распоряжении легко программируемую задержку или интервал между импульсами, равные 1 минуте.
SM0.5	Этот бит обеспечивает тактовый импульс, который 0,5 секунды включен и 0,5 секунды выключен, то есть время цикла, равное, равное 1 секунде. Благодаря этому, Вы имеете в своем распоряжении легко программируемую задержку или интервал между импульсами, равные 1 секунде.
SM0.6	Этот бит в одном цикле включен, а в следующем цикле выключен. Вы можете использовать этот бит как вход счетчика циклов.
SM0.7	Этот бит показывает положение переключателя режимов работы (TERM – выключен, RUN – включен). Если Вы используете этот бит для деблокировки свободно программируемой связи, когда переключатель стоит в положении RUN, то Вы можете разрешить нормальную связь с устройством программирования, переставляя переключатель в TERM.

### SMB1: Биты состояния

Как описано в таблице D–2, байт SMB1 содержит различные индикаторы возможных ошибок. Эти биты устанавливаются и сбрасываются во время обработки операций.

Таблица D–2. Специальные меркеры SMB1 (SM1.0 - SM1.7)

Специальные меркеры	Описание
SM1.0	Этот бит активизируется при обработке определенных операций, если результат равен нулю.
SM1.1	Этот бит активизируется при обработке определенных операций, если происходит переполнение или если обнаруживается недопустимое числовое значение.
SM1.2	Этот бит активизируется, если результат арифметической операции отрицателен.
SM1.3	Этот бит активизируется при делении на нуль.
SM1.4	Этот бит активизируется, если вследствие операции “записать значение в таблицу” таблица переполняется.
SM1.5	Этот бит активизируется, если с помощью операции LIFO или FIFO пытаются считывать значение из пустой таблицы.
SM1.6	Этот бит активизируется, если значение, представленное не в BCD–формате, должно преобразовываться в двоичное значение.
SM1.7	Этот бит активизируется, если значение ASCII не может быть преобразовано в допустимое шестнадцатиричное значение.

### SMB2: Прием в случае свободно программируемой связи

SMB2 служит в качестве буфера принимаемых символов в случае свободно программируемой связи. Как описано в таблице D–3, символы, принимаемые во время свободно программируемой связи, записываются по этому адресу, чтобы обеспечить легкий доступ к этим данным из KOP–программы.

Таблицы D–3. Специальные меркеры SMB2

Специальные меркеры	Описание
SMB2	Этот байт содержит каждый символ, принятый во время свободно программируемой связи.

### SMB3: Ошибка четности в случае свободно программируемой связи

SMB3 используется при свободно программируемой связи и содержит бит, который устанавливается, когда в одном из принятых кадров обнаруживается ошибка четности. Как описано в таблице D–4, бит SM3.0 включается, когда обнаруживается ошибка четности. С помощью данного бита Вы можете отвергнуть сообщение.

Таблица D–4. Специальные меркеры SMB3 (SM3.0 - SM3.7)

Специальные меркеры	Описание
SM3.0	Ошибка четности (0 = нет ошибки; 1 = ошибка)
SM3.1 - SM3.7	Резервные.



#### SMB4: Переполнение очереди

Как описано в таблице D–5, байт SMB4 содержит биты пополнения очереди прерываний, индикатор состояния, который показывает, разрешены или запрещены прерывания, а также меркер, который показывает, не занят ли передатчик. Биты пополнения очереди указывают либо на то, что прерывания возникают с более высокой скоростью, чем они могут обрабатываться, либо на то, что они были запрещены командой “Запретить все прерывания”.

Таблица D–5. Специальные меркеры SMB4 (SM4.0 - SM4.7)

Специальные меркеры	Описание
SM4.0 <sup>1</sup>	Активизируется, когда происходит пополнение очереди прерываний по связи.
SM4.1 <sup>1</sup>	Активизируется, когда происходит пополнение очереди прерываний по входам.
SM4.2 <sup>1</sup>	Активизируется, когда происходит пополнение очереди прерываний, управляемых временем.
SM4.3	Активизируется, когда обнаруживается ошибка программирования во время выполнения.
SM4.4	Показывает разблокированное состояние. Активизируется, когда разрешены все прерывания.
SM4.5	Активизируется, когда передатчик не занят.
SM4.6 и SM4.7	Резервные.
<sup>1</sup> Используйте биты состояния 4.0, 4.1 и 4.2 только в программах обработки прерываний. Эти биты состояния сбрасываются, когда очередь обработана и снова начинается обработка главной программы	

#### SMB5: Состояние входов и выходов

Как описано в таблице D–6, байт SMB5 содержит биты состояния по сбойным ситуациям, обнаруженным в системе ввода/вывода. Эти биты предоставляют обзор обнаруженных ошибок вода/вывода.

Таблица D–6. Специальные меркеры SMB5 (SM5.0 - SM5.7)

Специальные меркеры	Описание
SM5.0	Этот бит включается, когда возникает любая ошибка ввода/вывода.
SM5.1	Этот бит включается, когда к шине ввода/вывода подключено слишком много цифровых входов и выходов.
SM5.2	Этот бит активизируется, когда к шине ввода/вывода подключено слишком много аналоговых входов и выходов.
SM5.3 - SM5.7	Резервные.

### SMB6: Регистр идентификации CPU

Как описано в таблице D-7, байт SMB6 является регистром идентификации CPU. Биты SM6.4 - SM6.7 содержат идентификатор типа CPU. SM6.0 - SM6.3 зарезервированы для будущих функций.

Таблица D-7 Специальные меркеры SMB6

Специальные меркеры	Описание						
Формат	<table border="0"> <tr> <td>MSB</td> <td>LSB</td> <td></td> </tr> <tr> <td>7</td> <td>0</td> <td>Регистр идентификации CPU</td> </tr> </table>	MSB	LSB		7	0	Регистр идентификации CPU
MSB	LSB						
7	0	Регистр идентификации CPU					
SM6.4 - SM6.7	xxxx = 0000 = CPU 212 0010 = CPU 214 1000 = CPU 215 1001 = CPU 216						
SM6.0 - SM6.3	Резервные.						

### SMB7: Резервный

SMB7 зарезервирован для будущих функций.

### SMB8 - SMB21: Регистры идентификации и регистры ошибок модулей расширения

Байты SMB8 - SMB21 разбиты на пары, соответствующие модулям расширения от 0 до 6. Как описано в таблице D-8, байт с четным номером в паре является регистром идентификации модуля расширения. Этот байт идентифицирует тип модуля, а также вид и количество входов и выходов. Байт с нечетным номером в паре является регистром ошибок модуля. Этот байт отображает каждую ошибку, обнаруженную во входах и выходах модуля.

Таблица D-8. Специальные меркеры SMB8 - SMB21

Специальные меркеры	Описание																				
Формат	<table border="0"> <tr> <td>Байт с четным номером: регистр идентификации модуля</td> <td>Байт с нечетным номером: регистр ошибок модуля</td> </tr> <tr> <td> <table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table> </td> <td> <table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table> </td> </tr> </table> <table border="0"> <tr> <td>M: Наличие модуля 0 = имеется 1 = отсутствует</td> <td>C: Ошибка конфигурации</td> </tr> <tr> <td>tt: 00 Не интеллектуальный модуль 01 Резерв 10 Не модуль расширения 11 Резерв</td> <td>R: Выход за верхнюю границу диапазона</td> </tr> <tr> <td>A Вид ввода/вывода 0 = цифровой 1 = аналоговый</td> <td>P: Нет питания от пользователя</td> </tr> <tr> <td>ii 00 Нет входов 01 2AI или 8DI 10 4AI или 16DI 11 8AI или 32DI</td> <td>гг: Резерв QQ 00 Нет выходов 01 2 AO или 8DO 10 4 AO или 16DO 11 8 AO или 32 DO</td> </tr> </table>	Байт с четным номером: регистр идентификации модуля	Байт с нечетным номером: регистр ошибок модуля	<table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table>	MSB	LSB	7	0	<table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table>	MSB	LSB	7	0	M: Наличие модуля 0 = имеется 1 = отсутствует	C: Ошибка конфигурации	tt: 00 Не интеллектуальный модуль 01 Резерв 10 Не модуль расширения 11 Резерв	R: Выход за верхнюю границу диапазона	A Вид ввода/вывода 0 = цифровой 1 = аналоговый	P: Нет питания от пользователя	ii 00 Нет входов 01 2AI или 8DI 10 4AI или 16DI 11 8AI или 32DI	гг: Резерв QQ 00 Нет выходов 01 2 AO или 8DO 10 4 AO или 16DO 11 8 AO или 32 DO
Байт с четным номером: регистр идентификации модуля	Байт с нечетным номером: регистр ошибок модуля																				
<table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table>	MSB	LSB	7	0	<table border="0"> <tr> <td>MSB</td> <td>LSB</td> </tr> <tr> <td>7</td> <td>0</td> </tr> </table>	MSB	LSB	7	0												
MSB	LSB																				
7	0																				
MSB	LSB																				
7	0																				
M: Наличие модуля 0 = имеется 1 = отсутствует	C: Ошибка конфигурации																				
tt: 00 Не интеллектуальный модуль 01 Резерв 10 Не модуль расширения 11 Резерв	R: Выход за верхнюю границу диапазона																				
A Вид ввода/вывода 0 = цифровой 1 = аналоговый	P: Нет питания от пользователя																				
ii 00 Нет входов 01 2AI или 8DI 10 4AI или 16DI 11 8AI или 32DI	гг: Резерв QQ 00 Нет выходов 01 2 AO или 8DO 10 4 AO или 16DO 11 8 AO или 32 DO																				
SMB8 SMB9	Регистр идентификации модуля 0 Регистр ошибок модуля 0																				
SMB10 SMB11	Регистр идентификации модуля 1 Регистр ошибок модуля 1																				

Таблица D–8. Специальные меркеры SMB8 - SMB21, продолжение

Специальные меркеры	Описание
SMB12 SMB13	Регистр идентификации модуля 2 Регистр ошибок модуля 2
SMB14 SMB15	Регистр идентификации модуля 3 Регистр ошибок модуля 3
SMB16 SMB17	Регистр идентификации модуля 4 Регистр ошибок модуля 4
SMB18 SMB19	Регистр идентификации модуля 5 Регистр ошибок модуля 5
SMB20 SMB21	Регистр идентификации модуля 6 Регистр ошибок модуля 6

### SMW22 - SMW26: Времена цикла

Как описано в таблице D–9, слова SMW22, SMW24 и SMW26 дают информацию о времени цикла: наименьшее время цикла, наибольшее время цикла и последнее время цикла в миллисекундах.

Таблица D–9. Специальные меркеры SMW22 - SMW26

Специальные меркеры	Описание
SMW22	Это слово хранит время последнего цикла.
SMW24	Это слово хранит наименьшее время цикла с начала режима работы RUN.
SMW26	Это слово хранит наибольшее время цикла с начала режима работы RUN.

### SMB28 и SMB29: Аналоговые потенциометры

Байт SMB28 хранит цифровое значение, представляющее положение аналогового потенциометра 0. Байт SMB29 хранит цифровое значение, представляющее положение аналогового потенциометра 1.

Таблица D–10. Специальные меркеры SMB28 и SMB29

Специальные меркеры	Описание
SMB28	Этот байт хранит значение, которое вводится с помощью аналогового потенциометра 0. Это значение актуализируется один раз за цикл в режиме STOP/RUN.
SMB29	Этот байт хранит значение, которое вводится с помощью аналогового потенциометра 1. Это значение актуализируется один раз за цикл в режиме STOP/RUN.

**SMB30 и SMB130: Управляющие регистры свободно программируемой связи**

SMB30 управляет свободно программируемой связью для порта 0. SMB130 управляет свободно программируемой связью для порта 1. Вы можете считывать данные из SMB30 и SMB130 и записывать данные в них. Как описано в таблице D–11, эти байты конфигурируют соответствующий коммуникационный порт для свободно программируемой связи. Выбираться могут поддержка протокола для свободно программируемой связи, а также системный протокол.

Таблица D–11 Специальные меркеры SMB30 и SMB130

Интерфейс 0	Интерфейс 1	Описание							
Формат SMB30	Формат SMB130	MSB <span style="margin-left: 150px;">LSB</span> Управляющий байт для свободно программируемой связи <div style="display: flex; justify-content: space-between; width: 200px;"> <span>7</span> <span>0</span> </div> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">p</td> <td style="padding: 2px 5px;">d</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">b</td> <td style="padding: 2px 5px;">m</td> <td style="padding: 2px 5px;">m</td> </tr> </table>	p	d	b	b	b	m	m
p	d	b	b	b	m	m			
SM30.6 и SM30.7	SM130.6 и SM130.7	pp Выбор контроля четности 01 = контроль четности 10 = контроль четности отсутствует 11 = контроль нечетности							
SM30.5	SM130.5	d Биты данных на кадр 0 = 8 битов на кадр 1 = 7 битов на кадр							
с SM30.2 по SM30.4	с SM130.2 по SM130.4	bbb Скорость передачи данных 000 = 38.400 бод (в CPU 212: 19.200 бод) 001 = 19.200 бод 010 = 9.600 бод 011 = 4.800 бод 100 = 2.400 бод 101 = 1.200 бод 110 = 600 бод 111 = 300 бод							
SM30.0 и SM30.1	SM130.0 и SM130.1	mm Выбор протокола 00 = протокол интерфейса “точка-точка” (режим PPI/Slave) 01 = протокол свободно программируемой связи 10 = режим PPI/Master 11 = резервный (по умолчанию режим PPI/Slave)							

**SMB31 и SMW32: Управление записью в энергонезависимую память (EEPROM)**

Вы можете посредством Вашей программы записать значение, находящееся в памяти переменных, в энергонезависимую память (EEPROM). Для этого загрузите в SMW32 адрес значения, которое должно храниться независимо от наличия напряжения питания. Затем загрузите SMB31 командой “Сохранить значение”. После того, как Вы загрузили команду для сохранения значения, Вы не можете изменить значение в памяти переменных до тех пор, пока CPU не сбросит специальный меркер SM31.7 и тем самым не объявит конец операции сохранения.

В конце каждого цикла CPU проверяет, была ли выдана команда сохранить значение в энергонезависимой памяти. Если эта команда была, то заданное значение записывается в энергонезависимую память.

Как описано в таблице D–12, байт SMB31 определяет размер данных, которые должны записываться в энергонезависимую память, и, кроме того, предоставляет команду, запускающую обработку операции записи в память. SMW32 хранит начальный адрес данных в памяти переменных, которые должны храниться в энергонезависимой памяти.

Таблица D–12. Специальные меркеры SMB31 и SMW32

Специальные меркеры	Описание
Формат	<p>SMB31: MSB LSB</p> <p>программная команда 7 0</p> <p>SMW32: адрес в памяти переменных MSB LSB</p> <p>15 0</p>
SM31.0 и SM31.1	<p>ss: Размер записываемого в память значения</p> <p>00 = байт</p> <p>01 = байт</p> <p>10 = слово</p> <p>11 = двойное слово</p>
SM30.7	<p>c: Запись в энергонезависимую память (EEPROM)</p> <p>0 = Нет запроса на запись в память.</p> <p>1 = Программа пользователя требует от CPU записать значение в энергонезависимую память.</p> <p>CPU сбрасывает этот бит после каждой операции записи в память.</p>
SMW32	<p>Адрес данных в памяти переменных, которые должны храниться независимо от наличия напряжения питания, записан в SMW32. Это значение задается как смещение от V0. Если выполняется операция записи в память, то значение из этого адреса в памяти переменных записывается по адресу, соответствующему адресу памяти переменных, в постоянную память (EEPROM).</p>

### SMB34 и SMB35: Регистр интервалов для прерываний, управляемых временем

Как описано в таблице D–13, байт SMB34 задает интервал для управляемого временем прерывания 0, а байт SMB35 задает интервал для управляемого временем прерывания 1. Вы можете задавать интервал (с шагом в 1 мс) в диапазоне от 5 до 255 мс. CPU получает значение интервала, когда соответствующее прерывание, управляемое временем, поставлено в соответствие программе обработки прерываний. Чтобы изменить значение интервала, Вы должны заново прикрепить управляемое временем прерывание либо той же самой, либо другой программе обработки прерываний. Вы можете завершить событие управляемого временем прерывания, открепляя событие.

Таблица D–13. Специальные меркеры SMB34 и SMB35

Специальные меркеры	Описание
SMB34	Этот байт задает интервал (с шагом величиной 1 мс в диапазоне от 5 до 255 мс) для управляемого временем прерывания 0.
SMB35	Этот байт задает интервал (с шагом величиной 1 мс в диапазоне от 5 до 255 мс) для управляемого временем прерывания 1.

**SMB36 - SMB65: Регистр HSC**

Как описано в таблице D–14, байты SMB36 - SMB65 используются для того, чтобы контролировать работу быстрых счетчиков и управлять их работой.

Таблица D–14. Специальные меркеры SMB36 - SMB65

Специальные меркеры	Описание
SM36.0 - SM36.4	Резервные.
SM36.5	Бит состояния HSC0 - Текущее направление счета: 1 = прямой счет
SM36.6	Бит состояния HSC0 - Текущее значение равно предварительно установленному значению: 1 = равно
SM36.7	Бит состояния HSC0 - Текущее значение больше, чем предварительно установленное значение: 1 = больше
SM37.0 - SM37.2	Резервные.
SM37.3	Управляющий бит HSC0 - Направление счета: 1 = прямой счет
SM37.4	HSC0 - Актуализация направления: 1 = актуализировать направление
SM37.5	HSC0 - Актуализация предварительно установленного значения: 1 = записать новое значение в предварительную установку HSC0.
SM37.6	HSC0 -- Актуализация текущего значения: 1 = записать новое значение в текущее значение HSC0.
SM37.7	HSC0 - Бит разблокировки: 1 = разблокировать.
SMB38 SMB39 SMB40 SMB41	HSC0 - Новое текущее значение. SMB38 является старшим байтом, а SMB41 - младшим байтом.
SMB42 SMB43 SMB44 SMB45	HSC0 - Новое предварительно установленное значение. SMB42 является старшим байтом, а SMB45 - младшим байтом.
SM46.0 - SM46.4	Резервные.
SM46.5	Бит состояния HSC1 - Текущее направление счета: 1 = прямой счет
SM46.6	Бит состояния HSC1 - Текущее значение равно предварительно установленному значению: 1 = равно
SM46.7	Бит состояния HSC1 - Текущее значение больше, чем предварительно установленное значение: 1 = больше
SM47.0	Управляющий бит HSC1 - Степень активности для сброса: 0 = высокая; 1 = низкая
SM47.1	Управляющий бит HSC1 - Уровень активности для запуска: 0 = высокий; 1 = низкий
SM47.2	HSC1 - Скорость счета А/В-счетчика: 0 = четырехкратная скорость счета, 1 = однократная скорость счета
SM47.3	Управляющий бит HSC1 - Направление счета: 1 = прямой счет
SM47.4	HSC1 - Актуализация направления: 1 = актуализировать направление
SM47.5	HSC1 - Актуализация предварительно установленного значения: 1 = записать новое значение в предварительную установку HSC1.
SM47.6	HSC1 - Актуализация текущего значения: 1 = записать новое значение в текущее значение HSC1.
SM47.7	HSC1 - Бит разблокировки: 1 = разблокировать.

Таблица D–14. Специальные меркеры SMB36 - SMB65, продолжение

Специальные меркеры	Описание
SMB48 SMB49 SMB50 SMB51	HSC1 - Новое текущее значение. SMB48 является старшим байтом, а SMB51 - младшим байтом.
SMB52 - SMB55	HSC1 - Новое предварительно установленное значение. SMB52 является старшим байтом, а SMB55 - младшим байтом.
SM56.0 - SM56.4	Резервные.
SM56.5	Бит состояния HSC2 - Текущее направление счета: 1 = прямой счет
SM56.6	Бит состояния HSC2 - Текущее значение равно предварительно установленному значению: 1 = равно
SM56.7	Бит состояния HSC2 - Текущее значение больше, чем предварительно установленное значение: 1 = больше
SM57.0	Управляющий бит HSC2 - Уровень активности для сброса: 0 = высокий; 1 = низкий
SM57.1	Управляющий бит HSC2 - Уровень активности для запуска: 0 = высокий; 1 = низкий
SM57.2	HSC2 - Скорость счета А/В-счетчика: 0 = четырехкратная скорость счета, 1 = однократная скорость счета
SM57.3	Управляющий бит HSC2 - Направление счета: 1 = прямой счет
SM57.4	HSC2 - Актуализация направления: 1 = актуализировать направление
SM57.5	HSC2 - Актуализация предварительно установленного значения: 1 = записать новое значение в предварительную установку HSC2.
SM57.6	HSC2 - Актуализация текущего значения: 1 = записать новое значение в текущее значение HSC2.
SM57.7	HSC2 - Бит разблокировки: 1 = разблокировать
SMB58 SMB59 SMB60 SMB61	HSC2 - Новое текущее значение. SMB58 является старшим байтом, а SMB61 - младшим байтом.
SMB62 SMB63 SMB64 SMB65	HSC2 - Новое предварительно установленное значение. SMB62 является старшим байтом, а SMB65 - младшим байтом.

### SMB66 - SMB85: Регистр РТО/PWM

Как описано в таблице D–15, байты SMB66 - SMB85 служат для контроля операций и управления функциями “Последовательность импульсов” (РТО) и “Широтно-импульсная модуляция” (PWM). Подробную информацию по этим битам Вы найдете в описании быстрых выходных операций в главе 9.

Таблица D–15. Специальные меркеры SMB66 - SMB85

Специальные меркеры	Описание
SM66.0 - SM66.5	Резервные.
SM66.6	РТО0 - Переполнение конвейера: 0 = нет переполнения; 1 = переполнение
SM66.7	РТО0 - Бит холостого хода: 0 = РТО обрабатывается; 1 = РТО в состоянии холостого хода
SM67.0	РТО0/PWM0 - Актуализация времени цикла: 1 = записать новое время цикла

Таблица D-15. Специальные меркеры SMB66 - SMB85, продолжение

Специальные меркеры	Описание
SM67.1	PWMO- Актуализация длительности импульса: 1 = записать новую длительность импульса
SM67.2	PTO0 - Актуализация значения счетчика импульсов: 1 = записать новое значение счетчика импульсов
SM67.3	PTO0/PWM0 - База времени: 0 = 1 мкс/такт; 1 = 1 мс/такт
SM67.4 и SM67.5	Резервные.
SM67.6	PTO0/PWM0 - Выбор режима: 0 = PTO; 1 = PWM
SM67.7	PTO0/PWM0 - Разблокировка: 1 = разблокировать
SMB68 SMB69	PTO0/PWM0 - Время цикла SMB68 является старшим байтом, а SMB69 - младшим байтом.
SMB70 SMB71	PWM0 - Длительность импульса SMB70 является старшим байтом, а SMB71 - младшим байтом.
SMB72 SMB73 SMB74 SMB75	PTO0 - Значение счетчика импульсов SMB72 является старшим байтом, а SMB75 - младшим байтом.
SM76.0 - SM76.5	Резервные.
SM76.6	PTO1 - Переполнение конвейера: 0 = нет переполнения; 1 = переполнение
SM76.7	PTO1 - Бит холостого хода: 0 = PTO обрабатывается; 1 = PTO в состоянии холостого хода
SM77.0	PTO1/PWM1 - Актуализация времени цикла: 1 = записать новое время цикла
SM77.1	PWM1 - Актуализация длительности импульса: 1 = записать новую длительность импульса
SM77.2	PTO1 - Актуализация значения счетчика импульсов: 1 = записать новое значение счетчика импульсов
SM77.3	PTO1/PWM1 - База времени: 0 = 1 мкс/такт; 1 = 1 мс/такт
SM77.4 и SM77.5	Резервные.
SM77.6	PTO1/PWM1 - Выбор режима: 0 = PTO; 1 = PWM
SM77.7	PTO1/PWM1 - Разблокировка: 1 = разблокировать
SMB78 SMB79	PTO1/PWM1 - Время цикла SMB78 является старшим байтом, а SMB79 - младшим байтом.
SMB80 SMB81	PWM1 - Длительность импульса SMB80 является старшим байтом, а SMB81 - младшим байтом.
SMB82 SMB83 SMB84 SMB85	PTO1 - Значение счетчика импульсов SMB82 является старшим байтом, а SMB85 - младшим байтом.







# Преобразование программ из STEP 7– Micro/DOS в STEP 7–Micro/WIN

# E

С помощью STEP 7–Micro/WIN Вы можете импортировать программы, созданные с помощью программного обеспечения STEP 7–Micro/DOS, в проекты в STEP 7–Micro/WIN.

## Импортирование программ из STEP 7–Micro/DOS

Для импортирования программы из STEP 7–Micro/DOS в проект в STEP 7–Micro/WIN действуйте следующим образом:

1. Выберите пункт меню **Projekt** → **Öffnen** [**Проект** → **Открыть**].
2. В поле "Dateiformat" ["Формат файла"] выберите файлы проекта STEP 7–Micro/DOS (\*.vpu).
3. Выберите каталог в STEP 7–Micro/DOS, в котором записана программа, которую Вы хотите импортировать. Дважды щелкните мышью по каталогу, чтобы отобразить содержание каталога в списке слева (см. рис. E–1).
4. Выберите в списке программу или введите имя программы в поле "Dateiname" ["Имя файла"].
5. Щелкните мышью на кнопке "OK". В ответ на это открываются программа из STEP 7–Micro/DOS и соответствующие файлы в качестве безымянного проекта.

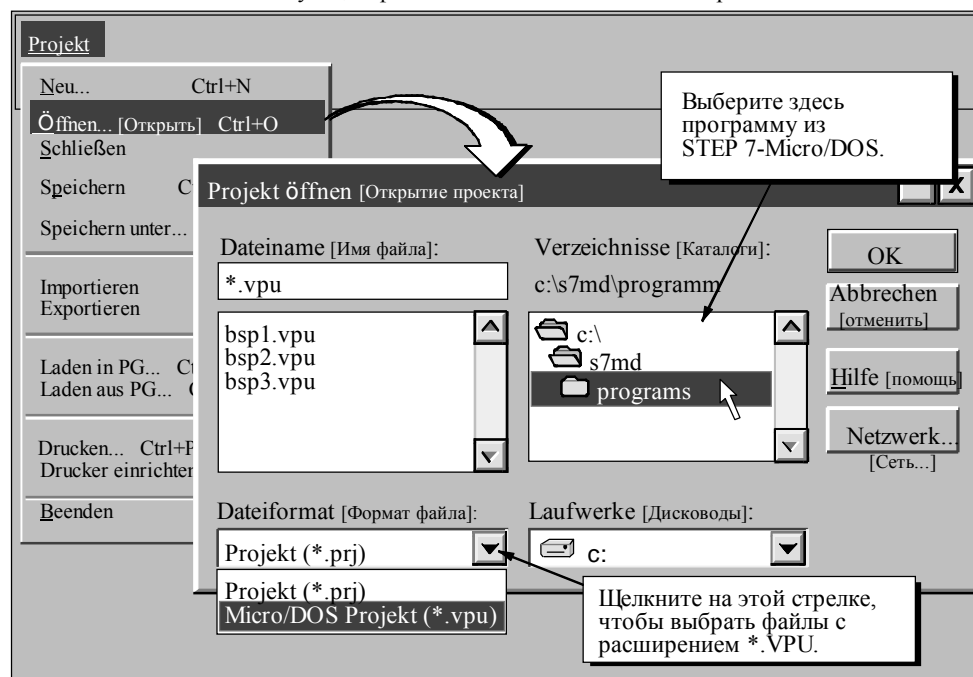


Рис. E-1. Импортирование программ из STEP 7-Micro/DOS в STEP 7-Micro/WIN

### Правила и ограничения при импортировании

Если Вы импортируете программный файл Micro/DOS.VPU, то копии следующих файлов преобразуются из STEP 7–Micro/DOS в STEP 7–Micro/WIN после их сохранения:

- программные файлы
- память переменных и данные
- синонимы и дескрипторы
- таблица состояний с тем же именем, что и проект.

При импортировании программ из STEP 7–Micro/DOS в проект в STEP 7–Micro/WIN протекают следующие процессы:

- Константы, которые были определены в памяти переменных, остаются неизменными.
- Синонимы из STEP 7–Micro/DOS преобразуются в символы STEP 7–Micro/WIN и, если требуется, сокращаются до максимальной длины 23 знака. Дескрипторы синонимов, которые могут включать в себя до 144 знаков, сокращаются в STEP 7–Micro/WIN до максимальной длины 79 символов на комментарий к символу.
- Комментарии к сегментам STEP 7–Micro/DOS (максимум 16 строк по 60 знаков) остаются неизменными в редакторах AWL и KOP.
- Таблица состояний из STEP 7–Micro/DOS, имеющая такое же имя, как и программа в STEP 7–Micro/DOS, преобразуется в таблицу состояний STEP 7–Micro/WIN. Например, если существуют программа с именем TEST.VPU и две таблицы состояний к ней TEST.CH2 и TEST2.CH2, то таблица состояний с именем TEST импортируется, а таблица состояний с именем TEST2, в отличие от этого, не импортируется.
- Сетевой адрес, пароль, установки выходов и реманентные области настраиваются соответственно файлам из STEP 7–Micro/DOS. Вы можете вызвать отображение этих параметров с помощью пункта меню CPU → **Konfigurieren...** [CPU → **Конфигурирование...**].

### Запись импортированной программы в память

Если Вы хотите сохранить импортированную программу в том же каталоге, в котором хранятся Ваши текущие проекты в STEP 7–Micro/WIN, то действуйте следующим образом:

1. Выберите пункт меню **Projekt** → **Speichern unter...** [**Проект** → **Сохранить как...**] и выберите в диалоговом окне "Speichern unter" ["Сохранить как"] текущий каталог в STEP 7–Micro/WIN.
2. В поле "Dateiname" ["Имя файла"] укажите имя, которое хотите присвоить импортированному программному файлу. Используйте расширение .PRJ.
3. Подтвердите, щелкнув на "OK".

---

#### Указание

Если Вы записали в память или изменили импортированную программу, то Вы не можете снова экспортировать ее в формате STEP 7–Micro/DOS. Однако первоначальные файлы в STEP 7–Micro/DOS не изменяются. Вы можете и дальше использовать их в STEP 7–Micro/DOS.

---

# Времена выполнения команд AWL

## Влияние потока сигнала на времена выполнения



Основное время выполнения команды AWL (см. табл. F-4) - это время, требующиеся для выполнения логики или функции команды, когда поток сигнала присутствует (когда значение в вершине стека равно "1" или включено). Выполнение некоторых операций зависит от того, имеется ли в наличии поток сигнала, то есть CPU обрабатывает функцию только тогда, когда активизирован поток сигнала операции (когда в вершине стека находится "1"). Если поток сигнала отсутствует (в вершине стека "0"), то для расчета времени обработки Вашей программы Вам нужно использовать времена выполнения "без потока сигнала". В табл. F-1 показано время выполнения команд AWL без потока сигнала (когда в вершине стека находится "0") для отдельных CPU S7-200.

Таблица F-1. Времена выполнения команд без потока сигнала

Команда без потока сигнала	CPU 212	CPU 214/ 215/216
Все команды AWL	10 мкс	6 мкс

## Влияние косвенной адресации на времена выполнения

Основное время выполнения команды AWL (см. табл. F-4) - это время, требующееся для выполнения команды, когда операнды или константы адресуются прямо. Если Вы используете в Вашей программе косвенную адресацию, то Вам нужно прибавить к времени выполнения указанное в таблице F-2 значение на каждый косвенно адресованный операнд.

Таблица F-2. Дополнительное время, которое должно прибавляться при косвенной адресации

Операция с косвенной адресацией	CPU 212	CPU 214/ 215/216
Все операции, кроме R, RI, S и SI	76 мкс	47 мкс
R, RI, S и SI	185,3 мкс	120,2 мкс

## Влияние аналоговых входов и выходов на времена выполнения

Доступ к аналоговым входам и выходам влияет на время выполнения команды. В табл. F-3 показаны минимальный и максимальный факторы, которые должны прибавляться к основному времени выполнения команды, когда команда обращается к аналоговому значению. Этот фактор определяется разрешающей способностью цифрового значения.

## Времена выполнения AWL-операций

Таблица F-3. Влияние аналоговых входов и выходов на времена выполнения

Аналоговые входы и выходы, влияющие на времена выполнения		CPU 212		CPU 214/215/216	
		Мин.	Макс.	Мин.	Макс.
<b>Аналоговые входы</b>	Преобразователь, 8 бит	117 мкс	192 мкс	78 мкс	153 мкс
	Преобразователь, 12 бит	96 мкс	171 мкс	64 мкс	139 мкс
	Преобразователь, 16 бит	72 мкс	147 мкс	48 мкс	123 мкс
<b>Аналоговые выходы</b>	Преобразователь, 8 бит	-	121 мкс	-	80 мкс
	Преобразователь, 12 бит	-	99 мкс	-	66 мкс
	Преобразователь, 16 бит	-	75 мкс	-	50 мкс

### Основные времена выполнения команд AWL

В табл. F-4 приведены основные времена выполнения отдельных команд AWL для различных CPU S7-200.

Таблица F-4. Времена выполнения команд AWL (в мкс)

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
=	Основное время выполнения: E, A M SM, T, Z, V, S	1,3 5,2 6,5	0,8 3,2 4,0	0,8 3,2 4,0	0,8 3,2 4,0
+D	Основное время выполнения	154	95	95	95
-D	Основное время выполнения	156	96	96	96
+I	Основное время выполнения	119	73	73	73
-I	Основное время выполнения	120	74	74	74
=I	Основное время выполнения	68	42	42	42
+R	Основное время выполнения Максимальное время выполнения	-	220 350	220 350	220 350
-R	Основное время выполнения Максимальное время выполнения	-	225 355	225 355	225 355
*R	Основное время выполнения Максимальное время выполнения	-	255 320	255 320	255 320
/R	Основное время выполнения Максимальное время выполнения	-	810 870	810 870	810 870
AE	Основное время выполнения	59	36	36	36
ATCH	Основное время выполнения	52	32	32	32
ATH	Общее = Основное время выполнения + (Длина)* (Умножитель длины) Основное время выполнения Умножитель длины	791 67	486 41	486 41	486 41
ATT	Основное время выполнения	-	268	268	268
BCDI	Основное время выполнения	271	166	166	166

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212	CPU 214	CPU 215	CPU 216
----------	----------	---------	---------	---------	---------

Времена выполнения AWL-операций

		(в мкс)	(в мкс)	(в мкс)	(в мкс)
BMB	Общее = Основное время выполнения + (Длина)* (Умножитель длины) Основное время выполнения Умножитель длины	686 34	422 21	422 21	422 21
BMD	Общее = Основное время выполнения + (Длина)* (Умножитель длины) Основное время выполнения Умножитель длины	- -	- -	446 43	446 43
BMW	Общее = Основное время выполнения + (Длина)* (Умножитель длины) Основное время выполнения Умножитель длины	691 56	424 34	424 34	424 34
CALL	Основное время выполнения	38	23	23	23
CRET	Основное время выполнения	28	17	17	17
CRETI	Основное время выполнения	82	50	50	50
DECB	Основное время выполнения	-	-	37	37
DECD	Основное время выполнения	105	65	65	65
DECO	Основное время выполнения	91	56	56	56
DECW	Основное время выполнения	90	55	55	55
DISI	Основное время выполнения	28	24	24	24
DIV	Основное время выполнения	444	273	273	273
DTCH	Основное время выполнения	42	26	26	26
DTR	Основное время выполнения Максимальное время выполнения	-	108 135	108 135	108 135
ED	Основное время выполнения	34	21	21	21
ENCO	Минимальное время выполнения Максимальное время выполнения	81 100	50 62	50 62	50 62
END	Основное время выполнения	2,0	1,2	1,2	1,2
ENI	Основное время выполнения	38	24	24	24
EU	Основное время выполнения	35	21	21	21
FIFO	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	234 29	234 29	234 29
FILL	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	627 20	385 12	385 12	385 12
FND <	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	424 28	424 28	424 28
FND <>	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	423 29	423 29	423 29

## Времена выполнения AWL-операций

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
FND =	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	431 25	431 25	431 25
FND >	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	428 28	428 28	428 28
FOR	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	135 0,12	135 0,12	135 0,12
HDEF	Основное время выполнения	86	53	53	53
HSC	Основное время выполнения	107	67	67	67
HTA	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	778 38	476 23	476 23	476 23
IBCD	Основное время выполнения	202	124	124	124
INCB	Основное время выполнения	-	-	34	34
INCD	Основное время выполнения	104	64	64	64
INCW	Основное время выполнения	89	54	54	54
INT	Типовое время выполнения с прерыванием	196	120	120	120
INVB	Основное время выполнения	-	-	40	40
INVD	Основное время выполнения	107	66	66	66
INVW	Основное время выполнения	92	56	56	56
JMP	Основное время выполнения	1,3	0,8	0,8	0,8
LBL	Основное время выполнения	0	0	0	0
LD	Основное время выполнения: E, A M SM, T, Z, V, S	1,3 3,2 5,2	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
LDB <=	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	69 72	42 44	42 44	42 44
LDB =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	69 72	42 44	42 44	42 44
LDB >=	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	69 72	42 44	42 44	42 44
LDD <=	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	147 150	90 92	90 92	90 92
LDD =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	147 150	90 92	90 92	90 92
LDD > =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	147 150	90 92	90 92	90 92
LDI	Основное время выполнения	53	33	33	33



Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
LDN	Основное время выполнения: E, A M SM, T, Z, V, S	2,0 3,9 5,9	1,2 2,4 3,6	1,2 2,4 3,6	1,2 2,4 3,6
LDNI	Основное время выполнения	54	33	33	33
LDR=	Основное время выполнения	-	98	98	98
LDR<=	Основное время выполнения	-	98	98	98
LDR>=	Основное время выполнения	-	98	98	98
LDW <=	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	117 120	72 74	72 74	72 74
LDW =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	117 120	72 74	72 74	72 74
LDW >=	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	117 120	72 74	72 74	72 74
LIFO	Основное время выполнения	-	261	261	261
LPP	Основное время выполнения	0,7	0,4	0,4	0,4
LPS	Основное время выполнения	1,3	0,8	0,8	0,8
LRD	Основное время выполнения	0,7	0,4	0,4	0,4
LSCR	Основное время выполнения	18	12	12	12
MEND	Основное время выполнения	1,3	0,8	0,8	0,8
MOVB	Основное время выполнения	48	30	30	30
MOVD	Основное время выполнения	89	54	54	54
MOVR	Основное время выполнения	89	54	54	54
MOVW	Основное время выполнения	72	44	44	44
MUL	Основное время выполнения	228	140	140	140
NEXT	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	135 0,12	135 0,12	135 0,12
NETR	Основное время выполнения	-	478	478	478
NETW	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	-	460 16,8	460 16,8	60 16,8
NOP	Основное время выполнения	0	0	0	0
NOT	Основное время выполнения	1,3	0,8	0,8	0,8
O	Основное время выполнения: E, A M SM, T, Z, V, S	1,3 3,2 5,2	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
OB < =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	70 73	43 45	43 45	43 45
OB =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	70 73	43 45	43 45	43 45

## Времена выполнения AWL-операций

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
OB > =	Время выполнения, если сравнение истинно	70	43	43	43
	Время выполнения, если сравнение ложно	73	45	45	45
OD < =	Время выполнения, если сравнение истинно	149	92	92	92
	Время выполнения, если сравнение ложно	151	93	93	93
OD =	Время выполнения, если сравнение истинно	149	92	92	92
	Время выполнения, если сравнение ложно	151	93	93	93
OD > =	Время выполнения, если сравнение истинно	149	92	92	92
	Время выполнения, если сравнение ложно	151	93	93	93
OI	Основное время выполнения	58	36	36	36
OLD	Основное время выполнения	1,3	0,8	0,8	0,8
ON	Основное время выполнения: E, A	1,3	0,8	0,8	0,8
	M	3,2	2,0	2,0	2,0
	SM, T, Z, V, S	5,2	3,2	3,2	3,2
ONI	Основное время выполнения	59	36	36	36
OR=	Основное время выполнения	-	98	98	98
OR<=	Основное время выполнения	-	98	98	98
OR>=	Основное время выполнения	-	98	98	98
ORB	Основное время выполнения	-	-	49	49
ORD	Основное время выполнения	148	91	91	91
ORW	Основное время выполнения	119	73	73	73
OW < =	Время выполнения, если сравнение истинно	118	72	72	72
	Время выполнения, если сравнение ложно	121	74	74	74
OW =	Время выполнения, если сравнение истинно	118	72	72	72
	Время выполнения, если сравнение ложно	121	74	74	74
OW > =	Время выполнения, если сравнение истинно	118	72	72	72
	Время выполнения, если сравнение ложно	121	74	74	74
PID	Основное время выполнения	-	-	2000	2000
	Слагаемое к перерасчету ( $K_c \cdot T_s / T_i$ ) и ( $K_c \cdot T_d / T_s$ ) перед расчетом PID-регулятора. Перерасчет производится, если значение $K_c$ , $T_d$ , $T_i$ , или $T_s$ изменилось после предыдущего выполнения операции или при переходе на автоматический режим.	-	-	2600	2600
PLS	Основное время выполнения	-	153	153	153

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
R	Общее = Время для операндов + (LM)*(Длина)				
	Время выполнения для счетчика	36,9	22,6	22,6	22,6
	Время выполнения для таймера	35,7	21,9	21,9	21,9
	Время выполнения для других операндов	43,4	26,6	26,6	26,6
	Мультипликатор длины для счетчика (LM)	31,3	19,2	19,2	19,2
	Мультипликатор длины для таймера (LM)	54,0	33,1	33,1	33,1
	Мультипликатор длины для других операндов (LM)	43,4	3,7	3,7	3,7
Если длина хранится в переменной, а не является константой, увеличьте основное время выполнения путем сложения с:	119,3	73,2	73,2	73,2	
RCV	Основное время выполнения	-	-	126	126
RET	Основное время выполнения	29	18	18	18
RETI	Основное время выполнения	82	50	50	50
RI	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	34,2	21,0	21,0	21,0
	Умножитель длины (LM)	65,2	40,0	40,0	40,0
	Если длина хранится в переменной, а не является константой, увеличьте основное время выполнения путем сложения с:	119,3	73,2	73,2	73,2
RLB	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	-	-	62	62
	Умножитель длины (LM)	-	-	1,2	1,2
RLD	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	140	86	86	86
	Умножитель длины (LM)	11,5	7,1	7,1	7,1
RLW	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	125	77	77	77
	Умножитель длины (LM)	7,4	4,6	4,6	4,6
RRB	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	-	-	62	62
	Умножитель длины (LM)	-	-	1,2	1,2
RRD	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	146	90	90	90
	Умножитель длины (LM)	11,2	6,9	6,9	6,9
RRW	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	128	78	78	78
	Умножитель длины (LM)	7,2	4,4	4,4	4,4
S	Общее = Основное время выполнения + (LM)* (Длина)				
	Основное время выполнения	40,8	25,0	25,0	25,0
	Умножитель длины (LM)	6,0	3,7	3,7	3,7
	Если длина хранится в переменной, а не является константой, увеличьте основное время выполнения путем сложения с:	119,3	73,2	73,2	73,2
SBR	Основное время выполнения	0	0	0	0
SCRE	Основное время выполнения	0	0	0	0
SCRT	Основное время выполнения	31	21	21	21

Времена выполнения AWL-операций

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение					
Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
SEG	Основное время выполнения	51	31	31	31
SHRB	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	486 2,4	299 1,5	299 1,5	299 1,5
SI	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM) Если длина хранится в переменной, а не является константой, увеличьте основное время выполнения путем сложения с:	34,2 63,0 119,3	21,0 38,4 73,2	21,0 38,4 73,2	21,0 38,4 73,2
SLB	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	- -	- -	64 1,6	64 1,6
SLD	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	142 9,6	87 5,9	87 5,9	87 5,9
SLW	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	128 5,6	79 3,4	79 3,4	79 3,4
SQRT	Основное время выполнения Максимальное время выполнения	-	1830 2110	1830 2110	1830 2110
SRB	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	- -	- -	64 1,6	64 1,6
SRD	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	148 9,3	91 5,7	91 5,7	91 5,7
SRW	Общее = Основное время выполнения + (LM)* (Длина) Основное время выполнения Умножитель длины (LM)	130 5,3	80 3,3	80 3,3	80 3,3
STOP	Основное время выполнения	13,7	8,4	8,4	8,4
SWAP	Основное время выполнения	70	43	43	43
TODR	Основное время выполнения	-	282	282	282
TODW	Основное время выполнения	-	489	489	489
TON	Основное время выполнения	51	32	32	32
TONR	Основное время выполнения	80	49	49	49
TRUNC	Основное время выполнения Максимальное время выполнения	-	258 420	258 420	258 420
U	Основное время выполнения: E, A M SM, T, Z, V, S	1,3 3,2 5,2	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
UB < =	Время выполнения, если сравнение истинно Время выполнения, если сравнение ложно	70 74	43 45	43 45	43 45

Таблица F-4. Времена выполнения команд AWL (в мкс), продолжение

Операция	Описание	CPU 212 (в мкс)	CPU 214 (в мкс)	CPU 215 (в мкс)	CPU 216 (в мкс)
UB =	Время выполнения, если сравнение истинно	70	43	43	43
	Время выполнения, если сравнение ложно	74	45	45	45
UB > =	Время выполнения, если сравнение истинно	70	43	43	43
	Время выполнения, если сравнение ложно	74	45	45	45
UD < =	Время выполнения, если сравнение истинно	148	91	91	91
	Время выполнения, если сравнение ложно	151	93	93	93
UD =	Время выполнения, если сравнение истинно	148	91	91	91
	Время выполнения, если сравнение ложно	151	93	93	93
UD > =	Время выполнения, если сравнение истинно	148	91	91	91
	Время выполнения, если сравнение ложно	151	93	93	93
ULD	Основное время выполнения	1,3	0,8	0,8	0,8
UN	Основное время выполнения: E, A M SM, T, Z, V, S	1,3	0,8	0,8	0,8
		3,2	2,0	2,0	2,0
		5,2	3,2	3,2	3,2
UNDD	Основное время выполнения	148	91	91	91
UNDW	Основное время выполнения	119	73	73	73
UNI	Основное время выполнения	59	36	36	36
UR=	Основное время выполнения	-	98	98	98
UR<=	Основное время выполнения	-	98	98	98
UR>=	Основное время выполнения	-	98	98	98
UW < =	Время выполнения, если сравнение истинно	119	73	73	73
	Время выполнения, если сравнение ложно	122	75	75	75
UW=	Время выполнения, если сравнение истинно	119	73	73	73
	Время выполнения, если сравнение ложно	122	75	75	75
UW > =	Время выполнения, если сравнение истинно	119	73	73	73
	Время выполнения, если сравнение ложно	122	75	75	75
WDR	Основное время выполнения	23	14	14	14
XMT	Основное время выполнения	295	181	181	181
XORB	Основное время выполнения	-	-	49	49
XORD	Основное время выполнения	148	91	91	91
XORW	Основное время выполнения	119	73	73	73
ZV	Основное время выполнения	84	52	52	52
ZVR	Основное время выполнения	114	70	70	70

## *Времена выполнения AWL-операций*

## S7–200 Номера заказа



<b>CPU</b>	<b>Номер заказа</b>
CPU 212, источник питания пост. тока, входы пост. тока, выходы пост. тока	6ES7 212-1AA01-0XB0
CPU 212, источник питания перемен. тока, входы пост. тока, релейные выходы	6ES7 212-1BA01-0XB0
CPU 212, источник питания перемен. тока, входы перемен. тока, выходы перемен. тока	6ES7 212-1CA01-0XB0
CPU 212, источник питания перемен. тока, входы пост. тока (токовыводящие), релейные выходы	6ES7 212-1BA10-0XB0
CPU 212, источник питания перемен. тока, входы перемен. тока 24 В, выходы перемен. тока	6ES7 212-1DA01-0XB0
CPU 214, источник питания пост. тока, входы пост. тока, выходы пост. тока	6ES7 214-1AC01-0XB0
CPU 214, источник питания перемен. тока, входы пост. тока, релейные выходы	6ES7 214-1BC01-0XB0
CPU 214, источник питания перемен. тока, входы перемен. тока, выходы перемен. тока	6ES7 214-1CC01-0XB0
CPU 214, источник питания перемен. тока, входы пост. тока (токовыводящие), релейные выходы	6ES7 214-1BC10-0XB0
CPU 214, источник питания перемен. тока, входы перемен. тока 24 В, выходы перемен. тока	6ES7 214-1DC01-0XB0
CPU 215, источник питания пост. тока, входы пост. тока, выходы пост. тока	6ES7 215-2AD00-0XB0
CPU 215, источник питания перемен. тока, входы пост. тока, релейные выходы	6ES7 215-2BD00-0XB0
CPU 216, источник питания пост. тока, входы пост. тока, выходы пост. тока	6ES7 216-2AD00-0X00
CPU 216, источник питания перемен. тока, входы пост. тока, релейные выходы	6ES7 216-2BD00-0X00

<b>Модули расширения</b>	<b>Номер заказа</b>
EM 221, цифровой ввод 8 x 24 В пост. тока	6ES7 221-1BF00-0XA0
EM 221, цифровой ввод 8 x 120 В перемен. тока	6ES7 221-1EF00-0XA0
EM 221, цифровой ввод 8 x 24 В пост. тока (токовыводящий)	6ES7 221-1BF10-0XA0
EM 221, цифровой ввод 8 x 24 В перемен. тока	6ES7 221-1JF00-0XA0
EM 222, цифровой вывод 8 x 24 В пост. тока	6ES7 222-1BF00-0XA0
EM 222, цифровой вывод 8 x реле	6ES7 222-1HF00-0XA0
EM 222, цифровой вывод 8 x 120/230 В перемен. тока	6ES7 222-1EF00-0XA0
EM 223, цифровой ввод/вывод, 4 входа 24 В пост. тока /4 выхода 24 В пост. тока	6ES7 223-1BF00-0XA0

Модули расширения	Номер заказа
EM 223, цифровой ввод/вывод, 4 входа 24 В пост. тока /4 релейных выхода	6ES7 223-1HF00-0XA0
EM 223, цифровой ввод/вывод 4 входа 120 В перем. тока /4 выхода 120/230 В перем. тока	6ES7 223-1EF00-0XA0
EM 223, цифровой ввод/вывод 8 входов 24 В пост. тока /8 релейных выходов	6ES7 223-1PH00-0XA0
EM 223, цифровой ввод/вывод 16 входов 24 В пост. тока /16 релейных выходов	6ES7 223-1PL00-0XA0
EM 231, аналоговый ввод, AI 3 x 12 бит	6ES7 231-0HC00-0XA0
EM 235, аналоговый ввод/вывод AI 3/AO 1 x 12 бит	6ES7 235-0KD00-0XA0

Общее	Номер заказа
Модуль памяти 8 К x 8	6ES7 291-8GC00-0XA0
Модуль памяти 16 К x 8	6ES7 291-8GD00-0XA0
Батарейный модуль	6ES7 291-8BA00-0XA0
Кабель PC/PPI	6ES7 901-3BF00-0XA0
Кабель для модулей расширения	6ES7 290-6BC50-0XA0
Клеммы заземления для профильной шины	6ES5 728-8MA11
Клеммный блок с 12 соединительными зажимами для полевого монтажа проводки	6ES7 290-2AA00-0XA0
Клеммный блок с 14 соединительными зажимами для полевого монтажа проводки	6ES7 290-2CA00-0XA0
Клеммный блок с 18 соединительными зажимами для полевого монтажа проводки	6ES7 290-2BA00-0XA0
Имитатор входных сигналов пост. тока для CPU 212	6ES7 274-1XF00-0XA0
Имитатор входных сигналов пост. тока для CPU 214	6ES7 274-1XH00-0XA0
Имитатор входных сигналов пост. тока для CPU 215/216	6ES7 274-1XK00-0XA0
Шинный штекер с портом для устройства программирования	6ES7 972-0BB10-0XA0
Шинный штекер без порта для устройства программирования	6ES7 972-0BA10-0XA0
Соединительный кабель	6XV1 830-0AH10
Плата MPI	6ES7 793-2AA00-0AA0

Программное обеспечение для программирования	Номер заказа
STEP 7-Micro/WIN Отдельная лицензия	6ES7 810-2AA00-0YX0
STEP 7-Micro/WIN Лицензия на копирование	6ES7 810-2AA00-0YX1
STEP 7-Micro/WIN Update (обновленная редакция)	6ES7 810-2AA00-0YX3

Руководства	Номер заказа
Контроллер S7–200, Системное руководство	6ES7 298-8FA00-8BH0
Децентрализованное периферийное устройство ET 200, Руководство	6ES5 998-3ES22