

SIMATIC

Модуль последовательного интерфейса ET 200S

Руководство

Содержание

Обзор **1**

Модуль последовательного
интерфейса **2**

Modbus/USB **3**

Предметный указатель

Настоящее руководство является
частью пакета документации, определяемой
заказным номером

6ES7151-1AA00-8BA0

Выпуск 10/2001

A5E00124881-01

Классификация указаний по безопасности

В настоящем руководстве содержатся указания, на которые следует обратить внимание в целях обеспечения собственной безопасности, равно как и безопасности и сохранности оборудования. Эти указания помечаются в руководстве с помощью предупреждающих знаков (треугольник с восклицательным знаком). По степени важности различают следующие указания:



Опасность

Указывает, что несоблюдение надлежащих мер предосторожности **приведет** к смерти, серьезной травме или значительному материальному ущербу.



Предупреждение

Указывает, что несоблюдение надлежащих мер предосторожности **может привести** к смерти, серьезной травме или значительному материальному ущербу.



Предостережение

Указывает, что несоблюдение надлежащих мер предосторожности может привести к травме или материальному ущербу.

Предостережение

Указывает, что несоблюдение надлежащих мер предосторожности может привести к материальному ущербу.

Примечание

Привлекает внимание читателя к особенно важной и полезной информации о продукте, обращении с ним или к определенной части документации.

Квалифицированный персонал

К монтажу и эксплуатации данного оборудования может допускаться только **квалифицированный персонал**. В данном руководстве под квалифицированным персоналом понимаются лица, имеющие допуск к выполнению работ по вводу в эксплуатацию, заземлению и маркировке электрических цепей, устройств и систем в соответствии с установленными правилами и стандартами безопасности.

Эксплуатация по назначению

Имейте в виду следующее:



Предупреждение

Торговые знаки

SIMATIC®, SIMATIC HMI®, SIMATIC NET® - зарегистрированные торговые знаки фирмы SIEMENS AG.

Третьи лица, использующие в своих целях любые другие наименования, приводимые в настоящем документе и относящиеся к торговым знакам, могут быть привлечены к ответственности за нарушение прав владельцев торговых знаков.

Авторские права © Siemens AG 2001. Все права сохраняются

Воспроизведение, передача или использование настоящего документа или его части допускается лишь с письменного разрешения. Нарушители будут привлекаться к ответственности за нанесенные убытки. Все права, включая права, вытекающие из патента или регистрации промышленной модели или разработки, зарезервированы.

Siemens AG
Bereich Automatisierungstechnik
Geschäftsgebiet Industrie-Automatisierung
Postfach 4848, D-90327 Nürnberg

Ответственность

Содержание данного руководства было проверено на соответствие описанным в нем техническим и программным продуктам. Поскольку возможные изменения в последних невозможно предвидеть в полном объеме, полное соответствие не может быть гарантировано. Материалы данного руководства регулярно проверяются, а необходимые изменения вносятся в последующие выпуски. Мы рады любым предложениям по улучшению качества наших руководств.

Выпуск **10/2001**

© Siemens AG 2001

Технические данные могут быть изменены.

Содержание

1	Обзор	1–1
2	Модуль последовательного интерфейса	2–1
2.1	Обзор изделия	2–2
2.2	Краткое описание ввода в эксплуатацию модуля последовательного интерфейса	2–6
2.3	Структурные схемы с указанием назначения контактов	2–14
2.4	Интерфейс RS–232C	2–20
2.5	Интерфейс RS–422/485	2–21
2.6	Основные принципы последовательной передачи данных	2–22
2.7	Передача данных с использованием протокола 3964(R)	2–32
2.8	Передача данных с помощью ASCII драйвера	2–41
2.9	Конфигурирование и параметрирование модуля последовательного интерфейса	2–52
2.10	Обмен данными с использованием функциональных блоков	2–58
2.11	Выполняемые функции и режимы при запуске	2–76
2.12	Справочные данные по другим ведущим устройствам (не S7–PROFIBUS)	2–79
2.13	Диагностика	2–92
2.14	Технические характеристики	2–99
3	Modbus/US\$	3–1
3.1	Обзор изделия	3–2
3.2	Схемы с указанием назначения контактов	3–5
3.2.1	Назначение клемм	3–5
3.2.2	Интерфейс RS–232C	3–10
3.2.3	Интерфейс RS–422/485	3–13
3.3	Протокол связи Modbus	3–14
3.3.1	Структура сообщения	3–14
3.3.2	Адрес ведомого устройства	3–14
3.3.3	Функции ведущего и ведомого устройств	3–15
3.3.4	Поле данных (ДАННЫЕ)	3–15
3.3.5	Завершение сообщения и контрольная сумма CRC	3–16
3.3.6	Ответы в случае ошибок	3–16
3.4	Драйвер Modbus-Master	3–18
3.4.1	Используемые интерфейсы и протоколы	3–18
3.4.2	Передача данных модулю ET 200S, являющемуся ведущим устройством Modbus	3–18
3.4.3	Конфигурирование и настройка параметров Modbus-Master	3–27
3.4.4	Коды функций, используемые Modbus-Master	3–30
3.4.5	Код функции 01 – Чтение логической ячейки	3–31

3.4.6	Код функции 02 – Чтение дискретного входа	3–32
3.4.7	Код функции 03 – Чтение регистров хранения	3–33
3.4.8	Код функции 04 – Чтение входных регистров	3–33
3.4.9	Код функции 05 – Изменение состояния логической ячейки	3–34
3.4.10	Код функции 06 – Запись нового значения в регистр	3–35
3.4.11	Код функции 07 – Чтение состояния ошибок	3–36
3.4.12	Код функции 08 – Проверка связи	3–36
3.4.13	Код функции 11 – Чтение счётчика событий связи	3–37
3.4.14	Код функции 12 – Чтение протокола событий связи	3–38
3.4.15	Код функции 15 – Изменение состояния нескольких логических ячеек	3–39
3.4.16	Код функции 16 – Запись новых значений в несколько регистров	3–40
3.5	Драйвер Modbus-Slave	3–41
3.5.1	Элементы канала связи Modbus-Slave (ведомое устройство Modbus)	3–41
3.5.2	Передача данных ведомому устройству Modbus ET 200S	3–42
3.5.3	Области данных в CPU SIMATIC	3–43
3.5.4	Конфигурирование параметров канала связи	3–45
3.5.5	Коды функций ведомого устройства	3–48
3.5.6	Код функции 01 – Чтение логической ячейки (выхода)	3–49
3.5.7	Код функции 02 – Чтение дискретного входа	3–52
3.5.8	Код функции 03 – Читать выходные регистры	3–55
3.5.9	Код функции 04 – Читать входные регистры	3–58
3.5.10	Код функции 05 – Изменение состояния логической ячейки	3–61
3.5.11	Код функции 06 – Запись нового значения в регистр	3–63
3.5.12	Код функции 08 – Проверка связи	3–65
3.5.13	Код функции 15 – Изменение состояния нескольких логических ячеек	3–66
3.5.14	Код функции 16 – Запись новых значений в несколько регистров	3–68
3.5.15	Преобразования в бит-ориентированных функциях	3–71
3.5.16	Преобразования в функциях, ориентированных на регистры	3–71
3.5.17	Разрешение/запрещение записи	3–73
3.5.18	Преобразование адресов Modbus для бит-ориентированных функций	3–74
3.5.19	Преобразование адресов Modbus для функций, предназначенных для регистров	3–79
3.5.20	Ограничения для функций записи	3–82
3.6	Диагностика	3–84
3.6.1	Диагностическая информация, предоставляемая светодиодами состояния	3–84
3.6.2	Структура диагностических сообщений функциональных блоков	3–85
3.6.3	Диагностика ведомых устройств PROFIBUS	3–94
3.6.4	Функции диагностики ведомого устройства Modbus	3–94
3.6.5	Ошибки	3–96
3.7	USS-Master (Ведущее устройство USS)	3–99
3.7.1	Протокол USS	3–99
3.7.2	Конфигурирование и параметрирование	3–100
3.7.3	Обзор функций	3–101
3.7.4	FC17 S_USST: Передача данных ведомому устройству	3–103
3.7.5	FC18 S_USSR: Прием данных от ведомого устройства	3–106
3.7.6	FC19 S_USSI: Инициализация	3–109
3.7.7	Блок неинтерпретируемых данных	3–112
3.7.8	Блок данных настройки параметров	3–118

3.7.9	Блок данных коммуникационного процессора	3–120
3.8	Процедура запуска модуля ET 200S SI Modbus/USS и его рабочие режимы	3–121
3.8.1	Загрузка конфигурации и параметров	3–121
3.8.2	Рабочие режимы модуля ET 200S SI Modbus/USS	3–122
3.8.3	Процедура запуска модуля ET 200S SI Modbus/USS	3–122
3.8.4	Действия модуля ET 200S SI Modbus/USS при изменении режимов работы CPU	3–123
3.9	Технические характеристики	3–125
Предметный указатель		Index–1

Что входит в состав руководства

Настоящий документ является дополнением к руководству "Станция распределенной периферии ET 200S".

В нем содержится описание отдельных модулей ET 200S, специально предназначенных для выполнения определенных функций.

Поиск информации в руководстве

В начале каждой главы приводится **Обзор изделия**, в котором перечисляются свойства и возможности применения описываемых модулей. Там же указывается заказной номер модуля, а также название и версия требуемого программного обеспечения, а также приводится интернет-адрес, по которому хранится текущий файл DDB.

В каждой главе также имеется раздел под названием **Краткие указания по вводу в эксплуатацию** для соответствующего модуля. В этих кратких указаниях приводится последовательность действий по монтажу и конфигурированию модуля, а также его интеграции в программу пользователя и проверки с помощью программы пользователя.

Предметный указатель

В предметном указателе содержатся термины и ключевые слова, встречающиеся в руководстве.

Модуль последовательного интерфейса

2

Обзор главы

В данной главе описывается модуль последовательного интерфейса, протоколы связи, поддерживаемые модулем, назначение модулю параметров, а также его интеграция в систему распределенного ввода/вывода ET 200S (система децентрализованной периферии). В настоящем документе освещаются следующие темы:

Раздел	Описание	Стр.
2.1	Обзор изделия	2–2
2.2	Краткое описание ввода в эксплуатацию модуля последовательного интерфейса	2–6
2.3	Структурные схемы с указанием назначения контактов	2–14
2.4	Интерфейс RS–232C	2–20
2.5	Интерфейс RS–422/485	2–21
2.6	Основные принципы последовательной передачи данных	2–22
2.7	Передача данных с использованием протокола 3964(R)	2–32
2.8	Передача данных с помощью ASCII драйвера	2–41
2.9	Конфигурирование и параметрирование модуля последовательного интерфейса	2–52
2.10	Обмен данными с использованием функциональных блоков	2–58
2.11	Выполняемые функции и режимы при запуске	2–76
2.12	Справочные данные по другим ведущим устройствам (не S7–PROFIBUS)	2–79
2.13	Диагностика	2–92
2.14	Технические характеристики	2–99

Подробную информацию по конфигурированию, настройке, выполнению кабельных и проводных соединений, по вводу в эксплуатацию и диагностике, а также технические характеристики системы распределенного ввода/вывода ET 200S можно найти в руководстве "Станция распределенной периферии ET 200S".

2.1 Обзор изделия

Заказной номер

6ES7 138-4DF00-0AB0

Описание изделия

Модуль последовательного интерфейса ET 200S является одним из встраиваемых модулей линейки ET 200S. Он предоставляет возможность последовательного обмена данными по одному из трех аппаратных протоколов (RS-232C, RS-422 и RS-485) с использованием одного из двух программных протоколов (ASCII и 3964(R)).

Интерфейсный модуль ET 200S можно использовать для организации обмена данными между программируемыми логическими контроллерами (ПЛК) или компьютерами по соединению "точка-точка". Обмен данными состоит из серии сеансов последовательной асинхронной передачи данных.

Тип связи выбирается при параметрировании модуля в программе конфигурирования аппаратных средств STEP 7 или с помощью другого приложения конфигурирования. В каталоге аппаратных средств отображаются следующие версии модуля:

- ASCII (8B)
- ASCII (4B)
- 3964R (8B)
- 3964R (4B)

Передача данных по 8 байт позволяет повысить скорость передачи, но требует большего адресного пространства ввода/вывода в корзине ET 200S. Передача данных по 4 байта требует меньшего адресного пространства ввода/вывода в корзине ET 200S, но скорость передачи при этом меньше. Выбор варианта модуля зависит от требований конкретного применения.

Функциональные возможности модуля последовательного интерфейса ET 200S

Модуль последовательного интерфейса ET 200S обладает следующими функциями:

- Встроенный интерфейс RS–232C, RS–422 или RS–485
- Скорость передачи до 19.2 кбит/с, полудуплексный режим
- В микропрограмму модуля встроены следующие протоколы связи
 - Протокол 3964(R)
 - ASCII драйвер

Функционирование драйверов зависит от параметров, назначенных модулю.

В таблице 2-1 перечислены функции драйверов при работе через соответствующие интерфейсы.

Таблица 2–1 Функции драйверов модуля последовательного интерфейса ET 200S			
Функция	RS–232C	RS–422	RS–485
ASCII драйвер	Да	Да	Да
Использование дополнительных сигналов RS–232C	Да	Нет	Нет
Управление/чтение дополнительных сигналов RS–232C с помощью функциональных блоков	Да	Нет	Нет
Управление потоком данных с помощью RTS/CTS	Да	Нет	Нет
Управление потоком данных с помощью XON/XOFF	Да	Да	Нет
Протокол 3964(R)	Да	Да	Нет

Обмен данными

Модуль последовательного интерфейса ET 200S позволяет устанавливать соединение "точка-точка" с различными модулями Siemens, а также устройствами других производителей, в том числе:

- с SIMATIC S5 с помощью драйвера 3964(R) с соответствующим интерфейсным модулем на стороне S5
- с терминалами сбора данных семейства ES2 производства Siemens с помощью драйвера 3964(R)
- с MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) и ES 030K с помощью драйвера 3964(R)
- с SIMOVERT и SIMOREG (протокол USS) с помощью ASCII драйвера (ET 200S SI RS 422/485) с соответствующей настройкой протокола в программе STEP 7
- с ПК по протоколу 3964(R) (для программирования на ПК имеются инструменты разработки: PRODAVE DOS 64R (6ES5 897-2UD11) для MS-DOS, PRODAVE WIN 64R (6ES5 897-2VD01) для Windows, или драйвер ASCII)
- со считывателями штрихкода с помощью драйвера 3964(R) или ASCII драйвера
- с программируемыми логическими контроллерами других производителей с помощью драйвера 3964(R) или ASCII драйвера
- с другими устройствами, имеющими простую структуру протокола, с помощью ASCII драйвера, путем соответствующей настройки протокола
- с другими устройствами, также имеющими драйвер 3964(R)

Светодиодные индикаторы

На передней панели интерфейсного модуля ET 200S имеются следующие светодиодные индикаторы состояний:

- SF (красный) Ошибка
- TX (зеленый) Передача по интерфейсу
- RX (зеленый) Прием по интерфейсу

Режимы и ошибки, индицируемые данными светодиодами, описаны в Разделе 2.13.

Передняя панель

На рисунке 2-1 показана маркировка, нанесенная на переднюю панель интерфейсного модуля ET 200S SI.

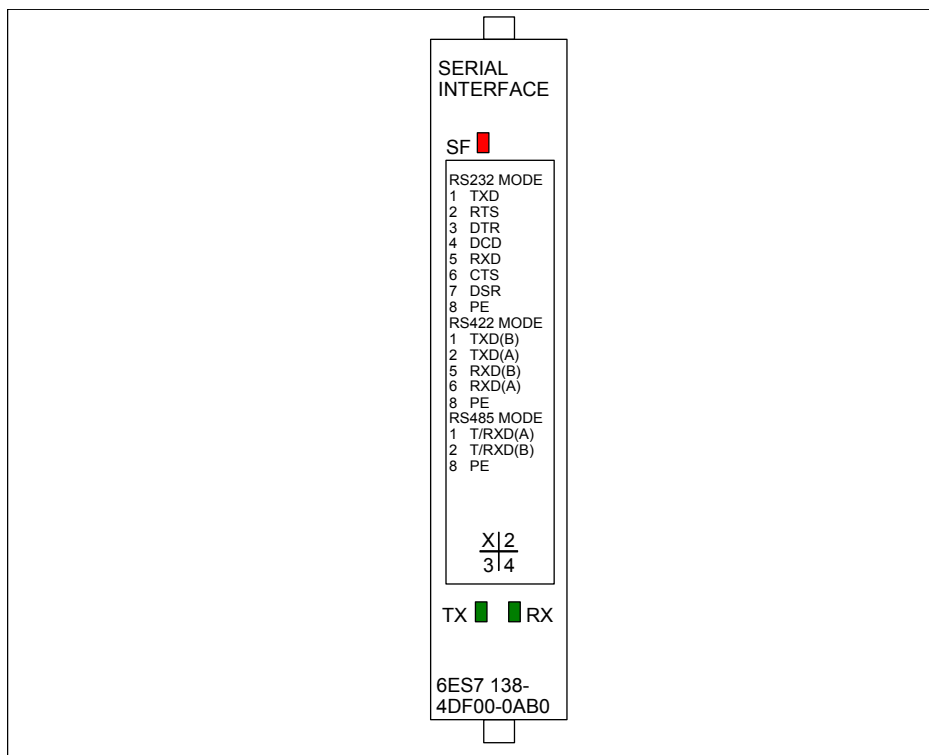


Рисунок 2–1 Маркировка на передней панели модуля последовательного интерфейса ET 200S

2.2 Краткие указания по вводу в эксплуатацию модуля последовательного интерфейса

Введение

В данном разделе на конкретном примере будет показано, как следует передавать и принимать данные между модулями последовательного интерфейса, как создать функционирующее приложение, как работают основные функции модуля последовательного интерфейса (аппаратные и программные), а также как проверить устройство и программу.

В качестве примера будут использоваться два модуля последовательного интерфейса ET 200S в режиме RS-232C ASCII.

Требования

Должны выполняться следующие требования:

- На S7 станции с ведущим устройством DP (DP-ведущим) должна быть приведена в рабочее состояние станция ET 200S.
- Необходимо наличие следующих компонентов:
 - Два терминальных модуля TM-E15S24-01
 - Два модуля последовательного интерфейса ET 200S
 - Материалы, необходимые для проводных соединений

Установка, проводные соединения и крепление

Установите два клеммных (терминальных) модуля ТМ-Е15S24-01 и подсоедините к ним провода (см. рисунок 2-2). Подсоедините два модуля последовательного интерфейса ET 200S к клеммным модулям (подробное описание данной процедуры можно найти в Главе 5 руководства "Станция распределенной периферии ET 200S").

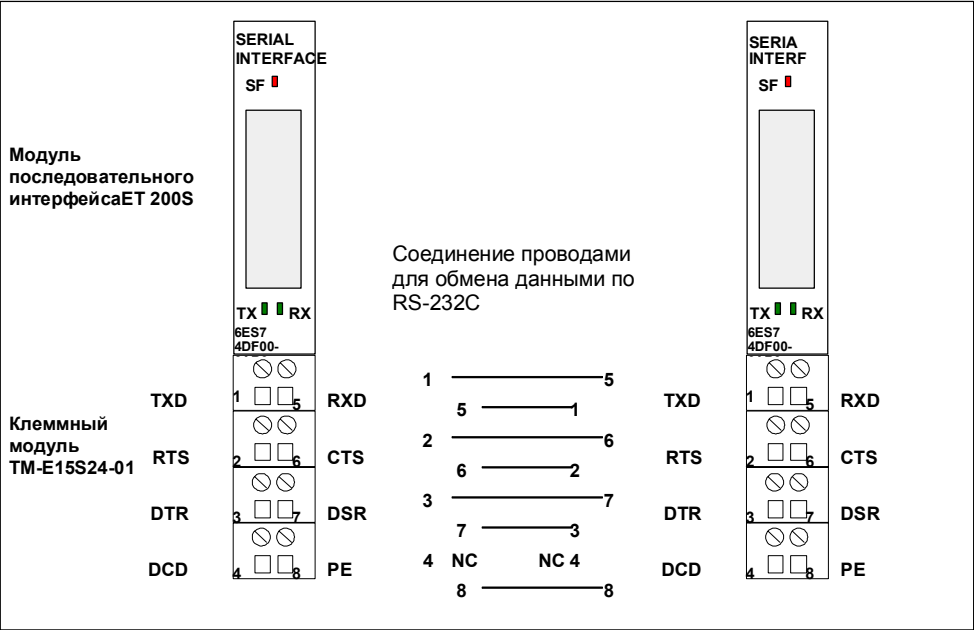


Рисунок 2-2 Назначение клемм, используемых в примере

Конфигурирование

Приступим к адаптации конфигурации аппаратных средств под существующую станцию ET 200S.

1. Откройте соответствующий проект в SIMATIC Manager и вызовите таблицу конфигурации в HWCONFIG.
2. Выберите в каталоге аппаратных средств модуль последовательного интерфейса ET 200S (6ES7 138-4DF00-0AB0 ASCII (8B)) и перетяните (с помощью мышки) данный номер на слот, в который был установлен первый модуль.
3. Двойным щелчком на данном номере откройте диалоговое окно DP Slave Properties (Свойства DP-ведомого).
4. Откройте вкладку Address/ID (Адрес/Идентификатор). Там будет указан начальный адрес для слота, в который был вставлен модуль. Запомните данный адрес, чтобы в дальнейшем использовать его при написании программы.
5. Повторите это действие для второго модуля.
6. Откройте вкладку Parameter Assignment (Назначение параметров). В этой вкладке содержатся детальные настройки модуля. Назначьте параметры двум модулям в соответствии с таблицей 2-2.
7. Загрузите конфигурацию в CPU в режиме STOP, выбрав команду меню **PLC → Download (ПЛК → Загрузить)**.

Таблица 2-2 Параметры, назначаемые в примере		
Параметры		Значения
Group diagnosis	Групповая диагностика	Disable (Отключено)
Interface	Интерфейс	RS232-C
Receive line initial state	Исходное состояние линии приема	R(A)5V / R(B)0V
Transmission rate	Скорость передачи	4800
Data bits	Количество битов данных	8
Stop bits	Количество стоп-битов	1
Parity	Четность	None (Нет)
Data flow control (default setting)	Управление потоком данных (значение по умолчанию)	None (Нет)
Indicator for end of receive message frame	Признак завершения принимаемого кадра сообщения	Expiration of character delay time (Превышение времени ожидания символа)
Character delay time (ms)	Время ожидания символа (мс)	10
End-of-text character 1	Символ завершения текста 1	3
End-of-text character 2	Символ завершения текста 2	0
Number of characters received	Количество принимаемых символов	100
Dynamic Message frame buffer	Динамический буфер кадров сообщений	Disable (Отключено)
Prevent message frame buffer overwrite	Предотвращать перезапись буфера кадров сообщений	Yes (Да)
Delete receive buffer at startup	Стирать буфер приема при запуске	Yes (Да)

Создание программы пользователя

Чтобы создать программу пользователя в STEP 7 для нашего примера, скопируйте FB3 S_SEND и FB2 S_RCV из библиотеки, которая входит в комплект поставки рассматриваемого модуля (см. примечание к таблице 2.12). В данной программе также потребуется создать экземплярный блок данных (DB) для каждого отдельного функционального блока (FB) при их вводе. Вводить следует те адреса, которые перечислены в разделе, посвященном конфигурированию аппаратных средств.

Введите следующую программу в OB1 (см. Рисунок 2–3). Выходные состояния FB3 и FB2 будем сохранять, поскольку они доступны лишь в течение одного цикла. (В нашем примере модуль, на который будут передаваться данные, имеет адрес 256, а модуль, от которого данные будут приниматься, имеет адрес 264. Обязательно используйте сконфигурированные адреса).

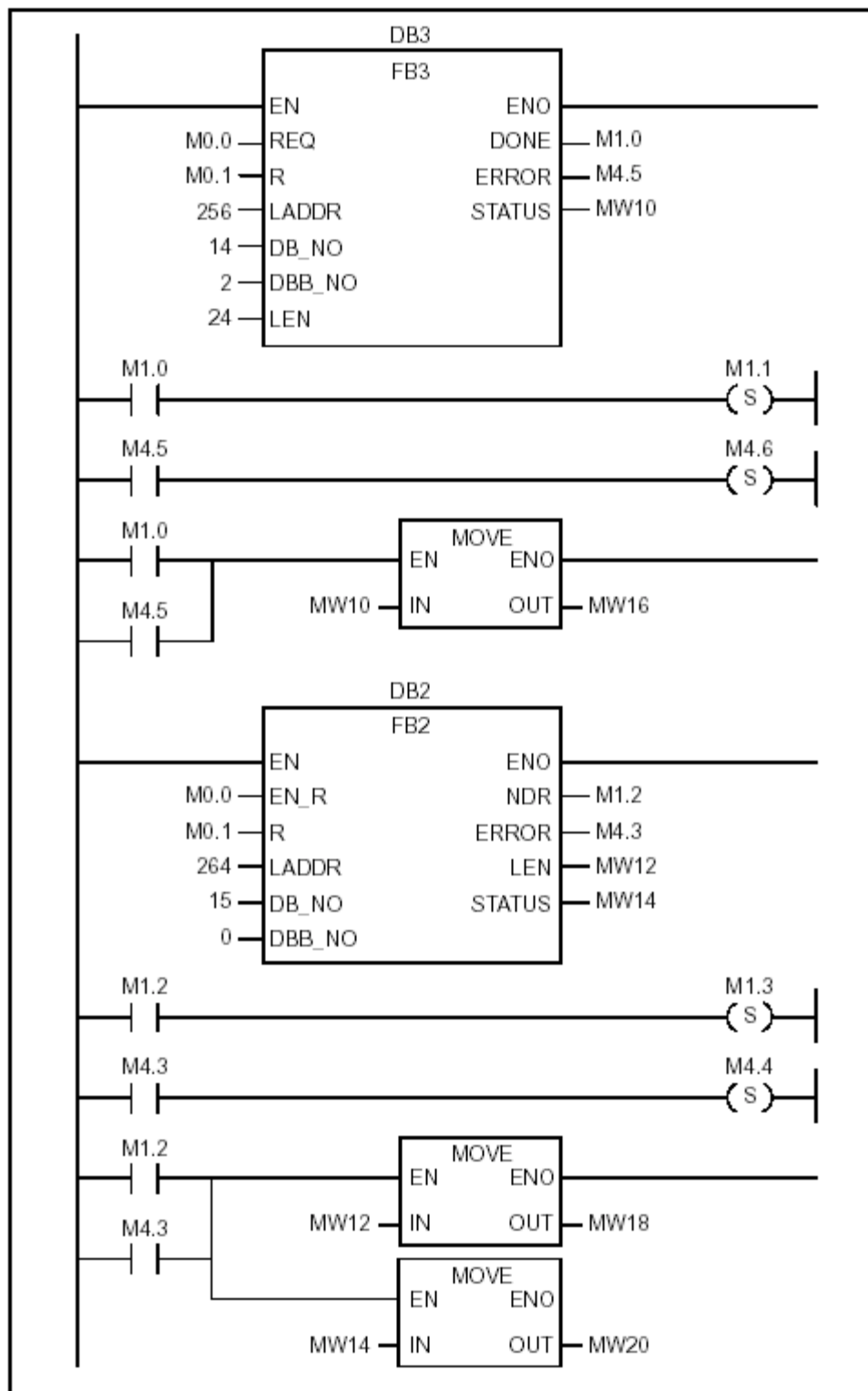


Рисунок 2–3 Пример программы для рассматриваемого случая применения

Создайте DB14 в качестве глобального блока данных в соответствии с таблицей 2–3.

Таблица 2–3 DB14 для рассматриваемого примера			
Адрес	Название	Тип	Начальное значение
0.0		STRUCT	
+0.0	Message	STRING[24]	‘Сообщение длиной в 24 символа’
=26.0		END_STRUCT	

Создайте DB15 в качестве глобального блока данных в соответствии с таблицей 2–4.

Таблица 2–4 DB15 для рассматриваемого примера			
Адрес	Название	Тип	Начальное значение
0.0		STRUCT	
+0.0	Message	ARRAY[1..200]	
*1.0		CHAR	
=200.0		END_STRUCT	

Сохраните блоки и загрузите их в CPU.

Тестирование программы-примера

Выполните следующие действия, чтобы протестировать программу рассматриваемого примера:

1. Выберите в своем проекте контейнер "Block" ("Блок").
2. Выберите команду меню **Insert → S7 Block > Variable Table (Вставить → S7 блок > Таблица переменных)**, чтобы вставить таблицу переменных VAT 1, и подтвердите с помощью ОК.
3. Откройте таблицу переменных VAT 1 и выберите команду меню **Insert → Range of Variables... (Вставить → Набор переменных)**. Введите в различных полях следующие значения:
Начиная с адреса DB15.DBW0
Количество 14
Формат отображения CHAR
Подтвердите с помощью ОК.
4. Введите следующие значения перед DB15.DBW0:

M0.0	Send/receive job	Задание на передачу/прием
M0.1	Reset	Сброс
M1.1	Send terminated	Передача завершена
M4.6	Send error	Ошибка передачи
MW16	Send status	Состояние передачи
M1.3	Receive new data	Прием новых данных
M4.4	Receive error	Ошибка приема
MW18	Receive length	Длина приема
MW20	Receive status	Состояние приема

5. Установите связь с ПЛК (режим Online), выбрав команду меню **PLC → Connect to → Configured CPU (ПЛК → Подключиться к → Сконфигурированный CPU)**.
6. Включите режим мониторинга, выбрав команду меню **Variable → Monitor (Переменная → Мониторинг)**.
7. Переключите CPU в режим RUN.
8. Сбросьте следующие биты в колонке Modify Value (Модифицировать значение) в состояние 0:
 - Передача завершена (M1.1)
 - Ошибка передачи (M4.6)
 - Прием новых данных (M1.3)
 - Ошибка приема (M4.4)
9. Измените следующие биты в колонке Change Value (Изменить значение) в следующем порядке:
 - 1) Бит для сброса M0.1 = **установите в 1**
 - 2) Бит для сброса M0.1 = **сбросьте в 0**
 - 3) Бит для задания на передачу/прием M0.0 = **сбросьте в 0** (если он еще не в 0)
 - 4) Бит для задания на передачу/прием M0.0 = **установите в 1**

Теперь можно:

- Увидеть, что бит завершенной передачи (M1.1) и бит приема новых данных (M1.3) установлены.
- Увидеть, что два бита ошибок были сброшены и два слова состояния обнулены.
- Увидеть, что DB15 содержит ожидавшиеся символы в DBW0..DBW22, и имеет ожидавшуюся длину приема 24 (W#16#18).

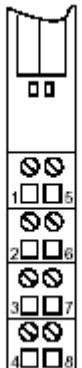
2.3 Схемы с указанием назначения контактов

Подключение проводов

Кабели (клеммы 1...8) должны быть экранированы. Экраны должны быть зафиксированы с обеих сторон. Для этого следует использовать опорные элементы (см. раздел, касающийся дополнительных принадлежностей для ET 200S в руководстве "Станция распределенной периферии ET 200S").

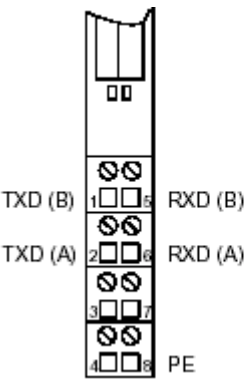
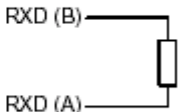
Назначение клемм для связи по RS-232C

В таблице 2-5 указано назначение клемм модуля последовательного интерфейса ET 200S для случая, когда выбран протокол связи RS-232C.

Таблица 2-5 Назначение контактов модуля последовательного интерфейса ET 200S для связи по RS-232C		
Вид	Назначение контактов	Замечания
		<p>Режим: полудуплексный и дуплексный</p> <p>Клеммы</p> <p>1 TXD Передаваемые данные</p> <p>5 RXD Принимаемые данные</p> <p>2 RTS Запрос на передачу</p> <p>6 CTS Готовность к передаче</p> <p>3 DTR Готовность терминала</p> <p>7 DSR Готовность данных</p> <p>4 DCD Детектор несущей</p> <p>8 PE Земля</p>

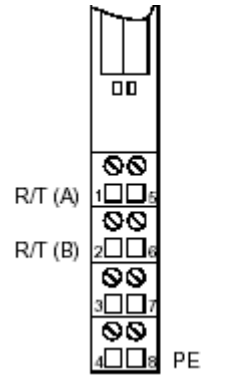
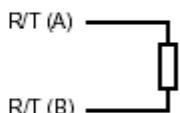
Назначение клемм для связи по RS-422

В таблице 2–6 указано назначение клемм модуля последовательного интерфейса ET 200S для случая, когда выбран протокол связи RS-422.

Таблица 2–6 Назначение контактов модуля последовательного интерфейса ET 200S для связи по RS-422		
Вид	Назначение контактов	Замечания
 <p>TXD (B) 1 □ □ 5</p> <p>TXD (A) 2 □ □ 6</p> <p>3 □ □ 7</p> <p>4 □ □ 8 PE</p> <p>Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом.</p> 		Режим: полудуплексный Клеммы 1 TXD (B) 5 RXD (B) 2 TXD (A) 6 RXD (A) 8 PE Земля

Назначение клемм для связи по RS-485

В таблице 2–7 указано назначение клемм модуля последовательного интерфейса ET 200S для случая, когда выбран протокол связи RS-485.

Таблица 2–7 Назначение контактов модуля последовательного интерфейса ET 200S для связи по RS-485		
Вид	Назначение контактов	Замечания
 <p>R/T (A) 1 □ □ 5</p> <p>R/T (B) 2 □ □ 6</p> <p>3 □ □ 7</p> <p>4 □ □ 8 PE</p> <p>Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом.</p> 		Режим: полудуплексный Клеммы 1 R/T (A) 2 R/T (B) 8 PE Земля

Подключение кабеля с 9-контактным разъемом к RS-232C

На рисунке 2-4 показано подключение кабеля для соединения "точка-точка" по RS-232C между модулем последовательного интерфейса ET 200S и коммуникационным партнером с использованием 9-контактного гнезда D типа.

- На стороне ET 200S сигнальные провода подсоединяются в соответствии с нумерацией клемм.
- На стороне коммуникационного партнера следует использовать 9-контактное гнездо sub-D типа.

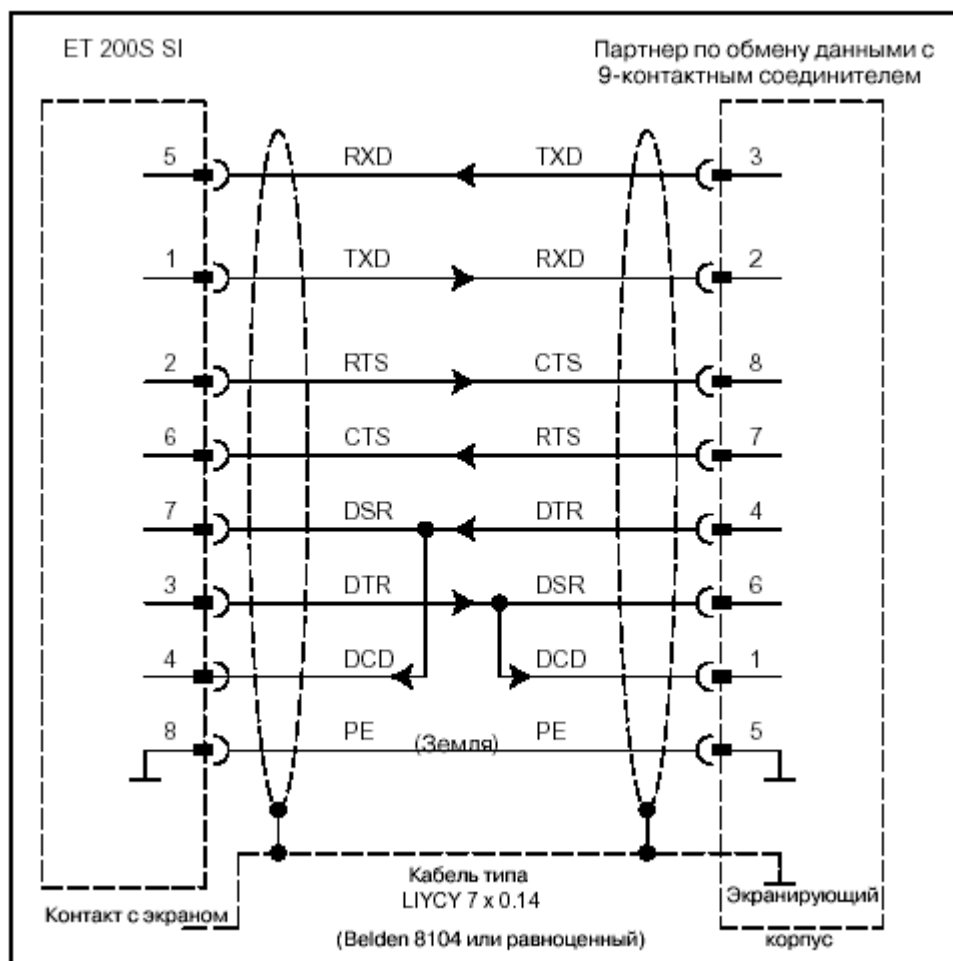


Рисунок 2-4 Разводка контактов соединительного кабеля RS-232C для 9-контактного разъема

Разводка контактов соединительного кабеля RS-232C с 25-контактным разъемом

На рисунке 2-5 показано подключение кабеля для соединения "точка-точка" по RS-232C между модулем последовательного интерфейса ET 200S и коммуникационным партнером с использованием 25-контактного гнезда D типа.

- На стороне ET 200S сигнальные провода подсоединяются в соответствии с нумерацией клемм.
- На стороне коммуникационного партнера следует использовать 25-контактное гнездо sub-D типа.

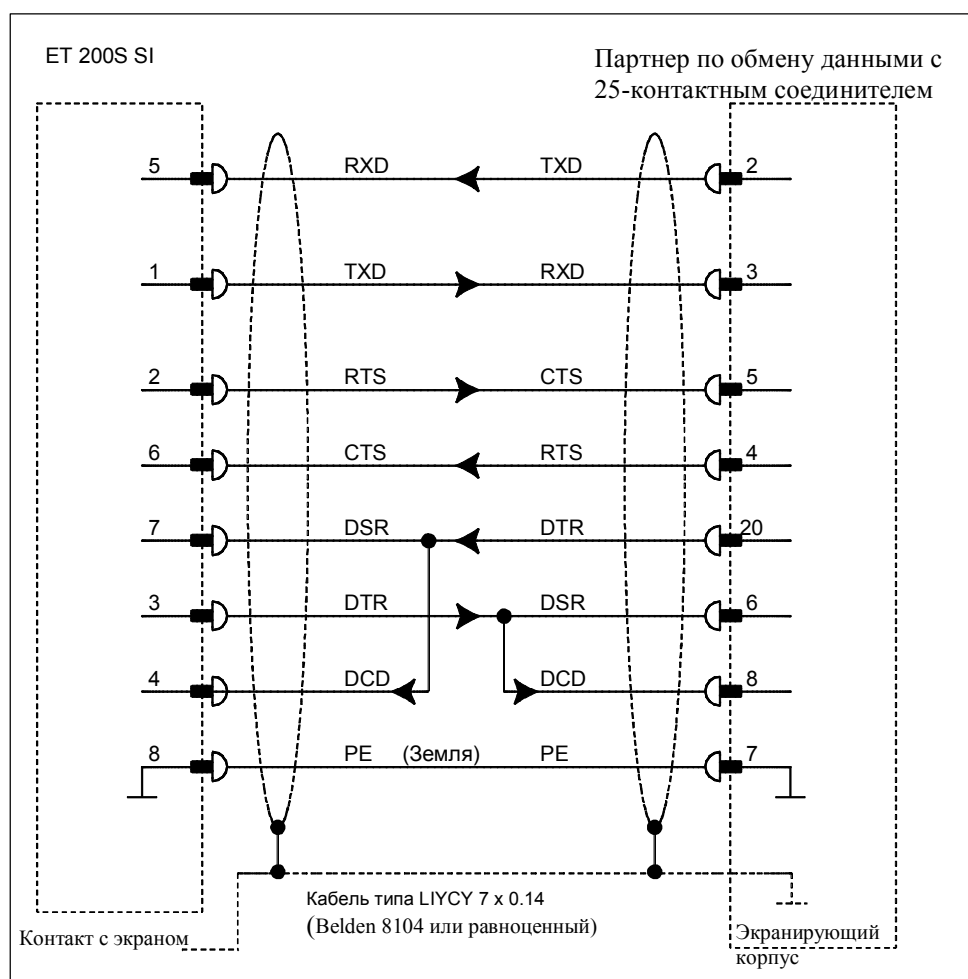


Рисунок 2-4 Разводка контактов соединительного кабеля RS-232C с 25-контактным разъемом

Разводка контактов соединительного кабеля RS-422 с 15-контактным разъемом

На рисунке 2-6 показано подключение кабеля для соединения "точка-точка" по RS-422 между модулем последовательного интерфейса ET 200S и коммуникационным партнером с использованием 15-контактного разъема D типа.

- На стороне ET 200S сигнальные провода подсоединяются в соответствии с нумерацией клемм.
- На стороне коммуникационного партнера следует использовать 15-контактный разъем sub-D типа.

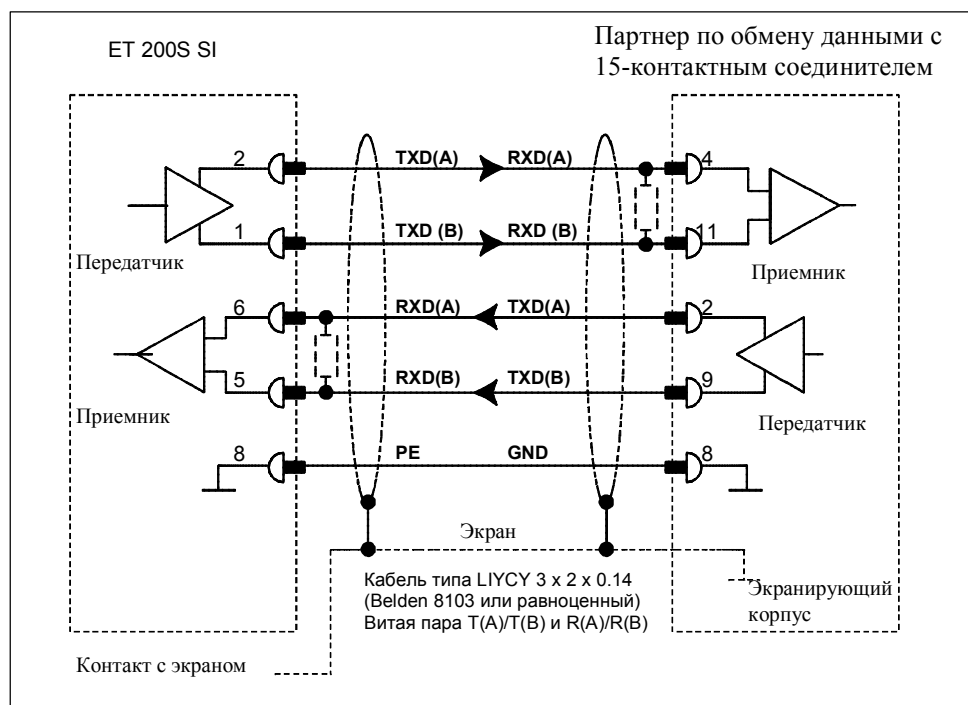


Рисунок 2-6 Разводка контактов соединительного кабеля RS-422 с 15-контактным разъемом

Примечание

Чтобы обмен данными происходил без ошибок, следует включить нагрузочное сопротивление, приблизительно, 330 Ом (см. рисунок 2-6).

Максимальная длина кабеля данного типа при скорости 19 200 бит/с составляет 1200 м.

Разводка контактов соединительного кабеля RS-485 с 15-контактным разъемом

На рисунке 2-7 показано подключение кабеля для соединения "точка-точка" по RS-485 между модулем последовательного интерфейса ET 200S и коммуникационным партнером с использованием 15-контактного разъема D типа.

- На стороне ET 200S сигнальные провода подсоединяются в соответствии с нумерацией клемм.
- На стороне коммуникационного партнера следует использовать 15-контактный разъем sub-D типа.

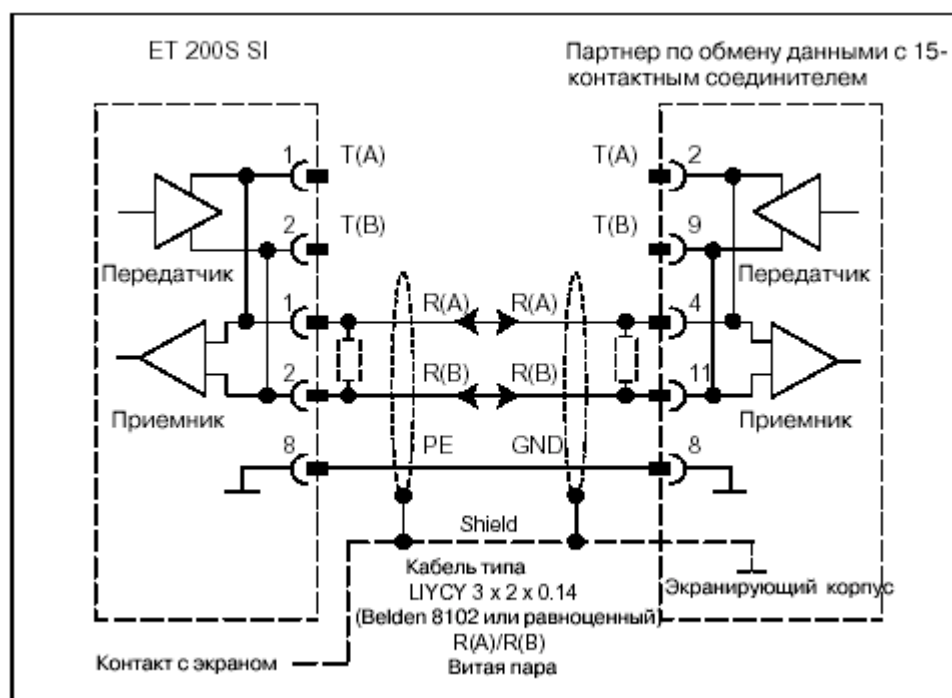


Рисунок 2-7 Разводка контактов соединительного кабеля RS-485C с 15-контактным разъемом

Примечание

Чтобы обмен данными происходил без ошибок, следует включить нагрузочное сопротивление, приблизительно, 330 Ом (см. рисунок 2-7).

Максимальная длина кабеля данного типа при скорости 19 200 бит/с составляет 1200 м.

2.4 Интерфейс RS–232C

Определение

Интерфейс RS–232C – интерфейс последовательной передачи данных в соответствии со стандартом RS–232C. Для передачи сигналов используется напряжение.

Свойства

Интерфейс RS–232C характеризуется следующими свойствами:

Тип:	Интерфейс напряжений
Фронтальный соединитель:	8-контактный стандартный клеммный соединитель модуля ET 200S
Сигналы RS–232C:	TXD, RXD, RTS, CTS, DTR, DSR, DCD, GND
Скорость передачи:	Максимум 19.2 кбит/с (протокол 3964(R)) Максимум 19.2 кбит/с (ASCII драйвер)
Длина кабеля:	Максимум 15 м, LIYCY 7 x 0.14
Стандарты:	DIN 66020, DIN 66259, EIA RS–232C, CCITT V.24/V.28
Степень защиты:	IP 20

Сигналы RS–232C

В таблице описываются сигналы RS–232C.

Сигнал	Название	Значение
TXD	Transmitted Data (Передаваемые данные)	Передача данных; линия передачи удерживается в режиме ожидания в состоянии логической "1".
RXD	Received Data (Принимаемые данные)	Прием данных; линия приема должна удерживаться в состоянии логической "1" коммуникационным партнером.
RTS	Request To Send (Запрос на передачу)	ВКЛ: ET 200S SI готов к передаче. ВЫКЛ: ET 200S SI не передает данные.
CTS	Clear To Send (Готовность к передаче)	Коммуникационный партнер может принимать данные от ET 200S. Интерфейсный модуль считает этот сигнал ответом на RTS (ВКЛ).
DTR	Data Terminal Ready (Готовность терминала)	ВКЛ: На ET 200S SI подано питание и он готов к работе. ВЫКЛ: На ET 200S SI не подано питание и он не готов к работе.
DSR	Data Set Ready (Готовность данных)	ВКЛ: Коммуникационный партнер запитан и готов к работе. ВЫКЛ: Коммуникационный партнер не запитан и не готов к работе.
DCD	Data Carrier Detect (Детектор несущей)	Обнаружен сигнал несущей (если подключен модем).

2.5 Интерфейс RS–422/485

Определение

Интерфейсы RS–422/485 – это интерфейсы последовательной передачи данных в соответствии со стандартами RS–422/485. Для передачи данных используются дифференциальные напряжения.

Свойства

Интерфейс RS–422/485 характеризуется следующими свойствами:

Тип:	Интерфейс дифференциальных напряжений
Фронтальный соединитель:	8-контактный стандартный клеммный соединитель модуля ET 200S
Сигналы RS–422:	TXD (A), RXD (A), TXD (B), RXD (B), GND
Сигналы RS–485:	R/T (A), R/T (B), GND
Скорость передачи:	Максимум 19.2 кбит/с (протокол 3964(R)) Максимум 19.2 кбит/с (ASCII драйвер)
Длина кабеля:	Максимум 1200 м, LIYCY 7 x 0.14
Стандарты:	EIA RS–422/485, CCITT V.11/V.27
Степень защиты:	IP 20

2.6 Основные принципы последовательной передачи данных

Соединение "точка-точка"

Существуют различные варианты построения сетей, позволяющие двум или большему числу партнеров обмениваться между собой данными. Самым простым способом организации обмена данными между двумя партнерами является установление соединения "точка-точка".

В соединении "точка-точка" модуль последовательного интерфейса служит в качестве интерфейса между программируемым контроллером и коммуникационным партнером. Данные передаются последовательно с помощью модуля последовательного интерфейса ET 200S.

Последовательная передача данных

При последовательной передаче отдельные биты каждого байта данных передаются один за другим в определенном порядке.

Модуль последовательного интерфейса ET 200S управляет передачей данных независимо от подключенного к нему коммуникационного партнера посредством своего интерфейса (порта) последовательной передачи данных. Для этих целей у модуля имеется два различных драйвера для двунаправленной передачи данных.

- ASCII драйвер
- Протокол 3964(R)

Двунаправленная передача данных: режимы

Модуль ET 200S поддерживает два режима двунаправленной передачи данных:

- Полудуплексный режим (протокол 3964(R), ASCII драйвер)
В этом режиме данные, которыми обмениваются между собой коммуникационные партнеры, передаются одновременно лишь в одном направлении. Таким образом, в полудуплексном режиме в любой момент времени данные либо передаются, либо принимаются (не могут передаваться и приниматься одновременно). Исключение могут составлять лишь отдельные символы управления, используемые для управления потоком данных (напр., XON/XOFF), которые могут передаваться в момент приема данных, либо приниматься в момент передачи данных.
- Дуплексный режим (ASCII драйвер)
В этом режиме данные, которыми обмениваются между собой коммуникационные партнеры, могут передаваться одновременно в двух направлениях. Дуплексный режим означает, что данные могут передаваться и приниматься одновременно. Каждый коммуникационный партнер должен "уметь" передавать и принимать данные одновременно.

В таблице 2–8 перечислены режимы передачи данных для различных типов интерфейсов с драйвером ASCII.

Таблица 2–8 Режимы передачи данных для модуля последовательного интерфейса ET 200S			
Передача данных	RS–232C	RS–422	RS–485
Полудуплексный режим	Да	Да	Да
Дуплексный режим	Да	Да	Не возможен

Согласование условий

Перед осуществлением последовательной передачи данных между двумя партнерами должен быть согласован ряд условий. К этим условиям относятся:

- Скорость передачи
- Время ожидания символа и время ожидания подтверждения
- Паритет (четность)
- Количество битов данных
- Количество стоп-битов
- Количество попыток установления связи и передачи

Роль, которую играют данные условия в различных протоколах передачи, а также способы их параметрирования описываются в разделах 2.7 и 2.8.

Кадр символа

Данные, которыми обмениваются между собой модуль последовательного интерфейса ET 200S и коммуникационный партнер, передаются по последовательному интерфейсу и имеют формат кадров длиной 10 или 11 битов. Существует три формата кадра. Требуемый формат можно сконфигурировать в STEP 7.

Кадр символа длиной 10 бит

На рисунке 2-8 показано три формата для 10-битного кадра символа.

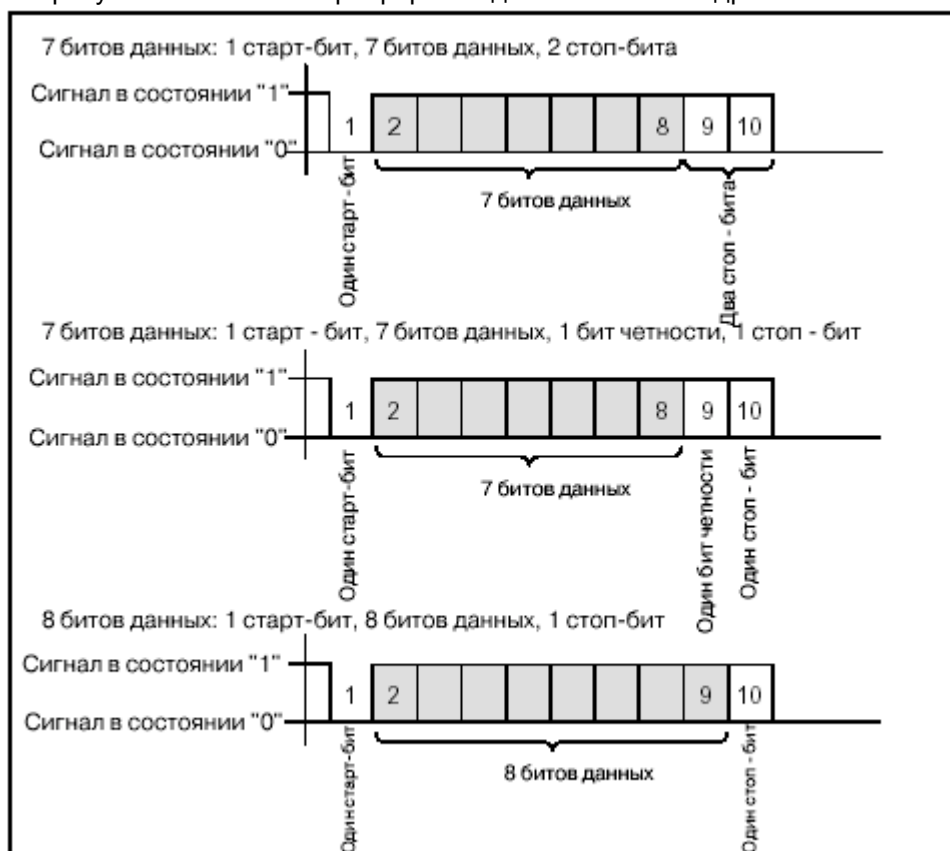


Рисунок 2–8 Кадр символа длиной 10 битов

Кадр символа длиной 11 битов

На рисунке 2-9 показано три формата данных для 11-битного кадра символа.

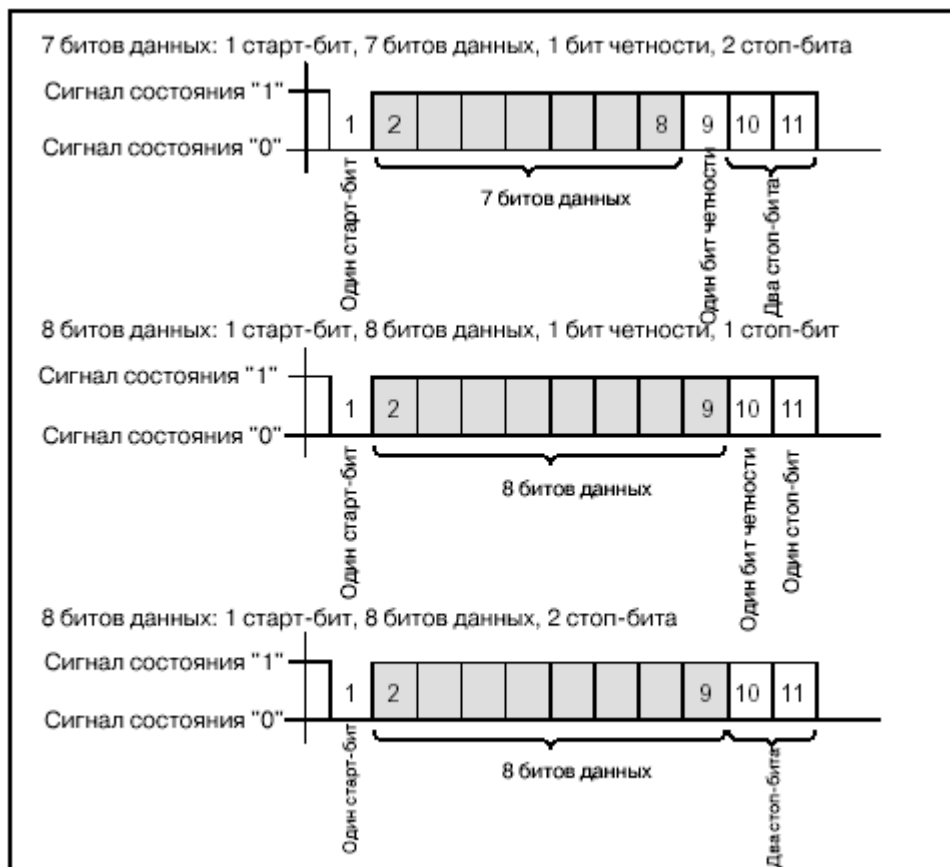


Рисунок 2-9 Кадр символа длиной 11 битов

Время ожидания символа

На рисунке 2-10 показан максимальный допустимый временной интервал между двумя символами, принимаемыми в пределах одного кадра сообщения. Этот временной интервал называют временем ожидания символа.

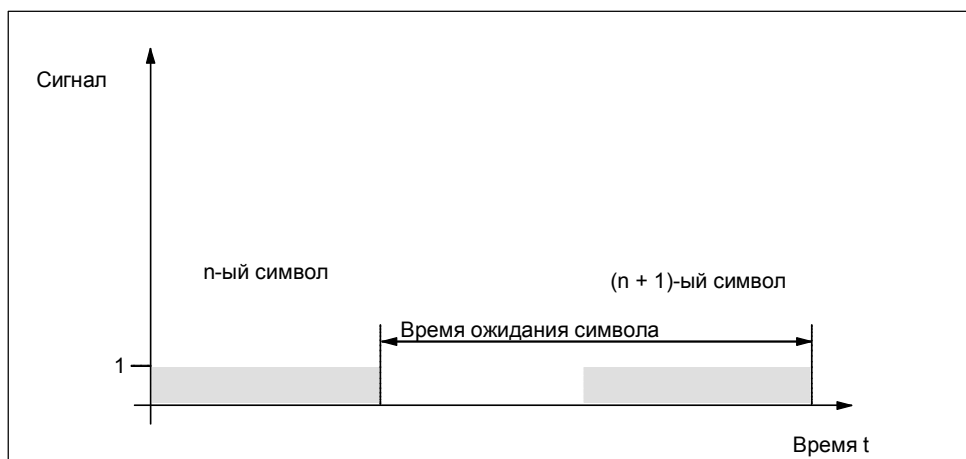


Рисунок 2–10 Время ожидания символа

7-уровневая эталонная модель ISO для передачи данных

В процессе передачи данных все партнеры, участвующие в обмене данными, должны следовать установленным правилам формирования и управления потоком данных (трафиком). Для этих целей организацией ISO была введена 7-уровневая модель, которая принимается за основу при разработке международных стандартов на протоколы передачи, используемые для организации связи между компьютерами.

Протокол

В процессе передачи данных все партнеры, участвующие в обмене данными, должны следовать установленным правилам формирования и управления потоком данных (трафиком). Такие правила называются протоколами.

Протоколом определяются следующие условия:

- **Режим**
Дуплексный или полудуплексный режимы
- **Инициирование**
Какой именно коммуникационный партнер может инициировать передачу данных, и при каких условиях.
- **Символы управления**
Какие символы управления будут использоваться для передачи данных.
- **Кадр символа**
Какой кадр символа будет использоваться для передачи данных.
- **Резервное копирование данных**
Какая процедура резервного копирования данных будет использоваться.
- **Время ожидания символа**
Временной интервал, в пределах которого должен быть принят передаваемый символ.
- **Скорость передачи**
Измеряется в битах в секунду (bps = бит/с)

7-уровневая эталонная модель ISO

Эталонная модель определяет алгоритм работы коммуникационных партнеров по отношению к внешним устройствам. Протокол каждого уровня, за исключением самого нижнего уровня, строится над протоколом предыдущего уровня.

В модели ISO различают следующие уровни:

1. **Физический уровень**
 - Условия, связанные с физической реализацией связи, например, среда передачи и скорость передачи
2. **Канальный уровень**
 - Процедура защиты передаваемых данных
 - Методы доступа
3. **Сетевой уровень**
 - Подключение к сети
 - Система адресации для обмена данными между двумя партнерами
4. **Транспортный уровень**
 - Процедура обнаружения ошибок
 - Устранение ошибок
 - Подтверждение установления связи
5. **Сеансовый уровень**
 - Начало и прекращение передачи данных
 - Сеанс
6. **Уровень представления**
 - Преобразование данных, представленных в стандартной форме системы связи, к форме, используемой конкретным устройством (правила интерпретации данных)
7. **Прикладной уровень**
 - Определение цели обмена данными и функций, необходимых для реализации этой цели

Реализация и обработка протоколов

Передающий коммуникационный партнер последовательно, один за другим, реализует протоколы, начиная с высшего (№7 – прикладной уровень) и заканчивая нижним (№1 – физический уровень) уровнем, а партнер, принимающий данные, обрабатывает эти протоколы в обратном порядке, т.е., начиная с уровня 1.

Не во всех протоколах охватываются все семь уровней эталонной модели. Если оба партнера, передающий и принимающий, используют один и тот же протокол, уровень 6 может не применяться.

Целостность передаваемых данных

При передаче данных и при выборе протокола (процедуры) передачи огромное значение играет требуемая целостность передаваемых данных. В общем случае, чем больше уровней эталонной модели охвачено, тем выше достоверность передаваемых данных.

Поддерживаемые протоколы

На рисунке 2-11 показано, какое место в эталонной модели ISO занимают протоколы ASCII и 3964(R), поддерживаемые модулем последовательного интерфейса ET 200S.

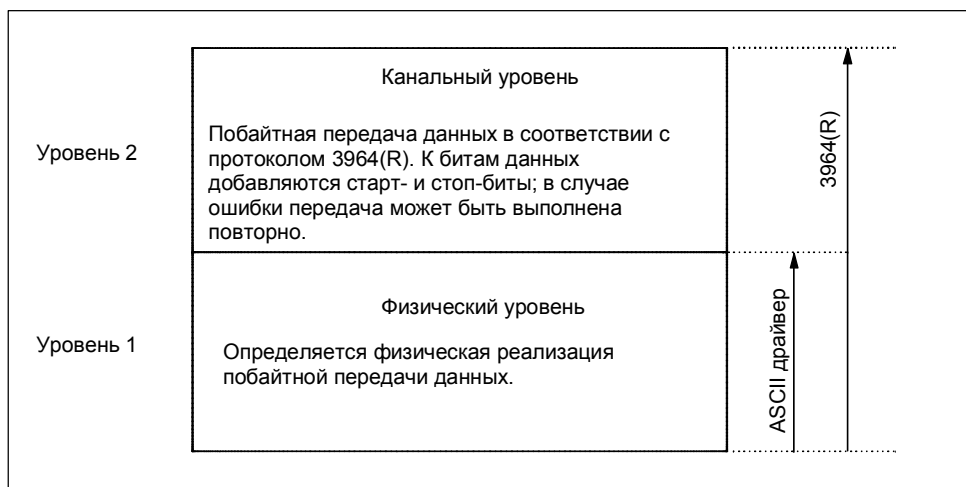


Рисунок 2–11 Место протоколов, поддерживаемых модулем ET 200 S SI, в эталонной модели ISO

Целостность передаваемых данных при использовании ASCII драйвера:

При использовании ASCII-драйвера необходимо руководствоваться следующими указаниями, чтобы повысить сохранность передаваемых данных:

- Помимо использования бита четности (который также можно отключить, сконфигурировав соответствующим образом кадр символа), не существует других способов, позволяющих повысить сохранность данных при их передаче с использованием ASCII-драйвера. Это означает, что передача данных с использованием ASCII драйвера является эффективной с точки зрения скорости передачи, но сохранность данных полностью гарантирована быть не может.
- Использование бита четности позволяет обнаружить инверсию бита в передаваемом символе. Если произошла инверсия двух или большего количества битов в символе, такая ошибка обнаружена быть не может.
- Для повышения целостности передаваемых данных можно указывать контрольную сумму и длину передаваемого кадра сообщения. Такие меры должны предусматриваться самим пользователем.
- Дополнительное повышение достоверности данных может быть достигнуто посредством применения кадров подтверждения, посылаемых в ответ на передаваемые или принимаемые кадры сообщений. Это также относится к протоколам верхнего уровня, применяемым для обмена данными (см. 7-уровневую эталонную модель ISO).

Целостность передаваемых данных при использовании протокола 3964(R)

Протокол 3964(R) обеспечивает более высокую надежность (целостность) передачи данных:

- Протокол 3964(R) характеризуется расстоянием Хэмминга 3. Расстояние Хэмминга является показателем надежности при передаче данных.
- Протокол 3964(R) обеспечивает высокую надежность передачи в линии данных. Высокая надежность (целостность) достигается за счет фиксированных моментов начала и завершения кадра сообщения, а также за счет использования символа контроля блока (BCC).

Для передачи данных могут использоваться две различные процедуры: с использованием или без использования символа контроля блока:

- Передача данных без символов контроля блока: **3964**
- Передача данных с символом контроля блока: **3964(R)**

В данном руководстве будет использоваться обозначение **3964(R)**, но дальнейшее описание и замечания будут относиться к обеим процедурам передачи данных.

Ограничения, связанные с использованием протокола 3964(R)

- Дальнейшая обработка передаваемых/принимаемых данных в программе ПЛК коммуникационного партнера не гарантируется. Для подтверждения можно использования программный механизм квитирования.
- Контроль блока в протоколе 3964(R) (логическая операция "исключающее ИЛИ") не позволяет обнаружить пропадание нулей (имеется в виду символ "0"), поскольку 0 в операции "исключающее ИЛИ" не влияет на результат вычислений.

Потеря целого символа (т.е., символа "0") маловероятна, но может произойти, например, при очень плохих условиях, сопутствующих передаче данных.

Мерой защиты от таких ошибок при передаче данных является передача длины кадра сообщения вместе с передаваемыми данными, и проверка длины коммуникационным партнером.

2.7 Передача данных с использованием протокола 3964(R)

Протокол 3964(R) служит для управления передачей данных в случае соединения “точка-точка” между модулем ET 200S и коммуникационным партнером. Помимо физического уровня (уровень 1) протокол 3964(R) также охватывает канальный уровень (уровень 2).

Символы управления

В процессе передачи данных в рамках протокола 3964(R) данные пользователя дополняются символами управления (функция канального уровня). С помощью символов управления партнер по обмену данными может проверить целостность прибывших данных и отсутствие в них ошибок. В протоколе 3964(R) анализируются следующие символы управления:

- **STX** Начало текста
Начало передаваемой строки символов
- **DLE** Смена направления в канале
Изменение направления передачи данных
- **ETX** Конец текста
Конец передаваемой строки символов
- **BCC** Символ контроля блока (только для 3964R)
- **NAK** Отрицательное подтверждение

Примечание

Если строка "DLE" передается в качестве информации, она передается дважды, чтобы ее можно было отличить от символа управления DLE в процессе установления и сброса соединения в линии передачи данных (дублирование DLE). Принимающая сторона устраняет дублирование DLE.

Приоритет

В рамках протокола 3964(R) одному коммуникационному партнеру должен быть присвоен более высокий приоритет, а другому – более низкий. Если оба партнера выставляют запрос на передачу одновременно, партнер, которому назначен более низкий приоритет, снимает свой запрос.

Контрольная сумма блока

При передаче данных по протоколу 3964R надежность передачи (целостность данных) повышается за счет передачи дополнительного символа контроля блока (BCC) (см. рисунок 2–12).

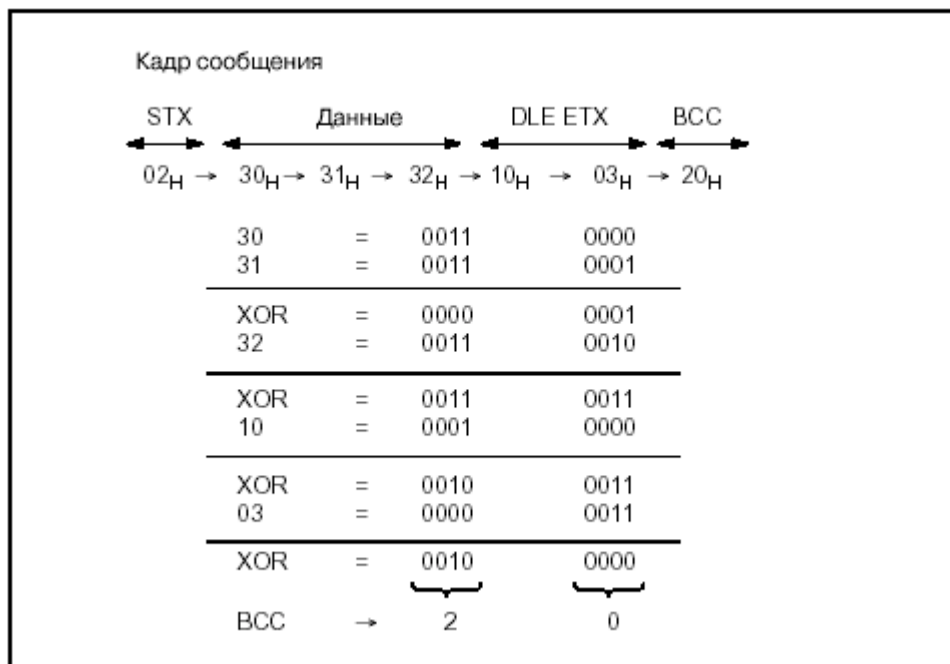


Рисунок 2–12 Контрольная сумма блока

Контрольная сумма блока вычисляется путем взятия "исключающего ИЛИ" по всем байтам данных передаваемого или принимаемого блока. Вычисление начинается с первого байта данных пользователя (первый байт кадра сообщения) после установления соединения, и заканчивается после DLE ETX по сбросу соединения.

Примечание

В случае дублирования DLE символ DLE используется при вычислении BCC дважды.

Передача данных с использованием 3964(R)

На рисунке 2-13 приводится последовательность передачи данных при использовании протокола 3964(R).

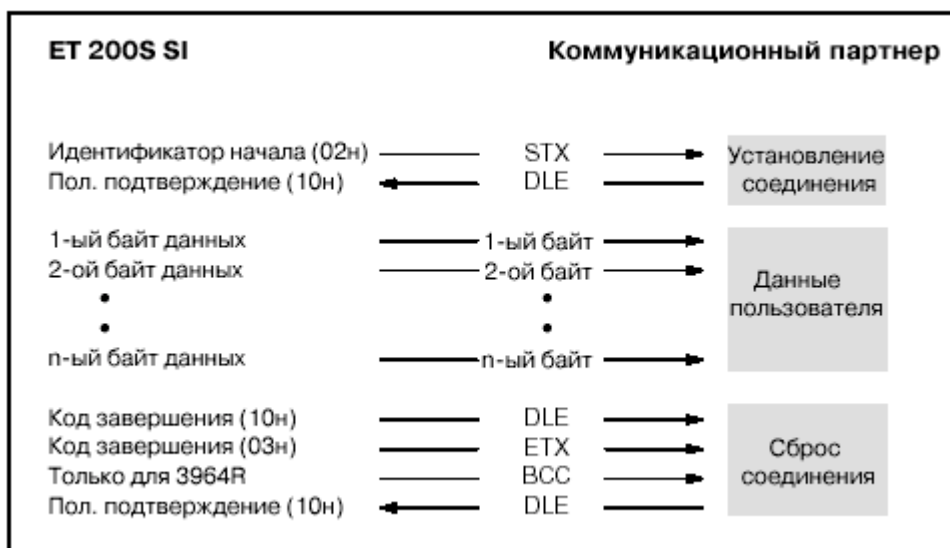


Рисунок 2–13 Последовательность передачи данных по протоколу 3964(R)

Установление соединения для передачи данных

Для установления соединения в рамках протокола 3964(R) передается символ управления STX. Если партнер по обмену данными возвращает в ответ символ DLE в пределах времени ожидания подтверждения (ADT), протокол переходит в режим передачи.

Если партнер по обмену данными возвращает символ NAK или любой другой символ управления (кроме DLE), либо в течение времени ожидания подтверждения не поступил ответ, в рамках протокола выполняется повтор установления соединения. Если количество неудачных попыток установления соединения превышает сконфигурированное значение, протокол прекращает попытки установления соединения и передает символ NAK коммуникационному партнеру. Системная программа сообщает об ошибке функциональному блоку S_SEND (выходной параметр STATUS).

Передача данных

Если соединение установлено успешно, данные пользователя, содержащиеся в выходном буфере ET 200S, передаются коммуникационному партнеру в соответствии с выбранными параметрами передачи. Партнер измеряет временные интервалы между поступающими символами. Временной интервал между двумя соседними символами не должен превышать по длительности время ожидания символа.

Освобождение соединения, установленного для передачи данных

Если в процессе активного сеанса передачи партнер по обмену данными передает символ управления NAK, драйвер протокола 3964(R) прерывает передачу блока и предпринимает повторную попытку передачи в соответствии с описанием выше. Если передается любой другой символ, драйвер протокола ожидает в течение времени ожидания символа, после чего передает символ NAK, чтобы перевести своего партнера по обмену данными в режим ожидания. После этого вся процедура передачи данных повторяется вновь, начиная с установления соединения по символу STX.

После того как содержимое буфера передано, к данным в рамках протокола добавляются символы DLE, ETX, а также контрольная сумма блока BCC (**только для 3964(R)**), являющиеся кодом завершения, и ожидается символ подтверждения. Если партнер по обмену данными передает символ DLE в пределах времени ожидания подтверждения, это означает, что блок данных был принят без ошибок. Если коммуникационный партнер отвечает символом NAK, любым другим символам (кроме DLE) или поврежденным символом, либо в течение времени ожидания подтверждения не поступает ответ, вся процедура передачи данных повторяется вновь, начиная с установления соединения по символу STX.

После установленного количества попыток передачи блока данных драйвер протокола прекращает попытки передачи и передает коммуникационному партнеру символ NAK. Системная программа сообщает об ошибке функциональному блоку S_SEND (выходной параметр STATUS).

Прием данных с помощью 3964(R)

На рисунке 2-14 показана последовательность передачи данных при приеме данных по протоколу 3964(R).

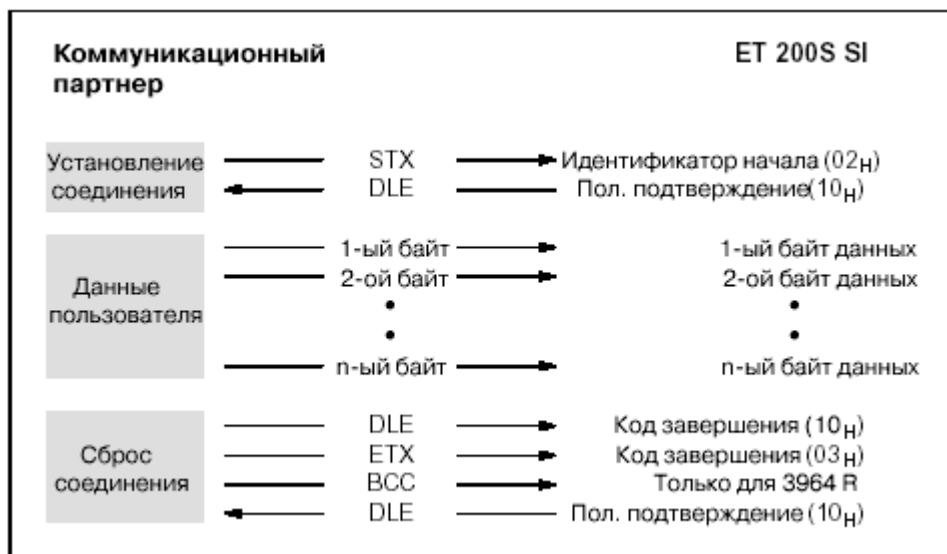


Рисунок 2–14 Поток данных при выполнении приема данных по протоколу 3964(R)

Установление соединения для приема данных

В режиме ожидания, когда не выполняется задание на передачу данных, драйвер протокола ожидает, когда коммуникационный партнер установит соединение.

Если в режиме ожидания драйвером протокола принимается любой символ управления, за исключением STX или NAK, протокол выдерживает паузу, равную по длительности времени ожидания символа, после чего передает символ NAK.

Прием данных

Если драйвером протокола принимается символ STX, и ему доступен пустой буфер приема, драйвер протокола отвечает символом DLE. После этого поступающие символы записываются в буфер приема. Если поступают два последовательных символа DLE, в буфер приема записывается только один из них.

Приняв очередной символ, драйвер ожидает следующий символ в течение времени ожидания символа. Если время ожидания истекает, а другой символ не поступает, коммуникационному партнеру высылается символ NAK.

Системная программа сообщает об ошибке функциональному блоку S_RCV (выходной параметр STATUS).

Если при установлении соединения по символу STX отсутствует пустой буфер приема, начинается отсчет интервала ожидания длительностью 400 мс. Если по истечении интервала ожидания по-прежнему отсутствует пустой буфер приема, системная программа сообщает об ошибке (сообщение об ошибке в выходном состоянии STATUS функционального блока). Драйвер протокола передает символ NAK и возвращается в режим ожидания. В других случаях драйвер протокола передает символ DLE и принимает данные в соответствии с описанием, приведенным выше.

Сброс соединения, установленного для приема данных

Если в процессе приема данных происходит ошибка (потеря символа, ошибка кадра, ошибка четности и т.п.), процедура приема данных продолжается до тех пор, пока не будет сброшено соединение, после чего партнеру по обмену данными передается символ NAK. После этого ожидается повторная передача. Если после определенного количества повторных попыток передачи, выбранного при назначении параметров, принять блок без ошибок не удалось, или коммуникационный партнер не повторил передачу блока в течение времени ожидания блока, равного 4 с, процедура приема данных прекращается. Системная программа сообщает об ошибке функциональному блоку S_RCV (выходной параметр STATUS).

Если драйвер протокола **3964(R)** обнаружил строку DLE ETX, прием прекращается, а коммуникационному партнеру передается символ DLE, означающий, что блок был принят без ошибок. Если блок поврежден, передается символ NAK. После этого ожидается повторная передача.

Если драйвер протокола **3964(R)** обнаружил строку DLE ETX BCC, прием прекращается. Принятый BCC сравнивается со значением, рассчитанным самим драйвером. Если рассчитанное значение совпало с BCC, и не произошло никаких других ошибок при приеме, драйвер протокола 3964(R) передает символ DLE, после чего происходит возврат в режим ожидания. Если сравнение с BCC дало ошибку, или произошла другая ошибка при приеме, партнеру по обмену данными возвращается символ NAK. После этого ожидается повторная передача.

Примечание

Как только драйвер протокола 3964(R) готов к работе, он передает коммуникационному партнеру одиночный символ NAK, устанавливая его в режим ожидания.

Обработка поврежденных данных

На рисунке 2-15 показана процедура обработки поврежденных данных в рамках протокола 3964(R).

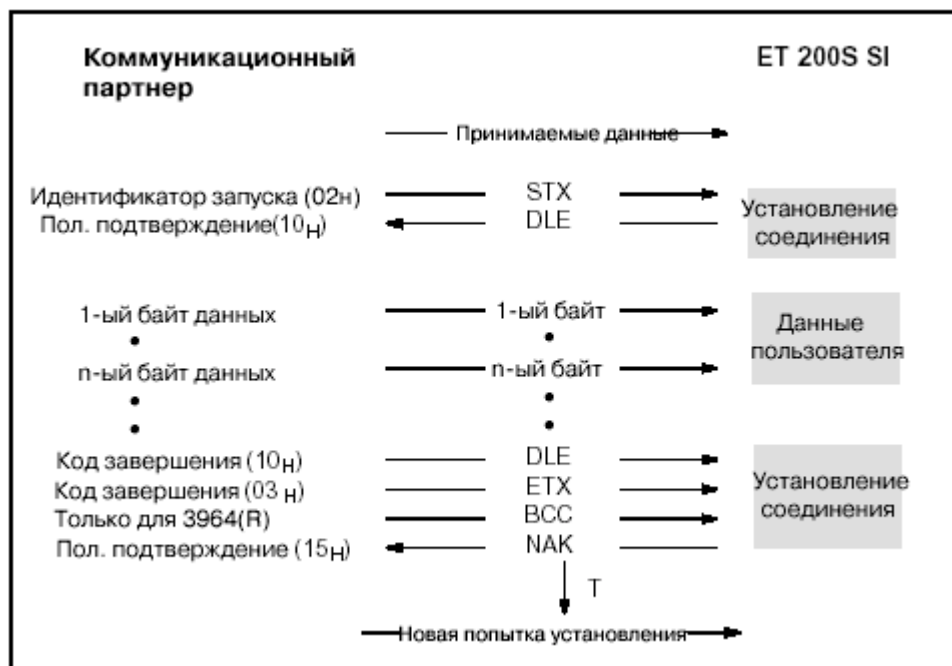


Рисунок 2–15 Последовательность передачи при приеме поврежденных данных

Когда приняты символы DLE, ETX и BCC, модуль ET 200S сравнивает BCC, переданный коммуникационным партнером, со значением, рассчитанным самим модулем. В случае совпадения рассчитанного значения с полученным BCC, и при отсутствии других ошибок при приеме, модуль ET 200S посылает в ответ символ DLE.

В противном случае модуль отвечает символом NAK и ожидает новой попытки передачи в течение времени ожидания блока (T), равного 4 сек. Если после сконфигурированного количества попыток передачи принять блок без ошибок не удалось, или если в течение времени ожидания блока других попыток передачи произведено не было, ET 200S прекращает процедуру приема.

Конфликт инициализации

На рисунке 2-16 показана последовательность передачи данных при конфликте инициализации.

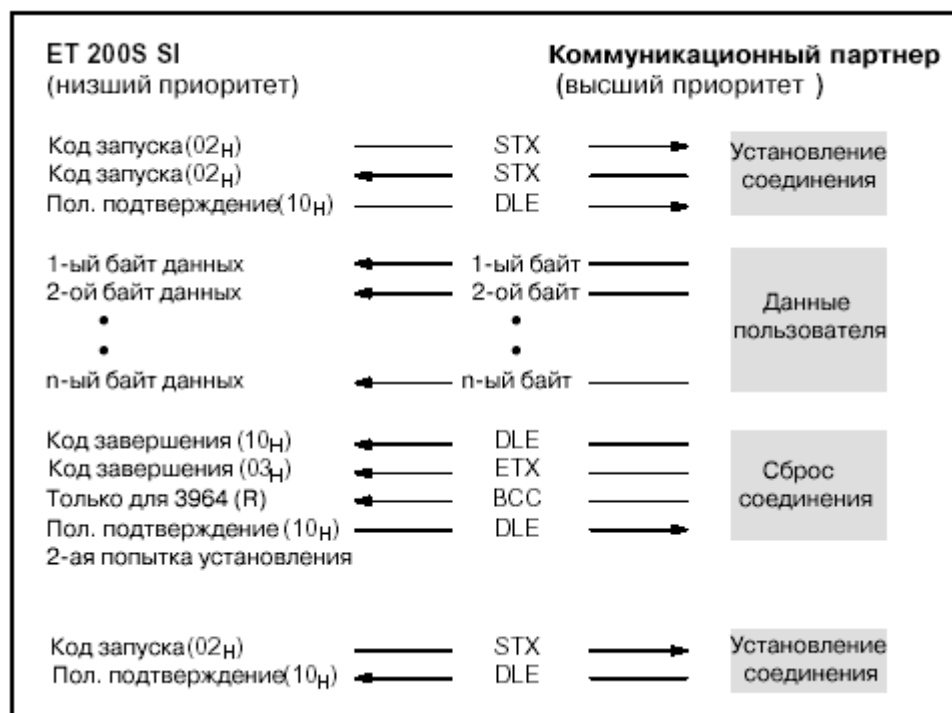


Рисунок 2–16 Последовательность передачи при конфликте инициализации

Если в ответ на запрос на передачу (символ STX), выставленный коммуникационным партнером, в пределах времени ожидания подтверждения (ADT) устройство высылает символ STX вместо символа подтверждения DLE или NAK, возникает конфликт инициализации. Оба устройства выставили запрос на передачу одновременно. Устройство с низшим приоритетом отзывает свое задание на передачу и отвечает символом DLE. Устройство с высшим приоритетом передает свои данные в соответствии с процедурой, описанной выше. Как только соединение будет сброшено, устройство с низшим приоритетом может выставить свое задание на передачу.

Чтобы конфликт инициализации мог быть разрешен, коммуникационным партнерам должны быть назначены различные приоритеты.

Ошибка процедуры

Драйвером протокола 3964(R) обнаруживаются как ошибки, вызванные партнером по обмену данными, так и ошибки, связанные с неисправностями в линии.

В обоих случаях драйвером протокола предпринимаются повторные попытки передачи/приема блока данных без ошибок. Если это оказалось невозможным за определенное количество повторных попыток (или если возникло новое ошибочное состояние), передача или прием драйвером протокола прекращается. Он сообщает номер первой обнаруженной ошибки и возвращает модуль в режим ожидания. Сообщения об ошибках отображаются в выходном состоянии STATUS функционального блока.

Если системная программа постоянно сообщает номер ошибки в выходном состоянии STATUS функционального блока (FB) при повторных попытках передачи и приема, это означает, что факторы, оказывающие воздействие на передаваемые данные, носят продолжительный характер. Способом борьбы является увеличение количества повторных попыток передачи и приема. В таких случаях рекомендуется проверить линию передачи на наличие возможных источников помех, поскольку увеличение количества повторов приводит к снижению скорости передачи и целостности передаваемых данных. Повреждение данных также может быть вызвано неисправностью на стороне партнера по обмену данными.

Обрывы в линии приема соответствуют состоянию BREAK (сигнализация BREAK посредством диагностического прерывания модуля ET 200S) (см. Раздел 2.13). Повторная передача не предпринимается. Состояние BREAK сбрасывается автоматически, как только в линии восстанавливается соединение.

Для каждой обнаруженной ошибки передачи (потеря символов, ошибка кадра или четности) сообщается стандартное число, независимо от того, была ошибка обнаружена во время передачи или приема блока данных. Об ошибке сообщается только в том случае, если повторные попытки оказались unsuccessful.

2.8 Передача данных с помощью ASCII драйвера

Введение

ASCII драйвер управляет передачей данных, осуществляемой через соединение "точка-точка", установленное между модулем ET 200S SI и партнером по обмену данными. Данный драйвер охватывает физический уровень (уровень 1 эталонной модели ISO).

Структура кадров сообщений оставлена открытой, чтобы пользователь S7 мог передать на модуль ET 200S SI полностью весь кадр передаваемого сообщения. Для принимающей стороны должен быть выбран критерий завершения сообщения. Структура передаваемых кадров сообщений может отличаться от структуры принимаемых кадров.

ASCII драйвер позволяет передавать и принимать данные любого состава: все выводимые на печать символы ASCII, а также все другие символы в пределах 00...FF_h (если используются кадры символа с 8-ю битами данных) или 00...7F_h (если используются кадры символа с 7-ю битами данных).

Передача данных с использованием ASCII драйвера

Для передачи данных при вызове функционального блока S_SEND в параметре LEN указывается количество байтов данных пользователя, которое должно быть передано. Данные пользователя должны содержать необходимые символы начала и завершения.

Если при приеме данных используется критерий завершения "Expiration of character delay time" ("Превышение времени ожидания символа"), ASCII драйвер выдерживает паузу между двумя передаваемыми кадрами сообщений. Функциональный блок (FB) S_SEND можно вызвать в любой момент, но ASCII драйвер не приступит к выводу данных, пока с момента передачи последнего кадра сообщения не истечет время ожидания символа.

Примечание

Если сконфигурировано управление потоком XON/XOFF, данные пользователя не должны содержать ни один из сконфигурированных символов XON или XOFF. По умолчанию в качестве XON используется DC1 = 11_h, а в качестве XOFF – DC3 = 13_h.

Передача данных

На рисунке 2-17 показана последовательность действий при выполнении процедуры передачи.

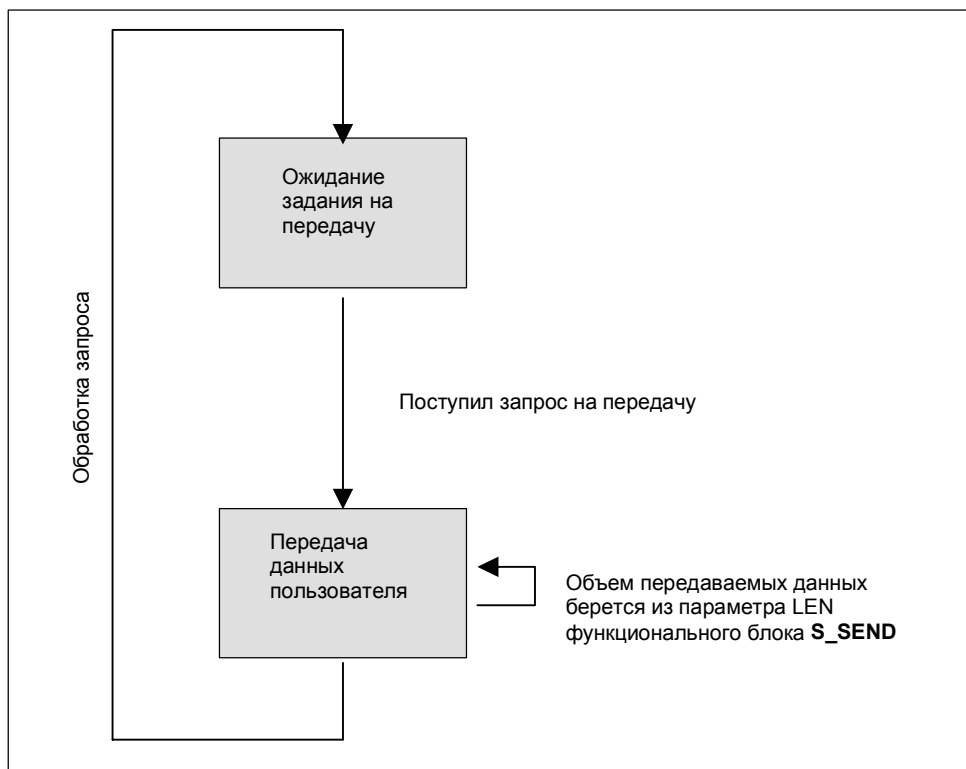


Рисунок 2–17 Последовательность действий при выполнении передачи

Прием данных с помощью ASCII драйвера

При передаче данных с использованием ASCII драйвера можно выбрать один из трех критериев завершения передачи. Критерий завершения определяет момент, когда кадр сообщения принят полностью. Возможны следующие критерии завершения:

- **Превышение времени ожидания символа**

У кадра сообщения нет какой-либо фиксированной длины или определенного символа завершения текста; сообщение считается завершенным, если в линии наступает пауза (превышающая по длительности время ожидания символа). В таблице 2-9 (стр. 2-43) приводятся минимальные значения для отдельных скоростей передачи.

- **По приему символа завершения текста**

Конец кадра сообщения помечается одним или двумя установленными символами завершения текста.

- **По приему определенного количества символов**

Все принимаемые кадры сообщений имеют одну и ту же длину.

Прозрачность кода

Прозрачность (неинтерпретируемость) кода в рамках протокола зависит от выбранного критерия завершения, а также от управления потоком данных:

- Выбраны один или два символа завершения текста
 - Нет прозрачности кода
- В качестве критерия завершения установлено превышение времени ожидания символа или фиксированная длина кадра сообщения
 - Код прозрачен (не интерпретируется)
- Прозрачность кода при работе не достигается, если используется управление потоком данных XON/XOFF.

Прозрачность (неинтерпретируемость) кода означает, что данные пользователя могут состоять из любой комбинации символов, при этом ни одна из них не будет принята за критерий завершения.

Минимальное время ожидания символа в зависимости от скорости передачи

Минимальное значение времени ожидания символа зависит от скорости передачи. В таблице 2-9 приводятся значения для минимального времени ожидания символа для различных скоростей передачи (в миллисекундах).

Таблица 2–9 Минимальное время ожидания символа	
Скорость передачи	Минимальное время ожидания символа
19 200	4 мс
9600	4 мс
4800	9 мс
2400	18 мс
1200	34 мс
600	68 мс
300	134 мс
110	364 мс

Критерий завершения – "Превышение времени ожидания символа"

При приеме данных признаком завершения кадра сообщения является пауза в линии передачи продолжительностью свыше времени ожидания символа.

Принимаемые данные поступают в CPU с помощью функционального блока S_RCV.

В данном случае время ожидания символа должно быть установлено таким образом, чтобы оно было надежно превышено между двумя соседними кадрами сообщений. В то же время, оно должно быть достаточно большим, чтобы не произошло ложного определения завершения кадра сообщения в случае, когда наблюдаются паузы при передаче данных коммуникационным партнером в пределах кадра сообщения. В таблице 2-9 (стр. 2-43) приводятся минимальные значения для различных скоростей передачи.

На рисунке 2-18 показана последовательность действий при приеме данных с использованием критерия завершения "Превышение времени ожидания символа".

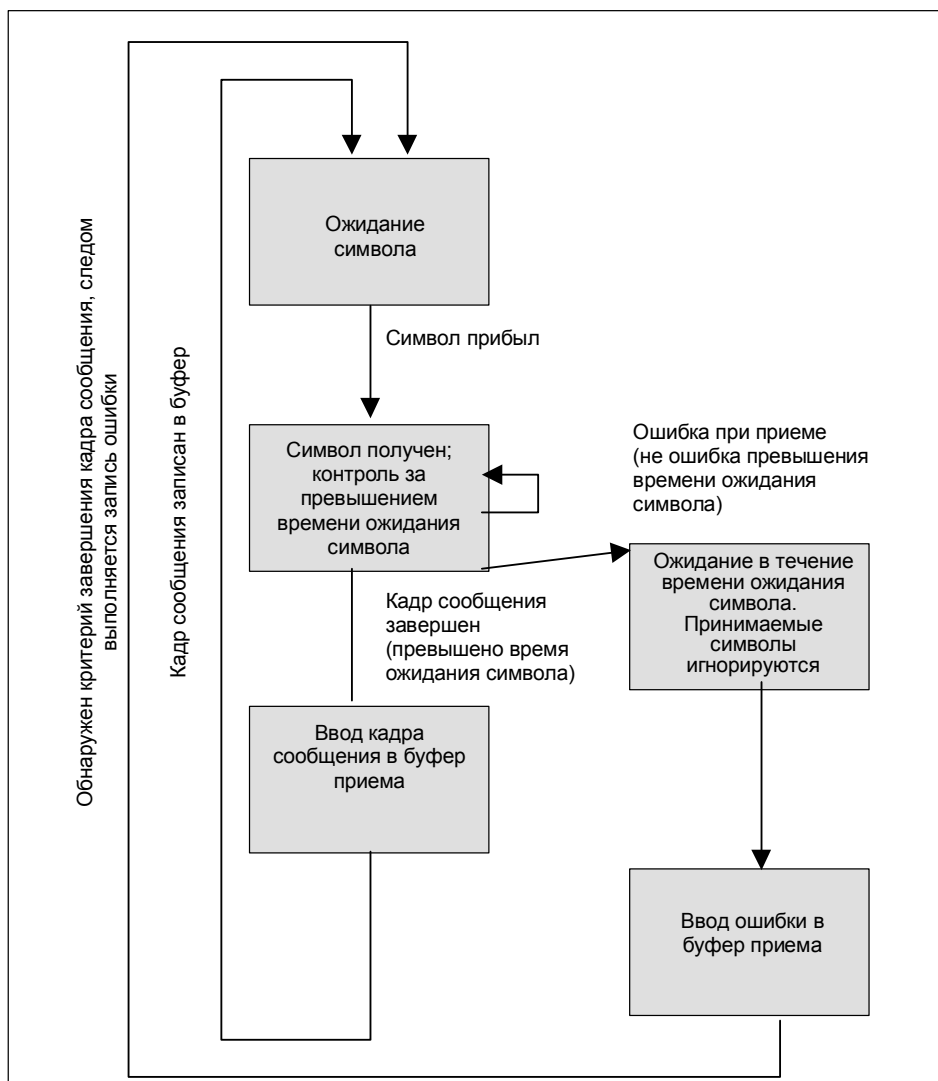


Рисунок 2–18 Последовательность действий при приеме данных с использованием критерия завершения "Превышение времени ожидания символа".

Критерий завершения – "Символ завершения текста"

При приеме данных признаком завершения кадра сообщения является поступление сконфигурированного символа(-ов) завершения текста. Принятые данные, включая символы завершения текста, поступают в CPU с помощью функционального блока S_RCV.

Если в процессе приема данных оказалось превышенным время ожидания символа, операция приема данных прекращается. Генерируется сообщение об ошибке, фрагмент принятого кадра сообщения игнорируется.

Если применяются символы завершения текста, передаваемые данные не являются прозрачными (неинтерпретируемыми), поэтому необходимо следить за тем, чтобы код (коды) завершения не входили в состав данных пользователя.

На рисунке 2-19 показана последовательность действий при приеме данных с использованием критерия завершения "Символ завершения текста".

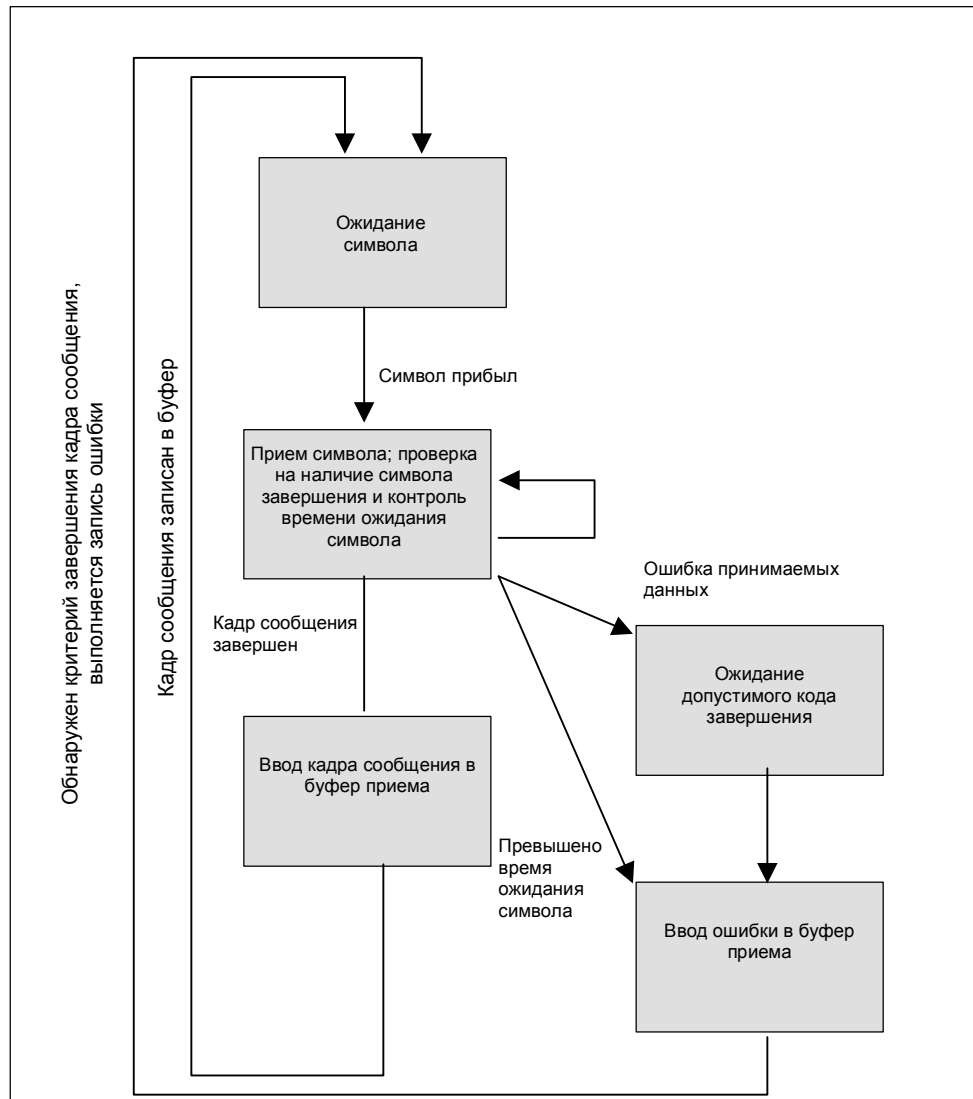


Рисунок 2–19 Последовательность действий при приеме данных с использованием критерия завершения "Символ текста"

Критерий завершения – "Фиксированная длина кадра сообщения"

При приеме данных признаком завершения кадра сообщения является факт приема сконфигурированного количества символов. Принятые данные поступают в CPU с помощью функционального блока S_RCV.

Если время ожидания символа оказалось превышенным до того, как было получено сконфигурированное количество символов, операция приема завершается. Генерируется сообщение об ошибке, принятый фрагмент кадра сообщения игнорируется. На рисунке 2-20 показана последовательность действий при приеме данных с использованием критерия завершения "Фиксированная длина кадра сообщения".

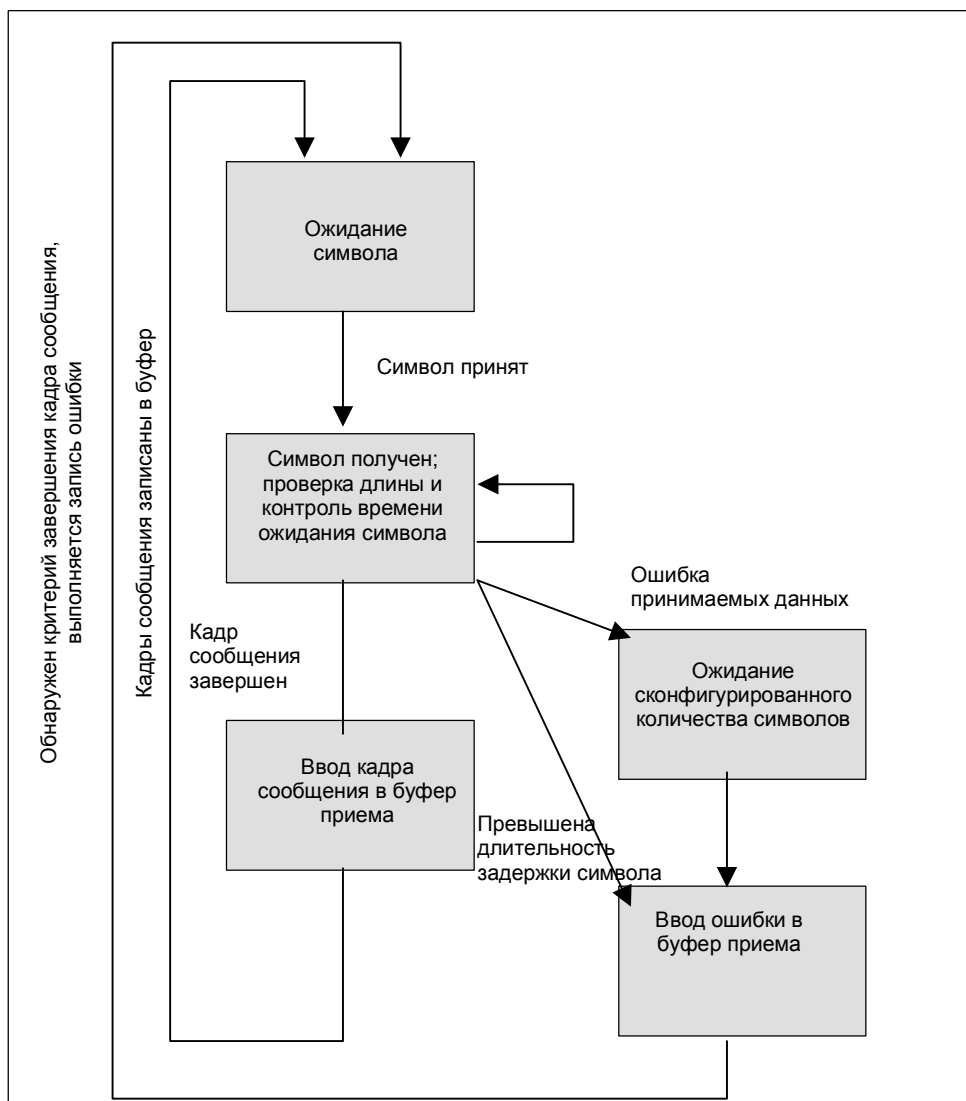


Рисунок 2–20 Последовательность действий при приеме данных с использованием критерия завершения "Фиксированная длина кадра сообщения".

Буфер приема модуля ET 200S

Буфер приема модуля ET 200S SI имеет объем 1024 байта. При назначении параметров можно указать, будет ли буфер приема очищаться при запуске, и будет ли в нем предотвращаться перезапись данных. Можно также отключить или включить функцию резервного копирования принимаемых кадров сообщений.

Буфер приема последовательного интерфейса ET 200S SI является циклическим:

- Если в буфер приема модуля ET 200S SI записывается несколько кадров сообщений, применяется следующее правило: в CPU передается наиболее старый кадр сообщения модуля ET 200S SI.

- Если в CPU требуется передавать лишь самый последний кадр сообщения, необходимо отключить динамические кадры сообщений и отключить защиту от перезаписи.

Примечание

Если непрерывное считывание принимаемых данных, предусмотренное в программе пользователя, прерывается на некоторое время, может случиться так, что при следующем запросе принимаемых данных первым в CPU поступает старый кадр сообщения от модуля ET200S, и лишь потом поступает более новый.

Старый кадр сообщения поступил либо потому, что процесс его передачи из ET200S SI в CPU уже был инициирован, либо он был уже принят в момент прерывания передачи.

Дополнительную информацию о буферах приема можно найти в Разделе 2.11.

Дополнительные сигналы интерфейса RS-232C

Модуль ET 200S SI поддерживает следующие дополнительные сигналы RS-232C:

- **DCD** (Вход) Детектор несущей;
- **DTR** (Выход) Готовность терминала;
ET 200S SI готов к работе.
- **DSR** (Вход) Готовность данных;
Партнер по обмену данными готов к работе
- **RTS** (Выход) Запрос на передачу;
ET 200S SI готов к передаче.
- **CTS** (Вход) Готовность к передаче;
Партнер по обмену данными может принимать данные от модуля ET 200S SI
(Ответ на RTS = ВКЛ модуля ET 200S SI)

После того, как на модули ET 200S было подано питание, выходные сигналы устанавливаются в состояние ВЫКЛ (неактивное состояние).

Способ использования сигналов управления DTR/DSR и RTS/CTS можно сконфигурировать с помощью интерфейса конфигурирования или с помощью функций (FC) в программе пользователя.

Функционирование дополнительных сигналов RS-232C

Дополнительные сигналы RS 232C могут использоваться следующим образом:

- Конфигурирование автоматического функционирования всех дополнительных сигналов RS 232C
- Конфигурирование управления потоком данных (RTS/CTS)
- Использование функциональных блоков (FB) S_VSTAT и S_VSET

Примечание

Если выбрано автоматическое функционирование дополнительных сигналов RS-232C, управление потоком данных с использованием RTS/CTS или управление сигналами RTS и DTR с помощью S_VSET становится невозможным.

Если выбрано управление потоком данных с помощью RTS/CTS, управление сигналом RTS с помощью S_VSET невозможно.

С другой стороны, состояние всех дополнительных сигналов RS-232C всегда можно прочитать с помощью S_VSTAT.

В последующих разделах поясняются основные принципы управления и анализа дополнительных сигналов RS-232C.

Автоматическое функционирование дополнительных сигналов RS–232C

Автоматическое функционирование дополнительных сигналов RS–232C в модуле ET 200S SI реализуется следующим образом:

- Как только модуль ET 200S SI переводится в режим автоматического функционирования дополнительных сигналов RS–232C (в результате конфигурирования), он устанавливает линию RTS в состояние ВЫКЛ, а линию DTR – во ВКЛ (ET 200S SI готов к работе).
Передача и прием кадров сообщений становятся возможными только после того, как линия DTR установлена во ВКЛ. Пока DTR находится в состоянии ВЫКЛ, данные через интерфейс RS–232C приниматься не будут. Задание на передачу данных прерывается, генерируется соответствующее сообщение об ошибке.
- Если **задание на передачу** поставлено в очередь, RTS переводится во ВКЛ и начинается отсчет сконфигурированного времени ожидания вывода данных. По истечении времени ожидания вывода данных и при условии, что CTS = ВКЛ, данные передаются по интерфейсу RS 232C.
- Если во время передачи данных сигнал CTS не установлен в состояние ВКЛ в течение времени ожидания вывода данных, или состояние CTS меняется в процессе передачи на ВЫКЛ, задание на передачу данных прерывается, генерируется соответствующее сообщение об ошибке.
- После того, как данные переданы, линия RTS отключается по истечении сконфигурированного времени передачи RTS. Модуль ET 200S SI не ожидает переключения линии CTS в ВЫКЛ.
- **Прием** данных через интерфейс RS–232C становится возможным, как только линия DSR устанавливается во ВКЛ. Если ожидается переполнение буфера приема модуля ET 200S SI, ответ от модуля ET 200S SI не поступает.
- Когда DSR из ВКЛ переводится в состояние ВЫКЛ, активное задание на передачу, а также процедура приема данных прерываются с генерацией соответствующего сообщения об ошибке.

Примечание

Если выбрано автоматическое функционирование дополнительных сигналов RS-232C, управление потоком данных с помощью RTS/CTS и управление RTS и DTR с использованием S_VSET становится невозможным.

Временная диаграмма

На рисунке 2-21 приводится временная диаграмма обработки задания на передачу.

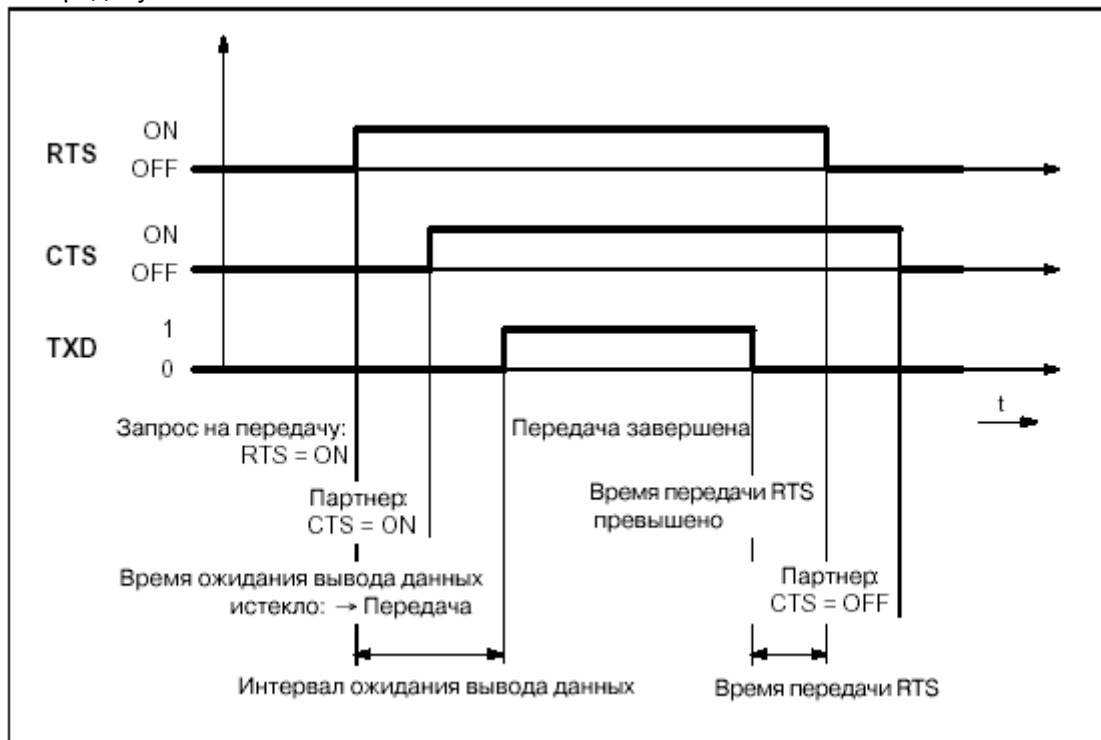


Рисунок 2-21 Временная диаграмма при автоматическом функционировании дополнительных сигналов RS-232C

Процедуры управления потоком данных/синхронизации данных

Для управления потоком данных между двумя коммуникационными партнерами используется процедура синхронизации данных (handshaking). Процедура синхронизации данных предотвращает потерю данных при их передаче между устройствами, функционирующими с разными скоростями передачи. Различают следующие процедуры синхронизации данных:

- Программная синхронизация данных (напр., XON/XOFF)
- Аппаратная синхронизация данных (напр., RTS/CTS)

Управление потоком данных в модуле ET 200S SI реализуется следующим образом:

- После того, как модуль ET 200S SI переведен при конфигурировании в режим управления потоком данных, он передает символ XON и устанавливает линию RTS в состояние ВКЛ.
- Когда достигнуто сконфигурированное количество кадров сообщений, либо когда до заполнения буфера приема остается 50 символов (размер буфера приема = 1024 байта), модуль ET 200S SI передает символ XOFF или устанавливает линию RTS в состояние ВЫКЛ. Если партнер по обмену данными игнорирует эти действия и продолжает передавать данные, буфер приема переполняется и генерируется сообщение об ошибке. Данные, принятые с последним кадром сообщения, будут утеряны.
- Как только кадр сообщения передан в CPU S7 и буфер приема готов к приему, ET 200S SI передает символ XON или устанавливает линию RTS в состояние ВКЛ.
- Если модуль ET 200S SI принимает символ XOFF или если сигнал управления CTS установлен в ВЫКЛ, модуль ET 200S SI прерывает передачу данных. Если в пределах сконфигурированного временного интервала ни символ XON не поступил, ни линия CTS не была установлена во ВКЛ, передача данных прерывается, а в выходном состоянии STATUS функционального блока генерируется соответствующее сообщение об ошибке (0708_n).

Чтение/управление с использованием функциональных блоков S_VSTAT и S_VSET

С помощью функционального блока S_VSTAT можно определить состояние любого дополнительного сигнала RS–232C. С помощью функционального блока S_VSET можно управлять выходными сигналами DTR и RTS. Информация по использованию функциональных блоков в качестве интерфейса между CPU и модулем ET 200S SI приводится в разделе 2-10.

2.9 Конфигурирование и параметрирование модуля последовательного интерфейса

Конфигурирование модуля последовательного интерфейса

Если связь между интерфейсным модулем ET 200S SI и ведущим устройством S7 осуществляется по сети PROFIBUS, для включения модуля в сеть PROFIBUS, а также для настройки параметров связи модуля используются средства конфигурирования аппаратных средств пакета STEP 7.

Если в каталоге аппаратных средств выбрать модуль ET 200S SI и вставить его в корзину ET 200S, в конфигурационную таблицу будут автоматически внесены заказной номер модуля, номер установочного места (слота), а также адреса входов и выходов (адреса ввода/вывода). После этого можно вызвать диалоговое окно свойств модуля ET 200S SI и выбрать в нем тип связи и другие параметры.

Назначение параметров драйверу ASCII

В таблице 2-10 перечислены параметры, которые можно сконфигурировать для ASCII драйвера модуля последовательного интерфейса.

Таблица 2-10 Параметры ASCII драйвера			
Параметры	Описание	Возможные значения	Значения по умолчанию
Диагностическое прерывание	Укажите, должен ли модуль генерировать диагностическое прерывание в случае возникновения серьезной ошибки.	Нет Да	Нет
Тип интерфейса	Укажите используемый интерфейс (см. разделы 2.4 и 2.5).	RS-232C RS-422 (дуплексный режим) RS-485 (полудуплексный режим)	RS-232C
Полудуплексный и дуплексный режим. Исходное состояние линии приема	Укажите исходное состояние линии приема для интерфейсов RS-422 и RS-485. Для интерфейса RS-232C не указывается.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V
Управление потоком данных (с предустановленными параметрами; изменение параметров, принимаемых по умолчанию, в программе пользователя)	Передачу и прием данных можно осуществлять с использованием функции управления потоком данных. Функция управления потоком данных используется для синхронизации передаваемых данных, если один из партнеров по обмену данными передает данные на большей скорости, чем другой партнер. Выберите тип управления потоком данных и задайте соответствующие параметры (см. раздел 2.8). Примечание: При использовании интерфейса RS 485 управление потоком данных невозможно. Управление потоком данных с помощью RTS/CTS и автоматическое управление сигналами V.24 возможно только при использовании интерфейса RS-232C.	Нет XON/OFF RTS/CTS Автоматическое функционирование сигналов V.24	Нет

Таблица 2–10 Параметры ASCII драйвера (продолжение)

Скорость передачи	Выберите скорость передачи данных (бит/с).	110 300 600 1200 2400 4800 9600 19200	9600
Количество битов данных	Выберите количество битов, используемых для передачи символов.	7 8	8
Стоп-биты	Выберите количество стоп-битов, которое будет добавляться к каждому передаваемому символу в процессе передачи для обозначения завершения символа.	1 2	1
Четность	<p>Последовательность передаваемых битов данных можно дополнить одним символом, что позволяет добавить бит четности (паритета). Дополнительный бит (0 или 1) переводит общее значение всех битов (битов данных и бита четности) в определенное состояние.</p> <p>Нет: данные передаются без бита четности.</p> <p>Нечет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было нечетным.</p> <p>Чет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было четным.</p> <p>Любой: состояние бита четности не играет роли. Четность не проверяется при приеме данных, но при передаче данных всегда устанавливается = "0".</p>	None – нет Odd – нечет Even – чет Any – любой	Even – чет

Таблица 2–10 Параметры ASCII драйвера (продолжение)

Признак завершения принимаемого кадра сообщения	<p>Если данные передаются с использованием ASCII драйвера, в качестве признака завершения принимаемого кадра сообщения можно использовать одно из трех возможных событий. В данном параметре можно выбрать один из трех признаков завершения и ввести специальные параметры.</p> <p>Примечание: если время ожидания символа истекает, прежде чем данные приняты, процедура приема прерывается во всех трех режимах. Кадр сообщения будет потерян (за исключением режима "Превышение времени ожидания символа").</p> <p>Превышение времени ожидания символа: признаком завершения кадра сообщения является истечение времени ожидания символа.</p> <p>По приему символа завершения текста: признаком завершения кадра сообщения является поступление заданного символа (-ов) завершения текста.</p> <p>По приему установленного количества символов: признаком завершения кадра сообщения является поступление кадра сообщения сконфигурированной длины. Все принимаемые кадры сообщений имеют одинаковую длину.</p>	<p>Превышение времени ожидания символа</p> <p>По приему символа завершения текста</p> <p>По приему установленного количества символов</p>	Превышение времени ожидания символа
Превышение времени ожидания символа, мс	Максимальный интервал между двумя принимаемыми символами. ¹	4 ... 65535 мс	4 мс
Символ завершения текста 1 ²	<p>Для приема данных с символами завершения текста можно выбрать до двух символов завершения текста. Выбранные символы завершения текста ограничивают длину кадра сообщения.</p>	<p>7 битов данных:³ 1 ... 7F_H</p> <p>8 битов данных:³ 1 ... FF_H</p>	3
Символ завершения текста 2 ²	<p>Для приема данных с символами завершения текста можно выбрать до двух символов завершения текста. Выбранные символы завершения текста ограничивают длину кадра сообщения. Второй код завершения (если указан).</p>	<p>7 битов данных:³ 0 ... 7F_H</p> <p>8 битов данных:³ 0 ... FF_H</p>	0
Длина принимаемого кадра сообщения ⁴	Если принимаемые данные должны иметь фиксированное количество символов, необходимо указать длину кадра. Длина кадра должна точно соответствовать количеству байтов данных, принимаемых партнером по обмену данными.	1 ... 200 байт	100

Таблица 2–10 Параметры ASCII драйвера (продолжение)

Динамический кадр сообщения	Можно указать, будет ли при приеме выполняться резервное сохранение только одного сообщения, либо динамическое резервное сохранение нескольких сообщений. Если выбраны динамические кадры, модуль может выполнять резервное сохранение нескольких сообщений различной длины. Для этих целей используется циклический буфер. В случае переполнения буфера запись выполняется поверх наиболее старого сообщения, если при параметрировании не было выбрана защита от перезаписи буфера. В последнем случае самое последнее сообщение будет утеряно. В обоих случаях признаком потери данных является диагностическое прерывание.	Включен Отключен	Включен
Предотвращение перезаписи буфера	Данный параметр позволяет предотвратить перезапись кадров сообщений, хранящихся в буфере резервного хранения, в случае поступления на модуль нового кадра сообщения в момент, когда буфер приема полон и его содержимое не удалено. Данный параметр позволяет предотвратить стирание старых кадров сообщений.	Нет Да	Да
Очистка приемного буфера SI при запуске	Укажите, должен ли буфер приема модуля автоматически очищаться, когда CPU переходит из режима STOP в режим RUN (запуск CPU). Это позволяет организовать работу таким образом, чтобы в буфер приема модуля поступали только те кадры сообщения, которые были приняты после запуска CPU.	Нет Да	Да
¹ Наименьшее значение времени ожидания символа зависит от скорости передачи. Смотрите таблицу 2-9 (стр. 2-43). ² Можно установить лишь тогда, когда критерием завершения является символ завершения текста. ³ Зависит от параметров, выбранных для кадров символа (7 или 8 битов данных). ⁴ Может быть выбрано, если критерием завершения является прием кадра фиксированной длины.			

Назначение параметров для драйвера протокола 3964(R)

В таблице 2-11 перечислены параметры, которые могут быть установлены для протокола 3964(R) модуля последовательного интерфейса.

Таблица 2–11 Параметры протокола 3964(R)			
Параметры	Описание	Возможные значения	Значение по умолчанию
Диагностическое прерывание	Укажите, должен ли модуль генерировать диагностическое прерывание в случае возникновения серьезной ошибки.	Нет Да	Нет
Тип интерфейса	Выберите используемый интерфейс (см. разделы 2.4 и 2.5).	RS–232C RS–422	RS–232C
Исходной состояние линии приема	Укажите исходное состояние линии приема для интерфейса RS–422. Для интерфейса RS–232C не указывается.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V

Таблица 2–11 Параметры протокола 3964(R) (продолжение)

Режим протокола	<p>Укажите, должен ли вместе с данными передаваться символ контроля блока (BCC), позволяющий повысить целостность передаваемых данных.</p> <p>Контрольная сумма блока вычисляется путем взятия "исключающего ИЛИ" по всем байтам данных передаваемого или принимаемого блока. Если партнер по обмену данными обнаруживает символ контроля блока при приеме данных (BCC), он сравнивает BCC со значением, рассчитанным им самим по принимаемым данным. Если принятый символ контроля блока не совпал с рассчитанным значением, партнер выдерживает паузу длительность 4 секунды (время ожидания блока), после чего передача данных повторяется.</p> <p>Если за сконфигурированное количество попыток передачи блок данных не смог быть принят, или в течение времени ожидания блока не было предпринято других попыток передачи, процедура приема прекращается.</p>	<p>No block check (Без символа контроля блока)</p> <p>Block check (С символом контроля блока)</p>	<p>Block check (С символом контроля блока)</p>
Скорость передачи	Выберите скорость передачи данных (бит/с).	<p>110</p> <p>300</p> <p>600</p> <p>1200</p> <p>2400</p> <p>4800</p> <p>9600</p> <p>19200</p>	9600
Количество битов данных	Выберите количество битов, из которых будет состоять символ.	<p>7</p> <p>8</p>	8
Стоп-биты	Выберите количество стоп-битов, которое будет добавляться к каждому передаваемому символу в процессе передачи для индикации завершения символа.	<p>1</p> <p>2</p>	1
Четность	<p>Последовательность передаваемых битов данных можно дополнить одним символом, что позволяет добавить бит четности (паритета). Дополнительный бит (0 или 1) переводит общее значение всех битов (битов данных и бита четности) в определенное состояние.</p> <p>Нет: данные передаются без бита четности.</p> <p>Нечет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было нечетным.</p> <p>Чет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было четным.</p> <p>Любой: состояние бита четности не играет роли. Четность не проверяется при приеме данных, но при передаче данных всегда устанавливается = "0".</p>	<p>None – Нет</p> <p>Odd – Нечет</p> <p>Even – Чет</p> <p>Any - Любой</p>	Even – Чет

Таблица 2–11 Параметры протокола 3964(R) (продолжение)

Время ожидания символа (мс)	Максимальный временной интервал между приемом двух символов. Введите наименьшее для своего приложения значение времени ожидания символа. Следует помнить, что время ожидания символа не должно быть меньше определенного значения, зависящего от скорости передачи.	20 ... 655350 мс с шагом 10 мс	220 мс
Время ожидания подтверждения (мс)	Выберите максимальное время, в течение которого от партнера по обмену данными может быть получено подтверждение при установке или сбросе соединения. Следует помнить, что время ожидания подтверждения не должно быть меньше определенного минимального значения, зависящего от скорости передачи.	10 ... 655350 мс с шагом 10 мс	2000 мс (550 мс без контроля блока)
Количество попыток установления	Укажите количество (<i>n</i>) попыток установления соединения. (После <i>n</i> неуспешных попыток функция прерывается и в выходном состоянии STATUS функционального блока S_SEND отображается ошибка).	1 ... 255	6
Количество попыток передачи	Укажите количество (<i>n</i>) попыток передачи кадра сообщения (после <i>n</i> неуспешных попыток передачи кадра сообщения функция прерывается и в выходном состоянии STATUS функционального блока S_SEND отображается ошибка). Возможными причинами прерывания являются: Ошибка четности Ошибка BCC, ошибка четности Различаются настройки коммуникационных партнеров (скорость передачи, четность, кадр символа, символ контроля блока, различные протоколы и т.п.).	1 ... 255	6
Приоритет	Если оба партнера выставляют задания на передачу одновременно, партнер с более низким приоритетом отзовет свое задание на передачу. Наивысший приоритет должен быть назначен только одному партнеру по обмену данными, другому партнеру должен быть назначен более низкий приоритет передачи данных.	High – Высший Low – Низший	Низший
Стирать буфер приема SI при запуске	Укажите, должен ли приемный буфер модуля автоматически очищаться при переходе CPU из режима STOP в режим RUN (запуск CPU). Это позволяет организовать работу таким образом, что буфер приема модуля будет принимать только кадры сообщения, которые были получены после запуска CPU.	Нет Да	Да

2.10 Обмен данными с использованием функциональных блоков

Обзор

Связь между CPU, ET 200S SI и партнером по обмену данными осуществляется посредством функциональных блоков и протоколов модуля ET 200S SI (сведения об обмене данными с CPU других производителей (не S7) приводятся в разделе 2.12).

Функциональные блоки служат для формирования программного интерфейса между CPU и модулем последовательного интерфейса ET 200S SI. Их следует вызывать циклически из программы пользователя.

Установка связи с CPU

При каждом новом запуске CPU системная служба CPU назначает модулю ET 200S SI актуальные параметры. После того, как соединение между CPU и модулем ET 200S SI установлено, последний нуждается в инициализации.

Каждый функциональный блок обладает своим собственным механизмом запуска. Задания могут выполняться в активном режиме лишь после завершения соответствующей процедуры запуска.

Модуль ET 200S SI может вызвать диагностическое прерывание в CPU. Операционная система предоставляет пользователю 2 байта данных при таком прерывании. Эти данные должны анализироваться в программе пользователя (OB82). Функциональные блоки не могут вызываться при обработке прерывания или из программы диагностического прерывания. В функциональных блоках вызовы прерываний не отключаются.

Смена протокола происходит в модуле ET 200S SI. В зависимости от выбранного протокола (протокол 3964(R) или ASCII драйвер), параметры интерфейса модуля ET 200S SI настраиваются в соответствии с настройками интерфейса коммуникационного партнера.

Функциональные блоки модуля ET 200S SI

Программируемый логический контроллер S7-300 предоставляет пользователю ряд функциональных блоков, с помощью которых в программе пользователя можно инициировать и управлять обменом данными между CPU и модулем последовательного интерфейса ET 200S SI. Функциональные блоки (FB), используемые модулем ET 200S SI, перечислены в таблице 2-12.

Таблица 2–12 Функциональные блоки модуля ET 200S SI			
FB	Прежнее название	Новое название	Назначение
FB2	P_RCV	S_RCV	Функциональный блок S_RCV позволяет принимать данные от коммуникационного партнера и записывать их в блок данных.
FB3	P_SEND	S_SEND	С помощью функционального блока S_SEND блок данных или его часть может быть передана коммуникационному партнеру.
FB4	V24_STAT	S_VSTAT	Функциональный блок S_VSTAT позволяет считывать состояния сигналов интерфейса RS–232C модуля ET 200S SI.
FB5	V24_SET	S_VSET	С помощью функционального блока S_VSET можно устанавливать/сбрасывать выходные состояния отдельных сигналов интерфейса RS–232C модуля ET 200S SI.
FB6	F_XON	S_XON	Функциональный блок S_XON позволяет настраивать дополнительные параметры, если при конфигурировании модуля было выбрано управление потоком данных XON/XOFF.
FB7	F_RTS	S_RTS	Функциональный блок S_RTS позволяет настраивать дополнительные параметры, если при конфигурировании модуля было выбрано управление потоком данных RTS/CTS.
FB8	F_V24	S_V24	Функциональный блок S_V24 позволяет настраивать дополнительные параметры, если при конфигурировании модуля было выбрано автоматическое функционирование сигналов V.24.

Примечание переводчика: в колонке "Прежнее название" указаны устаревшие наименования FB, использовавшиеся в английской версии руководства. В русской версии используются наименования, указанные в колонке "Новое название".

FB3 S_SEND: Передача данных коммуникационному партнеру

Функциональный блок S_SEND выполняет передачу блока данных, определяемого параметрами DB_NO, DBB_NO и LEN, на модуль ET 200S SI. Функциональный блок S_SEND вызывается циклически и статически (безусловный вызов). Как альтернатива, допускается также вызов по временному прерыванию.

Передача данных инициируется по переднему фронту на входе REQ. Для передачи данных может потребоваться несколько вызовов (циклов программы), в зависимости от объема передаваемых данных.

Если при циклическом вызове функционального блока S_SEND входной параметр R находится в состоянии "1", передача на ET 200S SI прерывается, а функциональный блок S_SEND сбрасывается в свое исходное состояние. Данные, которые уже были приняты модулем ET 200S SI, продолжают передаваться коммуникационному партнеру. Неизменное (статическое) состояние "1" на входе R означает отключение передачи.

Адрес модуля ET 200S SI, к которому происходит обращение, указывается в параметре LADDR.

Выход DONE указывает на завершение задания без ошибок. Наличие ошибок индицируется выходом ERROR. При наличии ошибок номер соответствующей ошибки отображается в выходном состоянии STATUS (см. раздел 2.13). Если ошибок не произошло, в STATUS содержится "0". Состояние выходов DONE и ERROR/STATUS также актуализируется по сбросу (вход R) функционального блока S_SEND (см. рис. 2-22 на стр. 2-62). В случае ошибки сбрасывается бит бинарного результата BR. Если работа блока завершается без ошибок, бит BR принимает состояние "1".

Примечание

Проверка параметров в функциональном блоке S_SEND не предусмотрена. В случае задания недопустимых параметров CPU переходит в режим STOP.

Выполнение инициированного задания модулем ET200S SI после перехода CPU из режима STOP в RUN возможно только после завершения процедуры, предусмотренной в модуле ET200S при запуске CPU для функционального блока P_PRINT (см. стр. 2-58). Задания, инициированные в процессе работы механизма запуска CPU, утеряны не будут. Они будут переданы модулю ET200S SI по завершении процедуры запуска.

Вызов FB3

Представление в форме STL	Представление в форме LAD
<pre> CALL S_SEND, I_SEND REQ: = R: = LADDR: = DB_NO: = DBB_NO: = LEN: = DONE: = ERROR: = STATUS: = </pre>	

Примечание

Параметры EN и ENO отображаются только при графическом представлении блока (LAD или FBD). Для обработки данных параметров компилятор использует бит BR.

Бит BR устанавливается в состояние "1", если работа блока была завершена без ошибок. В случае ошибки BR переводится в "0".

Присвоения в области данных

Функциональный блок S_SEND работает с экземплярным блоком данных (DB) (I_SEND). Номер DB указывается при вызове. Обращение к данным в экземплярном DB невозможно.

Примечание

Исключение: в случае ошибки, при которой выход STATUS == W#16#1E0F, более подробные сведения об ошибке можно получить из переменной SFCERR (см. раздел 2.12). Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символьного доступа к экземпляру DB.

Параметры FB3 S_SEND

В таблице 2–13 приводятся параметры функционального блока S_SEND (FB3).

Таблица 2–13 FB3: Параметры функционального блока S_SEND				
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
REQ	INPUT	BOOL	Инициирование задания по положительному фронту	
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес, взятый из STEP 7.
DB_NO	INPUT	INT	Номер блока данных	Номер передаваемого блока: зависит от CPU ("0" указывать не допускается).
DBB_NO	INPUT	INT	Номер байта данных	$0 \leq \text{DBB_NO} \leq 8190$ данные передаются в формате слов
LEN	INPUT	INT	Длина данных	$1 \leq \text{LEN} \leq 200$, указывается количество байтов
DONE ¹	OUTPUT	BOOL	Завершение задания без ошибок	Параметр STATUS == 16#00
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибками	В параметре STATUS содержатся сведения об ошибке
STATUS ¹	OUTPUT	WORD	Спецификация ошибки	Если ERROR == 1, в параметре STATUS содержатся сведения об ошибке

¹ Параметр DONE доступен в течение **одного** цикла CPU после успешного выполнения задания.

Временная диаграмма для FB3 S_SEND

На рисунке 2-22 показана последовательность переключения параметров DONE и ERROR в зависимости от того, какие сигналы подаются на входы REQ и R.

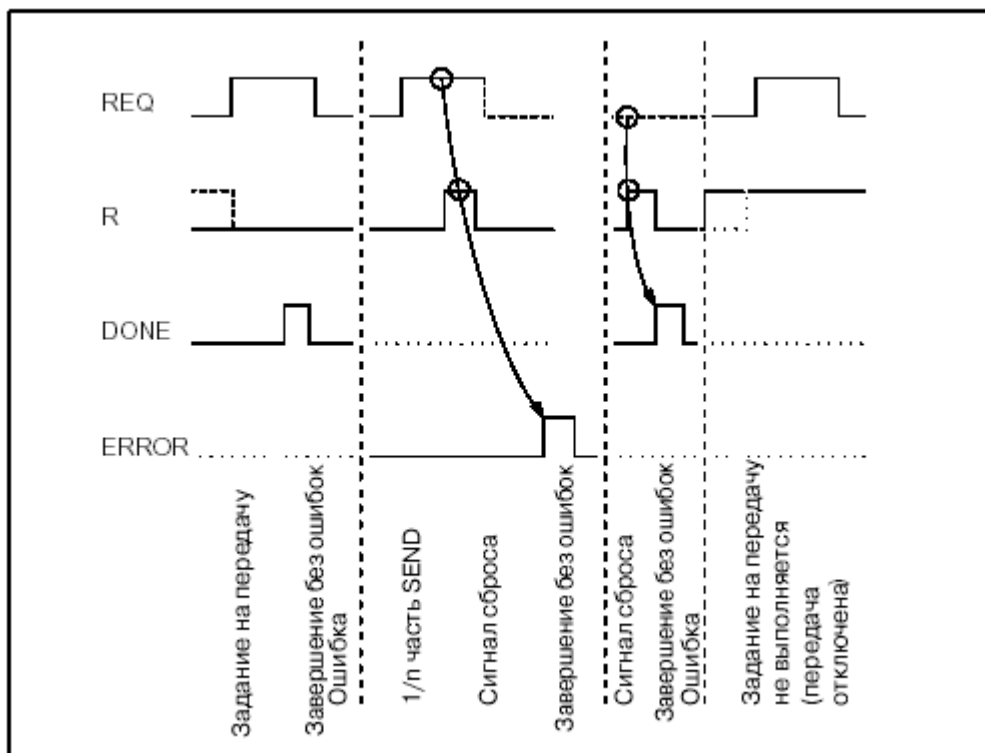


Рисунок 2-22 Временная диаграмма для FB3 S_SEND

Примечание

Запуск происходит по переднему фронту на входе REQ. Переключение сигнала из "0" в "1" на входе REQ (передний фронт) является достаточным условием для инициирования передачи. Значение RLO (RLO = результат логической операции) не обязательно должно быть = "1" в течение всего времени передачи данных.

FB2 S_RCV: Прием данных от коммуникационного партнера

Функциональный блок S_RCV выполняет передачу данных из модуля ET 200S SI в область данных S7, определяемую параметрами DB_NO и DBB_NO. Передача данных осуществляется в результате циклического вызова функционального блока S_RCV. Как альтернатива, этот блок можно вызывать статически по временному прерыванию (безусловный вызов).

Статическое (неизменное) состояние "1" параметра EN_R означает, что модуль ET 200S SI будет проверять необходимость чтения данных. Активную процедуру передачи данных можно прервать, сбросив параметр EN_R в состояние "0". Прерванное задание на прием завершается сообщением об ошибке (выход STATUS). Прием будет выключен в течение всего времени, пока параметр EN_R находится в состоянии "0". Для передачи данных может потребоваться несколько вызовов (программных циклов), в зависимости от объема передаваемых данных.

Если функциональный блок обнаружит состояние "1" на входе R, текущее задание на передачу прерывается и функциональный блок S_RCV переводится в свое исходное состояние. Прием будет выключен в течение всего времени, пока параметр R находится в состоянии "1". Когда вход R вновь возвращается в состояние "0", прерванный кадр сообщения принимается вновь с самого начала.

Адрес модуля ET 200S SI, к которому происходит обращение, указывается в параметре LADDR.

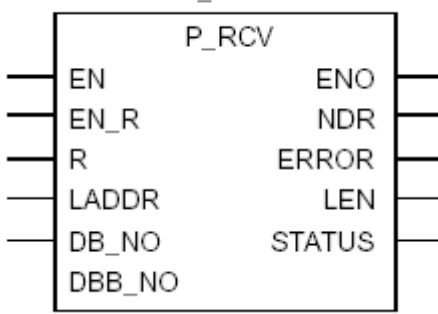
Выход NDR указывает, что задание было завершено без ошибок (все данные прочитаны). Выход ERROR указывает на наличие ошибки. В случае возникновения ошибки в выходном состоянии STATUS содержится соответствующий номер ошибки. Если приемный буфер заполнен более чем на 2/3, на выход STATUS поступает предупреждение при каждом вызове S_RCV, если выход ERROR не установлен в "1". При отсутствии ошибок или предупреждений в STATUS содержится "0".

Сигналы NDR и ERROR/STATUS также актуализируются при сбросе функционального блока S_RCV (параметр LEN == 16#00). В случае ошибки сбрасывается бит BR. Если работа блока завершается без ошибок, бит BR принимает состояние "1".

Примечание

В функциональном блоке S_RCV не предусмотрена проверка параметров. Если заданы недопустимые параметры, CPU переключается в режим STOP. Инициированное задание может быть принято модулем ET200S SI после переключения CPU из режима STOP в RUN только после того, как завершится процедура, предусмотренная в модуле ET200S при запуске CPU для функционального блока S_RCV (см. стр. 2-58).

Вызов FB 2

Представление в формате STL	Представление в формате LAD
CALL S_RCV, I_RCV EN_R: = R: = LADDR: = DB_NO: = DBB_NO: = NDR: = ERROR: = LEN: = STATUS: =	

Примечание

Параметры EN и ENO отображаются только при графическом представлении блока (LAD или FBD). Для обработки данных параметров компилятор использует бит BR.

Бит BR устанавливается в состояние "1", если работа блока была завершена без ошибок. В случае ошибки бит BR переводится в "0".

Присвоение в области данных

Функциональный блок S_RCV работает с экземплярным DB (I_RCV). Номер DB указывается при вызове. Обращение к данным экземплярного DB невозможно.

Примечание

Исключение: в случае ошибки, когда STATUS == W#16#1E0D, более подробные сведения об ошибке можно получить из переменной SFCERR. Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземплярному DB.

Параметры FB2 S_RCV

В таблице 2-14 приводятся параметры функционального блока S_RCV.

Таблица 2–14 FB2: параметры S_RCV				
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
EN_R	INPUT	BOOL	Разрешение чтения данных	
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Прием прекращен.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес, взятый из STEP 7.
DB_NO	INPUT	INT	Номер блока данных	Номер принимаемого блока данных: зависит от CPU, "0" указывать не допускается
DBB_NO	INPUT	INT	Номер байта данных	$0 \leq \text{DBB_NO} \leq 8190$ принимаемые данные в формате слов
NDR ¹	OUTPUT	BOOL	Задание завершено без ошибок, данные приняты	Параметр STATUS == 16#00
ERROR ¹	OUTPUT	BOOL	Задание завершено с ошибками	Параметр STATUS содержит сведения об ошибке
LEN ¹	OUTPUT	INT	Длина принятого кадра сообщения	$1 \leq \text{LEN} \leq 200$, указывается количество байтов
STATUS ¹	OUTPUT	WORD	Описание ошибки	Если ERROR == 1, в параметре STATUS содержатся сведения об ошибке
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания.				

Временная диаграмма для функционального блока S_RCV FB2

На рисунке 2-23 показана последовательность переключения параметров NDR, LEN и ERROR, в зависимости от подаваемых сигналов на входы EN_R и R.

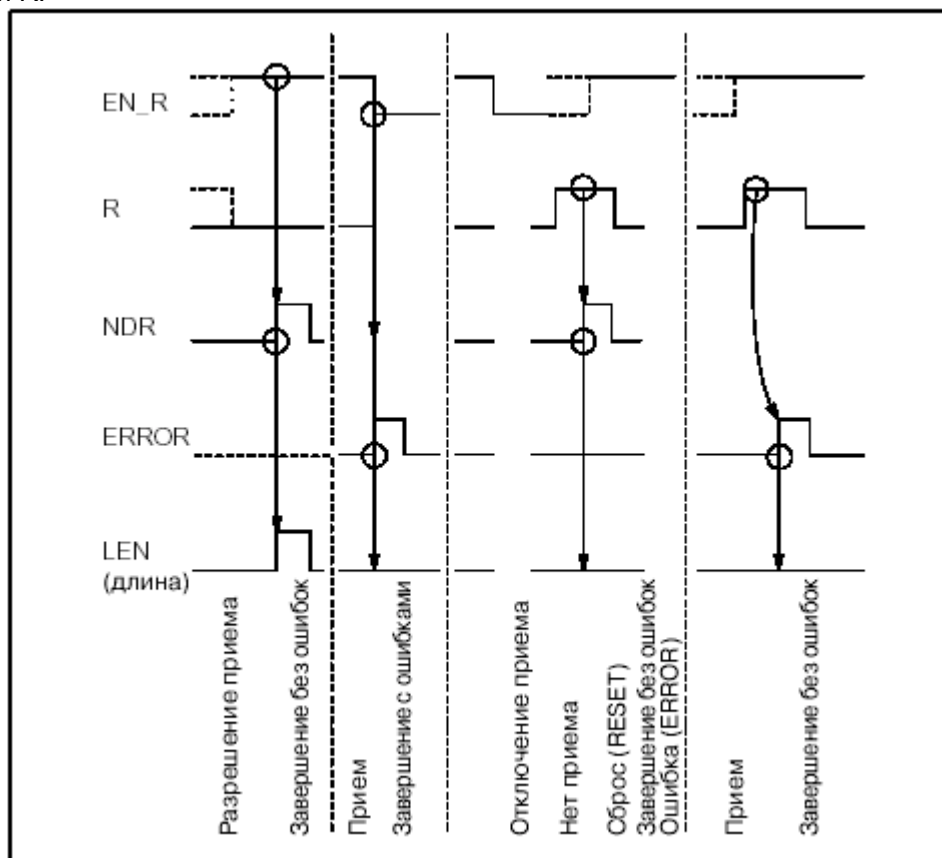


Рисунок 2–23 Временная диаграмма для FB2 S_RCV

Примечание

Вход EN_R должен быть постоянно (статически) установлен в "1". В течение выполнения всего задания на прием для параметра EN_R необходимо, чтобы RLO (результат логической операции) был = "1".

Функции назначения параметров для различных вариантов управления потоком данных

Если модуль последовательного интерфейса ET 200S используется совместно с CPU S7, и конфигурирование модуля выполняется с помощью программы конфигурирования аппаратных средств STEP 7, можно выбрать один из следующих методов управления потоком данных:

- None (Управление отсутствует)
- XON/XOFF
- RTS/CTS
- Automatic operation of the V.24 signals
(Автоматическое функционирование сигналов V.24)

Для каждого из данных методов можно настроить дополнительные параметры. По умолчанию эти параметры имеют некоторые типичные для большинства задач значения. Данные параметры можно изменить с помощью программы пользователя и функциональных блоков, описание которых приводится ниже.

FB6 S_XON: настройка символов для XON/XOFF

С помощью функционального блока S_XON можно настроить дополнительные параметры (см. таблицу 2-15), если при конфигурировании модуля было выбрано управление потоком XON/XOFF.

Представление в виде STL	Представление в виде LAD
<pre>CALL S_XON, I_XON REQ: = R: = LADDR: = XON: = XOFF: = WAIT_FOR_XON: = DONE: = ERROR: = STATUS: =</pre>	

Присвоение в области данных

Функциональный блок S_XON работает с экземплярным DB (I_XON). Номер DB указывается при вызове. Обращение к данным экземплярного DB невозможно.

Примечание

Исключение: в случае ошибки, STATUS == W#16#1E0D, более подробные сведения об ошибке можно получить из переменной SFCERR (см. раздел 2.12). Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземпляру DB.

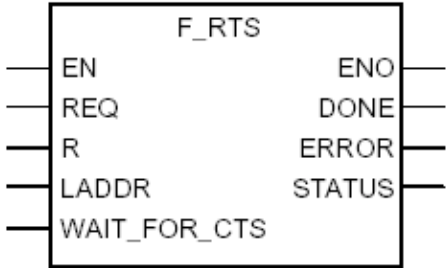
Параметры FB6 S_XON

В таблице 2-15 приводятся параметры FB6.

Таблица 2–15 FB6: параметры					
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии	Значение по умолч.
REQ	INPUT	BOOL	Инициирование задания по переднему фронту		
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.	
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес берется из STEP 7.	
XON	INPUT	BYTE	Символ XON	0 ... 7F _H (7 битов данных) 0 ... FF _H (8 битов данных)	11 (DC1)
XOFF	INPUT	BYTE	Символ XOFF	0 ... 7F _H (7 битов данных) 0 ... FF _H (8 битов данных)	13 (DC3)
WAIT_FOR_XON	INPUT	TIME	Интервал ожидания переключения из XON в XOFF	20 мс ... 10 мин 55 с 350 мс	2 с
DONE ¹	OUTPUT	BOOL	Завершение задания без ошибок	Параметр STATUS == 16#00;	
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибкой	В параметре STATUS содержатся сведения об ошибке	
STATUS ¹	OUTPUT	WORD	Спецификация ошибки	Если ERROR == 1, в параметре STATUS содержатся сведения об ошибке	
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания					

FB7 S_RTS: Настройка параметров для RTS/CTS

С помощью функционального блока S_RTS можно настроить дополнительные параметры (см. таблицу 2-16), если при конфигурировании модуля было выбрано управление потоком RTS/CTS.

Представление в виде STL	Представление в виде LAD
<pre>CALL S_RTS, I_RTS REQ: = R: = LADDR: = WAIT_FOR_CTS: = DONE: = ERROR: = STATUS: =</pre>	

Присвоение в области данных

Функциональный блок S_RTS работает с экземплярным DB (I_RTS). Номер DB указывается при вызове. Обращение к данным экземплярного DB невозможно.

Примечание

Исключение: в случае ошибки, STATUS == W#16#1E0D, более подробные сведения об ошибке можно получить из переменной SFCERR (см. раздел 2.13). Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземплярному DB.

Параметры FB7

В таблице 2-16 приводятся параметры FB7.

Таблица 2–16 FB7: параметры для S_RTS					
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии	Значение по умолч.
REQ	INPUT	BOOL	Инициирование задания по переднему фронту		
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.	
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес берется из STEP 7.	
WAIT_FOR_CTS	INPUT	TIME	Интервал ожидания переключения CTS в сост. ВКЛ	20 мс ... 10 мин 55 с 350 мс	2 с
DONE ¹	OUTPUT	BOOL	Завершение задания без ошибок	Параметр STATUS == 16#00;	
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибками	В параметре STATUS содержатся сведения об ошибке	
STATUS ¹	OUTPUT	WORD	Спецификация ошибок	Если ERROR == 1, то в параметре STATUS содержатся сведения об ошибке	
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания					

FB8 S_V24: настройка параметров для автоматического функционирования дополнительных сигналов RS–232C

С помощью функционального блока S_V24 можно настроить дополнительные параметры (см. табл. 2-17), если при конфигурировании модуля было выбрано автоматическое функционирование дополнительных сигналов RS–232C.

Представление в формате STL	Представление в формате LAD
<pre> ВЫЗОВ S_V24, I_V24 REQ: = R: = LADDR: = TIME_RTS_OFF: = DATA_WAIT_TIME: = DONE: = ERROR: = STATUS: = </pre>	

Присвоение в области данных

Функциональный блок S_V24 работает с экземплярным DB (I_V24). Номер DB указывается при вызове. Обращение к данным в экземплярном DB невозможно.

Примечание

Исключение: в случае ошибки, STATUS == W#16#1E0D, более подробные сведения об ошибке можно получить из переменной SFCERR (см. раздел 2.12). Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземпляру DB.

Параметры FB8

В таблице 2-17 приводятся параметры FB8.

Таблица 2–17 Параметры FB8 для S_V24					
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии	Значение по умолч.
REQ	INPUT	BOOL	Инициирование задания по переднему фронту		
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.	
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес берется из STEP 7.	
TIME_RTS_OFF	INPUT	TIME	Время, которое должно истечь до отключения RTS	0 мс10 мин 55 с 350 мс	10 мс
DATA_WAIT_TIME	INPUT	TIME	Время, в течение которого следует ожидать установления партнером CTS в сост. ВКЛ после установления RTS во ВКЛ.	0 мс10 мин 55 с 350 мс	10 мс
DONE ¹	OUTPUT	BOOL	Завершение задания без ошибок	Параметр STATUS == 16#00;	
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибками	В параметре STATUS содержатся сведения об ошибке	
STATUS ¹	OUTPUT	WORD	Спецификация ошибок	Если ERROR == 1, то в параметре STATUS содержатся сведения об ошибке	
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания.					

Чтение и управление дополнительными сигналами RS–232C

Для чтения и управления дополнительными сигналами RS–232C можно использовать функциональный блок FB4 S_VSTAT, служащий для чтения состояний линий интерфейса, и FB S_VSET, позволяющий устанавливать/сбрасывать состояния линий интерфейса.

FB4 S_VSTAT: Чтение состояния линий интерфейса модуля ET 200S SI

Функциональный блок S_VSTAT выполняет чтение состояния дополнительных сигналов интерфейса RS–232C модуля ET 200S SI и предоставляет их пользователю в параметрах блока. Функциональный блок S_VSTAT можно вызывать статически и циклически (безусловный вызов). В качестве альтернативы возможен также вызов по временному прерыванию. Состояния дополнительных сигналов RS–232C обновляются при каждом вызове функции (циклический опрос).

Адрес модуля ET 200S SI, к которому производится обращение, указывается в параметре LADDR.

Вызов FB4

Представление в виде STL	Представление в виде LAD
<pre> CALL S_VSTAT, I_STAT REQ: = R: = LADDR: = DONE: = ERROR: = STATUS: = DTR_OUT: = DSR_IN: = RTS_OUT: = CTS_IN: = DCD_IN: = </pre>	

Примечание

Параметры EN и ENO отображаются только при графическом представлении блока (LAD или FBD). Для обработки этих параметров компилятором используется значение бита BR.

Значение бита BR устанавливается в состояние "1", если работа блока была завершена без ошибок. В случае ошибки BR сбрасывается в "0".

Присвоение в области данных

Функциональный блок S_V24 STAT работает с экземплярным DB (I_V24 STAT). Номер DB указывается при вызове. Обращение к данным в экземплярном DB невозможно.

Примечание

Изменение состояния сигнала может быть обнаружено лишь в том случае, если длительность импульса больше определенного минимального значения. К определяющим факторам относятся длительность цикла CPU, интервал обновления модуля ET200S SI и время отклика партнера по обмену данными.

Параметры FB4 S_VSTAT

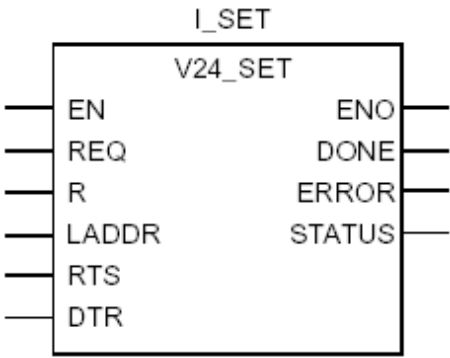
В таблице 2-18 приводятся параметры функционального блока S_VSTAT (FB4).

Таблица 2-18 Параметры FB4 S_VSTAT				
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
REQ	INPUT	BOOL	Инициирование задания по переднему фронту	
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес берется из STEP 7.
DONE ¹	OUTPUT	BOOL	Индикация завершения работы функционального блока	(Выход ET 200S SI)
ERROR ¹	OUTPUT	BOOL	Завершение задания без ошибок	В параметре STATUS содержатся сведения об ошибке
STATUS ¹	OUTPUT	WORD	Спецификация ошибок	Если ERROR == 1, то в параметре STATUS содержатся сведения об ошибке
DTR_OUT ¹	OUTPUT	BOOL	Готовность терминала: ET 200S SI готов к работе	(Выход ET 200S SI)
DSR_IN ¹	OUTPUT	BOOL	Готовность данных: партнер по обмену данными готов к работе.	(Вход ET 200S SI)
RTS_OUT ¹	OUTPUT	BOOL	Запрос на передачу: ET 200S SI готов к передаче данных.	(Выход ET 200S SI)

CTS_IN ¹	OUTPUT	BOOL	Готовность к передаче: партнер по обмену данными может принимать данные от модуля ET 200S SI (ответ на состояние ВКЛ линии RTS модуля ET 200S SI)	(Вход ET 200S SI)
DCD_IN ¹	OUTPUT	BOOL	Детектор несущей.	(Вход ET 200S SI)
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания.				

FB5 S_VSET: Установка/сброс состояний сигналов интерфейса модуля ET 200S SI

Выходные сигналы интерфейса можно устанавливать и сбрасывать с помощью соответствующих входов функционального блока S_VSET. Функциональный блок V24_SET вызывается статически (безусловный вызов) в пределах цикла. Кроме того, возможен вызов по временному прерыванию. Адрес модуля ET 200S SI, к которому выполняется обращение, указывается в параметре LADDR.

Представление в формате STL	Представление в формате LAD
<pre>CALL S_VSET, I_SET REQ = R = LADDR: = RTS: = DTR: = DONE: = ERROR: = STATUS: =</pre>	

Примечание

Параметры EN и ENO отображаются только при графическом представлении блока (LAD или FBD). Для обработки данных параметров компилятором используется значение бита BR. Значение бита BR устанавливается в состояние "1", если работа блока была завершена без ошибок. В случае ошибки BR сбрасывается в "0".

Присвоение в области данных

Функциональный блок S_VSET работает с экземплярным DB (I_SET). Номер DB указывается при вызове. Доступ к данным экземплярного DB невозможен.

Параметры FB5 S_VSET

В таблице 2-19 приводятся параметры функционального блока S_VSET (FB5).

Таблица 2–19		FB5: параметры S_VSET		
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
REQ	INPUT	BOOL	Инициирование задания по переднему фронту	
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес берется из STEP 7.
RTS	INPUT	BOOL	Запрос на передачу, ET 200S SI готов к передаче данных.	(Выход ET 200S SI)
DTR	INPUT	BOOL	Готовность терминала, ET 200S SI готов к работе	(Выход ET 200S SI)
DONE ¹	OUTPUT	BOOL	Индикация завершения работы функционального блока	(Выход ET 200S SI)
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибками	В параметре STATUS содержатся сведения об ошибке
STATUS ¹	OUTPUT	WORD	Спецификация ошибок	Если ERROR == 1, то в параметре STATUS содержатся сведения об ошибке
¹ Параметр DONE доступен в течение одного цикла CPU после успешного выполнения задания.				

2.11 Выполняемые функции и режимы при запуске

Режимы модуля последовательного интерфейса ET 200S SI

Модуль ET 200S SI может находиться в следующих режимах:

- **STOP:** когда модуль ET 200S SI находится в режиме STOP (СТОП), ни один из протоколов не активен, и на все задания на передачу и прием CPU отвечает отказом (отрицательным подтверждением). Модуль ET 200S SI пребывает в режиме STOP до тех пор, пока не будет устранена причина, вызывающая режим STOP (обрыв провода, неверно заданный параметр и т.п.).

- **Изменение конфигурации:** при изменении конфигурации модуля ET 200S SI происходит инициализация драйвера протокола. При изменении конфигурации светится светодиод SF.

В этом режиме передача и прием невозможны, а хранящиеся в модуле ET 200S SI передаваемые и принимаемые кадры будут утеряны при последующем запуске драйвера протокола. Обмен данными между модулем ET 200S SI и CPU перезапускается (текущие кадры сообщений отменяются).

По завершении изменения конфигурации модуль ET 200S SI переводится в режим RUN и готов к передаче и приему.

- **RUN:** модуль ET 200S SI выполняет задания на передачу, выставляемые CPU. Кадры, принятые от партнера по обмену данными, предоставляются CPU и считываются последним.

Характеристики процедуры запуска модуля ET 200S SI

Запуск выполняется в два этапа:

- **Инициализация:** при подаче на модуль ET 200S SI напряжения последовательный интерфейс инициализируется и ожидает конфигурационных данных от CPU.
- **Назначение параметров:** на данном этапе модуль ET 200S SI получает параметры, которые были присвоены текущему слоту в STEP 7.

Поведение модуля ET 200S SI при переключении режимов CPU

По завершении запуска модуля ET 200S SI начинает выполняться обмен данными между CPU и ET 200S SI, который осуществляется посредством функциональных блоков.

- **CPU STOP:** в режиме CPU STOP связь по PROFIBUS невозможна. Любые активные сеансы передачи данных между модулем и CPU, включая задания на передачу и прием данных, прерываются, соединение устанавливается заново.

При использовании ASCII драйвера, для которого не было выбрано управление потоком данных, передача данных по интерфейсу RS-232C модуля ET 200S SI продолжается. Другими словами, текущее задание на передачу завершается. Поступающие кадры сообщений принимаются драйвером ASCII вплоть до заполнения буфера приема.

- **Запуск CPU:** при своем запуске CPU передает параметры модулю ET 200S SI.

Сконфигурировав соответствующий параметр должным образом, можно добиться того, чтобы буфер приема модуля ET 200 S SI автоматически очищался при запуске CPU.

- **CPU RUN:** когда CPU находится в режиме RUN, никаких ограничений на передачу и прием не накладывается. В течение первых циклов FB, следующих за перезапуском CPU, выполняется синхронизация работы модуля ET 200S SI и соответствующих функциональных блоков. Только после этого будут выполняться новые функциональные блоки S_SEND или S_RCV.

Что следует учитывать при передаче кадров сообщений.

Кадры сообщений могут быть переданы только в режиме CPU RUN.

Если CPU переходит в режим STOP во время передачи данных от CPU модулю, функциональный блок S_SEND сообщает об ошибке (05) 02_n после перезапуска. Чтобы этого не произошло, функциональный блок S_SEND можно вызвать в программе пользователя со входом RESET из OB запуска.

Примечание

Модуль ET200S SI приступает к передаче данных коммуникационному партнеру только после получения всех данных от CPU.

Что следует помнить при приеме кадров сообщений

В STEP 7 можно указать, должен ли стираться буфер приема при запуске.

- Если выбрана очистка буфера, буфер приема модуля ET 200S SI будет автоматически очищен при переходе CPU из режима STOP в RUN.
- Если очистка не выбрана, то в буфере приема модуля ET 200S SI будет храниться указанное количество кадров сообщений.

Если CPU переходит в режим STOP во время передачи данных от CPU модулю ET 200S SI, функциональный блок сообщает об ошибке (05) 02_n после перезапуска. Чтобы этого не произошло, в программе пользователя можно вызвать функциональный блок S_SEND со входом RESET из OB запуска. Если не выбрана автоматическая очистка буфера приема ET 200S SI при запуске, кадр сообщения передается вновь из модуля на CPU.

2.12 Справочные данные по другим ведущим устройствам (не S7–PROFIBUS)

Обмен данными между DB-ведущим и модулем ET 200S SI

При конфигурировании модуля ET 200S SI выбирается передача данных по 4 или 8 байтов (ввод или вывод данных) с поддержанием консистентности (целостности) по всей длине. Для передачи данных из и в CPU по каналу связи PROFIBUS–DP модуль ET 200S SI использует 4- или 8-байтное адресное пространство ввода/вывода.

CPU может производить запись данных ввода/вывода в любое время, равно как и считывать данные ввода/вывода, следующим образом:

- CPU передает задание модулю ET 200S SI в первом байте адресного пространства вывода модуля.
- Модуль ET 200S SI получает задание в результате передачи кода задания в адресном пространстве ввода.
- Данные из/в CPU поступают в виде сегментов длиной 3 или 7 байтов (количество сегментов зависит от размера адресного пространства ввода/вывода), пока не будут переданы все данные, относящиеся к текущему заданию.

Первый байт сегмента называется байтом координирования и служит для синхронизации передачи каждого сегмента между CPU и модулем ET 200S SI (см. рис. 2-24). В остальных байтах адресного пространства ввода/вывода содержатся данные текущего задания.

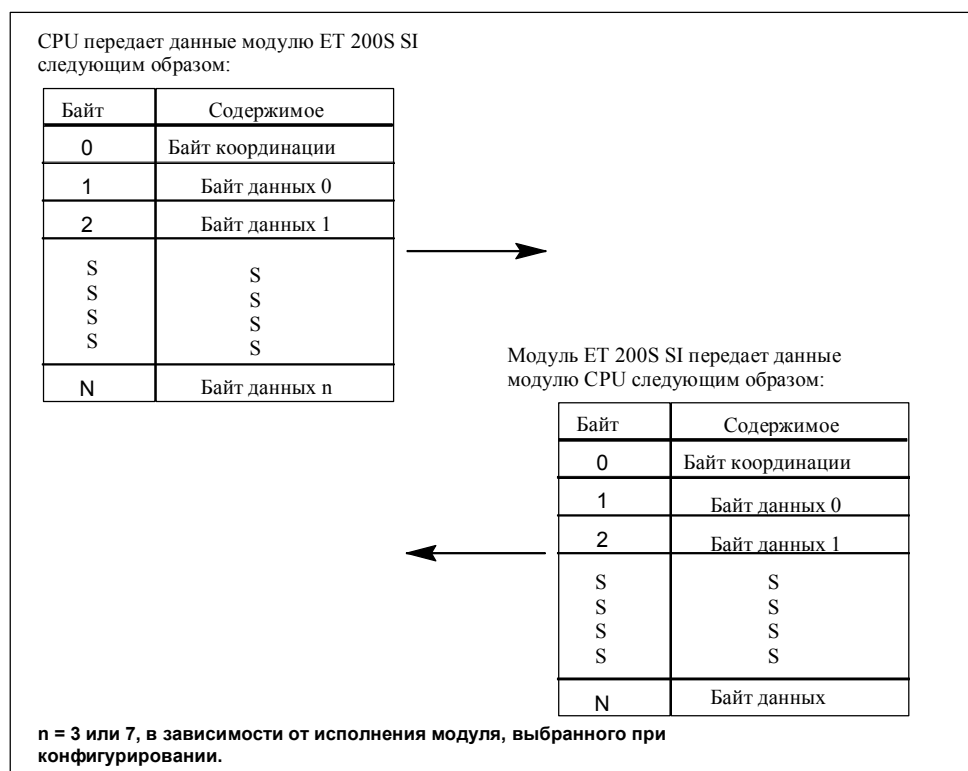


Рисунок 2–24 Обмен данными между CPU и модулем ET200S SI

Описание байта координирования

В таблице 2-20 приводится содержимое байта координирования (байта 0), который служит для синхронизации обмена данными между CPU и модулем последовательного интерфейса ET 200S SI.

Таблица 2–20 Содержимое байта координирования (байта 0) при передаче данных	
Сегмент байта	Описание
Выходной байт, записываемый CPU	<div> <div>Бит</div> <div>7</div> <div>6</div> <div>5</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> <div>0</div> </div> <div> <div>Рез.</div> <div>Код задания</div> <div>Ошибка</div> <div>Номер исполнения</div> </div>
Бит 7	Резерв
Код задания	Устанавливается модулем CPU для инициирования задания
Порядковый номер исполнения	<p>Задание на передачу: увеличивается модулем CPU на 1, когда CPU передает следующий сегмент модулю ET 200S SI...или</p> <p>Задание на прием: принимается каждый раз во входном байте 0 модуля CPU, когда CPU принимает новый сегмент от интерфейсного модуля в правильном порядке. При установленном бите ошибки указывает порядковый номер последнего действительного исполнения. (значение находится в пределах 1...7).</p>
Ошибка	Устанавливается CPU в качестве признака того, что сегмент не был принят в требуемом порядке. В поле порядкового номера исполнения указывается порядковый номер последнего действительного исполнения.
Байт задания, записываемый модулем ET 200S SI	<div> <div>Бит</div> <div>7</div> <div>6</div> <div>5</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> <div>0</div> </div> <div> <div>Рез.</div> <div>Код задания</div> <div>Ошибка</div> <div>Номер исполнения</div> </div>
Бит 7	Резерв
Код задания	Принимается модулем ET 200S SI в качестве подтверждения того, что задание было принято.
Порядковый номер исполнения	<p>Задание на передачу: принимается в каждом выходном байте (байте 0) модуля, когда модуль принимает новый сегмент от CPU в правильном порядке. При установленном бите ошибки указывает порядковый номер последнего действительного исполнения.</p> <p>Задание на прием: увеличивается модулем на 1, когда модуль передает другой сегмент модулю CPU (значение находится в пределах 1...7).</p>
Ошибка	Устанавливается модулем в качестве признака того, что сегмент не был принят в требуемом порядке. В поле порядкового номера исполнения указывается порядковый номер последнего действительного исполнения.

Коды заданий

В таблице 2-21 приводятся задания, определяемые сочетанием битов 4...6 в байте координирования (в байте 0).

Таблица 2–21		Коды заданий
Биты 6, 5, 4	Шестнадц. значение	Определение
0 0 0	0 _н	Состояние ожидания (холостой ход)
0 0 1	1 _н	Передача
0 1 0	2 _н	Прием
0 1 1	3 _н	Чтение состояния сигналов V.24
1 0 0	4 _н	Запись сигналов V.24
1 0 1	5 _н	Параметры передачи: с помощью данного задания можно настроить дополнительные параметры, которые не указаны в файле DDB.
1 1 0	6 _н	Резерв
1 1 1	7 _н	Подтверждение завершения задания

Правила записи кодов заданий

При записи кодов заданий в байт координирования, когда CPU и модуль ET 200S SI могут осуществлять синхронизацию передачи данных, применяются следующие правила:

- Прежде чем программа пользователя в CPU может записывать код задания в выходной байт координирования, она должна получить код состояния ожидания во входном байте координирования модуля ET 200S SI.
- Прежде чем программа пользователя в CPU может осуществлять запись первого сегмента в байты вывода 1...n, она должна получить код подтверждения задания (т.е., код получения задания) во входном байте координирования модуля.
- Если в программу пользователя поступают коды подтверждения задания, отличающиеся от кодов, переданных программой, программа не может производить запись в байты вывода 0...n до тех пор, пока она вновь не получит код состояния ожидания во входном байте координирования модуля ET 200S SI.

Такая ситуация может возникнуть, например, когда в пределах одного и того же цикла выполняются два отдельных задания. Оба задания получили код состояния ожидания и оба записывают различные коды заданий в выходной байт. Поскольку цикл CPU и цикл PROFIBUS-DP не являются синхронными, нельзя предсказать, какое задание достигнет модуля первым. Поэтому каждое задание должно иметь возможность ожидания завершения другого задания, прежде чем будет исполняться само.

Слово состояния

Далее приводятся примеры передачи данных, в которых модуль ET 200S SI сообщает CPU о своем состоянии, используя для этого байты 1 и 2. В таблице 2-28 (стр. 2-93) приводится перечень значений слова состояния.

Порядок байтов в слове

Если для обмена данными между CPU и модулем ET 200S SI используются слова длиной 16 битов (например, состояние и длина), первым передается байт, содержащий большее значение.

Пример данных, передаваемых от CPU модулю

В таблице 2-22 приводится пример передачи модулем CPU сообщения, состоящего из первых 22-х букв алфавита. Используется адресное пространство ввода/вывода размером 8 байт. Цикл DP, приблизительно, совпадает с циклом CPU, поэтому при возврате модулем порядкового номера исполнения наблюдается запаздывание длительностью в один цикл.

Таблица 2–22		Пример передачи данных																							
Цикл CPU	CPU записывает в ET 200S SI				CPU считывает из ET 200S SI																				
1-ый	Программа пользователя получает от модуля следующий код состояния ожидания:	<div>Байт 0 1 2 3 4 5 6 7</div> <div>←<table><tr><td>00_H</td><td>nnnn_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table></div>								00 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Состояние	Не имеет значения					
		00 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																
Квит. зад.	Состояние	Не имеет значения																							
		<div>(Байт 0 1 2 3 4 5 6 7)</div> <div><table><tr><td>10_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td></tr></table></div> <div>→</div>								10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Задание	Не имеет значения						
10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																		
Задание	Не имеет значения																								
2-ой	Программа пользователя по-прежнему "видит" код состояния ожидания модуля:	<div>Байт 0 1 2 3 4 5 6 7</div> <div>←<table><tr><td>00_H</td><td>nnnn_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table></div>								00 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Состояние	Не имеет значения					
		00 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																
Квит. зад.	Состояние	Не имеет значения																							
		<div>CPU повторяет задание на передачу :</div> <div>Байт 0 1 2 3 4 5 6 7</div> <div><table><tr><td>10_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td></tr></table></div> <div>→</div>								10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Задание	Не имеет значения						
10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																		
Задание	Не имеет значения																								
3-ий	Программа пользователя получает следующий ответ от модуля:	<div>Байт 0 1 2 3 4 5 6 7</div> <div>←<table><tr><td>10_H</td><td>nnnn_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table></div>								10 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Состояние	Не имеет значения					
		10 _H	nnnn _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																
Квит. зад.	Состояние	Не имеет значения																							

11 _H	0016 _H	'a'	'b'	'c'	'd'	'e'
Задание	Длина	Данные				

→

	CPU записывает задание на передачу :
--	---

Таблица 2–22 Пример передачи данных (продолжение)

Таблица 2.22 Пример передачи данных (продолжение)																		
4-ый	Программа пользователя читает следующий ответ модуля (порядковые номера передачи и исполнения по-прежнему нулевые, что указывает на запаздывание ответа от модуля):	<table><tr><td>10_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="7">Не имеет значения</td></tr></table>	10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Не имеет значения						
10 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H											
Квит. зад.	Не имеет значения																	
	CPU передает второй сегмент, поскольку ошибки отсутствуют:																	
	<table><tr><td>12_H</td><td>'f'</td><td>'g'</td><td>'h'</td><td>'i'</td><td>'j'</td><td>'k'</td><td>'l'</td></tr><tr><td>Задание</td><td colspan="7">Данные</td></tr></table>	12 _H	'f'	'g'	'h'	'i'	'j'	'k'	'l'	Задание	Данные							→
12 _H	'f'	'g'	'h'	'i'	'j'	'k'	'l'											
Задание	Данные																	
5-ый	Программа пользователя читает следующий ответ модуля:	<table><tr><td>11_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="7">Не имеет значения</td></tr></table>	11 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Не имеет значения						
11 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H											
Квит. зад.	Не имеет значения																	
	CPU передает третий сегмент, поскольку ошибки отсутствуют:																	
	<table><tr><td>13_H</td><td>'m'</td><td>'n'</td><td>'o'</td><td>'p'</td><td>'q'</td><td>'r'</td><td>'s'</td></tr><tr><td>Задание</td><td colspan="7">Данные</td></tr></table>	13 _H	'm'	'n'	'o'	'p'	'q'	'r'	's'	Задание	Данные							→
13 _H	'm'	'n'	'o'	'p'	'q'	'r'	's'											
Задание	Данные																	
6-ой	Программа пользователя читает следующий ответ модуля:	<table><tr><td>12_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="7">Не имеет значения</td></tr></table>	12 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Не имеет значения						
12 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H											
Квит. зад.	Не имеет значения																	
	CPU передает четвертый сегмент, поскольку ошибки отсутствуют:																	
	<table><tr><td>14_H</td><td>'t'</td><td>'u'</td><td>'v'</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="3">Данные</td><td colspan="4">Не имеет значения</td></tr></table>	14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H	Задание	Данные			Не имеет значения				→
14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H											
Задание	Данные			Не имеет значения														
7-ой	Программа пользователя читает следующий ответ модуля:	<table><tr><td>13_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="7">Не имеет значения</td></tr></table>	13 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Не имеет значения						
13 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H											
Квит. зад.	Не имеет значения																	
	CPU передает пятый сегмент, поскольку ошибки отсутствуют:																	
	<table><tr><td>14_H</td><td>'t'</td><td>'u'</td><td>'v'</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="3">Данные</td><td colspan="4">Не имеет значения</td></tr></table>	14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H	Задание	Данные			Не имеет значения				→
14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H											
Задание	Данные			Не имеет значения														

Таблица 2–22 Пример передачи данных (продолжение)

8-ой	Программа пользователь читает следующий ответ модуля:	<table><tr><td>14_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="7">Не имеет значения</td></tr></table>	14 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Не имеет значения																	
14 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																						
Квит. зад.	Не имеет значения																												
	CPU не передает новой информации (передает прежние значения) и ожидает последнего подтверждения от модуля; это указывает на то, что сообщение было передано партнеру по обмену данными.	<table><tr><td>14_H</td><td>'t'</td><td>'u'</td><td>'v'</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="2">Данные</td><td colspan="5">Не имеет значения</td></tr></table>	14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H	Задание	Данные		Не имеет значения															
14 _H	't'	'u'	'v'	xx _H	xx _H	xx _H	xx _H																						
Задание	Данные		Не имеет значения																										
п-ый	Через несколько циклов CPU программа пользователя получает следующий ответ от модуля:	<table><tr><td>Байт</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td></td><td>74_H</td><td colspan="2">nnnn_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="2">Состояние</td><td colspan="6">Не имеет значения</td></tr></table>	Байт	0	1	2	3	4	5	6	7		74 _H	nnnn _H		xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Состояние		Не имеет значения					
Байт	0	1	2	3	4	5	6	7																					
	74 _H	nnnn _H		xx _H	xx _H	xx _H	xx _H	xx _H																					
Квит. зад.	Состояние		Не имеет значения																										
—	CPU записывает код режима ожидания в задание и прекращает задание																												

Пример приема данных от модуля в CPU

В таблице 2-23 приводится пример приема сообщения от модуля последовательного интерфейса модулем CPU. Используется адресное пространство ввода/вывода размером 8 байт. Цикл DP короче цикла CPU, поэтому задержка (ожидание) в модуле отсутствует.

Таблица 2–23			Пример приема данных																											
Цикл CPU	CPU записывает в ET 200S SI	CPU считывает из ET 200S SI																												
п	Программа пользователя в течение нескольких циклов получает от модуля код состояния ожидания, пока не будет получено состояние, соответствующее наличию принятого сообщения:	<div>←</div> <table><tr><td>Байт</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>00_H</td><td colspan="2">nnnn_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Квит. зад.</td><td colspan="2">Состояние</td><td colspan="6">Не имеет значения</td></tr></table> <div>0000_H = принятых сообщений нет 0001_H = имеется принятое сообщение 0B01_H = буфер приема переполнен на 2/3</div>	Байт	0	1	2	3	4	5	6	7	00 _H	nnnn _H		xx _H	xx _H	xx _H	xx _H	xx _H	Квит. зад.	Состояние		Не имеет значения							
			Байт	0	1	2	3	4	5	6	7																			
			00 _H	nnnn _H		xx _H	xx _H	xx _H	xx _H	xx _H																				
Квит. зад.	Состояние		Не имеет значения																											
CPU записывает задание на прием :																														
	<div>Байт</div> <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>20_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td><td>xx_H</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td></tr></table> <div>→</div>	0	1	2	3	4	5	6	7	20 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	Задание	Не имеет значения											
0	1	2	3	4	5	6	7																							
20 _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H	xx _H																							
Задание	Не имеет значения																													

Таблица 2–23 Пример приема данных (продолжение)

Таблица 2.20 Пример приема данных (продолжение)																																														
Следующий цикл (n + 1)	<div>Программа пользователя получает от модуля следующий ответ (модуль подтверждает прием, возвращает в ответ первый сегмент и увеличивает на 1 порядковый номер исполнения):</div> <div><table><tr><td>21H</td><td>0006H</td><td>'a'</td><td>'b'</td><td>'c'</td><td>'d'</td><td>'e'</td><td>→</td></tr><tr><td>Квит. зад.</td><td>Длина</td><td colspan="5">Данные</td><td></td></tr></table></div>	21H	0006H	'a'	'b'	'c'	'd'	'e'	→	Квит. зад.	Длина	Данные																																		
21H	0006H	'a'	'b'	'c'	'd'	'e'	→																																							
Квит. зад.	Длина	Данные																																												
	<div>CPU записывает задание, чтобы подтвердить первый сегмент:</div> <div><table><tr><td>Байт 0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td></td></tr><tr><td>21H</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>→</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td><td></td></tr></table></div>	Байт 0	1	2	3	4	5	6	7		21H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	→	Задание	Не имеет значения																									
Байт 0	1	2	3	4	5	6	7																																							
21H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	→																																						
Задание	Не имеет значения																																													
Следующий цикл (n + 2)	<div>Программа пользователя читает 2-ой сегмент от модуля</div> <div><table><tr><td>22H</td><td>'f'</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td></td></tr><tr><td>Квит. зад.</td><td>Данные</td><td colspan="6">Не имеет значения</td><td></td></tr></table></div> <div><table><tr><td>Байт 0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td></td></tr><tr><td>21H</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>→</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td><td></td></tr></table></div>	22H	'f'	xxH	xxH	xxH	xxH	xxH	xxH		Квит. зад.	Данные	Не имеет значения							Байт 0	1	2	3	4	5	6	7		21H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	→	Задание	Не имеет значения							
22H	'f'	xxH	xxH	xxH	xxH	xxH	xxH																																							
Квит. зад.	Данные	Не имеет значения																																												
Байт 0	1	2	3	4	5	6	7																																							
21H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	→																																						
Задание	Не имеет значения																																													
Следующий цикл (n + 3)	<div>Модуль возвращается в состояние ожидания по завершении первой операции приема</div> <div><table><tr><td>Байт 0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td></td></tr><tr><td>00H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>←</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td><td></td></tr></table></div>	Байт 0	1	2	3	4	5	6	7		00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	←	Квит. зад.	Состояние	Не имеет значения																								
Байт 0	1	2	3	4	5	6	7																																							
00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	←																																						
Квит. зад.	Состояние	Не имеет значения																																												
—	CPU прекращает задание.																																													

Пример чтения состояния сигнала V.24

В таблице 2-24 приводится пример чтения модулем CPU состояния сигналов V.24 модуля последовательного интерфейса. Используется адресное пространство ввода/вывода размером 8 байт.

Таблица 2–24		Пример чтения состояния сигналов V.24																							
Цикл CPU	CPU записывает в ET 200S SI	CPU считывает из ET 200S SI																							
1-ый	Программа пользователя получает от модуля следующий код состояния ожидания:	Байт 0 1 2 3 4 5 6 7																							
		<table><tr><td>00H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table>								00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																
Квит. зад.	Состояние	Не имеет значения																							
CPU записывает задание на чтение состояния сигналов V.24: Байт 0 1 2 3 4 5 6 7																									
2-ой	Программа пользователя получает от модуля следующий ответ:	<table><tr><td>30H</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Задание</td><td colspan="7">Не имеет значения</td></tr></table>								30H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	Задание	Не имеет значения						
		30H	xxH	xxH	xxH	xxH	xxH	xxH	xxH																
		Задание	Не имеет значения																						
CPU записывает подтверждение и принимает порядковый номер исполнения: Байт 0 1 2 3 4 5 6 7																									
3-ий	Модуль возвращается в состояние ожидания по завершении первой операции:	Байт 0 1 2 3 4 5 6 7																							
		<table><tr><td>00H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table>								00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																
Квит. зад.	Состояние	Не имеет значения																							
CPU прерывает задание.																									

Пример изменения состояния сигналов V.24

В таблице 2-25 приводится пример изменения модулем CPU состояния сигналов V.24 модуля последовательного интерфейса. Используется адресное пространство ввода/вывода размером 8 байт.

Таблица 2–25		Пример изменения состояния сигналов V.24																							
Цикл CPU	CPU записывает в ET 200S SI	CPU считывает из ET 200S SI																							
1-ый	Программа пользователя получает от модуля код состояния ожидания:	Байт 0 1 2 3 4 5 6 7																							
		<table><tr><td>00H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table>								00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																
Квит. зад.	Состояние	Не имеет значения																							
CPU записывает задание на изменение состояния сигналов V.24:																									
2-ой	Программа пользователя получает от модуля следующий ответ:	Байт 0 1 2 3 4 5 6 7																							
		<table><tr><td>40H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table>								40H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		40H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																
Квит. зад.	Состояние	Не имеет значения																							
CPU записывает режим ожидания в выходной байт:																									
3-ий	Программа пользователя получает от модуля следующий ответ (модуль возвращается в состояние ожидания по завершении операции):	Байт 0 1 2 3 4 5 6 7																							
		<table><tr><td>00H</td><td>nnnnH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td><td>xxH</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td colspan="6">Не имеет значения</td></tr></table>								00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		00H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																
Квит. зад.	Состояние	Не имеет значения																							
CPU записывает код режима ожидания в задание и прекращает задание.																									

Параметры управления потоком данных

Код задания на передачу параметров (1 0 1 в табл. 2-21) при использовании драйвера ASCII позволяет настраивать дополнительные параметры. Параметры зависят от типа управления потоком данных, выбранного в файле DDB. В таблице 2-26 приводится описание трех типов управления потоком данных.

Таблица 2–26 Параметры управления потоком данных			
Диапазон установки параметров для управления потоком данных XON/XOFF			
Байт	Описание	Диапазон настройки	Значение по умолчанию
1	Номер блока параметров	20 _H	
2 и 3	Длина	0004 _H	0004 _H
4	Символ XON	0 ... 127 (7 битов данных) 0 ... 255 (8 битов данных)	11 (DC1)
5	Символ XOFF	0 ... 127 (7 битов данных) 0 ... 255 (8 битов данных)	13 (DC3)
6 и 7	Время ожидания перехода из XON в XOFF	20 ... 655350 с шагом 10 мс	200 (2000 мс)
Диапазон установки параметров для управления потоком данных RTS/CTS			
Байт	Описание	Диапазон настройки	Значение по умолчанию
1	Номер блока параметров	21 _H	
2 и 3	Длина	0002 _H	0002 _H
4 и 5	Время ожидания включения CTS	20 ... 655350 с шагом 10 мс	200 (2000 мс)
Диапазон установки параметров для автоматического функционирования дополнительных сигналов RS–232C			
Байт	Описание	Диапазон настройки	Значение по умолчанию
1	Номер блока параметров	22 _H	
2 и 3	Длина	0004 _H	0004 _H
4 и 5	Время ожидания выключения RTS после передачи	0 ... 655350 с шагом 10 мс	1 (10 мс)
6 и 7	Время ожидания включения CTS после включения RTS	0 ... 655350 с шагом 10 мс	1 (10 мс)

Пример последовательности действий для XON/XOFF

В таблице 2-27 приводится пример установки модулем CPU параметров XON/XOFF. Используется адресуемое пространство ввода/вывода размером 4 байта.

Таблица 2–27		Пример последовательности действий для XON/XOFF													
Цикл CPU	CPU записывает в ET 200S SI	CPU считывает из ET 200S SI													
1-ый	Программа пользователя получает от модуля следующий код режима ожидания: →	<table><tr><td>00 н</td><td>nnnn н</td><td>xx н</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td>Не имеет значения</td></tr></table>		00 н	nnnn н	xx н	Квит. зад.	Состояние	Не имеет значения						
	00 н	nnnn н	xx н												
Квит. зад.	Состояние	Не имеет значения													
	<table><tr><td>Байт 0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>50 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Задание</td><td colspan="3">Не имеет значения</td></tr></table>	Байт 0	1	2	3	50 н	xx н	xx н	xx н	Задание	Не имеет значения			←Задание: передается код параметра (1 0 1 или 5н) + порядковый номер исполнения 0	
Байт 0	1	2	3												
50 н	xx н	xx н	xx н												
Задание	Не имеет значения														
2-ой	Программа пользователя получает следующий ответ от модуля: →	<table><tr><td>50 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Квит. задания</td><td colspan="3">Не имеет значения</td></tr></table>		50 н	xx н	xx н	xx н	Квит. задания	Не имеет значения						
	50 н	xx н	xx н	xx н											
Квит. задания	Не имеет значения														
	CPU передает 1-ый сегмент, поскольку задание было принято.														
	<table><tr><td>51 н</td><td>20 н</td><td colspan="2">0004 н</td></tr><tr><td>Задание</td><td>Поток данных</td><td colspan="2">Длина передаваемых данных</td></tr></table>	51 н	20 н	0004 н		Задание	Поток данных	Длина передаваемых данных		←Задание: продолжается передача параметров + инкремент порядкового номера исполнения ←Поток данных: код для параметров потока данных					
51 н	20 н	0004 н													
Задание	Поток данных	Длина передаваемых данных													
3-ий	Программа пользователя получает следующий ответ от модуля: →	<table><tr><td>51 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Квит. задания</td><td colspan="3">Не имеет значения</td></tr></table>		51 н	xx н	xx н	xx н	Квит. задания	Не имеет значения						
	51 н	xx н	xx н	xx н											
Квит. задания	Не имеет значения														
	CPU передает 2-ой сегмент, поскольку признаки ошибки отсутствуют:														
	<table><tr><td>52 н</td><td>0B н</td><td>0D н</td><td>00 н</td></tr><tr><td>Задание</td><td>DC1</td><td>DC3</td><td>Время ожидания перехода XON в XOFF (MSB)</td></tr></table>	52 н	0B н	0D н	00 н	Задание	DC1	DC3	Время ожидания перехода XON в XOFF (MSB)						
52 н	0B н	0D н	00 н												
Задание	DC1	DC3	Время ожидания перехода XON в XOFF (MSB)												
4-ый	Программа пользователя получает следующий ответ от модуля: →	<table><tr><td>52 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Квит. задания</td><td colspan="3">Не имеет значения</td></tr></table>		52 н	xx н	xx н	xx н	Квит. задания	Не имеет значения						
	52 н	xx н	xx н	xx н											
Квит. задания	Не имеет значения														
	CPU передает 3-ий сегмент, поскольку признаки ошибки отсутствуют:														
	<table><tr><td>53 н</td><td>C8 н</td><td>xx н</td><td>xx н</td></tr><tr><td>Задание</td><td>Время ожидания перехода XON в XOFF (LSB)</td><td colspan="2">Не имеет значения</td></tr></table>	53 н	C8 н	xx н	xx н	Задание	Время ожидания перехода XON в XOFF (LSB)	Не имеет значения							
53 н	C8 н	xx н	xx н												
Задание	Время ожидания перехода XON в XOFF (LSB)	Не имеет значения													

Таблица 2–27 Пример последовательности действий для XON/XOFF (продолжение)

Таблица 1.2. Пример последовательности действий для XON/XOFF (продолжение)												
5-ый	Программа пользователя получает следующий ответ от модуля: →			<table><tr><td>53 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Квит. задания</td><td colspan="3">Не имеет значения</td></tr></table>	53 н	xx н	xx н	xx н	Квит. задания	Не имеет значения		
	53 н	xx н	xx н	xx н								
Квит. задания	Не имеет значения											
CPU повторяет 3-ий сегмент и ожидает подтверждения задания.												
	<table><tr><td>53 н</td><td>C8 н</td><td>xx н</td><td>xx н</td></tr><tr><td>Задание</td><td>Время ожидания перехода XON в XOFF (LSB)</td><td colspan="2">Не имеет значения</td></tr></table>	53 н	C8 н	xx н	xx н	Задание	Время ожидания перехода XON в XOFF (LSB)	Не имеет значения				
	53 н	C8 н	xx н	xx н								
Задание	Время ожидания перехода XON в XOFF (LSB)	Не имеет значения										
6-ой	Программа пользователя получает следующий ответ от модуля: →			<table><tr><td>73 н</td><td>nnnn н</td><td>xx н</td></tr><tr><td>Квит. зад.</td><td>Состояние</td><td>Не имеет значения</td></tr></table>	73 н	nnnn н	xx н	Квит. зад.	Состояние	Не имеет значения		
	73 н	nnnn н	xx н									
Квит. зад.	Состояние	Не имеет значения										
CPU записывает код режима ожидания в задание и прекращает задание.												
	<table><tr><td>00 н</td><td>xx н</td><td>xx н</td><td>xx н</td></tr><tr><td>Задание</td><td colspan="3">Не имеет значения</td></tr></table>	00 н	xx н	xx н	xx н	Задание	Не имеет значения					
	00 н	xx н	xx н	xx н								
Задание	Не имеет значения											

Условия возникновения ошибки

Модуль последовательного интерфейса возвращает сообщение об ошибке в следующих ситуациях:

- Если длина задания на передачу превышает 200 байт, модуль возвращает в ответ подтверждение завершения задания, а в слове состояния содержится код ошибки. После этого CPU записывает код режима ожидания задания и прерывает задание.
- Если модулю было отправлено задание на прием, а в принятом сообщении содержится ошибка, модуль принимает код задания на прием с порядковым номером исполнения 0, а в слове состояния содержится код ошибки. После этого CPU записывает код режима ожидания в задание и прерывает задание.
- Если модулю было передано задание на прием, а принятые сообщения отсутствуют, модуль принимает код задания на прием с порядковым номером исполнения 0, а слово состояния содержит значение 0101_н. Это не является ошибкой, но не позволяет модулю отключиться в режиме задания на прием, заставляя его ожидать принятое сообщение, что позволяет выполнять задания на передачу. CPU записывает код режима ожидания в задание и прекращает задание.

Исключения

Ранее упоминалось, что ни одна операция (например, задание на передачу) не может быть инициирована из программы пользователя, пока модуль не будет находиться в состоянии ожидания. После того, как задание передано, необходимо дождаться, пока модуль примет код задания, и только после этого требуемая операция может быть выполнена. Если в процессе выполнения операций происходит фрагментирование данных, могут произойти следующие ошибки (исключения):

Примечание

В приведенных ниже примерах операций передачи или назначения параметров под передающим устройством понимается CPU, а под приемным устройством – модуль последовательного интерфейса. В случае операции приема под передающим устройством понимается модуль последовательного интерфейса, а под принимающим устройством – CPU.

- **Ошибка:** при передаче фрагментированных данных передающее устройство контролирует бит ошибки принимающего устройства. Если бит ошибки установлен, передающее устройство передает фрагменты повторно, начиная с фрагмента, следующего за фрагментом, номер которого указан принимающим устройством.
- **Нарушение порядка номеров исполнения:** если в режиме фрагментированной передачи на принимающее устройство поступает фрагмент, порядковый номер исполнения которого не равен предыдущему номеру исполнения + 1, принимающее устройство должно сообщить об ошибке и указать в ответе последний принятый порядковый номер исполнения.
- **Измененный код задания:**
 - Если на принимающее устройство в режиме фрагментированной передачи поступает код задания, отличающийся от кода, соответствующего началу фрагментированной передачи, и если он не равен 000 или 111, принимающее устройство не воспринимает этот отличающийся код и игнорирует связанные с ним данные.
 - Если на принимающее устройство в режиме фрагментированной передачи поступает фрагмент с кодом задания состояния ожидания, операция прерывается и выполняется переход в режим ожидания, при этом бит ошибки не устанавливается.
 - Если в режиме фрагментированной передачи на принимающее устройство поступает код задания, соответствующий подтверждению завершения задания, операция прерывается и выполняется переход в состояние ожидания, при этом бит ошибки не взводится.
 - Если в процессе фрагментированной передачи передающее устройство принимает ответ, содержащий другой код задания, сообщение должно быть отменено. После этого вновь устанавливается код режима ожидания, модуль должен перейти в режим ожидания, а операция должна быть выполнена повторно.

2.13 Диагностика

Обзор

Функция диагностики модуля ET 200S SI служит для быстрого обнаружения ошибок, в том числе и в рабочем режиме. Предлагаются следующие способы диагностики:

- Светодиоды состояния на передней панели модуля ET 200S SI
- Выход STATUS функциональных блоков
- Использование диагностики ведомых устройств PROFIBUS

Диагностика по светодиодам состояния

На передней панели интерфейсного модуля ET 200S располагаются следующие светодиоды состояний:

- **TX** (зеленый) Светится, когда через интерфейс модуля передаются данные
- **RX** (зеленый) Светится, когда через интерфейс модуля принимаются данные
- **SF** (красный) Указывает на одну из возможных ошибок:
 - Аппаратная неисправность
 - Ошибка встроенной микропрограммы
 - Ошибка назначения параметров
 - Обрыв провода или отсутствие контакта в кабеле между модулем и партнером по обмену данными
Только для соединений по интерфейсу RS-422 в случае исходного состояния линии приема R(A) 5V/R(B) 0V.
 - Ошибка связи (четность, ошибка кадра, переполнение буфера).

Структура диагностических сообщений функциональных блоков

В каждом функциональном блоке имеется параметр STATUS, служащий для диагностики ошибок. Независимо от используемого функционального блока, за сообщениями в слове STATUS закреплены одни и те же номера. Структура параметра STATUS показана на рисунке 2-25.



Рисунок 2–25 Структура параметра STATUS

Пример: на рисунке 2-26 показано содержимое параметра STATUS в случае, когда задание было прервано в результате “холодного” перезапуска, “горячего” перезапуска или простого перезапуска (класс события 1Е_н, номер события 0D_н).

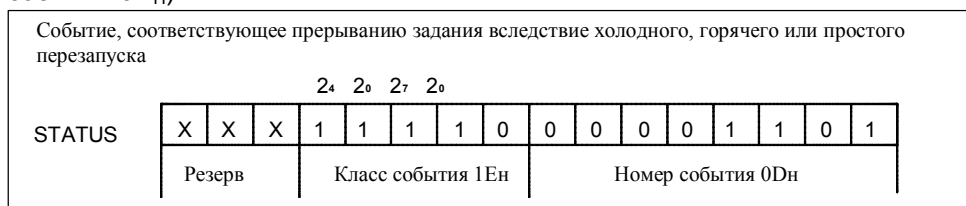


Рисунок 2–26 Пример: параметр STATUS для класса события 1Е_н, событие 0D_н

Диагностические сообщения функциональных блоков

В таблице 2-28 описаны классы событий, а также приведены номера событий и рекомендуемые меры по устранению каждого из условий возникновения ошибки.

Таблица 2–28 Диагностические сообщения в параметре STATUS		
Класс события 5 (05 _н): "Ошибка при выполнении задания CPU"		
Номер события	Событие	Меры по устранению
(05) 02 _н	Задание не может быть выполнено в данном режиме модуля ET 200S SI (например, не выполнено параметрирование интерфейса устройства).	Обработайте диагностическое прерывание и устраните ошибку соответствующим образом.
(05) 0E _н	Недопустимая длина кадра сообщения.	Длина кадра сообщения > 200 байтов. Модулем ET 200S SI будет приниматься оставшаяся часть кадра сообщения (> 200 байтов), а первый его фрагмент, таким образом, будет утерян. Выберите меньшую длину кадра сообщения.
(05) 50 _н	Запрос на обновление параметров не действителен для текущего управления потоком данных модуля ET 200S SI.	Либо измените параметры функционального блока (FB6 S_XON, FB7 S_RTS, FB8 S_V24), либо измените управление потоком данных модуля ET 200S SI в конфигурации аппаратных средств, чтобы эти параметры соответствовали друг другу.

Таблица 2–28 Диагностические сообщения в параметре STATUS (продолжение)

(05) 51 _H	В процессе связи между модулем ET 200S SI и программируемым логическим контроллером произошла ошибка выполнения кадра. Ошибка произошла, когда принятый кадр сообщения модуля ET 200S SI передавался программируемому логическому контроллеру.	Модуль и программируемый логический контроллер прервали передачу. Повторите задание на прием; модуль ET 200S SI передаст принятое сообщение вновь.
Класс события 7 (07_H): "Ошибка передачи"		
(07) 02 _H	Только для 3964(R): Ошибка при установлении соединения: После передачи STX был принят код NAK или любой другой код (за исключением DLE или STX).	Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(07) 03 _H	Только для 3964(R): Превышено время ожидания подтверждения: После передачи STX не поступил ответ от партнера по обмену данными в пределах времени ожидания подтверждения.	Коммуникационный партнер реагирует слишком медленно или в линии передачи имеется обрыв. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(07) 04 _H	Только для 3964(R): Прерывание партнером: В процессе текущей процедуры передачи партнером был принят один или несколько символов.	Проверьте, сигнализируется ли ошибка также и партнером, например, из-за того, что не все передаваемые данные поступили (напр., из-за обрыва в линии передачи), или из-за серьезных неисправностей, например, из-за выхода из строя коммуникационного партнера. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(07) 05 _H	Только для 3964(R): Отрицательное подтверждение во время передачи.	Проверьте, сигнализируется ли ошибка также и партнером, например, из-за того, что не все передаваемые данные поступили (напр., из-за обрыва в линии передачи), или из-за серьезных неисправностей, например, из-за выхода из строя коммуникационного партнера. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(07) 06 _H	Только для 3964(R): Ошибка по завершению соединения: <ul style="list-style-type: none"> Партнер отменил кадр сообщения в конце соединения с помощью символа NAK или случайной строки (кроме DLE), или Код подтверждения (DLE) был принят слишком рано. 	Проверьте, сигнализируется ли ошибка также и партнером, например, из-за того, что не все передаваемые данные поступили (напр., из-за обрыва в линии передачи), или из-за серьезных неисправностей, например, из-за выхода из строя коммуникационного партнера. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.

Таблица 2–28 Диагностические сообщения в параметре STATUS (продолжение)

(07) 07 _H	Только для 3964(R): В конце соединения превышено время ожидания подтверждения, либо после передачи кадра сообщения превышено время ожидания ответа: После сброса соединения с помощью символа DLE ETX от партнера по обмену данными не поступил ответ в течение времени ожидания подтверждения.	Коммуникационный партнер реагирует слишком медленно или неисправен. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(07) 08 _H	Только для ASCII драйвера: Превышен интервал ожидания перехода XON или CTS в состояние ВКЛ.	Партнер по связи неисправен, функционирует слишком медленно или разорвал связи (ушел в off-line). Проверьте партнерское устройство или, если это необходимо, измените его параметры.
(07) 0B _H	Только для 3964(R): Неразрешимый конфликт инициализации из-за того, что оба партнера имеют наивысший приоритет.	Измените параметры.
(07) 0C _H	Только для 3964(R): Неразрешимый конфликт инициализации из-за того, что оба партнера имеют наимизший приоритет.	Измените параметры.
Класс события 8 (08_H): "Ошибка приема"		
(08) 02 _H	Только для 3964(R): Ошибка при установлении соединения: <ul style="list-style-type: none"> В режиме ожидания был принят один или несколько случайных кодов (не NAK или STX), или После того, как был принят STX, партнер по обмену данными передал другие коды, не ожидая в ответ символ DLE. После подачи питания на партнерское устройство: <ul style="list-style-type: none"> В процессе активизации партнера на модуль поступил недопустимый код. 	Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(08) 05 _H	Только для 3964(R): Логическая ошибка при приеме: После того, как был принят символ DLE, поступил другой случайный код (не DLE или ETX).	Проверьте, всегда ли коммуникационный партнер дублирует символ DLE в заголовке кадра сообщения и в строке данных, а также, сбрасывается ли соединение с помощью DLE ETX. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(08) 06 _H	Превышена длительность ожидания символа: <ul style="list-style-type: none"> После приема символа следующий символ не был принят в пределах времени ожидания символа, или Только для 3964(R): <ul style="list-style-type: none"> В процессе установления соединения после передачи DLE не был принят 1-ый символ в пределах времени ожидания символа. 	Коммуникационный партнер реагирует слишком медленно или неисправен. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(08) 07 _H	Только для 3964(R): Недопустимая длина кадра сообщения: Принят кадр сообщения нулевой длины.	Прием кадра сообщения нулевой длины не является ошибкой. Проверьте, почему коммуникационный партнер передает кадры, не содержащие данных пользователя.

Таблица 2–28 Диагностические сообщения в параметре STATUS (продолжение)

(08) 08 _H	Только для 3964(R): Ошибка символа контроля блока (BCC) Расчетное значение BCC не совпадает с BCC, принятым партнером по завершению соединения.	Проверьте, не подвергается ли соединение каким-либо возмущениям. Если это подтверждается, могут наблюдаться случайные коды ошибок. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(08) 09 _H	Только для 3964(R): Следует немедленно задать количество попыток повтора.	Установите время ожидания блока для коммуникационного партнера таким же, как и у модуля. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
(08) 0A _H	Отсутствует свободный буфер приема: Отсутствует пустой буфер приема для приема данных.	Блок S_RCV следует вызывать чаще.
Класс события 8 (08_H): "Ошибка приема"		
(08) 0C _H	Ошибка передачи: <ul style="list-style-type: none"> Обнаружена ошибка передачи (ошибка четности, ошибка стоп-бита, ошибка переполнения). Только для 3964(R): <ul style="list-style-type: none"> Если ошибка происходит при передаче или приеме, инициируются повторы. Если в режиме ожидания принят сбойный символ, об ошибке сообщается немедленно, чтобы неисправности в линии передачи могли быть обнаружены как можно раньше. Если включен светодиод SF (красный), в соединительном кабеле между двумя коммуникационными партнерами имеется обрыв. 	Возмущения в линии передачи приводят к повтору кадров сообщений, снижая тем самым пропускную способность канала. При этом возрастает опасность пропуска ошибок. Неисправность такого рода можно устранить, если изменить настройки системы или саму линию связи. Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов.
(08) 0D _H	ОБРЫВ: обрыв в линии приема, подключенной к партнеру.	Переустановите соединение или включите партнерское устройство.
(08) 10 _H	Только для ASCII драйвера: Ошибка четности <ul style="list-style-type: none"> Если включен светодиод SF (красный), в соединительном кабеле между двумя коммуникационными партнерами произошел обрыв. 	Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов. Устраните ошибку, изменив системные настройки или саму линию.
(08) 11 _H	Только для ASCII драйвера: Ошибка кадра символа <ul style="list-style-type: none"> Если включен светодиод SF (красный), в соединительном кабеле между двумя коммуникационными партнерами произошел обрыв. 	Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов. Устраните ошибку, изменив системные настройки или саму линию.
(08) 12 _H	Только для ASCII драйвера: После передачи модулем символов XOFF или сброса CTS были приняты другие символы.	Настройте параметры коммуникационного партнера или производите чтение данных быстрее.

Таблица 2–28 Диагностические сообщения в параметре STATUS (продолжение)

(08) 18 _H	Только для ASCII драйвера: DSR = ВЫКЛ или CTS = ВЫКЛ	Сигналы DSR или CTS были сброшены партнером до или в процессе операции передачи. Проверьте управление вспомогательными сигналами RS–232C у коммуникационного партнера.
Класс события 30 (1E_H): "Ошибки связи между модулем и CPU"		
(1E) 0D _H	"Задание прервано из-за холодного перезапуска, горячего перезапуска или сброса"	
(1E) 0E _H	Статическая ошибка при вызове DP_RDDAT SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 0F _H	Статическая ошибка при вызове DP_WRDAT SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 10 _H	Статическая ошибка при вызове RD_LGADR SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 20 _H	Параметр превышает допустимый диапазон.	Измените вход функционального блока таким образом, чтобы он принадлежал допустимому диапазону.
(1E) 41 _H	В параметре LEN функциональных блоков задано недопустимое количество байтов.	Укажите длину в диапазоне 1 ... 200 байт.
Класс события 11 (0B_H): "Предупреждение"		
(0B) 01 _H	Буфер приема переполнен более чем на 2/3.	

Вызов переменной SFCERR

Дополнительные сведения об ошибках 14 (1E 0E_H) и 15 (1E 0F_H) для класса события 30 можно получить из переменной SFCERR.

Переменную SFCERR можно загрузить из экземплярного DB, принадлежащего соответствующему функциональному блоку.

Сообщения об ошибках, содержащиеся в переменной SFCERR, описаны в разделе, посвященном системным функциям "DPRD_DAT" и SFC15 "DPWR_DAT" в руководстве *System Software for S7 300/400, System and Standard Functions*.

Диагностика ведомых устройств PROFIBUS

Диагностика ведомых устройств соответствует стандарту PROFIBUS EN 50170, Том 2. В зависимости от конкретного DP-ведущего, диагностические данные могут быть прочитаны либо с помощью STEP 5, либо с помощью STEP 7 для всех DP-ведомых, соответствующих указанному стандарту.

Диагностические данные ведомых устройств PROFIBUS состоят из диагностических данных модуля, сведений о состоянии модуля и диагностических данных канала. Подробные сведения о диагностических данных ведомых устройств DP можно найти в главе "Ввод в эксплуатацию и диагностика" в руководстве *ET 200S Distributed I/O Device*.

Диагностические данные канала: диагностические данные канала следуют за состоянием модуля и содержат сведения об ошибках канала модулей. В таблице 2-29 перечислены типы ошибок канала.

Таблица 2–29 Типы ошибок канала модуля последовательного интерфейса ET 200S		
Тип ошибки	Значение	Чего делать
00110: Обрыв провода	Оборвался или отсоединился провод.	Проверьте подключение проводов к клеммам. Проверьте кабель коммуникационного партнера.
00111: Переполнение	Переполнение буфера; превышение длины сообщения.	Функциональный блок S_RCV следует вызывать чаще.
01000: Нулевая длина	Только для 3964(R): передано сообщение нулевой длины.	Проверьте, почему партнер по связи передает кадры, не содержащие данных пользователя.
01001: Ошибка	Произошла внутренняя ошибка модуля.	Замените модуль.
10000: Ошибка назначения параметра	Модулю не были назначены параметры.	Исправьте назначение параметров.
10110: Ошибка сообщения	Ошибка кадра, ошибка четности.	Проверьте настройки связи.

2.14 Технические характеристики

Технические характеристики протоколов и интерфейсов

Таблица 2–30 Общие технические характеристики модуля ET 200S	
Общие технические характеристики	
Элементы отображения	Светодиоды: зеленый, TX (передача) зеленый, RX (прием) красный, SF (системная ошибка)
Поддерживаемые драйверы протоколов	Драйвер протокола 3964(R) ASCII драйвер
Скорости передачи для протокола 3964(R)	110, 300, 600, 1200, 2400, 4800, 9600, 19200 бит/с (полудуплекс)
Скорости передачи для ASCII драйвера	110, 300, 600, 1200, 2400, 4800, 9600, 19200 бит/с (дуплекс)
Кадр символа (10 битов или 11 битов)	Количество битов на символ: 7 или 8 Количество старт/стоп-битов: 1 или 2 Четность: нет, чет, нечет, любая
Требования к памяти для стандартных блоков (FB)	Передача и прием: приблиз., 4300 байт
Технические характеристики интерфейса RS–232C	
Интерфейс	RS–232C, 8 выводов
Сигналы RS–232C	TXD, RXD, RTS, CTS, DTR, DSR, DCD, PE Все указанные выводы электрически развязаны от внутреннего источника питания модуля ET 200S.
Максимальное расстояние передачи	15 м
Технические характеристики интерфейса RS–422/485	
Интерфейс	RS–422, 5 выводов RS–485, 3 вывода
Сигналы RS–422 Сигналы RS–485	TXD (A), RXD (A), TXD (B), RXD (B), PE R/T (A), R/T (B), PE Все указанные выводы электрически развязаны от внутреннего источника питания модуля ET 200S.
Максимальное расстояние передачи	1200 м

Технические характеристики

Габариты и вес	
Габариты Ш x В x Г (мм)	15 x 81 x 52
Вес	Приблиз., 50 г
Данные для отдельных модулей	
RS-232C	
• Количество входов	4
• Количество выходов	3
RS-422	
• Кол-во входных пар	1
• Кол-во выходных пар	1
RS-485	
• Кол-во входных/ выходных пар	1
Длина кабеля	
• Экранированного (RS-232C)	Макс. 15 м
• Экранированного (RS-422/485)	Макс. 1200 м
Степень защиты ¹	IEC 801-5
Напряжение, токи, потенциалы	
Номинальное напряжение питания электронных модулей (L+)	24 В DC
• Защита от неправильной полярности	Да
Развязка	
• Между каналами и задней шиной	Да
• Между каналами и цепями питания электронных устройств	Да
• Между каналами	Нет
• Между каналами и PROFIBUS-DP	Да
Испытания на пробой изоляции между	
• Каналами и задней шиной и между каналами и линией L+ (напряжение нагрузки)	500 В DC
• Линией L+ (напряжение нагрузки) и задней шиной	500 В AC
Источник тока	
• От задней шины	Макс. 10 мА
• От линии L+ (напряжение нагрузки)	Макс. 120 мА, Тип. 50 мА
Мощность, рассеиваемая модулем	Тип. 1,2 Вт

Состояния, прерывание, диагностика	
Отображение состояний	Зеленый светодиод (TX) Зеленый светодиод (RX)
Диагностические функции	
• Отображение групповой ошибки	Красный светодиод "SF"
• Возможность отображения диагностической информации	Возможно
Выходы	
Выходы, уровни RS-232C	± Макс. 10 В
• Для емкостной нагрузки	Макс. 2500 пФ
• Защита от короткого замыкания	Да
• Ток короткого замыкания	Приблиз., 60 мА
• Напряжение между выходами или входами и РЕ (земля)	Макс. 25 В
Выходы, RS-422/485	
Соппротивление нагрузки	Мин. 50 кОм
• Защита от короткого замыкания	Да
• Ток короткого замыкания	Приблиз., 60 мА

- ¹ При подключении цепей напряжения пользователя требуется внешнее защитное оборудование:
- Молниеотвод, адаптер для установки на стандартную рейку
 - Молниеотвод, защитный модуль типа KTAD-24B

Обзор главы

Раздел	Описание	Страница
3.1	Обзор изделия	3–2
3.2	Назначение клемм	3–5
3.3	Протокол передачи Modbus	3–14
3.4	Драйвер Modbus-Master	3–18
3.5	Драйвер Modbus-Slave	3–41
3.6	Диагностика	3–84
3.7	USS-Master	3–99
3.8	Функции, выполняемые при запуске, и режимы работы драйвера Modbus/USS модуля последовательного интерфейса ET 200S	3–121
3.9	Технические характеристики	3–125

Подробную информацию по конфигурированию, настройке, выполнению кабельных и проводных соединений, по вводу в эксплуатацию и диагностике, а также технические характеристики системы распределенного ввода/вывода ET 200S можно найти в руководстве “Станция распределенной периферии ET 200S”.

3.1 Обзор изделия

Модуль последовательного интерфейса ET 200S SI Modbus/USS

Заказной номер

6ES7 138-4DF10-0AB0

Описание

Модуль последовательного интерфейса ET 200S Modbus/USS (ET 200S SI Modbus/USS) – это один из модулей семейства ET 200S, подключаемый в базовую корзину (?) ET 200S. Он позволяет реализовать последовательный обмен данными по одному из трёх аппаратных протоколов (интерфейсов) (RS-232C, RS-422 и RS-485) с использованием одного из двух программных протоколов:

- Modbus
- USS-Master (USS-Ведущий)

Модуль ET 200S SI Modbus/USS позволяет организовать обмен данными между программируемыми контроллерами (ПЛК) или компьютерами по соединению “точка-точка”. Обмен данными состоит из серии сеансов последовательной асинхронной передачи данных.

Режим связи выбирается при параметрировании модуля в программе конфигурирования аппаратных средств (hardware configuration) в STEP 7 или с помощью другой программы конфигурирования (не S7). В каталоге аппаратных средств отображаются следующие версии модуля:

- Modbus-Master (8 байт)
- Modbus-Master (4 байта)
- Modbus-Slave (8 байт)
- Modbus-Slave (4 байта)
- USS-Master (8 байт)
- USS-Master (4 байта)

Передача данных по 8 байт позволяет увеличить производительность (скорость), но требует большего адресного пространства ввода/вывода в ET 200S. Для передачи данных по 4 байта требуется меньший объём адресного пространства ввода/вывода в корзине ET 200S, но при этом производительность (скорость) меньше. Выбор версии модуля определяется требованиями решаемой задачи.

Модуль последовательного интерфейса ET 200S SI Modbus/USS показан на рисунке 3–1.

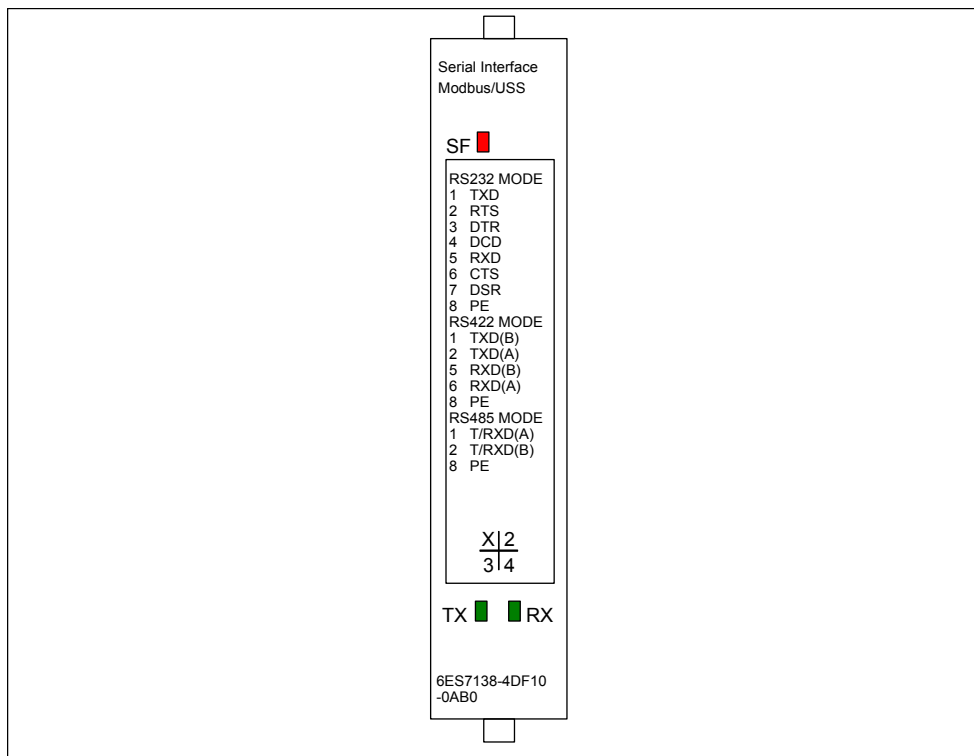


Рисунок 3–1 Модуль последовательного интерфейса ET 200S SI Modbus/USS

Модуль последовательного интерфейса ET 200S SI Modbus/USS обладает следующими функциями и свойствами:

- Встроенный интерфейс RS–232C, RS–422 или RS–485
- Скорость передачи до 38.4 кбод, полудуплексный режим
- В микропрограмму модуля встроены следующие протоколы связи:
 - Драйвер Modbus-Master (ведущее устройство Modbus)
 - Драйвер Modbus-Slave (ведомое устройство Modbus)
 - Драйвер USS-Master (ведущее устройство USS)

Функционирование драйверов определяется параметрами, выбранными для модуля. В таблице 3–1 перечислены возможные сочетания драйверов и интерфейсов, а также функции драйверов.

Таблица 3–1 Функции драйверов модуля Modbus/USS			
Функция	RS–232C	RS–422	RS–485
Драйвер Modbus	Да	Да	Да
Автоматическое использование сигналов RS 232C	Да	Нет	Нет
Драйвер USS-Master	Нет	Нет	Да

Таблица 3–2 содержит краткое описание светодиодов состояния модуля.

Таблица 3–2 Светодиоды	
Светодиоды	Описание
SF	Светодиод (красный) является признаком неисправности.
TX	Светодиод (зелёный) служит для индикации передачи через порт.
RX	Светодиод (зелёный) служит для индикации приёма через порт.

3.2 Схемы с указанием назначения контактов

3.2.1 Назначение клемм

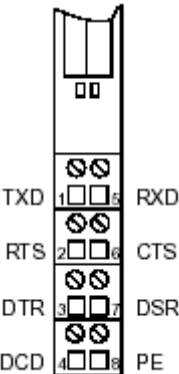
Указания по выполнению проводных соединений

Кабели (клеммы 1...8) должны быть экранированы. Экраны должны быть зафиксированы с обеих сторон. Для этого следует использовать опорные элементы (см. раздел, касающийся дополнительных принадлежностей для ET 200S в руководстве “Станция распределенной периферии ET 200S”).

Назначение клемм при обмене данными через RS–232C

С ведомой системой можно установить соединение “точка-точка”. При этом дополнительные линии интерфейса RS–232C не используются.

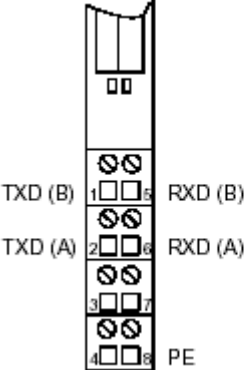
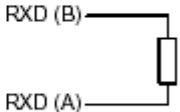
В таблице 3–3 приводится назначение клемм модуля последовательного интерфейса ET 200S SI Modbus/USS, когда выбран протокол связи RS–232C.

Таблица 3–3 Назначение клемм при обмене данными по интерфейсу RS–232C		
Вид	Назначение клемм	Замечания
		Режим: дуплексный Клеммы 1 TXD Передаваемые данные 5 RXD Принимаемые данные 2 RTS Запрос на передачу 6 CTS Готовность к передаче 3 DTR Готовность терминала 7 DSR Готовность данных 4 DCD Детектор несущей 8 PE Земля

Назначение клемм при обмене данными через интерфейс RS-422

С ведомой системой можно установить соединение “точка-точка”.

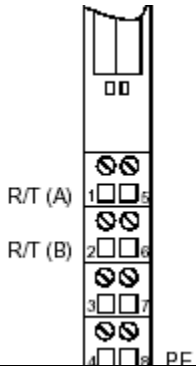
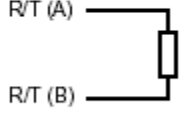
В таблице 3–4 приводится назначение клемм модуля ET 200S SI Modbus/USS для случая, когда выбран протокол связи RS-422.

Таблица 3–4 Назначение клемм при обмене данными по интерфейсу RS-422		
Вид	Назначение клемм	Замечания
	<p>Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом.</p> 	<p>Режим: дуплексный</p> <p>Клеммы</p> <p>1 TXD (B)</p> <p>5 RXD (B)</p> <p>2 TXD (A)</p> <p>6 RXD (A)</p> <p>8 PE Земля</p>

Назначение клемм при обмене данными через RS-485

К ведущей системе можно подключить до 32-х ведомых устройств, создав, таким образом, многоточечное соединение (сеть). Драйвер модуля переключает двухпроводную линию из режима передачи в режим приёма и наоборот.

В таблице 3–5 указано назначение клемм модуля ET 200S SI Modbus/USS для случая, когда выбран протокол связи RS-485.

Таблица 3–5 Назначение клемм при обмене данными по интерфейсу RS-485		
Вид	Назначение клемм	Замечания
	<p>Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом.</p> 	<p>Режим: дуплексный</p> <p>Клеммы</p> <p>1 R/T (A)</p> <p>2 R/T (B)</p> <p>8 PE Земля</p>

Разводка контактов соединительного кабеля RS-232C с 9-контактным разъемом

На рисунке 3-2 показано подключение кабеля для соединения “точка-точка” по RS-232C между модулем ET 200S SI и ведомым устройством с использованием 9-контактного гнезда D типа.

- На стороне ET 200S сигнальные провода подключаются к клеммам с соответствующими номерами.
- На стороне ведомого устройства используется 9-контактное гнездо типа sub D.

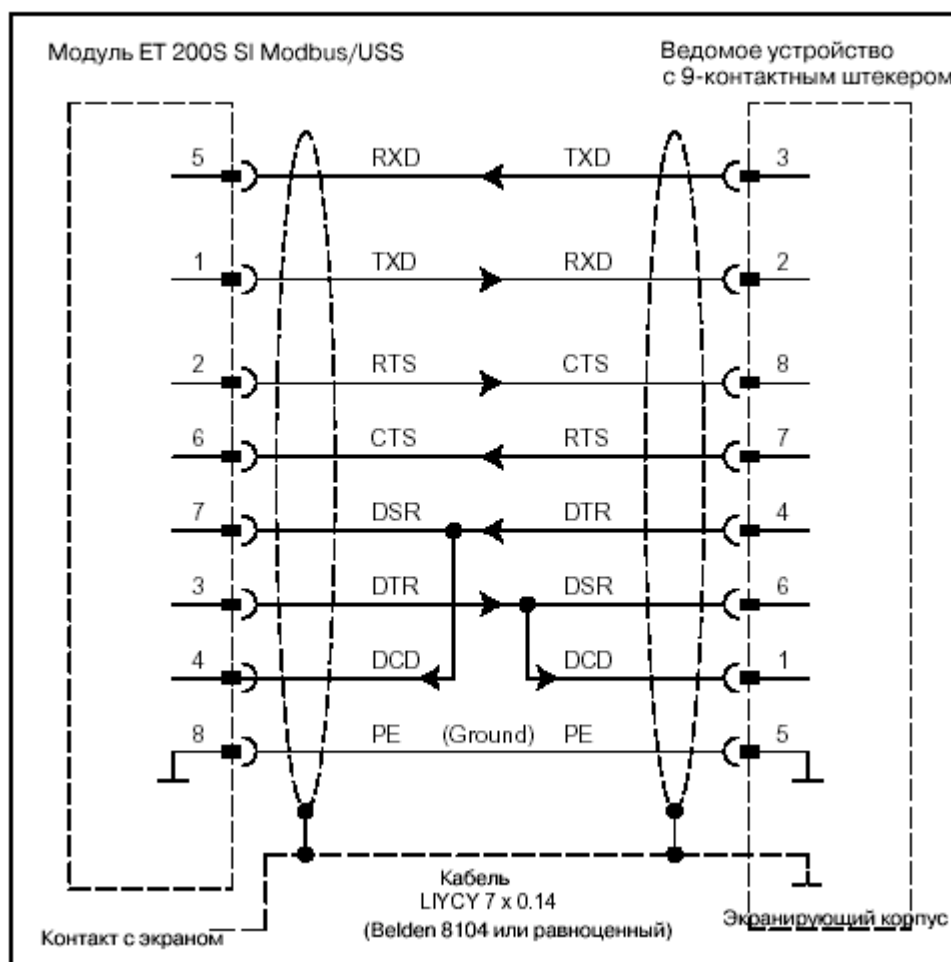


Рисунок 3–2 Разводка контактов соединительного кабеля RS-232C для 9-контактного разъема (1 ведущее устройство, 1 ведомая система)

Разводка контактов соединительного кабеля RS-232C с 25-контактным разъемом

На рисунке 3-3 показано подключение кабеля для соединения “точка-точка” по RS-232C между модулем ET 200S SI и ведомым устройством с использованием 25-контактного разъема D типа.

- На стороне ET 200S сигнальные провода подключаются к клеммам с соответствующими номерами.
- На стороне ведомого устройства используется 25-контактный разъем типа sub D.

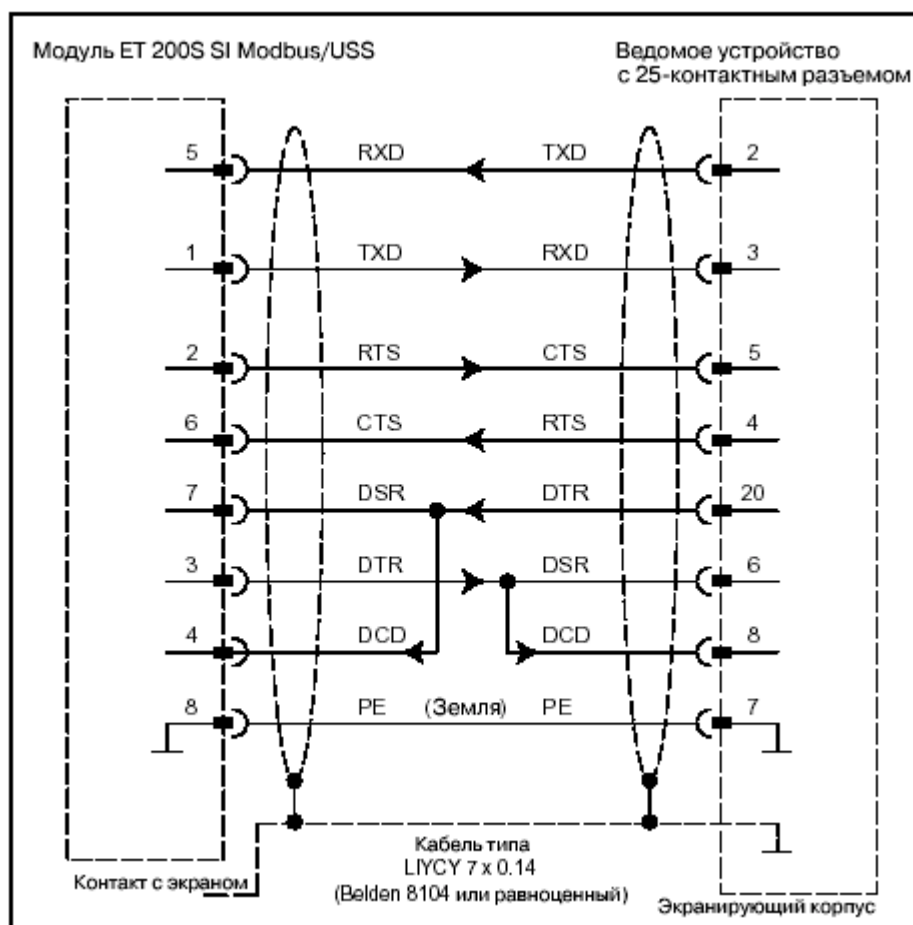


Рисунок 3-3 Разводка контактов соединительного кабеля RS-232C с 25-контактным разъемом (1 ведущее устройство, 1 ведомая система)

Разводка контактов соединительного кабеля RS-422 с 15-контактным разъемом

На рисунке 3-4 показано подключение кабеля для соединения “точка-точка” по RS-422 между модулем ET 200S SI и ведомым устройством с использованием 15-контактного разъема D типа.

- На стороне ET 200S сигнальные провода подключаются к клеммам с соответствующими номерами.
- На стороне ведомого устройства используется 15-контактный разъем типа sub D.

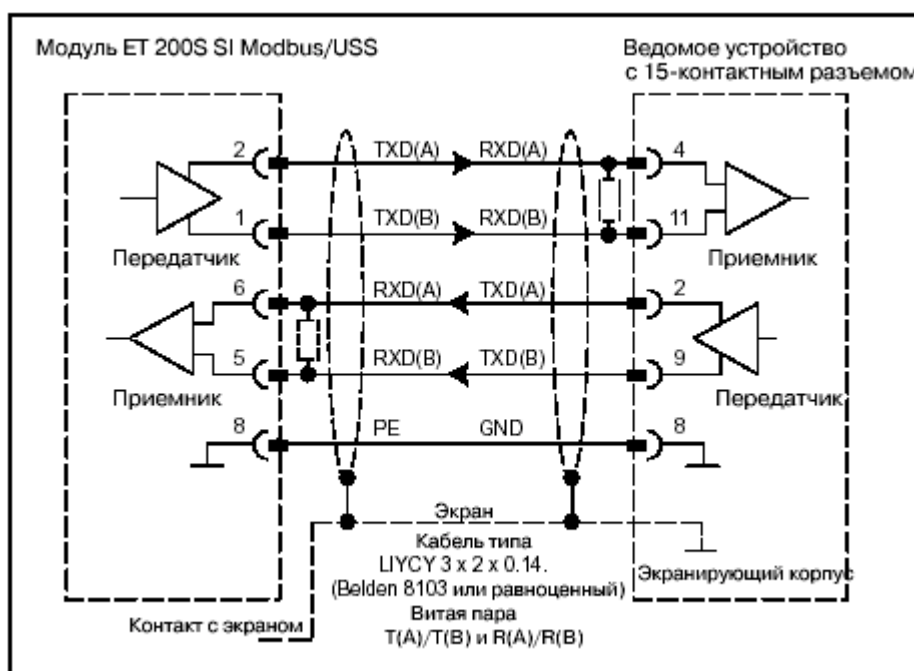


Рисунок 3-4 Разводка контактов соединительного кабеля RS-422 с 15-контактным разъемом (1 ведущее устройство, 1 ведомая система)

Примечание

Если длина кабеля превышает 50 м, следует подсоединить окончное сопротивление (приблизительно, 330 Ом), чтобы обмен данными не сопровождался ошибками. Смотрите рисунок 3-4.

Максимальная длина кабеля данного типа при скорости 38 400 бод составляет 1200 м.

Разводка контактов соединительного кабеля RS-485 с 15-контактным разъемом

На рисунке 3-5 показано подключение кабеля для соединения “точка-точка” по RS-485 между модулем ET 200S SI и ведомым устройством с использованием 15-контактного разъема D типа.

- На стороне ET 200S сигнальные провода подключаются к клеммам с соответствующими номерами.
- На стороне ведомого устройства используется 15-контактный разъем типа sub D.

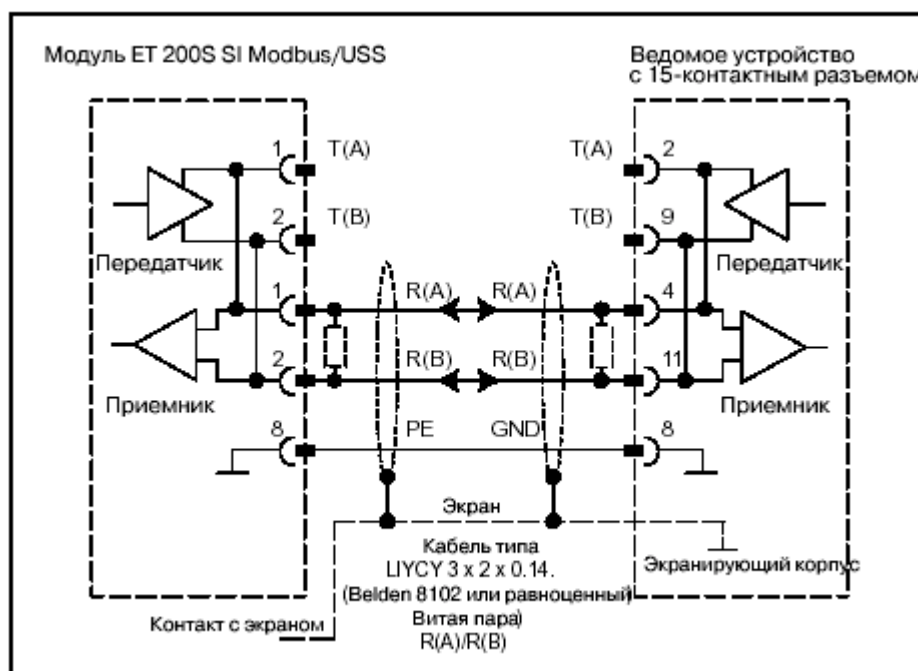


Рисунок 3-5 Разводка контактов соединительного кабеля RS-485 с 15-контактным разъемом (1 ведущее устройство, 1 ведомая система)

Примечание

Если длина кабеля превышает 50 м, следует подсоединить оконечное сопротивление (приблизительно, 330 Ом), чтобы обмен данными не сопровождался ошибками. Смотрите рисунок 3-4.

Максимальная длина кабеля данного типа при скорости 38 400 бод составляет 1200 м.

3.2.2 Интерфейс RS–232C

Интерфейс RS–232C – интерфейс последовательной передачи данных в соответствии со стандартом RS–232C. Для передачи сигналов используется напряжение. Свойства интерфейса RS–232C приводятся в таблице 3–6.

Таблица 3–6 Свойства интерфейса RS–232C	
Свойство	Описание
Тип:	Интерфейс напряжений
Фронтальный штекер:	8-контактный стандартный клеммный соединитель модуля ET 200S
Сигналы RS–232C:	TXD, RXD, RTS, CTS, DTR, DSR, DCD, GND
Скорость передачи:	Максимум 38.4 кбит/с
Длина кабеля:	Максимум 15 м, кабель LIYCY 7 x 0.14
Стандарты:	DIN 66020, DIN 66259, EIA RS–232C, CCITT V.24/V.28
Степень защиты:	IP 20

Сигналы RS–232C

Модуль Modbus/USB поддерживает сигналы RS–232C (см. таблицу 3–7).

Таблица 3–7 Сигналы интерфейса RS–232C		
Сигнал	Название	Значение
TXD	Transmitted Data (Передаваемые данные)	Передача данных; линия передачи удерживается в режиме ожидания в состоянии логической “1”.
RXD	Received Data (Принимаемые данные)	Прием данных; линия приема должна удерживаться в состоянии логической “1” коммуникационным партнером.
RTS	Request To Send (Запрос на передачу)	ВКЛ: ET 200S SI готов к передаче. ВЫКЛ: ET 200S SI не передает данные.
CTS	Clear To Send (Готовность к передаче)	Коммуникационный партнер может принимать данные от ET 200S. Модуль ET 200S SI считает этот сигнал ответом на RTS (ВКЛ).
DTR	Data Terminal Ready (Готовность терминала)	ВКЛ: На ET 200S SI подано питание и он готов к работе. ВЫКЛ: На ET 200S SI не подано питание и он не готов к работе.
DSR	Data Set Ready (Готовность данных)	ВКЛ: Коммуникационный партнер запитан и готов к работе. ВЫКЛ: Коммуникационный партнер не запитан и не готов к работе.
DCD	Data Carrier Detect (Детектор несущей)	Обнаружен сигнал несущей (если подключен модем).

Автоматическое использование вспомогательных сигналов

Автоматическое функционирование дополнительных сигналов RS-232C в модуле ET 200S SI реализуется следующим образом:

- Как только модуль ET 200S SI переводится в режим автоматического функционирования дополнительных сигналов RS-232C (в результате конфигурирования), он устанавливает линию RTS в состояние ВЫКЛ, а линию DTR – во ВКЛ (ET 200S SI готов к работе).
- Передача и прием кадров сообщений становятся возможными только после того, как линия DTR установлена во ВКЛ. Пока DTR находится в состоянии ВЫКЛ, данные через интерфейс RS-232C приниматься не будут. Любое задание на передачу данных прерывается, генерируется соответствующее сообщение об ошибке.
- Если **задание на передачу** поставлено в очередь, RTS переводится во ВКЛ и начинается отсчет сконфигурированного времени ожидания вывода данных. По истечении времени ожидания вывода данных и при условии, что CTS = ВКЛ, по интерфейсу RS 232C передаются данные.
- Если во время передачи данных сигнал CTS не установлен в состояние ВКЛ в течение времени ожидания вывода данных, или состояние CTS меняется в процессе передачи на ВЫКЛ, задание на передачу данных прерывается, генерируется соответствующее сообщение об ошибке.
- После того, как данные переданы, линия RTS отключается по истечении сконфигурированного времени ожидания перехода RTS в ВЫКЛ. Модуль ET 200S SI не ожидает переключения линии CTS в ВЫКЛ.
- **Прием** данных через интерфейс RS-232C становится возможным, как только линия DSR устанавливается во ВКЛ. Если ожидается переполнение буфера приема модуля ET 200S SI, ответ от модуля ET 200S SI не поступает.
- Когда DSR из ВКЛ переводится в состояние ВЫКЛ, активное задание на передачу, а также процедура приема данных прерываются с генерацией соответствующего сообщения об ошибке.

Примечание

Автоматическое использование вспомогательных сигналов RS-232C возможно только в полудуплексном режиме.

Примечание

Время ожидания переключения RTS в ВЫКЛ должно быть настроено в параметрах интерфейса таким образом, чтобы партнёр по связи мог принять последние символы кадра сообщения целиком, прежде чем будет сброшен сигнал RTS (сброшен запрос на передачу). Время ожидания вывода данных должно быть настроено таким образом, чтобы партнёр по связи мог перейти в состояние готовности к приёму до истечения данного времени.

Временная диаграмма переключения вспомогательных сигналов

На рисунке 3–6 показана последовательность переключения сигналов при запросе на передачу:

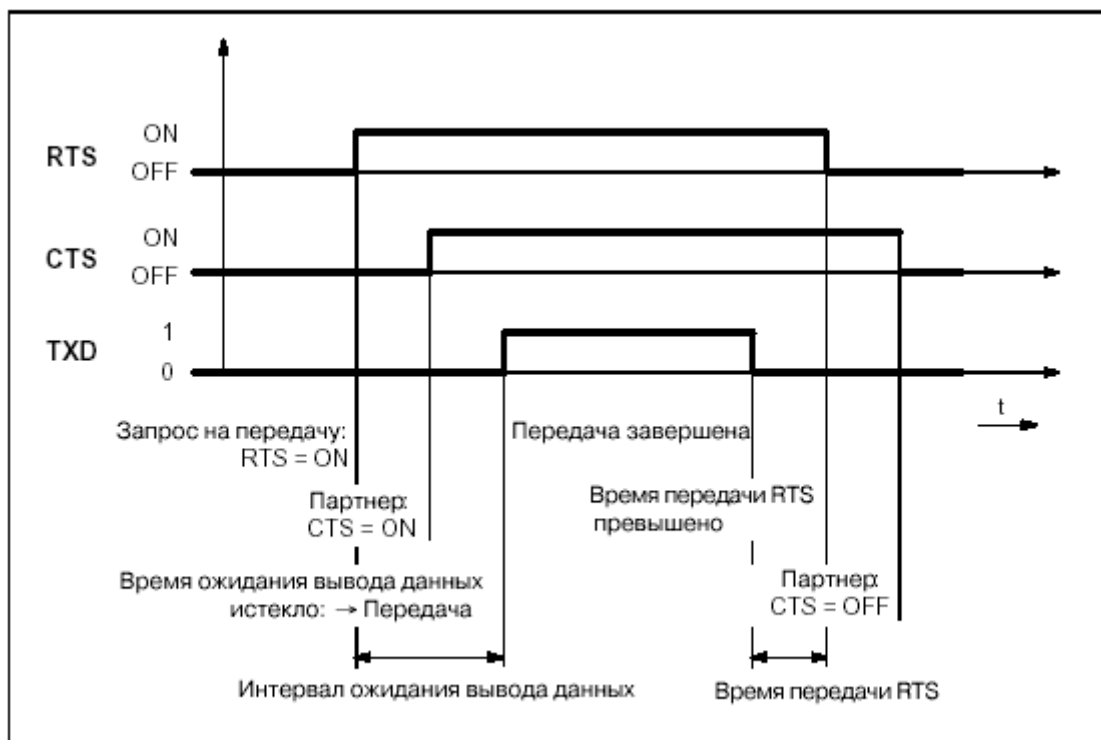


Рисунок 3–6 Временная диаграмма автоматического использования вспомогательных сигналов RS–232C

3.2.3 Интерфейс RS–422/485

Интерфейсы RS–422/485 – это интерфейсы последовательной передачи данных в соответствии со стандартами RS–422/485. Для передачи данных используются дифференциальные напряжения.

Таблица 3–8 Свойства интерфейса RS 422/485	
Характеристика	Описание
Тип:	Интерфейс дифференциальных напряжений
Фронтальный штекер:	8-контактный стандартный клеммный соединитель модуля ET 200S
Сигналы RS–422:	TXD (A), RXD (A), TXD (B), RXD (B), GND
Сигналы RS–485:	R/T (A), R/T (B), GND
Скорость передачи:	Максимум 38.4 кбит/с
Длина кабеля:	Максимум 1200 м, тип кабеля LIYCY 7 x 0.14
Стандарты:	EIA RS–422/485, CCITT V.11/V.27
Степень защиты:	IP 20

3.3 Протокол связи Modbus

3.3.1 Структура сообщения

Обмен данными в режиме “ведущий-ведомый” и/или “ведомый-ведущий” начинается с передачи адреса ведомого устройства, за которым следует код функции. После этого передаются данные. При обмене данными в режиме “ведущий-ведомый” и/или “ведомый-ведущий” используются следующие элементы:

АДРЕС ВЕДОМОГО УСТРОЙСТВА	Адрес ведомого устройства Modbus
КОД ФУНКЦИИ	Код функции Modbus
ДАННЫЕ	Количество байтов, количество логических ячеек, данные
CRC	Контрольная сумма

Структура поля данных зависит от используемого кода функции. В конце сообщения передается контрольная сумма CRC. В таблице 3–9 приводятся элементы сообщения.

Таблица 3–9 Структура сообщения			
АДРЕС	ФУНКЦИЯ	ДАННЫЕ	CRC
байт	байт	n байт	2 байта

3.3.2 Адрес ведомого устройства

Адрес ведомого устройства может быть указан в диапазоне 1 ... 255. Адрес используется для обращения к определённому ведомому устройству, подключенному к шине.

Широковещательное (групповое) сообщение

Для обращения ко всем ведомым устройствам сети (шины) ведущее устройство использует нулевой адрес (0).

Примечание

Широковещательные сообщения могут использоваться только для функций 05, 06, 15 и 16.

Ответные сообщения на широковещательные сообщения ведомыми устройствами не возвращаются.

3.3.3 Функции ведущего и ведомого устройств

Код функции определяет значение, а также структуру сообщения. Функции перечисляются в таблице 3–10. Там же указывается их поддержка ведущим и ведомым устройствами.

Таблица 3–10 Функции ведущего и ведомого устройств			
Код функции	Описание	Ведущий	Ведомый
01	Чтение логической ячейки	√	√
02	Чтение дискретного входа	√	√
03	Чтение регистров хранения	√	√
04	Чтение входных регистров	√	√
05	Изменение состояния логической ячейки	√	√
06	Запись нового значения в регистр	√	√
07	Чтение состояния ошибки	√	
08	Проверка связи	√	√
11	Счётчик событий связи	√	
12	Протокол событий связи	√	
15	Изменение состояния нескольких логических ячеек	√	√
16	Запись новых значений в несколько регистров	√	√

3.3.4 Поле данных (ДАННЫЕ)

Поле данных (ДАННЫЕ) используется для передачи следующих данных, связанных с определёнными кодами функций:

- Количество байтов
- Адрес первой ячейки
- Адрес первого регистра
- Количество логических ячеек
- Количество регистров

3.3.5 Завершение сообщения и контрольная сумма CRC

Признаком завершения сообщения служит контрольная сумма CRC 16, состоящая из 2-х байтов. Она рассчитывается с использованием следующего полинома:

$$x^{16} + x^{15} + x^2 + 1$$

Первым передаётся младший байт, за которым следует старший байт. Если в течение времени, требуемого для передачи трёх с половиной символов ($3.5 \times$ время ожидания символа), не происходит передачи данных, модуль Modbus/USB считает сообщение завершённым.

Время ожидания, являющееся признаком завершения сообщения, зависит от скорости передачи.

По истечении времени ожидания завершения сообщения выполняется обработка сообщения, возвращённого ведомым устройством, а также проверяется его формат (см. таблицу 3–11).

Таблица 3–11 Завершение сообщения	
Скорость передачи	Интервал ожидания
38400 бит/с	4 мс
19200 бит/с	4 мс
9600 бит/с	4 мс
4800 бит/с	8 мс
2400 бит/с	16 мс
1200 бит/с	32 мс
600 бит/с	64 мс
300 бит/с	128 мс

3.3.6 Ответы в случае ошибок

Если в запросном сообщении ведущего устройства распознана ошибка (например, обнаружен недопустимый адрес регистра), ведомое устройство выполняет следующие действия:

- Ведомое устройство устанавливает старший бит в поле кода функции ответного сообщения.
- Ведомое устройство передаёт один байт кода ошибки, описывающий причину ошибки.

Сообщение с кодом ошибки

Ответное сообщение ведомого устройства, содержащее код ошибки, может, например, иметь следующую структуру: адрес ведомого устройства 5, код функции 5, код ошибки 2.

Ответное сообщение от ведомого устройства КОД_ОШИБКИ_хх	05H	Адрес ведомого устройства
	85H	Код функции
	02H	Код ошибки (1 ... 7)
	ххН	Младший байт кода CRC
	ххН	Старший байт кода CRC

Если на драйвер поступает ответное сообщение с кодом ошибки, текущее задание (запрос) завершается с ошибкой.

Номер ошибки, соответствующий принятому коду ошибки (код ошибки 1–7), также вводится в область SYSTAT.

В адресуемый (конечный) блок данных S_RCV данные не записываются. В таблице 3–12 перечислены коды ошибок, посылаемые модулем.

Таблица 3–12 Коды ошибок		
Код ошибки	Описание	Возможная причина
01	Недопустимая функция	Принят недопустимый код функции
02	Недопустимый адрес данных	Попытка доступа в не активизированную область SIMATIC (см. <i>конфигурирование параметров – области, границы</i>)
03	Недопустимое значение в поле данных	Длина превышает 2040 бит или 127 регистров, поле данных не FF00 или 0000 для FC05, дополнительный диагностический код <> 0000 для FC08.
04	Неисправность соответствующего устройства	До сих пор не выполнена инициализация с помощью функционального блока (FB) Modbus-коммуникаций, либо FB сообщает об ошибке. Ошибка при передаче данных модулем CPU (например, DB не существует, превышена максимальная длина передаваемых данных (размер блока CPU <--> модуля)

3.4 Драйвер Modbus-Master

3.4.1 Используемые интерфейсы и протоколы

Можно использовать как RS–232, так и RS–422/485 (X27).

С помощью данного драйвера интерфейс RS–422/485 можно использовать как в 2–проводном режиме, так и в 4–проводном режиме. В 2–проводном режиме можно подключить до 32-х ведомых устройств к одному ведущему в режиме полудуплексной связи. Таким образом, можно создать многоточечное соединение (сеть). В случае 4–проводного режима (RS–422) можно соединить между собой только одно ведущее устройство и одно ведомое устройство в режиме полудуплексной связи.

3.4.2 Передача данных модулю ET 200S, являющемуся ведущим устройством Modbus

Обмен данными между модулем и CPU выполняется с помощью функциональных блоков (FB) S_SEND и S_RCV. Функциональный блок S_SEND запускается по переднему фронту на входе REQ, если требуется вывод данных. Функциональный блок S_RCV приводится в готовность путём установки EN_R в состояние “1”. S_RCV требуется для всех функций, связанных со чтением данных.

Функциональный блок FB3 S_SEND: Передача данных партнёру по связи

Чтобы ведущее устройство Modbus (Modbus Master) могло выполнить запрос, требуется запуск обоих FB: S_SEND и S_RCV. Функциональный блок S_SEND запускается по переднему фронту на входе REQ, если требуется вывод данных на модуль. Функциональный блок S_RCV приводится в готовность к приёму данных от модуля путём переключения EN_R в “1”. S_RCV требуется для всех функций, связанных со чтением данных. На рисунке 3-7 показана последовательность переключения сигналов блоков S_SEND и S_RCV, когда ведущее устройство Modbus выполняет запрос.

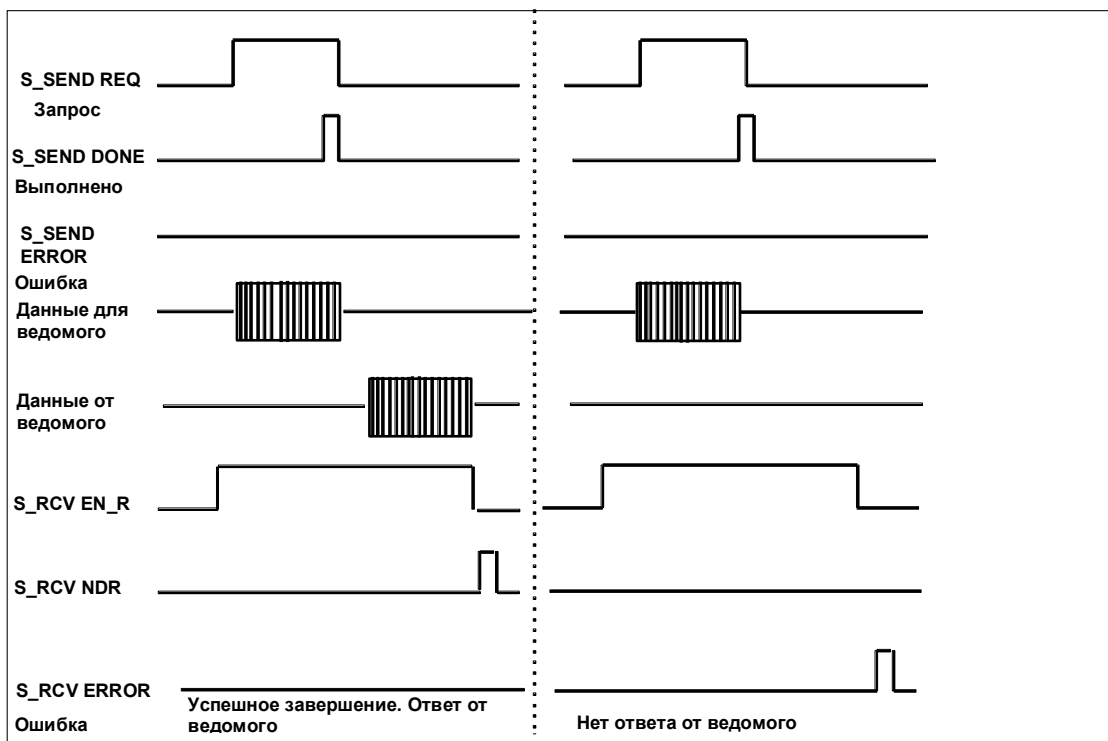


Рисунок 3–7 Временная диаграмма для запроса Modbus

Передача данных инициируется по переднему фронту на входе REQ. Для передачи данных может потребоваться несколько вызовов (циклов программы), в зависимости от объема передаваемых данных.

Функциональный блок S_SEND может быть вызван в пределах цикла, когда на входе параметра R присутствует "1". В результате передача данных модулю прекращается, и функциональный блок S_SEND возвращается в своё исходное состояние. Данные, которые уже были приняты модулем, продолжают передаваться коммуникационному партнёру. Если вход R непрерывно находится в состоянии 1, это приводит к отключению процедуры передачи.

Адрес модуля ET 200S SI, к которому производится обращение, указывается в параметре LADDR.

Сигнал на выходе DONE соответствует результату "запрос завершён без ошибок". Сигнал ERROR означает, что произошла ошибка. В случае ошибки на выходе STATUS отображается номер соответствующего события. Если ошибок не было, в STATUS лежит 0. Выходы DONE и ERROR/STATUS также актуализируются при сбросе функционального блока S_SEND. В случае ошибки также сбрасывается бит BR. Если работа блока завершается без ошибок, BR находится в состоянии 1.

В таблице 3–13 показано представление блока FB3 S_SEND в форматах STL и LAD.

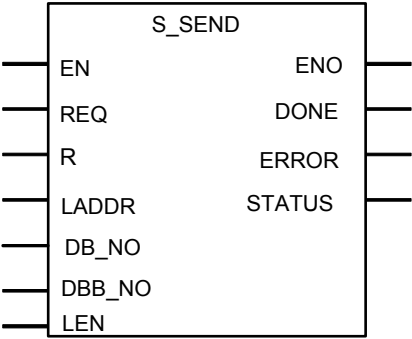
Примечание

В функциональном блоке S_SEND не предусмотрена проверка параметров. Если параметры заданы неверно, CPU перейдёт в режим STOP.

Выполнение иницированного задания модулем ET200S SI после перехода CPU из режима STOP в RUN возможно только после завершения процедуры, предусмотренной в модуле ET200S при запуске CPU для функционального блока S_SEND. Любые запросы, выставленные в момент выполнения данной процедуры, утрачены не будут, и будут переданы сразу же по завершении процедуры.

Вызов FB3

Таблица 3–13 Представление FB3 S_SEND в форматах STL и LAD

Представление в формате STL	Представление в формате LAD
CALL S_SEND, I_SEND	I_SEND
REQ: =	
R: =	
LADDR: =	
DB_NO: =	
DBB_NO: =	
LEN: =	
DONE: =	
ERROR: =	
STATUS: =	

Примечание

Параметры EN и ENO отображаются только в случае графического представления (LAD или FBD). Для обработки этих параметров компилятор использует бит BR.

Бит BR переводится в состояние “1”, если работа блока завершилась без ошибок. Если произошла ошибка, BR сбрасывается в 0.

Присвоения в области данных

Функциональный блок S_SEND работает с экземплярным DB I_SEND, номер которого указывается в вызове. Доступ к данным в экземплярном DB невозможен.

Примечание

Исключение: в случае ошибки, при которой выход STATUS == W#16#1E0F, более подробные сведения об ошибке можно получить из переменной SFCERR. Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземпляру DB.

Параметры FB3 S_SEND

В таблице 3–14 перечислены параметры S_SEND (FB3).

Таблица 3–14 FB3: Параметры функционального блока S_SEND				
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
REQ	INPUT	BOOL	Инициирование задания по положительному фронту	
R	INPUT	BOOL	Прерывание задания	Текущее задание прервано. Передача заблокирована.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес, взятый из STEP 7.
DB_NO	INPUT	INT	Номер блока данных	Номер передаваемого блока: зависит от CPU ("0" указывать не допускается).
DBB_NO	INPUT	INT	Номер байта данных	$0 \leq \text{DBB_NO} \leq 8190$ данные передаются в формате слов
LEN	INPUT	INT	Длина данных	$1 \leq \text{LEN} \leq 200$, указывается количество байтов
DONE ¹	OUTPUT	BOOL	Завершение задания без ошибок	Параметр STATUS == 16#00
ERROR ¹	OUTPUT	BOOL	Завершение задания с ошибками	В параметре STATUS содержатся сведения об ошибке
STATUS ¹	OUTPUT	WORD	Спецификация ошибки	Если ERROR == 1, в параметре STATUS содержатся сведения об ошибке

¹ Параметр DONE доступен в течение **одного** цикла CPU после успешного выполнения задания.

Временная диаграмма для FB3 S_SEND

На рисунке 3-8 показана последовательность переключения параметров DONE и ERROR в зависимости от того, какие сигналы подаются на входы REQ и R.

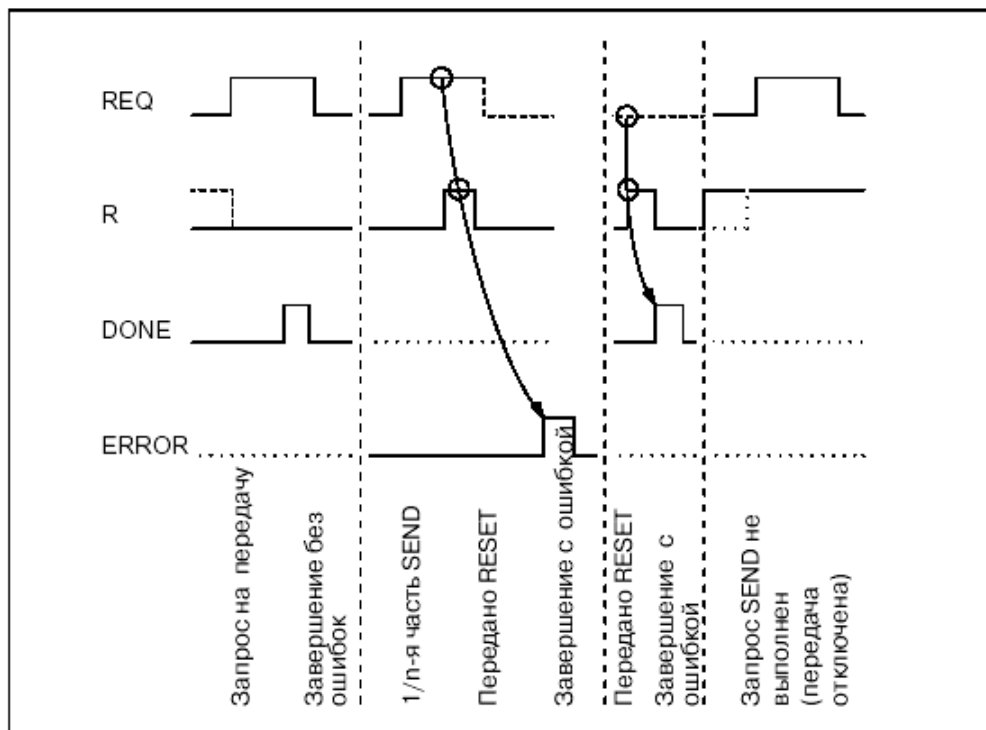


Рисунок 3–8 Временная диаграмма для FB3 S_SEND

Примечание

Запуск происходит по переднему фронту на входе REQ. Переключение сигнала из “0” в “1” на входе REQ (передний фронт) является достаточным условием для инициирования передачи. Значение RLO (RLO = результат логической операции) не обязательно должно быть = “1” в течение всего времени передачи данных.

FB2 S_RCV: Приём данных от партнёра по обмену данными

Функциональный блок S_RCV выполняет передачу данных из модуля ET 200S SI в область данных S7, определяемую параметрами DB_NO и DBB_NO. Функциональный блок S_RCV вызывается статически (безусловный вызов) в пределах цикла. Также можно осуществлять вызов по временному прерыванию.

Если параметр EN_R находится статически в состоянии "1", программа выполняет проверку, могут ли быть прочитаны данные модулем ET 200S SI. Активную процедуру передачи данных можно прервать, сбросив параметр EN_R в состояние "0". Прерванное задание на прием завершается сообщением об ошибке (выход STATUS). Прием будет выключен в течение всего времени, пока параметр EN_R находится в состоянии "0". Для передачи данных может потребоваться несколько вызовов (программных циклов), в зависимости от объема передаваемых данных.

Если функциональный блок обнаружит состояние "1" на входе R, текущее задание на передачу прерывается и функциональный блок S_RCV переводится в свое исходное состояние. Прием будет выключен в течение всего времени, пока параметр R находится в состоянии "1". Когда вход R вновь возвращается в состояние "0", прерванный кадр сообщения принимается вновь с самого начала.

Адрес модуля ET 200S SI, к которому происходит обращение, указывается в параметре LADDR.

Выход NDR указывает, что задание было завершено без ошибок (все данные прочитаны). Выход ERROR указывает на наличие ошибки. В случае возникновения ошибки в выходном состоянии STATUS содержится соответствующий номер ошибки. Если приемный буфер заполнен более чем на 2/3, на выход STATUS поступает предупреждение при каждом вызове S_RCV, если выход ERROR не установлен в "1". При отсутствии ошибок или предупреждений в STATUS содержится "0".

Сигналы NDR и ERROR/STATUS также актуализируются при сбросе функционального блока S_RCV (параметр LEN == 16#00). В случае ошибки сбрасывается бит BR. Если работа блока завершается без ошибок, бит BR принимает состояние "1".

В таблице 3–15 показано представление функционального блока FB2 S_RCV в формате STL и LAD.

Примечание

В функциональном блоке S_RCV не предусмотрена проверка параметров. Если заданы недопустимые параметры, CPU переключается в режим STOP.

Инициализированное задание может быть принято модулем ET200S SI после переключения CPU из режима STOP в RUN только после того, как завершится процедура, предусмотренная в модуле ET200S при запуске CPU для функционального блока S_RCV.

Таблица 3–15 Представление FB2 S_RCV в формате STL и LAD

Представление в форме STL	Представление в форме LAD
CALL S_RCV, I_RCV	I_RCV
EN_R: =	
R: =	
LADDR: =	
DB_NO: =	
DBB_NO: =	
NDR: =	
ERROR: =	
LEN: =	
STATUS: =	

Примечание

Параметры EN и ENO отображаются только при графическом представлении блока (LAD или FBD). Для обработки данных параметров компилятор использует бит BR.

Бит BR устанавливается в состояние “1”, если работа блока была завершена без ошибок. В случае ошибки бит BR переводится в “0”.

Присвоения в области данных

Функциональный блок S_RCV работает с экземплярным DB (I_RCV). Номер DB указывается при вызове. Обращение к данным экземплярного DB невозможно.

В таблице 3–16 приводятся параметры FB2 S_RCV.

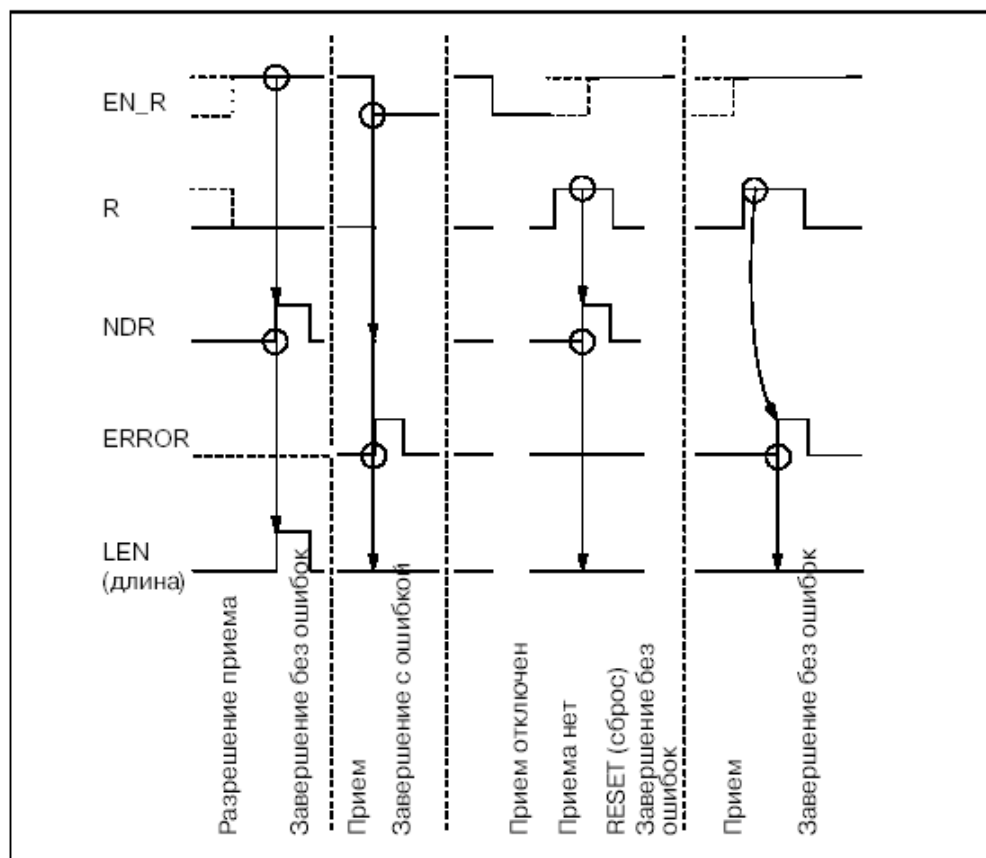
Примечание

Исключение: в случае ошибки, когда STATUS == W#16#1E0D, более подробные сведения об ошибке можно получить из переменной SFCERR. Эту переменную, содержащую сведения об ошибке, можно загрузить лишь с помощью символического доступа к экземплярному DB.

Таблица 3–16 FB2: параметры S_RCV				
Название	Тип	Тип данных	Описание	Допустимые значения, комментарии
EN_R	INPUT	BOOL	Разрешение чтения данных	
R	INPUT	BOOL	Прерывание задания	Текущее задание прерывается. Прием прекращается.
LADDR	INPUT	INT	Базовый адрес модуля ET 200S SI	Базовый адрес, взятый из STEP 7.
DB_NO	INPUT	INT	Номер блока данных	Номер принимаемого блока данных: зависит от CPU, "0" указывать не допускается
DBB_NO	INPUT	INT	Номер байта данных	$0 \leq \text{DBB_NO} \leq 8190$ принимаемые данные в формате слов
NDR ¹	OUTPUT	BOOL	Задание завершено без ошибок, данные приняты	Параметр STATUS == 16#00
ERROR ¹	OUTPUT	BOOL	Задание завершено с ошибками	Параметр STATUS содержит сведения об ошибке
LEN ¹	OUTPUT	INT	Длина принятого кадра сообщения	$1 \leq \text{LEN} \leq 200$, указывается количество байтов
STATUS ¹	OUTPUT	WORD	Описание ошибки	Если ERROR == 1, в параметре STATUS содержатся сведения об ошибке

¹ Параметр DONE доступен в течение **одного** цикла CPU после успешного выполнения задания.

Временная диаграмма для функционального блока S_RCV FB2



На рисунке 3-9 показана последовательность переключения параметров NDR, LEN и ERROR, в зависимости от подаваемых сигналов на входы EN_R и R.

Рисунок 3–9 Временная диаграмма для FB2 S_RCV

Примечание

Вход EN_R должен быть постоянно (статически) установлен в "1". В течение выполнения всего задания на прием для параметра EN_R необходимо, чтобы RLO (результат логической операции) был = "1".

3.4.3 Конфигурирование и настройка параметров Modbus-Master

Конфигурирование модуля Modbus

Если для связи с модулем по сети PROFIBUS используется ведущее устройство S7, следовательно, для конфигурирования модуля сети PROFIBUS и для настройки его параметров связи используется программа конфигурирования аппаратных средств пакета STEP 7.

Когда в Каталоге аппаратных средств выбран Modbus-Master (ведущее устройство Modbus), и модуль вставлен в базовую корзину ET 200S в конфигурации сети, в таблицу конфигурации автоматически вносится заказной номер модуля, номер установочного места (слота), а также адреса ввода/вывода. После этого можно вызвать диалоговое окно свойств Modbus-Master, выбрать режим связи и другие параметры.

Настройка параметров драйвера ведущего устройства

В таблице 3–17 приводятся параметры, которые можно выбрать для драйвера, функционирующего в режиме Modbus-Master (ведущее устройство Modbus).

Таблица 3–17 Параметры драйвера ведущего устройства Modbus			
Параметр	Описание	Диапазон установки	Значение по умолчанию
Diagnostic Alarm (Диагностическое сообщение)	Укажите, должен ли модуль генерировать диагностическое сообщение, если происходит серьёзная ошибка.	Нет Да	Нет
Interface Type (Тип интерфейса)	Выберите тип электрического интерфейса, который будет использоваться.	RS–232C RS–422 (дуплексный режим) RS–485 (полудуплексный режим)	RS–232C
Half–Duplex Receive Line Initial State (Исходное состояние линии для приёма в полудуплексном режиме)	Для режимов RS–422 и RS–485 укажите исходное состояние линии приёма. В режиме RS–232C не используется.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V
Data Flow Control (Управление потоком данных) (с принимаемыми по умолчанию параметрами; значения, принимаемые по умолчанию, можно изменить в программе пользователя)	Передачу и приём данных можно осуществлять с включённой функцией управления потоком данных, которая служит для синхронизации передачи данных в том случае, когда один коммуникационный партнёр работает быстрее другого. Выберите тип управления потоком данных и настройте соответствующие параметры. Примечание: управление потоком данных невозможно для интерфейса RS–485. Управление потоком данных "Automatic use of the V.24 Signals" ("Автоматическое использование сигналов V.24") возможно только для интерфейса RS–232C.	Нет Автоматическое использование сигналов V.24	Нет

Таблица 3–17 Параметры драйвера ведущего устройства Modbus (продолжение)

Параметр	Описание	Диапазон установки	Значение по умолчанию
Transmission Rate (Скорость передачи)	Укажите скорость передачи данных в бит/с.	110 300 600 1200 2400 4800 9600 19200 38400	9600
Stop Bits (Стоп-биты)	Выберите количество стоп-битов, которое будет добавляться к каждому символу, указывая на завершение передачи символа.	1 2	1
Parity (Чётность)	Последовательность передаваемых битов данных можно дополнить одним битом, что позволяет добавить бит чётности (паритета). Дополнительный бит (0 или 1) переводит общее значение всех битов (битов данных и бита чётности) в определенное состояние. Нет: данные передаются без бита чётности. Нечет: бит чётности устанавливается таким образом, чтобы общее количество битов данных (включая бит чётности) находящихся в состоянии "1", было нечетным. Чет: бит чётности устанавливается таким образом, чтобы общее количество битов данных (включая бит чётности), находящихся в состоянии "1", было четным.	None (Нет) Odd (Чёт) Even (Нечёт)	Even (Нечёт)
Response Time (Время отклика)	Время, в течение которого ожидается отклик от ведомого устройства.	50 мс...655000 мс	2000 мс
Operating Mode (Рабочий режим)	"Normal Operation" (“Нормальный режим”) "Interference Suppression" (“Подавление помех”)	Normal (Обычный) Interference Suppression (Подавление помех)	Normal (Обычный)
Character Delay Multiple (Коэффициент времени ожидания символа)	Множитель времени ожидания символа от 1 до 10.	от 1 до10	1
Delete Serial Interface Receive Buffer during Startup (Удалять буфер приёма ET 200S SI при запуске)	Укажите, должен ли буфер приема модуля автоматически очищаться, когда CPU переходит из режима STOP в режим RUN (запуск CPU). Это позволяет организовать работу таким образом, чтобы в буфер приема модуля поступали только те кадры сообщения, которые были	Нет Да	Да

	приняты после запуска CPU.		
--	----------------------------	--	--

- **Полудуплексный режим (RS422) с четырёхпроводным подключением** В таком режиме работы данные передаются по линии передачи T(A), T(B), а принимаются через линию приёма R(A), R(B). Обработка ошибок выполняется в соответствии с тем, какая функция выбрана в параметре “Рабочий режим драйвера” (“Обычный режим” или “Подавление помех”).
- **Полудуплексный режим (RS485) с двухпроводным подключением** В данном режиме работы драйвер переключает двухпроводную линию приёма R(A), R(B) из режима передачи в режим приёма и наоборот. При таком режиме работы все распознанные ошибки передачи, а также сигналы BREAK (ОСТАНОВ), поступающие до или после принимаемых сообщений, игнорируются. Уровень BREAK в перерыве между сообщениями также игнорируется. Начало сообщения, принимаемого от ведомого устройства, распознаётся по правильно принятому адресу ведомого устройства. Рекомендуется указывать в качестве начального состояния линии приёма состояние R(A) 0V, R(B) 5V (Высокий уровень).
- **Исходное состояние полудуплексной линии приёма**
С помощью данного параметра выбирается исходное состояние линии приёма для режимов RS-422 and RS-485. Он не используется для режима RS-232C.
Предварительная установка линии приёма:
Предварительная установка R(A) 5V, R(B) 0V (BREAK)
Исходное состояние двухпроводной линии R(A), R(B) устанавливается модулем следующим образом:
 $R(A) \rightarrow +5V, R(B) \rightarrow 0V \quad (V_A - V_B \geq +0.3V).$
Это означает, что в случае обрыва линии в модуле присутствует уровень BREAK.
Предварительная установка R(A) 0V, R(B) 5V (Высокий уровень)
Исходное состояние двухпроводной линии R(A), R(B) устанавливается модулем следующим образом:
 $R(A) \rightarrow 0V, R(B) \rightarrow +5V \quad (V_A - V_B \leq -0.3V).$
Это означает, что в случае обрыва линии (и/или когда модуль находится в режиме ожидания, т.е., когда ни одно из ведомых устройств не передаёт данные), в модуле присутствует уровень HIGH (высокий). Состояние линии BREAK не может быть распознано.
- **Скорость передачи.** Скорость передачи указывается как количество битов, передаваемых в секунду (бит/с). Максимальная скорость передачи модуля составляет 38400 бит/с в режиме полудуплексной связи.
- **Биты данных.** Данный параметр указывает, из какого количества битов будет состоять передаваемый символ. Следует всегда выбирать 8 битов. Всегда должен использоваться 11-битный кадр символа; если для чётности выбрано “нет”, необходимо выбрать 2 стоп-бита.
- **Стоп-биты.** Количество стоп-битов определяет наименьший возможный временной интервал между двумя передаваемыми символами. Всегда должен использоваться 11-битный кадр символа; если для чётности выбрано “нет”, необходимо выбрать 2 стоп-бита.
- **Чётность.** Бит чётности служит для повышения надёжности передачи данных. В зависимости от выбранного здесь параметра, передаваемый кадр дополняется до нечётного или чётного количества битов. Если для чётности выбрано “нет”, бит чётности не передаётся. В результате надёжность передачи данных снижается. Всегда должен использоваться 11-битный кадр символа; если для чётности выбрано “нет”, необходимо выбрать 2 стоп-бита.

- **Время отклика.** Контрольное время отклика – это время, в течение которого ведущее устройство ожидает ответное сообщение от ведомого устройства после передачи сообщения запроса.
- **Нормальная работа.** В данном режиме работы распознанные ошибки передачи и/или сигналы BREAK до или после принимаемых сообщений, поступающих от ведомого устройства, приводят к генерированию соответствующего сообщения об ошибке.
- **Подавление помех.** Если состояние BREAK распознано в линии приёма в начале принимаемого сообщения, либо если блок интерфейса модуля обнаруживает ошибки при передаче, драйвер считает принятое сообщение сбойным и игнорирует его. Начало сообщения, принимаемого от ведомого устройства, распознаётся по правильно принятому адресу ведомого устройства. Ошибки передачи и/или состояния BREAK также игнорируются, если они происходят в конце принимаемого сообщения (код CRC).
- **Коэффициент времени ожидания символа.** Если партнёр по связи не укладывается в требования ко времени, предъявляемые спецификацией Modbus, время ожидания символа t_{ZVZ} можно умножить на коэффициент f_{MUL} . Время ожидания символа следует регулировать только в том случае, если партнёр по связи не удовлетворяет требованиям, предъявляемым к данному параметру. В результате, время задержки символа t_{ZVZ} рассчитывается следующим образом:

$$t_{ZVZ} = t_{ZVZ_TAB} \cdot f_{MUL} ;$$

t_{ZVZ_TAB} : Табличное значение для t_{ZVZ}

f_{MUL} : Коэффициент

3.4.4 Коды функций, используемые Modbus-Master

В таблица 3–18 перечислены коды функций, которые поддерживаются драйвером ведущего устройства Modbus.

Таблица 3–18 Параметры драйвера ведущего устройства Modbus			
Код функции	Описание	Функция в SIMATIC S7	
01	В Modbus: чтение логической ячейки. Смысл: чтение выходного дискретного сигнала	Побитное чтение	Биты памяти M
		Побитное чтение	Выходы Q
		Побитное чтение (с интервалом в 16 битов)	Таймеры T
		Побитное чтение (с интервалом в 16 битов)	Счётчики C
02	Чтение дискретного входа	Побитное чтение	Биты памяти M
		Побитное чтение	Входы I
03	Чтение регистров хранения	Чтение по словам	Блок данных DB
04	Чтение входных регистров	Чтение по словам	Блок данных DB
05	В Modbus: изменение состояния логической ячейки Смысл: изменение выходного дискретного сигнала	Побитная запись	Биты памяти M
		Побитная запись	Выходы Q
06	Запись нового значения в регистр	Запись по словам	Блок данных DB

Таблица 3–18 Параметры драйвера ведущего устройства Modbus
(продолжение)

Код функции	Описание	Функция в SIMATIC S7	
07	Чтение состояния ошибок	Побитное чтение	Чтение 8-битного байта состояния
08	Проверка связи	–	–
11	Счётчик событий связи	Чтение 2-х слов	Состояние события и счётчик
12	Протокол событий связи	Чтение 70 байтов	Протокол событий
15	В Modbus: изменение состояния нескольких логических ячеек Смысл: изменение нескольких выходных дискретных сигналов	Побитная запись (1...2040 битов)	Биты памяти M
		Побитная запись (1...2040 битов)	Выходы Q
16	Запись новых значений в несколько регистров	Запись по словам (1...127 регистров)	Блок данных DB

3.4.5 Код функции 01 – Чтение логической ячейки

Функция	Данная функция служит для чтения отдельных битов ведомого устройства.
Начальный адрес	Адрес первого бита не проверяется драйвером и передаётся без изменений.
Количество битов	Любое значение от 1 до 2040 (бит = “логическая ячейка”).
LEN (байтов)	6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#1	Код функции
+2.0	Bit Start Address (Адрес первого бита)	WORD	W#16#0040	Адрес первого бита
+4.0	Bit Amount (Количество битов)	INT	16	Количество битов

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные [1])	WORD	W#16#1701	Данные

Драйвер записывает данные ответного сообщения в конечный блок данных по словам. Первый принятый байт записывается как младший байт первого слова "data[1]", третий принятый байт – как младший байт второго слова "data[2]" и т.д.

Если принято меньше 9-ти битов, либо принят только младший байт, в оставшийся старший байт последнего слова записывается значение 00H.

3.4.6 Код функции 02 – Чтение дискретного входа

Функция	Данная функция служит для чтения отдельных битов ведомого устройства.
Начальный адрес	Адрес первого бита не проверяется драйвером и передаётся без изменений.
Количество битов	Любое значение от 1 до 2040 (бит = "логическая ячейка").
LEN (байтов)	6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#2	Код функции
+2.0	Bit Start Address (Адрес первого бита)	WORD	W#16#0120	Адрес первого бита
+4.0	Bit Amount (Количество битов)	INT	24	Количество битов

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные [1])	WORD	W#16#2604	Данные
+2.0	data[2] (данные [2])	WORD	W#16#0048	Данные

Драйвер записывает данные ответного сообщения в конечный блок данных по словам. Первый принятый байт записывается как младший байт первого слова "data[1]", третий принятый байт – как младший байт второго слова "data[2]" и т.д. Если принято меньше 9-ти битов, либо прочитан только младший байт, в оставшийся старший байт последнего слова записывается значение 00H.

3.4.7 Код функции 03 – Чтение регистров хранения

Функция	Эта функция служит для чтения отдельных регистров ведомого устройства
Начальный адрес	Адрес первого регистра не проверяется драйвером и передаётся без изменений.
Количество регистров	Можно прочесть до 127 регистров (1 регистр = 2 байта).
LEN (байтов)	6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#3	Код функции
+2.0	Register Start Address (Адрес первого регистра)	WORD	W#16#0040	Адрес первого регистра
+4.0	Register Amount (Количество регистров)	INT	2	Количество регистров

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные [1])	WORD	W#16#2123	Данные
+2.0	data[2] (данные [2])	WORD	W#16#2527	Данные

3.4.8 Код функции 04 – Чтение входных регистров

Функция	Функция служит для чтения отдельных регистров ведомого устройства.
Начальный адрес	Адрес первого регистра не проверяется драйвером и передаётся без изменений.
Количество регистров	Можно прочесть до 127 регистров (1 регистр = 2 байта).

LEN
(байтов) 6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#4	Код функции
+2.0	Register Start Address (Адрес первого регистра)	WORD	W#16#0050	Адрес первого регистра
+4.0	Register Amount (Количество регистров)	INT	3	Количество регистров

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные[1])	WORD	W#16#2123	Данные
+2.0	data[2] (данные[2])	WORD	W#16#2527	Данные
+4.0	data[3] (данные[3])	WORD	W#16#3536	Данные

3.4.9 Код функции 05 – Изменение состояния логической ячейки

Функция Данная функция служит для установки или сброса отдельных битов ведомого устройства.

Адрес бита Адрес бита не проверяется драйвером и передаётся без изменений.

Состояние бита Для бита можно указать одно из следующих состояний:
 FF00H ⇒ бит установлен
 0000H ⇒ бит сброшен.

LEN 6
(байтов)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#5	Код функции
+2.0	Bit Address (Адрес бита)	WORD	W#16#0019	Адрес бита
+4.0	Bit State (Состояние бита)	WORD	W#16#FF00	Состояние бита

Ведомое устройство должно вернуть ведущему устройству сообщение запроса без каких-либо изменений.

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#5	Код функции
+2.0	Bit Address (Адрес бита)	WORD	W#16#0019	Адрес бита
+4.0	Bit State (Состояние бита)	WORD	W#16#FF00	Состояние бита

3.4.10 Код функции 06 – Запись нового значения в регистр

Функция Данная команда служит для записи нового значения в регистр ведомого устройства.

Адрес регистра Адрес регистра не проверяется драйвером и передаётся без изменений.

Значение регистра Можно записать любое значение.

LEN 6
(байтов)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#6	Код функции
+2.0	Reg Address (Адрес регистра)	WORD	W#16#0180	Адрес регистра
+4.0	Reg Value (Значение регистра)	WORD	W#16#3E7F	Значение регистра

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#6	Код функции
+2.0	Reg Address (Адрес регистра)	WORD	W#16#0180	Адрес регистра
+4.0	Reg Value (Значение регистра)	WORD	W#16#3E7F	Значение регистра

3.4.11 Код функции 07 – Чтение состояния ошибок

Функция Команда служит для чтения 8-ми битов событий подключенного ведомого устройства. Номер первого бита событий зависит от подключенного устройства, поэтому указывать его в программе пользователя SIMATIC не обязательно.

LEN 2
(байтов)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#7	Код функции

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные[1])	WORD	W#16#3Exx	Данные

Драйвер записывает отдельные биты ответного сообщения в старший байт конечного блока данных data[1]. Младший байт data[1] остаётся без изменений. Для длины указывается значение 1. Длина принимаемых данных всегда = 1.

3.4.12 Код функции 08 – Проверка связи

Функция Функция служит для проверки связи в установленных соединениях. Поддерживается только код диагностики 0000.

Код диагностики Единственное возможное значение кода диагностики - 0000.

Проверочное значение В качестве проверочного значения может использоваться любое число.

LEN (байтов) 6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#8	Код функции
+2.0	Diag Code (Код диагностики)	WORD	B#16#0000	Код диагностики
+4.0	Reg Value (Значение регистра)	WORD	B#16#A5C3	Проверочное значение

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#8	Код функции
+2.0	Diag Code (Код диагностики)	WORD	B#16#0000	Код диагностики
+4.0	Reg Value (Значение регистра)	WORD	B#16#A5C3	Проверочное значение

3.4.13 Код функции 11 – Чтение счётчика событий связи

Функция Данная функция служит для чтения слова состояния (2 байта), а также счётчика событий (2 байта) из ведомого устройства.

LEN 2
(байтов)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#0B	Код функции

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные[1])	WORD	W#16#FEDC	Слово состояния
+2.0	data[2] (данные[2])	WORD	W#16#0108	Счётчик событий

3.4.14 Код функции 12 – Чтение протокола событий связи

Функция Эта функция служит для чтения следующих данных:

- слово состояния (2 байта)
 - счётчик ошибок (2 байта)
 - счётчик сообщений (2 байта)
 - байты событий (64 байта)
- из ведомого устройства.

LEN 2
(байтов)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#0C	Код функции

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	data[1] (данные[1])	WORD	W#16#8765	Слово состояния
+2.0	data[2] (данные[2])	WORD	W#16#0108	Счётчик событий
+4.0	data[3] (данные[3])	WORD	W#16#0220	Счётчик сообщений
+6.0	bytedata[1] (байт данных[1])	BYTE	B#16#01	Байт событий 1
+7.0	bytedata[2] (байт данных[2])	BYTE	B#16#12	Байт событий 2
:	:			:
+68.0	bytedata[63] (байт данных[63])	BYTE	B#16#C2	Байт событий 63
+69.0	bytedata[64] (байт данных[64])	BYTE	B#16#D3	Байт событий 64

3.4.15 Код функции 15 – Изменение состояния нескольких логических ячеек

Функция	Данная функция служит для изменения до 2040 битов ведомого устройства.
Начальный адрес	Адрес первого бита не проверяется драйвером и передаётся без изменений.
Количество битов	Можно указать любое количество битов от 1 до 2040 (бит = “логическая ячейка”). Этот параметр указывает, какое количество битов ведомого устройства должно быть перезаписано. Параметр “Количество байтов” в сообщении запроса генерируется драйвером на основе передаваемого параметра “Количество битов”.
LEN (байтов)	>6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#0F	Код функции
+2.0	Bit Start Address (Адрес первого бита)	WORD	W#16#0058	Адрес первого бита
+4.0	Bit Amount (Количество битов)	INT	10	Количество битов
+6.0	coil_state[1] (состояние_бита[1])	WORD	W#16#EFCD	Состояние бита 5FH..58H/57H..50H

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#F	Код функции
+2.0	Bit Address (Адрес бита)	WORD	W#16#0058	Адрес бита
+4.0	Bit Amount (Количество битов)	INT	10	Количество битов

Драйвер передаёт данные из исходного блока данных по словам. Первым будет передан старший байт (байт 1) слова блока данных в позиции “EF”, после чего будет передан младший байт (байт 0) слова блока данных в позиции “CD”. Если передаётся нечётное количество байтов, последним байтом будет старший байт (байт 1).

3.4.16 Код функции 16 – Запись новых значений в несколько регистров

Функция	Данная функция служит для перезаписи до 127 регистров ведомого устройства за одно сообщение запроса.
Начальный адрес	Адрес первого регистра не проверяется драйвером и передаётся без изменений.
Количество регистров	Можно прочитать до 127 регистров (1 регистр = 2 байта). Параметр “Количество байтов” в сообщении запроса генерируется драйвером на основе передаваемого параметра “Количество регистров”.
LEN (байтов)	>6

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#10	Код функции
+2.0	Register Start Address (Адрес первого регистра)	WORD	W#16#0060	Адрес первого регистра
+4.0	Register Amount (Количество регистров)	INT	3	Количество регистров
+6.0	reg_data[1] (данные_регистра[1])	WORD	W#16#41A1	Данные регистра
+8.0	reg_data[2] (данные_регистра[2])	WORD	W#16#42A2	Данные регистра
+10.0	reg_data[3] (данные_регистра[3])	WORD	W#16#43A3	Данные регистра

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#10	Код функции
+2.0	Register Start Address (Адрес первого регистра)	WORD	W#16#0060	Адрес первого регистра
+4.0	Register Amount (Количество регистров)	INT	3	Количество регистров

3.5 Драйвер Modbus-Slave

3.5.1 Элементы канала связи Modbus-Slave (ведомое устройство Modbus)

Предоставляемый канал связи (data link) служит для доступа к определённым областям памяти CPU SIMATIC S7 по протоколу Modbus.

Структуры данных

Прежде чем приступить к конфигурированию структур данных S7 в проекте, необходимо проверить, совместимы ли они с программами пользователя, имеющимися в системах Modbus-Master (ведущих системах Modbus).

Канал связи Modbus-Slave (ведомое устройство Modbus)

Канал связи Modbus-Slave модуля состоит из двух компонентов:

- Драйвер ведомого устройства Modbus (Modbus-Slave)
- Функциональный блок протокола связи Modbus для CPU SIMATIC S7

Функциональный блок для обмена данными в режиме Modbus-Slave (ведомое устройство Modbus)

Помимо драйвера ведомого устройства Modbus каналу связи Modbus-slave требуется наличие в CPU S7 специального функционального блока обмена данными.

Функциональный блок протокола связи Modbus выполняет все функции, необходимые для канала связи.

Функциональный блок FB81 (S_MODB) принимает данные по протоколу Modbus и преобразует адреса Modbus в адреса областей памяти SIMATIC. FB81(S_MODB) должен вызываться циклически из программы пользователя. Функциональный блок протокола связи Modbus использует в качестве рабочей области экземплярный блок данных.

3.5.2 Передача данных ведомому устройству Modbus ET 200S

Запрос ведомого устройства Modbus осуществляется путем циклического вызова функционального блока S_MODB из программы пользователя. Запрос от модуля ET200S SI поступает в S_MODB, последний обрабатывает данный запрос и возвращает ответ модулю. Связь между CPU ПЛК и модулем реализуется с использованием функциональных блоков S_SEND и S_RCV, которые вызываются из S_MODB.

Всякий раз, когда CPU перезапускается, программа пользователя должна произвести инициализацию функционального блока протокола связи Modbus. Инициализация стартует по переднему фронту на входе START.

Функциональный блок записывает размеры областей операндов I, Q, M, T, и C модуля CPU в экземплярный блок данных функционального блока. Если инициализация завершается без ошибок, функциональный блок устанавливает выход START_OK.

Ошибка инициализации сигнализируется выходом START_ERROR. В этом случае обмен данными по Modbus невозможен, и на все запросы от ведущего устройства Modbus возвращается сообщение с кодом ошибки.

S_MODB использует Таблицу преобразования данных Modbus, расположенную в блоке данных, с целью приведения адресов Modbus к областям памяти ПЛК SIMATIC S7.

С помощью входного параметра OB_MASK в функциональном блоке Modbus можно включить режим маскирования ошибок обращения к вводу/выводу. В случае попытки записи по несуществующему адресу ввода/вывода CPU не переходит в режим STOP или не вызывает организационный блок (OB) ошибки. Ошибка обращения распознаётся функциональным блоком и функция завершается с возвратом сообщения об ошибке ведущему устройству Modbus.

Представление в формате STL	Представление в формате LAD
CALL S_MODB, I_MODB LADDR = START_TIMER = START_TIME = DB_NO = OB_MASK = START = START_FM = NDR = START_OK = START_ERROR = ERROR_NR = ERROR_INFO =	

Примечание

Параметры EN и ENO отображаются только в случае графического представления (LAD или FBD). Для обработки данных параметров компилятор использует бит BR. Бит BR устанавливается в состояние "1", если блок завершён без ошибок. Если произошла ошибка, BR сбрасывается в "0".

3.5.3 Области данных в CPU SIMATIC

Таблица преобразования данных Modbus

Функциональный блок FB81 (S_MODB) преобразует адреса Modbus, содержащиеся в сообщениях, к системе адресации, принятой в S7, то есть, приводит эти адреса к определённым областям памяти в SIMATIC. Обращение к отдельным областям памяти SIMATIC может быть указано пользователем путём подачи DB на вход FB81 (S_MODB) (см. таблицу 3–19).

Таблица 3–19 Таблица преобразования

Адрес	Название	Тип	Начальное значение	Фактическое значение	Комментарий	Код применяемой функции
0.0	aaaaa	WORD	W#16#0	W#16#0	Начальный адрес Modbus	01
2.0	bbbbb	WORD	W#16#0	W#16#7F7	Конечный адрес Modbus	
4.0	uuuuu	WORD	W#16#0	W#16#1F4	Память M	
6.0	ccccc	WORD	W#16#0	W#16#7F8	Начальный адрес Modbus	01
8.0	ddddd	WORD	W#16#0	W#16#FEF	Конечный адрес Modbus	
10.0	ooooo	WORD	W#16#0	W#16#15	Выход Q	
12.0	eeeeee	WORD	W#16#0	W#16#FF0	Начальный адрес Modbus	01
14.0	fffff	WORD	W#16#0	W#16#17E7	Конечный адрес Modbus	
16.0	ttttt	WORD	W#16#0	W#16#28	Память таймеров	

Таблица 3–19 Таблица преобразования (продолжение)

Адрес	Название	Тип	Начальное значение	Фактическое значение	Комментарий	Применяемый код функции
18.0	ggggg	WORD	W#16#0	W#16#17E8	Начальный адрес Modbus	01
20.0	hhhhh	WORD	W#16#0	W#16#1FDF	Конечный адрес Modbus	
22.0	zzzzz	WORD	W#16#0	W#16#28	Память счётчиков	
24.0	kkkkk	WORD	W#16#0	W#16#1FE0	Начальный адрес Modbus	02
26.0	lllll	WORD	W#16#0	W#16#27D7	Конечный адрес Modbus	02
28.0	vvvvv	WORD	W#16#0	W#16#320	Память M	02
30.0	nnnnn	WORD	W#16#0	W#16#27D8	Начальный адрес Modbus	02
32.0	rrrrr	WORD	W#16#0	W#16#2FCF	Конечный адрес Modbus	02
34.0	sssss	WORD	W#16#0	W#16#11	Вход I	02
36.0	DB_Number_FC_03_06_16	WORD	W#16#0	W#16#6	DB	03, 06, 13
38.0	DB_Number_FC_04	WORD	W#16#0	W#16#2	DB	04
40.0	DB_Min	WORD	W#16#0	W#16#1	Минимальный используемый номер DB	Пределы
42.0	DB_Max	WORD	W#16#0	W#16#6	Максимальный используемый номер DB	Пределы
44.0	M_Min	WORD	W#16#0	W#16#1F4	Минимальный используемый адрес памяти M	Пределы
46.0	M_Max	WORD	W#16#0	W#16#4B0	Максимальный используемый адрес памяти M	Пределы
48.0	Q_Min	WORD	W#16#0	W#16#0	Минимальный используемый адрес памяти Q	Пределы
50.0	Q_Max	WORD	W#16#0	W#16#64	Максимальный используемый адрес памяти Q	Пределы

3.5.4 Конфигурирование параметров канала связи

Для драйвера необходимо настроить следующие параметры и рабочие режимы, используя конфигуратор аппаратных средств.

- Скорость передачи, чётность
- Адрес модуля, являющегося ведомым устройством
- Рабочий режим (Нормальный, Подавление помех)
- Множитель для времени ожидания символа

Перечисленные ниже параметры следует настроить путём подачи DB на вход FB81 (S_MODB).

- Области адресов для кодов функции 01, 05, 15
- Области адресов для кода функции 02
- Номер базовой DB для кодов функции 03, 06, 16
- Номер базовой DB для кода функции 04
- Предельные значения для доступа в режиме “только для записи”

Настройка параметров драйвера ведомого устройства

В таблице 3–20 приводятся параметры, которые можно выбрать для модуля, работающего в режиме ведомого устройства Modbus.

Таблица 3–20 Параметры драйвера ведомого устройства Modbus			
Параметр	Описание	Диапазон значений	Значение по умолчанию
Диагностическое прерывание	Укажите, должен ли модуль генерировать диагностическое прерывание в случае возникновения серьезной ошибки.	Нет Да	Нет
Тип интерфейса	Укажите используемый интерфейс.	RS–232C RS–422 (дуплексный режим) RS–485 (полудуплексный режим)	RS–232C
Исходное состояние линии приема в полудуплексном режиме	Укажите исходное состояние линии приема для интерфейсов RS–422 и RS–485. Для интерфейса RS–232C не указывается.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V	R(A) 5V / R(B) 0V
Управление потоком данных (с параметрами, принимаемыми по умолчанию; изменение параметров, принимаемых по умолчанию, в программе пользователя)	Передачу и прием данных можно осуществлять с использованием функции управления потоком данных. Функция управления потоком данных используется для синхронизации передаваемых данных, если один из партнеров по обмену данными передает данные на большей скорости, чем другой партнер. Выберите тип управления потоком данных и задайте соответствующие параметры. Примечание: При использовании интерфейса RS 485 управление потоком данных невозможно. Автоматическое управление сигналами V.24 возможно только при использовании интерфейса RS–232C.	Нет Автоматическое функционирование сигналов V.24	Нет

Таблица 3–20 Параметры драйвера ведомого устройства Modbus (продолжение)

Скорость передачи	Выберите скорость передачи данных (бит/с).	110 300 600 1200 2400 4800 9600 19200 38400	9600
Стоп-биты	Выберите количество стоп-битов, которое будет добавляться к каждому передаваемому символу в процессе передачи для обозначения завершения символа.	1 2	1
Четность	Последовательность передаваемых битов данных можно дополнить одним символом, что позволяет добавить бит четности (паритета). Дополнительный бит (0 или 1) переводит общее значение всех битов (битов данных и бита четности) в определенное состояние. Нет: данные передаются без бита четности. Нечет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было нечетным. Чет: бит четности устанавливается таким образом, чтобы общее количество битов данных (включая бит четности), находящихся в состоянии "1", было четным.	None – нет Odd – нечет Even – чет	Even – чет
Адрес ведомого устройства	Собственный адрес модуля, выступающего в роли ведомого устройства	1–255	222
Режим работы	Нормальный режим Подавление помех	Нормальный Подавление помех	Нормальный режим
Множитель времени ожидания символа	Используется значение от 1 д 10.	1 ... 10	1
Очистка приемного буфера SI при запуске	Укажите, должен ли буфер приема модуля автоматически очищаться, когда CPU переходит из режима STOP в режим RUN (запуск CPU). Это позволяет организовать работу таким образом, чтобы в буфер приема модуля поступали только те кадры сообщения, которые были приняты после запуска CPU.	Нет Да	Да
¹ Наименьшее значение времени ожидания символа зависит от скорости передачи.			

- **Скорость передачи:** скорость передачи данных (количество битов в секунду (бит/с)). В полудуплексном режиме скорость передачи модуля составляет 38400 бит/с.
- **Количество битов данных:** количество битов в передаваемом символе. Для данного драйвера всегда следует выбирать 8 битов данных. Должен всегда использоваться кадр символа длиной 11 битов; если будет выбран параметр четности “нет”, следует выбрать два стоп-бита.
- **Количество стоп-битов:** количество стоп-битов задает наименьший возможный временной интервал между двумя соседними передаваемыми символами. Должен всегда использоваться кадр символа длиной 11 битов; если будет выбран параметр четности “Нет”, следует выбрать два стоп-бита.
- **Четность:** бит четности вводится для повышения надежности передачи данных; в зависимости от выполненных настроек, он дополняет передаваемые биты данных либо до четного, либо до нечетного количества битов в состоянии “1”. Если выбран параметр четности “Нет”, бит четности передаваться не будет. При этом надежность передачи снижается. Должен всегда использоваться кадр символа длиной 11 битов; если будет выбран параметр четности “Нет”, следует выбрать два стоп-бита.
- **Адрес ведомого устройства:** адрес ведомого устройства Modbus, на который должен откликаться модуль. Модуль отвечает только на те сообщения, которые содержат адрес ведомого устройства, совпадающий с адресом, установленным в этом параметре. Сообщения других ведомых устройств не интерпретируются, и ответ на них не возвращается.
- **Режим нормальной работы:** в этом режиме работы в ответ на все распознанные ошибки передачи и/или события BREAK, происходящие до или после приема сообщения от ведомого устройства, генерируется соответствующее сообщение об ошибке.
- **Подавление помех:** если в начале принимаемого сообщения в линии приема обнаружено событие BREAK, либо если интерфейсный блок модуля обнаружил ошибку передачи, драйвер считает принятое сообщение сбойным и игнорирует его. Признаком начала сообщения от ведомого устройства является адрес ведомого устройства, принятый без ошибок. Ошибки передачи и/или события BREAK также игнорируются, если они происходят после завершения принимаемого сообщения (код CRC).
- **Множитель времени ожидания:** если партнер по обмену данными не удовлетворяет требованиям к временным характеристикам, оговоренным в спецификациях Modbus, есть возможность увеличить время ожидания символа t_{ZVZ} , умножив его на множитель f_{MUL} . Время ожидания символа следует регулировать только в том случае, когда если партнер по обмену данными не укладывается в требуемое время. Конечное время ожидания символа t_{ZVZ} рассчитывается следующим образом:

$$t_{ZVZ} = t_{ZVZ_TAB} * f_{MUL} ;$$

$$t_{ZVZ_TAB} = \text{Табличное значение } t_{ZVZ} ;$$

f_{MUL} = Множитель.

3.5.5 Коды функций ведомого устройства

Драйвер Modbus-Slave (ведомое устройство Modbus) поддерживает коды функций, перечисленные в таблице 3–21.

Таблица 3–21 Коды функций ведомого устройства

Код функции	Описание	Функция в SIMATIC S7	
01	Чтение логической ячейки	Чтение отдельных битов	Биты памяти М
		Чтение отдельных битов	Выходы Q
		Чтение отдельных битов (с интервалом 16 битов)	Таймеры Т
		Чтение отдельных битов (с интервалом 16 битов)	Счетчики С
02	Чтение дискретного входа	Чтение отдельных битов	Биты памяти М
		Чтение отдельных битов	Входы I
03	Чтение регистров хранения	Чтение отдельных слов	Блок данных DB
04	Чтение входных регистров	Чтение отдельных слов	Блок данных DB
05	Изменение состояния логической ячейки	Запись отдельных битов	Биты памяти М
		Запись отдельных битов	Выходы Q
06	Запись нового значения в регистр	Запись отдельных слов	Блок данных DB
08	Проверка связи	–	–
15	Изменение состояния нескольких логических ячеек	Запись отдельных битов (1...2040 битов)	Биты памяти М
		Запись отдельных битов (1...2040 битов)	Выходы Q
16	Запись новых значений в несколько регистров	Запись отдельных слов (1...127 регистров)	Блок данных DB

Все адреса Modbus, перечисленные в таблице 3-21, относятся к уровню передачи сообщения, а не к уровню пользователя в ведущей системе Modbus. Это означает, что адреса Modbus в передаваемых сообщениях начинаются с 0000 Нех.

3.5.6 Код функции 01 – Чтение логической ячейки (выхода)

Функция Данная функция служит для чтения ведущей системой (ведущим устройством) Modbus отдельных битов из указанных ниже областей памяти SIMATIC.

Сообщение запроса

ADDR	FUNC	начальный_адрес	количество_битов	CRC
------	------	-----------------	------------------	-----

Ответное сообщение

ADDR	FUNC	начальный_адрес	n Байт ДАННЫХ	CRC
------	------	-----------------	---------------	-----

Длина в байтах LEN = 6

начальный_адрес

Параметр “начальный_адрес” интерпретируется драйвером, как адрес бита Modbus. Например, функциональный блок FB81(S_MODB) проверяет, находится ли данный “начальный_адрес” в пределах одной из областей, которые были указаны в DB преобразования для FC 01, 05, 15 (от /до : биты памяти, выходы, таймеры, счётчики).

Располагается ли начальный_адрес (адрес бита Modbus) в области	Обращение производится в следующую область памяти SIMATIC
aaaaa ... bbbbb	Начиная с бита памяти M uuuuu.0
ccccc ... ddddd	Начиная с выхода Q ooooo.0
eeeee ... fffff	Начиная с таймера T ttttt
ggggg ... hhhhh	Начиная со счётчика C zzzzz

Адрес обращения рассчитывается следующим образом (преобразование адреса):

Обращение в SIMATIC, начиная с	Формула преобразования
Байта памяти	$=((\text{начальный_адрес} - \text{aaaaa}) / 8) + \text{uuuuu}$
Выходного байта	$=((\text{начальный_адрес} - \text{ccccc}) / 8) + \text{ooooo}$
Таймера	$=((\text{начальный_адрес} - \text{eeeeee}) / 16) + \text{ttttt}$
Счётчика	$=((\text{начальный_адрес} - \text{ggggg}) / 16) + \text{zzzzz}$

Обращение к битам памяти и выходам

При обращении к битам памяти и выходам областей SIMATIC рассчитывается Количество_оставшихся_битов, которое используется для обращения к соответствующему биту первого/последнего байта памяти или выходного байта.

Обращение к таймерам и счётчикам

При вычислении адреса получившееся число:

- (начальный_адрес – eeeee) или
- (начальный_адрес– ggggg)

должно делиться на 16 без остатка (обращение только к отдельному слову целиком, позиции слов фиксированы).

количество_битов

“Количество_битов” (количество логических ячеек) можно задать в диапазоне 1 ... 2040. Именно такое количество битов будет прочитано.

При обращении к областям таймеров и счётчиков SIMATIC

“количество_битов” должно делиться на 16 (обращение только к слову целиком).

Пример использования

Пример преобразования адресов Modbus:

Преобразование адресов Modbus для функций FC 01, 05, 15			
Адрес Modbus в передаваемом сообщении		Область памяти SIMATIC	
от	0 до 2047	Начиная с бита	M 1000.0
от	2048 до 2559	Начиная с выхода	Q 256.0
от	4096 до 4607	Начиная с таймера	T 100
от	4608 до 5119	Начиная со счётчика	C 200

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#1	Код функции
+2.0	Bit Start Address (Адрес первого бита)	WORD	W#16#0040	Адрес первого бита
+4.0	Bit Amount (Количество битов)	INT	16	Количество битов

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Data[1] (Данные [1])	WORD	W#16#1701	Данные

Драйвер записывает данные ответного сообщения в конечный блок данных (DB) отдельно по одному слову. Первый принятый байт размещается в младший байт первого слова Data[1], третий принятый байт – в младший байт второго слова Data[2] и т.д. Если принято меньше 9-ти битов или прочитан только младший байт, в старший байт последнего слова записывается значение 00H.

Вычисление адреса:

Начальный_адрес (адрес Modbus) 0040 Нех (64 десят.) находится в области битов памяти:

Байт памяти	$=((\text{начальный_адрес} - \text{aaaaa}) / 8) + \text{uuuuu}$
	$=((64 - 0) / 8) + 1000$
	$=1008;$

Количество_оставшихся_битов равно:

Количество_оставшихся_битов	$=((\text{начальный_адрес} - \text{aaaaa}) \% 8)$	[По модулю 8]
	$=((64 - 0) \% 8)$	
	$= 0;$	

Обращение осуществляется к области от бита М 1008.0 до бита М 1011.7 включительно.

Количество битов:

Количество битов Modbus ("количество_битов"), равное 0020 Нех (32 десят.), означает, что необходимо прочитать 32 бита (= 4 байта).

В таблице 3–22 приводятся дополнительные примеры обращения к данным.

Таблица 3–22 Дополнительные примеры обращения к данным

начальный_адрес Нех Десят.		Расчёт адреса	Адрес
0000	0	Бит памяти $((0 - 0) / 8) + 1000$	→M1000.0
0021	33	Бит памяти $((33 - 0) / 8) + 1000$	→M1004.1
0400	1024	Бит памяти $((1024 - 0) / 8) + 1000$	→M1128.0
0606	1542	Бит памяти $((1542 - 0) / 8) + 1000$	→M1192.6
0840	2112	Выход $((2112 - 2048) / 8) + 256$	→Q264.0
09E4	2532	Выход $((2532 - 2048) / 8) + 256$	→Q316.4
1010	4112	Таймеры $((4112 - 4096) / 16) + 100$	→T 101
10C0	4288	Таймеры $((4288 - 4096) / 16) + 100$	→T112
1200	4608	Счётчики $((4608 - 4608) / 16) + 200$	→C200
13E0	5088	Счётчики $((5088 - 4608) / 16) + 200$	→C230

3.5.7 Код функции 02 – Чтение дискретного входа

Функция Функция служит для чтения ведущей системой (ведущим устройством) Modbus отдельных битов из указанных ниже областей памяти SIMATIC.

Сообщение запроса

ADDR	FUNC	начальный_адрес	количество_битов	CRC
------	------	-----------------	------------------	-----

Ответное сообщение

ADDR	FUNC	количество_байтов n	n Байт ДАННЫХ	CRC
------	------	---------------------	---------------	-----

Длина в байтах LEN = 6

начальный_адрес

Адрес бита Modbus, заданный параметром “начальный_адрес”, интерпретируется драйвером следующим образом:

Драйвер проверяет, находится ли начальный_адрес в пределах одной из следующих областей, которые были заданы в блоке данных преобразования для FC 02 (от/до : биты памяти, входы).

Располагается ли начальный_адрес (адрес бита Modbus) в области	Обращение производится в следующую область памяти SIMATIC	
От kkkkk до lllll	Начиная с бита памяти	M vvvvv.0
От nnnnn до rrrrr	Начиная со входа	I sssss. 0

Адрес обращения рассчитывается следующим образом (преобразование адреса):

Обращение в SIMATIC, начиная с	Формула преобразования
Байта памяти	$=((\text{начальный_адрес} - kkkkk) / 8) + vvvvv$
Входного байта	$=((\text{начальный_адрес} - nnnnn) / 8) + sssss$

Обращение к битам памяти и входам

При обращении к областям битов памяти и входам SIMATIC рассчитывается Количество_оставшихся_битов, которое используется для обращения к соответствующему биту первого/последнего байта памяти или входного байта.

количество_битов

Можно указать количество_битов (количество логических ячеек) от 1 до 2040. Именно такое количество битов будет прочитано.

Пример применения

Пример преобразования адреса Modbus:

Преобразование адреса Modbus для функции FC 02			
Адрес Modbus в передаваемом сообщении		SIMATIC memory area	
От 0 до 4095		Начиная с бита памяти	M 2000.0
От 4096 до 5119		Начиная со входа	I 128.0

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#2	Код функции
+2.0	Bit Start Address (Адрес первого бита)	WORD	W#16#0120	Адрес первого бита
+4.0	Bit Amount (Количество битов)	INT	24	Количество битов

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Data[1] (Данные [1])	WORD	W#16#2604	Данные
+2.0	Data[2] (Данные [2])	WORD	W#16#0048	Данные

Драйвер записывает данные ответного сообщения в конечный блок данных отдельно по одному слову. Первый принятый байт записывается в младший байт первого слова Data[1], третий принятый байт – в младший байт второго слова Data[2] и т.д.

Если принято меньше 9-ти битов или прочитан только младший байт, в старший байт последнего слова записывается значение 00H.

Вычисление адреса:

Начальный_адрес (адрес Modbus) 1030 Hex (4144 десят.) находится в области входов:

Входной байт	$=((\text{начальный_адрес} - nnnnn) / 8) + sssss$
	$=((4144 - 4096) / 8) + 128$
	$=134;$

Количество_оставшихся_битов равно:

Количество_оставшихся_битов	$=((\text{начальный_адрес} - aaaaa) \% 8)$	[По модулю 8]
	$=((4144 - 4096) \% 8)$	
	$= 0;$	

Обращение выполняется к области от входа I 134.0 до входа I 136.7 включительно.

Количество битов:

Количество битов Modbus ("количество_битов") 0018 Hex (24 десят.) означает, что необходимо прочитать 24 бита (= 3 байта).

В таблице 3–23 приводятся дополнительные примеры обращения к данным.

Таблица 3–23 Дополнительные примеры обращения к данным

начальный_адрес Hex Десятичный		Расчёт адреса	Адрес
0000	0	Бит памяти $((0 - 0) / 8) + 2000$	→M2000.0
0071	113	Бит памяти $((113 - 0) / 8) + 2000$	→M2014.1
0800	2048	Бит памяти $((2048 - 0) / 8) + 2000$	→M2256.0
0D05	3333	Бит памяти $((3333 - 0) / 8) + 2000$	→M2416.5
1000	4096	Вход $((4096 - 4096) / 8) + 128$	→I 128.0
10A4	4260	Вход $((4260 - 4096) / 8) + 128$	→I 148.4

3.5.8 Код функции 03 – Читать выходные регистры

Функция Функция служит для чтения ведущей системой (ведущим устройством) Modbus слов данных из блока данных.

Сообщение запроса

ADDR	FUNC	начальный_регистр	количество_регистров	CRC
------	------	-------------------	----------------------	-----

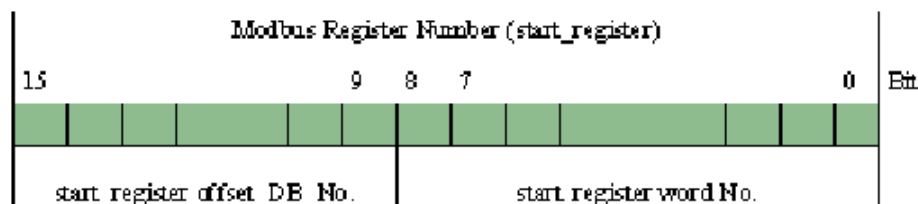
Ответное сообщение

ADDR	FUNC	количество_байтов n	n/2 – ДАННЫЕ регистра (старш. байт, младш. байт)	CRC
------	------	------------------------	---	-----

Длина в байтах 6

начальный_регистр

Адрес регистра Modbus (параметр “начальный_регистр”) интерпретируется драйвером следующим образом:



Для дальнейшей генерации адресов в функциональном блоке FB81(S_MODB) используется номер базового блока данных (начиная с DB xxxxx), введенный в блоке данных преобразования для FC 03, 06, 16.

Адрес обращения рассчитывается в два действия следующим образом (преобразование адреса):

Обращение к SIMATIC	Формула расчёта
Блок данных (конечный DB)	= (номер базового DB xxxxx + номер_DB (смещение от начального_регистра))
Слово данных DBW	= (номер_слова начального_регистра * 2

Формула расчёта для параметра “начальный_регистр”

Предположим, что номер конечного DB, который должен быть прочитан, нам известен, тогда адрес Modbus (параметр “начальный_регистр”), который требуется ведущей системе, можно рассчитать по следующей формуле:

$$\text{начальный_регистр} = ((\text{конечный DB} - \text{номер базового DB}) * 512) + (\text{слово данных_DBW} / 2)$$

В расчёте используются только слова, номера которых являются чётными.

количество_регистров

Можно указать любое “количество_регистров” от 1 до 127. Именно такое количество регистров будет прочитано.

Пример применения

Преобразование адреса Modbus для функций FC 03, 06, 16	
Адрес Modbus в передаваемом сообщении	Область памяти SIMATIC
0	Начиная с блока данных DB 800 (номер базового DB)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#3	Код функции
+2.0	Register Start Address (Адрес начального регистра)	WORD	W#16#0040	Адрес начального регистра
+4.0	Register Amount (Количество регистров)	INT	2	Количество регистров

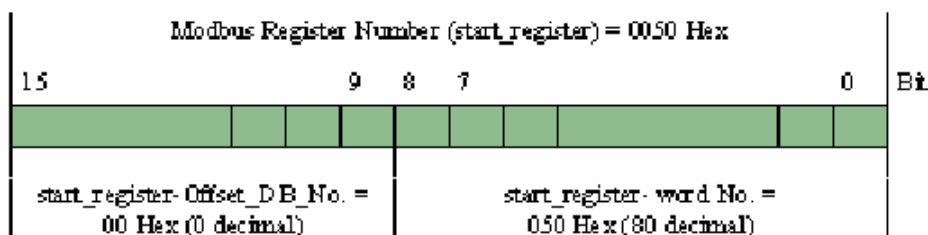
Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Data[1] (Данные [1])	WORD	W#16#2123	Данные
+2.0	Data[2] (Данные [2])	WORD	W#16#2527	Данные

Расчёт адреса:

Адрес Modbus ("начальный_регистр") 0050 Hex (80 десят.) интерпретируется следующим образом:



Блок данных DB (конечный DB)	= (номер базового DB xxxxx + номер_DB (смещение от начального_регистра)) =(800 + 0) =800;
------------------------------	--

Слово данных DBW	=(номер_слова начального_регистра * 2) =(80 * 2) =160;
------------------	--

Обращение выполняется к блоку данных DB 800, слову данных DBW 160.

Количество регистров:

Количество регистров Modbus ("количество_регистров") 0002 Hex (2 десят.) означает, что будет прочитано 2 регистра = 2 слова данных.

В таблице 3–24 приводятся дополнительные примеры обращения к данным.

Таблица 3–24 Дополнительные примеры обращения к данным

начальный_регистр		Номер базового DB	начальный_регистр		Конечный DB		DBW
			Номер_DB (смещение)	Номер слова			
Hex	Десят.	Десят.	Десят.	Hex	Десят.	Десят.	Десят.
0000	0	800	0	000	0	800	0
01F4	500	800	0	1F4	500	800	1000
0200	512	800	1	000	0	801	0
02FF	767	800	1	0FF	255	801	510
0300	768	800	1	100	256	801	512
03FF	1023	800	1	1FF	511	801	1022
0400	1024	800	2	000	0	802	0

3.5.9 Код функции 04 – Читать входные регистры

Функция Функция служит для чтения ведущей системой (ведущим устройством) Modbus слов данных из блока.

Сообщение запроса

ADDR	FUNC	начальный_регистр	количество_регистров	CRC
------	------	-------------------	----------------------	-----

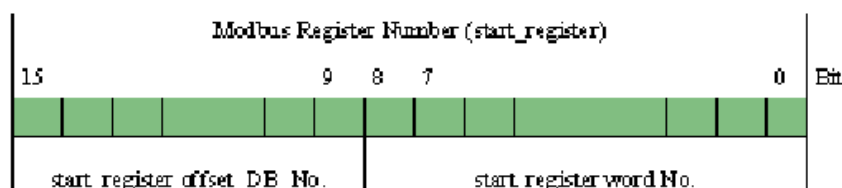
Ответное сообщение

ADDR	FUNC	количество_байтов n	n/2-ДАнные Регистров (Старший, Младший)	CRC
------	------	---------------------	---	-----

Длина в байтах 6

начальный_адрес

Адрес регистра Modbus, указанный в параметре “начальный_регистр”, интерпретируется драйвером следующим образом:



Для дальнейшей генерации адресов в функциональном блоке FB81(S_MODB) используется номер базового DB (начиная с DB xxxxx), введённый в блоке данных (DB) преобразования для FC 04.

Адрес обращения рассчитывается в два действия следующим образом (преобразование адреса):

Обращение к SIMATIC	Формула расчёта
Блок данных (конечный DB)	= (номер базового DB xxxxx + номер_DB (смещение от начального_регистра))
Слово данных DBW	= (начальный_регистр слово_№ * 2)

Предположим, что номер конечного DB, подлежащего чтению, нам известен, тогда адрес Modbus в параметре начальный_регистр, требующийся ведущей системе, можно рассчитать по следующей формуле:

$$\text{начальный_регистр} = ((\text{конечный DB} - \text{номер базового DB}) * 512) + (\text{слово данных_DBW} / 2)$$

В расчёте используются только слова с чётными номерами.

количество_регистров

Можно указать любое количество_регистров от 1 до 127. Именно такое количество регистров будет прочитано.

Пример применения

Преобразование адреса Modbus для функции FC 04	
Адрес Modbus в передаваемом сообщении	Область памяти SIMATIC
0	Начиная с блока данных DB 900 (номер базового DB)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#4	Код функции
+2.0	Register Start Address (Адрес начального регистра)	WORD	W#16#0050	Начальный адрес регистра
+4.0	Register Amount (Количество регистров)	INT	3	Количество регистров

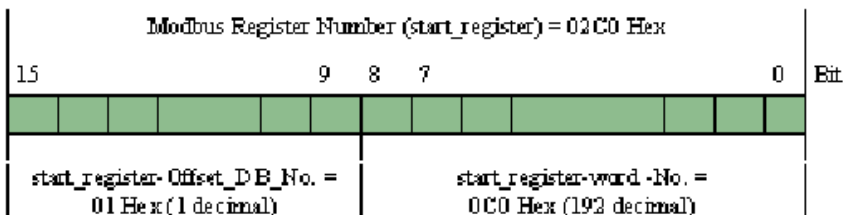
Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Data[1] (Данные[1])	WORD	W#16#2123	Данные
+2.0	Data[2] (Данные[2])	WORD	W#16#2527	Данные
+4.0	Data[3] (Данные[3])	WORD	W#16#3536	Данные

Расчёт адреса:

Адрес Modbus (параметр “начальный_регистр”) 02C0 Hex (704 десят.) интерпретируется следующим образом:



Блок данных DB (конечный DB)	=(номер базового DB xxxxx + номер_DB (смещение от начального_регистра)) =(900+ 0) =901;
------------------------------	--

Слово данных DBW	=(начальный_регистр слово_№* 2) =(192 * 2) =384;
------------------	--

Обращение выполняется к блоку данных DB 901, слову данных DBW 384.

Количество регистров:

Количество регистров Modbus (параметр “количество_регистров”), равное 0003 Hex (3 десят.), означает, что будет прочитано 3 регистра = 3 слова данных.

В таблице 3–25 приводятся дополнительные примеры обращения к данным.

Таблица 3–25 Дополнительные примеры обращения к данным

			начальный_регистр				
начальный_регистр		Номер базового DB	Номер_DB (смещение)	Номер слова		Конечный DB	DBW
Hex	Десят.	Десят.	Десят.	Hex	Десят.	Десят.	Десят.
0000	0	900	0	000	0	900	0
0064	100	900	0	064	100	900	200
00C8	200	900	0	0C8	200	900	400
0190	400	900	0	190	400	900	800
1400	5120	900	10	000	0	910	0
1464	5220	900	10	064	100	910	200
14C8	5320	900	10	0C8	200	910	400

3.5.10 Код функции 05 – Изменение состояния логической ячейки

Функция Данная функция служит для записи ведущей системой (ведущим устройством) Modbus отдельных битов в указанные ниже области памяти CPU SIMATIC.

Сообщение запроса

ADDR	FUNC	адрес_ячейки	БИТ_установка/сброс	CRC
------	------	--------------	---------------------	-----

Ответное сообщение

ADDR	FUNC	адрес_ячейки	БИТ_установка/сброс	CRC
------	------	--------------	---------------------	-----

Длина в байтах 6

адрес_катушки

Адрес бита Modbus, заданный в параметре “адрес_ячейки”, интерпретируется драйвером следующим образом:

Функциональный блок FB81 (S_MODB) проверяет, находится ли адрес_ячейки в пределах одной из областей, которые были введены в DB преобразования для FC 01, 05, 15 (от / до : биты памяти, выходы, таймеры, счётчики).

Располагается ли адрес бита Modbus (начальный_адрес) в области	Обращение выполняется в следующую область памяти SIMATIC	
от <i>aaaaa</i> до <i>bbbbbb</i>	начиная с бита памяти	M <i>iiii</i> .0
от <i>sssss</i> до <i>dddddd</i>	начиная с выхода	Q <i>oooo</i> .0

Адрес обращения рассчитывается в два действия следующим образом (преобразование адреса):

Обращения в SIMATIC	Формула преобразования
Байт памяти	$=((\text{начальный_адрес} - \text{sssss}) / 8) + \text{ooooo}$
Выходной байт	$=((\text{начальный_адрес} - \text{aaaaa}) / 8) + \text{iiiiii}$

Обращение к битам памяти и выходам

При обращении к областям битов памяти и выходам SIMATIC рассчитывается Количество_оставшихся_битов, которое используется для обращения к соответствующему биту первого/последнего байта памяти или выходного байта.

Обращение к тамерам и счётчикам

Функция FC 05 не позволяет обращаться к областям таймеров и счётчиков SIMATIC, и в ответ на такое обращение драйвер возвращает сообщение об ошибке.

БИТ установка/сброс

Данный параметр может иметь одно из двух значений:

FF00H = установить бит.

0000H = сбросить бит.

Пример применения

Преобразование адреса Modbus для функций FC 01, 05, 15			
Адрес Modbus в передаваемом сообщении		Область памяти SIMATIC	
от 0	до 2047	Начиная с бита памяти	M 1000.0
от 2048	до 2559	Начиная с выхода	Q 256.0

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#5	Код функции
+2.0	Bit address (Адрес бита)	WORD	W#16#0019	Адрес бита
+4.0	Bit state (Состояние бита)	WORD	W#16#FF00	Состояние бита

Ведомое устройство должно вернуть в ответ сообщение ведущего устройства без изменения (Эхо).

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#5	Код функции
+2.0	Bit address (Адрес бита)	WORD	W#16#0019	Адрес бита
+4.0	Bit state (Состояние бита)	WORD	W#16#FF00	Состояние бита

Расчёт адреса:

Адрес Modbus (параметр "адрес_ячейки") 0809 Hex (2057 десят.) располагается в области выходов:

Выходной байт	$= ((\text{адрес_ячейки} - \text{ссссс}) / 8) + 00000$
	$= ((2057 - 2048) / 8) + 256$
	$= 257$

Количество_оставшихся_битов имеет следующее значение:

Количество_оставшихся_битов	$= ((\text{адрес_ячейки} - \text{ссссс}) \% 8)$	[По модулю 8]
	$= ((2057 - 2048) \% 8)$	
	$= 1 ;$	

Обращение выполняется к выходу Q 257.1.

Другие примеры

Другие примеры обращения к битам памяти и выходам смотрите, пожалуйста, в FC 01.

3.5.11 Код функции 06 – Запись нового значения в регистр

Функция Данная функция служит для записи ведущей системой (ведущим устройством) Modbus слова данных в блок данных CPU.

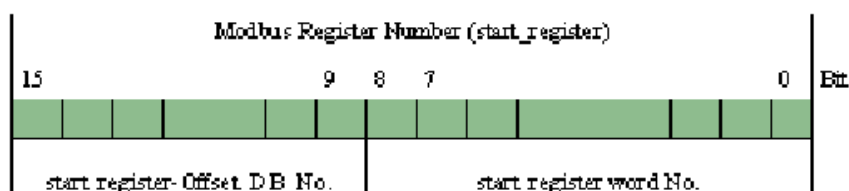
Сообщение запроса	ADDR	FUNC	начальный_регистр	ДАННЫЕ_значение (Старший, Младший)	CRC
--------------------------	------	------	-------------------	---------------------------------------	-----

Ответное сообщение	ADDR	FUNC	начальный_регистр	ДАННЫЕ_значение (Старший, Младший)	CRC
---------------------------	------	------	-------------------	---------------------------------------	-----

Длина в байтах 6

начальный_регистр

Адрес регистра Modbus, указанный в поле “начальный_регистр”, интерпретируется драйвером следующим образом:



Для дальнейшей генерации адресов в функциональном блоке FB81(S_MODB) используется номер базового DB (начиная с DB xxxxx), введённый в DB преобразования для FC 03, 06, 16.

Адрес обращения рассчитывается в два действия следующим образом (преобразование адреса):

Обращение к SIMATIC	Формула расчёта
Блок данных (конечный DB)	= (номер базового DB xxxxx + номер_DB (смещение от начального_регистра))
Слово данных DBW	= (начальный_регистр – слово_№*2)

Предположим, что номер конечного DB нам известен, тогда адрес Modbus (“начальный_регистр”), требующийся ведущей системе, можно рассчитать по следующей формуле:

$$\text{начальный_регистр} = ((\text{конечный DB} - \text{номер базового DB}) * 512) + (\text{слово данных_DBW} / 2)$$

В расчёте используются только слова, номера которых являются чётными.

ДАННЫЕ

Поле ДАННЫЕ может принимать любое значение (значение регистра).

Пример назначения параметров:

Преобразование адреса Modbus для функций FC 03, 06, 16	
Адрес Modbus в передаваемом сообщении	Область памяти SIMATIC
0	Начиная с блока данных DB 800 (номер базового DB)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#6	Код функции
+2.0	Reg Address (Адрес регистра)	WORD	W#16#0180	Адрес регистра
+4.0	Reg value (Значение регистра)	WORD	W#16#3E7F	Значение регистра

Конечный блок данных

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#6	Код функции
+2.0	Reg Address (Адрес регистра)	WORD	W#16#0180	Адрес регистра
+4.0	Reg value (Значение регистра)	WORD	W#16#3E7F	Значение регистра

Расчёт адреса

Адрес Modbus (“начальный_регистр”) 0180 Hex (384 десят.) интерпретируется следующим образом:

Modbus Register Number (start_register) = 0180 Hex															
15						9	8	7						0	Bit
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
start_register- Offset_DB_No. = 00 Hex (0 decimal)								start_register- word_No. = 180 Hex (384 decimal)							

Блок данных DB =(номер базового DB xxxxx + номер_DB (смещение от начального_регистра))
 =(800 + 0)
 = 800 ;

Слово данных DBW =(начальный_регистр–слово_No* 2)
 =(384 * 2)
 = 768 ;

Обращение выполняется к DB 800, слову данных DBW 768.

Дополнительные примеры

Дополнительные примеры обращения к данным смотрите, пожалуйста, в FC 03.

3.5.12 Код функции 08 – Проверка связи

Функция Функция служит для проверки связи. Она не оказывает воздействия на CPU S7, программу пользователя или данные пользователя. Принятое сообщение возвращается драйвером в ведущую систему (ведущему устройству) независимо от других событий.

Сообщение запроса	ADDR	FUNC	Диагностический Код (Старший, Младший)	Проверочные данные	CRC
--------------------------	------	------	--	--------------------	-----

Ответное сообщение	ADDR	FUNC	Диагностический Код (Старший, Младший)	Проверочные данные	CRC
---------------------------	------	------	--	--------------------	-----

Диагностический код Поддерживается только диагностический код 0000.

Проверочные данные Любое значение (16 бит).

Длина в байтах 6

Пример применения

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#8	Код функции
+2.0	Diag Code (Диагностический код)	WORD	B#16#0000	Диагностический код
+4.0	Reg Value (Значение регистра)	WORD	B#16#A5C3	Проверочное значение

Конечный блок данных RCV

Содержимое конечной области RCV:

Адрес	Название	Тип	Фактическое значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#8	Код функции
+2.0	Diag Code (Диагностический код)	WORD	B#16#0000	Диагностический код
+4.0	Test Value (Проверочное значение)	WORD	B#16#A5C3	Проверочное значение

3.5.13 Код функции 15 – Изменение состояния нескольких логических ячеек

Функция Данная функция служит для записи ведущей системой (ведущим устройством) Modbus нескольких битов в перечисленные ниже области памяти SIMATIC.

Сообщение запроса	ADDR	FUNC	начальный_адрес	количество	количество_байтов n	n - ДАННЫЕ	CRC
--------------------------	------	------	-----------------	------------	---------------------	------------	-----

Ответное сообщение	ADDR	FUNC	начальный_адрес	количество			CRC
---------------------------	------	------	-----------------	------------	--	--	-----

Длина в байтах >6

начальный_адрес

Адрес бита Modbus, указанный в параметре “начальный_адрес”, интерпретируется драйвером следующим образом:

Функциональный блок FB81(S_MODB) проверяет, располагается ли “начальный_адрес” в одной из областей, которые были указаны в DB преобразования для FC 01, 05, 15 (от/до : биты памяти, выходы, таймеры, счётчики).

Располагается ли адрес бита Modbus (“начальный_адрес”) в области	Обращение происходит к следующей области памяти SIMATIC	
От <i>aaaaa</i> до <i>bbbbbb</i>	Начиная с бита памяти	M <i>uuuu</i> .0
От <i>sssss</i> до <i>dddddd</i>	Начиная с выхода	Q <i>ooooo</i> .0

Адрес обращения рассчитывается следующим образом (преобразование адреса):

Доступ в SIMATIC	Формула расчёта
Начиная с байта памяти	$=((\text{начальный_адрес} - \text{sssss}) / 8) + \text{ooooo}$
Начиная с выходного байта	$=((\text{начальный_адрес} - \text{aaaaa}) / 8) + \text{uuuuu}$

Обращение к битам памяти и выходам

При обращении к областям битов памяти и выходам SIMATIC рассчитывается Количество_оставшихся_битов, которое используется для обращения к соответствующему биту байта памяти или выходного байта.

Обращение к таймерам и счётчикам

Функция FC 15 не позволяет обращаться к областям таймеров и счётчиков SIMATIC, и на попытки такого обращения драйвер возвращает сообщение об ошибке.

Количество

Можно указать любое значение от 1 до 2040 (количество битов).

ДАННЫЕ

В поле ДАННЫЕ содержатся состояния битов (любые значения).

Пример применения

Преобразование адресов Modbus для функций FC 01, 05, 15				
Адрес Modbus в передаваемом сообщении		Область памяти SIMATIC		
От	0 до 2047	Начиная с бита памяти	M 1000.0	
От	2048 до 2559	Начиная с выхода	Q 256.0	

Операция

Ведущая система (ведущее устройство) Modbus собирается установить следующие состояния битов памяти M 1144.0 ... M 1144.7 и M 1145.0 ... M 1145.3:

Бит памяти	7	6	5	4	3	2	1	0	Бит
M 1144	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	

Бит памяти	7	6	5	4	3	2	1	0	Бит
M 1145	–	–	–	–	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	(Address) Адрес	BYTE	B#16#5	Адрес ведомого устройства
+1.0	(Function) Функция	BYTE	B#16#0F	Код функции
+2.0	(Bit Start Address) Адрес начального бита	WORD	W#16#0058	Адрес начального бита
+4.0	(Bit Amount) Количество битов	INT	10	Количество битов
+6.0	(coil_state[1]) состояние_ячейки [1]	WORD	W#16#EFCD	Состояние_ячейки 5FH..58H/57H..50H

Вычисление адреса

Адрес Modbus 0480 Hex (1152 десят.), содержащийся в параметре “адрес_ячейки”, располагается в области битов памяти:

Байт памяти	$=((\text{начальный_адрес} - \text{aaaaa}) / 8) + \text{uuuuu}$
	$=((1152 - 0) / 8) + 1000$
	$=1144;$

Количество_оставшихся_битов вычисляется следующим образом:

Количество_оставшихся_битов	$=((\text{начальный_адрес} - \text{aaaaa})) \% 8)$	[Modulo 8]
	$=((1152 - 0) \% 8)$	
	$= 0;$	

Обращение выполняется к битам памяти, начиная с М 1144.0.

Другие примеры

Дополнительные примеры обращения к битам памяти и выходам смотрите в разделе, посвященном FC 01.

3.5.14 Код функции 16 – Запись новых значений в несколько регистров

Функция

Данная функция используется для записи ведущей системой (ведущим устройством) Modbus нескольких слов данных в блок данных CPU SIMATIC.

Сообщение запроса

ADDR	FUNC	начальный_регистр	количество	количество байтов_п	ДАнные-п (Старший, Младший)	CRC

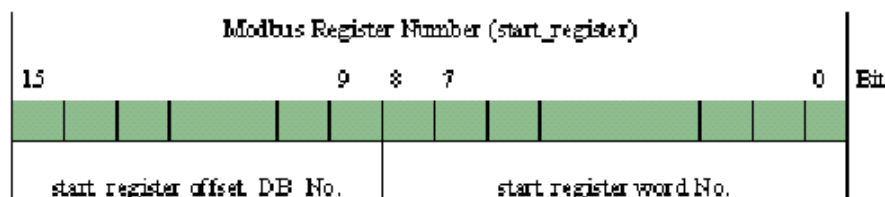
Ответное сообщение

ADDR	FUNC	начальный_регистр	количество	CRC

Длина в байтах >6

начальный_регистр

Адрес регистра Modbus в параметре “начальный_регистр” интерпретируется драйвером следующим образом:



Для дальнейшей генерации адресов в функциональном блоке FB81(S_MODB) используется номер базового DB (начиная с DB xxxxx), введенный в DB преобразования FC 03, 06, 16 при назначении параметров. Адрес обращения рассчитывается (преобразование адреса) в два действия следующим образом:

Обращение к SIMATIC	Формула расчета
Блок данных DB (конечный DB)	=(номер базового DB xxxxx + номер_DB (смещение от начального_регистра))
Слово данных DBW	=(начальный_регистр слово_№* 2)

При условии, что записываемый конечный блок данных (DB) известен, адрес Modbus в параметре “начальный_регистр”, необходимый для ведущей системы, можно рассчитать по следующей формуле:

$$\text{начальный_регистр} = ((\text{конечный DB} - \text{номер_базового DB}) * 512) + (\text{слово данных_DBW} / 2)$$

В расчете используются только слова данных с четными номерами.

Количество

Можно указать любое количество от 1 до 127 (количество регистров).

ДАННЫЕ (старший, младший байты)

В ДАННЫХ (старший, младший байт) может быть записано любое значение (значение регистра). Ведущая система (ведомое устройство) Modbus собирает записать значения CD09 Hex, DE1A Hex, EF2B Hex в слова данных DBW 100, DBW 102, DBW 104 блока данных DB 800.

Пример применения

Преобразование адресов Modbus для кодов функций FC03, 06, 16	
Адрес Modbus в передаваемом сообщении	Область памяти SIMATIC
0	начиная с блока данных DB 800 (номер базового DB)

Исходный блок данных SEND

Структура исходной области SEND:

Адрес	Название	Тип	Начальное значение	Комментарий
+0.0	Address (Адрес)	BYTE	B#16#5	Адрес ведомого устройства
+1.0	Function (Функция)	BYTE	B#16#10	Код функции
+2.0	Register Start Address (Адрес начального регистра)	WORD	W#16#0060	Адрес начального регистра
+4.0	Register Amount (Количество регистров)	INT	3	Количество регистров
+6.0	reg_data[1] данные_регистра[1]	WORD	W#16#41A1	Данные регистра
+8.0	reg_data[2] данные_регистра[2]	WORD	W#16#42A2	Данные регистра
+10.0	reg_data[3] данные_регистра[3]	WORD	W#16#43A3	Данные регистра

Расчет адреса

Адрес Modbus 0032 Hex (50 десят.) в параметре “начальный_регистр” интерпретируется следующим образом:

Modbus Register Number (start_register) = 0032 Hex															Bit
15						9	8	7						0	
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
start_register offset_DB_No. = 00 Hex (0 decimal)							start_register word No. = 32 Hex (50 decimal)								

Блок данных DB (конечный DB)	$= (\text{номер_базового DB } \text{xxxxx} + \text{номер_DB (смещение от начального_регистра)})$ $= (800 + 0)$ $= 800;$
------------------------------	--

Слово данных DBW	$= (\text{начальный_регистр слово_№.} * 2)$ $= (50 * 2)$ $= 100;$
------------------	---

Обращение выполняется к DB 800, слово данных DBW 100 .

Другие примеры

Другие примеры обращения к данным смотрите в разделе для FC 03.

3.5.15 Преобразования в бит-ориентированных функциях

Код функции 02

Бит-ориентированная функция 02 обеспечивает доступ к битам областей памяти и входам SIMATIC в режиме “только для чтения”.

Чтобы указать начальный и конечный адреса Modbus, в пределах которых будет осуществляться доступ к битам памяти и входам, можно использовать блок данных (DB) преобразования. Более того, можно назначить определенный элемент данных, начиная с которого будет осуществляться обращение к области памяти SIMATIC.

Области адресов Modbus и области адресов SIMATIC в FC 02 можно выбирать независимо от аналогичных областей в FC 01, 05 и 15.

Таблица 3–26 Области адресов		
Адрес Modbus в передаваемом сообщении		Область памяти SIMATIC
От kkkkk	Биты памяти	Начиная с M vvvv.0
До lllll		
От nnnnn	Входы	Начиная с I sssss.0
До rrrrr		

3.5.16 Преобразования в функциях, ориентированных на регистры

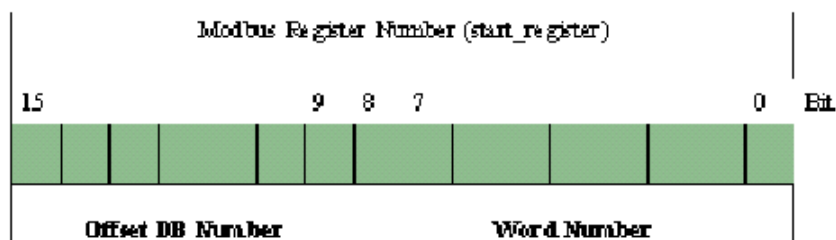
Коды функций 03, 06, 16

Функции 03, 06 и 16, предназначенные для работы с регистрами, позволяют производить запись и чтение блоков данных областей памяти SIMATIC.

Расчет требуемого номера блока данных выполняется в два действия.

1) Чтобы указать номер базового DB, можно использовать инструмент назначения параметров. Этим базовым DB является первый DB, к которому возможен доступ.

2) Адрес начального регистра Modbus (номер регистра), передаваемый в сообщении, интерпретируется следующим образом:



Номер адресуемого DB

Конечный (расчетный) номер блока данных, к которому будет выполняться обращение, рассчитывается следующим образом:

Номер базового DB + смещение от базового DB.

Это означает, что к области блока данных, состоящей из 128 последовательных блоков данных, можно обращаться в пределах всей адресуемой области блока данных (65535 блоков данных).

Номер слова в DB

Номер слова позволяет обращаться к области от DBW 0 до DBW 1022 в пределах каждого блока данных.

В этом случае драйвер обращается к отдельным словам блока данных, который, как правило, состоит из отдельных байтов.

Код функции 04

Функция 04, ориентированная на работу с регистрами, позволяет обращаться к блокам данных области памяти SIMATIC в режиме “только для чтения”.

Режим и осуществление такого доступа описане в разделе для кодов функций 03, 06 и 16.

Функции 04 принадлежит собственный базовый DB со своим номером, доступный для свободного параметрирования с помощью блока данных (DB) преобразования. С его помощью можно выбрать вторую независимую область, состоящую из 128-ми блоков данных.

Эти блоки данных также могут быть доступны в режиме “только для чтения”.

3.5.17 Разрешение/запрещение записи

Коды функций 05, 06, 15, 16

Для функций 05, 06, 15 и 16, предназначенных для записи, можно запретить или ограничить доступ к определенным областям памяти SIMATIC.

Чтобы указать область, в которую возможна запись ведущей системой (ведомым устройством) Modbus, можно использовать блок данных (DB) преобразования.

Если ведущее устройство пытается обратиться в любую из областей памяти SIMATIC, запись в которую не разрешена, в доступе будет отказано. При этом будет выдано сообщение об ошибке (исключение). В таблице 5–9 показано разрешение доступа для записи.

Таблица 3–27 Разрешение доступа для записи

38.0	DB_Number_ FC_04	WORD	W#16#0	W#16#2	DB	04
40.0	DB_Min	WORD	W#16#0	W#16#1	Мин. использ. номер DB	Предельные значения
42.0	DB_Max	WORD	W#16#0	W#16#6	Макс. использ. номер DB	
44.0	M_Min	WORD	W#16#0	W#16#1F4	Мин. использ. адрес памяти M	
46.0	M_Max	WORD	W#16#0	W#16#4B 0	Макс. использ. адрес памяти M	
48.0	Q_Min	WORD	W#16#0	W#16#0	Мин. использ. адрес выхода Q	
50.0	Q_Max	WORD	W#16#0	W#16#64	Макс. использ. адрес выхода Q	

3.5.18 Преобразование адресов Modbus для бит-ориентированных функций

Коды функций 01, 05, 15

Бит-ориентированные функции 01, 05 и 15 позволяют производить как запись, так и чтение битов памяти, выходов, таймеров и счетчиков SIMATIC. FC01 позволяет обращаться к таймерам и счетчикам только для чтения.

Чтобы указать начальный и конечный адреса Modbus, в пределах которых должно выполняться обращение к битам памяти, выходам, таймерам и счетчикам, можно использовать блок данных преобразования. Кроме того, можно назначить определенный элемент данных, начиная с которого будет производиться обращение к области памяти SIMATIC.

Обзор функций FC 01, 05, 15

Преобразование адресов Modbus для FC 01, 05, 15			
Блок данных параметров		Введенный параметр	Значение
Биты в памяти SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От аaaa	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До bbbb	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC (биты памяти)	Начиная с M uuuu.0	от 0 до 65535 (десят.)	Начиная с данного байта памяти
Область выходов SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От cccc	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До dddd	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC <i>Выходы</i> (номер выходного байта)	Начиная с Q ooooo.0	от 0 до 65535 (десят.)	Начиная с данного выхода
Область таймеров SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От eeee	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До ffff	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC <i>Таймеры</i> (номер таймеров)	Начиная с по tttt	от 0 до 65535 (десят.)	Начиная с данного таймера (= 16-битное слово)
Область счетчиков SIMATIC			

Преобразование адресов Modbus для FC 01, 05, 15 (продолжение)

Адрес Modbus в передаваемом сообщении (номер бита)	От gggg	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До hhhh	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC Счетчики (количество счетчиков)	Начиная с C zzzzz	от 0 до 65535 (десят.)	Начиная с данного таймера (= 16-битное слово)

От/до адреса Modbus

Чтобы указать начало определенной области (адрес Modbus), можно использовать начальный адрес (От); например, начало области битов памяти, выходов и т. д.
(= номер первого бита области).

Чтобы указать конец определенной области (адрес Modbus), можно использовать конечный адрес (До); например, конечный адрес области битов памяти, выходов и т. д.
(= номер последнего бита области).

Адреса “от/до” соответствуют адресам Modbus в передаваемом сообщении (номера битов, начиная с 0) для функций FC 01, 05 и 15.

Отдельные области, определяемые адресами “от/до”, не должны перекрываться.

Между отдельными областями, определяемыми адресами “от/до”, допускаются промежутки.

Начиная с элемента области памяти SIMATIC

Параметр “Начиная с” можно использовать для указания начальной области SIMATIC, к которой будет приводиться область Modbus, определяемая адресами “от/до” (= первый байт памяти-, выходной байт-/ таймер-/ номер счетчика области SIMATIC).

Пример для FC 01, 05, 15

Преобразование адресов Modbus для FC 01, 05, 15			
Блок данных параметров		Введенный параметр	Значение
Биты памяти области SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От 0	от 0 до 65535(десят.)	Начиная с данного адреса Modbus
	До 2047	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC Биты памяти (биты памяти)	Начиная с M 1000.0	от 0 до 65535 (десят.)	Начиная с данного байта памяти
Область выходов SIMATIC			

Преобразование адресов Modbus для **FC 01, 05, 15** (продолжение)

Адрес Modbus в передаваемом сообщении (номер бита)	От 2048	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До 2559	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC <i>Выходы</i> (номер выходного байта)	Начиная с Q 256.0	от 0 до 65535 (десят.)	Начиная с данного выхода
Область таймеров SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От 4096	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До 4255	от 0 до 65535 (десят.)	Заканчивая данным адресом Modbus включительно
Область памяти SIMATIC <i>Таймеры</i> (номер таймера)	Начиная с T 100	от 0 до 65535 (десят.)	Начиная с данного таймера (= 16-битное слово)
Область счетчиков SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От 4256	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До 4415	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC <i>Счетчики</i> (номер счетчика)	Начиная с C 120	от 0 до 65535 (десят.)	Начиная с данного счетчика (= 16-битное слово)

Адреса Modbus от 0 до 2047 соответствуют битам памяти SIMATIC, начиная с бита памяти M 1000.0. Другими словами, длина области = 2048 битов = 256 байт, что означает, что последний бит памяти = M 1255.7.

Адреса Modbus от 2048 до 2559 соответствуют выходам SIMATIC, начиная с выхода Q 256.0. Другими словами, длина области = 512 битов = 64 байта, что означает, что последний бит выхода = Q 319.7.

Адреса Modbus от 4096 до 4255 соответствуют таймерам SIMATIC, начиная с таймера T 100. Другими словами, длина области = 160 битов = 10 слов, что означает, что последний таймер = T 109.

Адреса Modbus от 4256 до 4415 соответствуют счетчикам SIMATIC, начиная со счетчика C 120. Другими словами, длина области = 160 битов = 10 слов, что означает, что последний счетчик = C 129.

Обзор FC02

Преобразование адресов Modbus для FC 02			
Блок данных параметров		Введенный параметр	Значение
Биты области памяти SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC Биты памяти	Начиная с	от 0 до 65535 (десят.)	Начиная с данного байта памяти
Область входов SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC Входы (номер входного байта)	Начиная с	от 0 до 65535 (десят.)	Начиная с данного входного байта

Адрес Modbus “от/до”

Чтобы указать начало определенной области (адрес Modbus), например, области битов памяти или входов, можно использовать начальный адрес “от” (= номер первого бита области).

Чтобы указать конец определенной области, можно использовать конечный адрес “до” (= номер последнего бита области).

Адреса “от/до” соответствуют адресам Modbus в передаваемом сообщении (номера битов, начиная с 0) для функции FC 02.

Отдельные области “от/до” не должны перекрываться.

Между отдельными областями “от/до” допускаются промежутки.

Начиная с области памяти SIMATIC

Чтобы указать начало области SIMATIC, к которой будет приведена область Modbus, определяемая адресами “от/до”, можно использовать параметр “Начиная с” (= первый байт памяти–, номер входного байта области SIMATIC).

Пример для FC 02

Преобразование адресов Modbus для FC 02			
Блок данных параметров		Введенный параметр	Значение
Биты в памяти SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От 0	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До 4095	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC биты памяти	Начиная с M 0.0	от 0 до 65535(десят.)	Начиная с данного байта памяти
Область входов SIMATIC			
Адрес Modbus в передаваемом сообщении (номер бита)	От 4096	от 0 до 65535 (десят.)	Начиная с данного адреса Modbus
	До 5119	от 0 до 65535 (десят.)	По данный адрес Modbus включительно
Область памяти SIMATIC Входы (номер входного байта)	Начиная с I 128.0	от 0 до 65535 (десят.)	Начиная с данного входного байта

Адреса Modbus от 0 до 4095 соответствуют битам памяти SIMATIC, начиная с бита памяти M 0.0. При этом длина области = 4096 битов = 512 байтов, что означает, что последний бит памяти = M 511.7.

Адреса Modbus от 4096 до 5119 соответствуют входам SIMATIC, начиная со входа I 128.0. При этом длина области = 1024 битов = 128 байтов, что означает, что последний входной бит = I 255.7.

Примечание

Начальный элемент битов области памяти, указанный для FC02, не зависит от начального элементов битов области памяти для кодов функций 01, 05 и 15. Это означает, что с помощью функции 02 можно использовать вторую область битов памяти SIMATIC (только для чтения), которая не будет зависеть от первой.

3.5.19 Преобразование адресов Modbus для функций, предназначенных для регистров

Обзор FC 03, 06, 16

Преобразование адресов Modbus для FC 03, 06, 16			
Блок данных параметров		Введенный параметр	Значение
Блоки данных области SIMATIC			
Адрес Modbus = 0 в передаваемом сообщении (номер регистра) означает обращение к:			
Блоку данных области памяти SIMATIC	Начиная с блока данных	от 1 до 65535 (десят.)	Начиная с данного блока данных Начиная с DBW 0 (= номер базового блока данных)

начиная с DB

Чтобы указать первый блок данных области SIMATIC, к которому будет осуществляться доступ, можно использовать параметр “Начиная с DB” (= номер базового блока данных).

Обращение к данному DB, начиная со слова данных DBW 0, осуществляется, когда в сообщении Modbus указан номер регистра 0.

Соответственно, последующие номера регистров Modbus означают доступ к последующим словам данных/блокам данных.

Можно обратиться к 127 последовательным блокам данных.

Биты 9...15 в номере регистра Modbus интерпретируются драйвером для обращения к отдельным, последовательно расположенным блокам данных.

Пример применения

Преобразование адресов Modbus для FC 03, 06, 16			
Блок данных параметров		Введенный параметр	Значение
Блоки данных области SIMATIC			
Адрес Modbus = 0 в передаваемом сообщении (номер регистра) означает обращение к:			
Блоку данных области памяти SIMATIC	Начиная с DB 800	от 1 до 65535 (десят.)	Начиная с этого блока данных Начиная с DBW 0 (= номер базового блока данных)

Для обращения к блоку данных 800, начиная с DBW 0 в системе SIMATIC, можно использовать адрес регистра Modbus = 0.

Последующие адреса регистров Modbus (начиная с 512 и т.д.) означают обращение к последующим блокам данных: DB 801 и т.д.

Обзор FC 04

Преобразование адресов Modbus для FC 04			
Блок данных параметров		Введенный параметр	Значение
Блоки данных области SIMATIC			
Адрес Modbus = 0 в передаваемом сообщении (номер регистра) означает обращение к:			
Блоку данных области памяти SIMATIC	Начиная с DB	от 1 до 65535 (десят.)	Начиная с данного блока данных Начиная с DBW 0 (= номер базового блока данных)

Начиная с DB

Чтобы указать первый блок данных области SIMATIC, начиная с которого будет осуществляться доступ, можно использовать параметр “Начиная с DB” (= номер базового блока данных).

Обращение к данному DB, начиная со слова данных DBW 0, осуществляется, когда в сообщении Modbus указан номер регистра 0.

Соответственно, последующие номера регистров Modbus означают обращение к последующим словам данных/блокам данных.

Можно обратиться к 127 следующим друг за другом блокам данных.

Биты 9...15 в номере регистра Modbus интерпретируются драйвером для обращения к отдельным, расположенным последовательно блокам данных.

Примечание

Значение “Начиная с DB” указывается независимо от соответствующего значения для функций 03, 06 и 16. Это означает, что FC04 позволяет использовать вторую область блоков данных SIMATIC (только для чтения), которая совершенно не зависит от первой.

Пример преобразования для FC 04

Преобразование адресов Modbus для FC 04			
Блок данных параметров		Введенный параметр	Значение
Блоки данных области SIMATIC			
Адрес Modbus = 0 в передаваемом сообщении (номер регистра) означает обращение к:			
Блоку данных области памяти SIMATIC	Начиная с DB 1200	от 1 до 65535 (десят.)	Начиная с данного блока данных Начиная с DBW 0 (= номер базового блока данных)

Для обращения к блоку данных 1200, начиная с DBW 0 в системе SIMATIC, можно указать адрес регистра Modbus 0.

Последующие адреса регистров Modbus (512, 1024 и т. д.) означают обращение к последующим блокам данным, то есть, к блокам DB 1201, 1202 и т.д.

3.5.20 Ограничения для функций записи

Обзор FC 05, 06, 15, 16

Ограничения доступа для записи в SIMATIC (FC 05, 06, 16)			
Блок данных параметров		Введенный параметр	Значение
Блоки данных (DB: расчетный номер DB)	DB MIN	от 1 до 65535	Первый DB, в который разрешена запись
	DB MAX	от 1 до 65535	Последний DB, в который разрешена запись MAX=0: запись запрещена во все DB
Биты памяти M (номер байта памяти)	M MIN	от 0 до 65535	Первый байт памяти, в который разрешена запись
	M MAX	от 1 до 65535	Последний байт памяти, в который разрешена запись MAX=0: запись запрещена во все биты памяти
Выходы Q (номер байта выхода)	Q MIN	от 0 до 65535	Первый выходной байт, в который разрешена запись
	Q MAX	от 1 до 65535	Последний выходной байт, в который разрешена запись MAX=0: запись запрещена во все выходы

MIN/MAX область памяти SIMATIC

Для функций, осуществляющих запись, можно указать нижнюю и верхнюю границы области, в которую разрешена запись (MIN/MAX). Обращение для записи будет разрешено только к области, определяемой данными границами.

Если указано нулевое значение верхней границы, запись запрещена во всю область целиком.

При выборе размера области, в которую разрешается запись, следует учитывать, к какому CPU она относится.

Если ведущее устройство пытается производить запись в область, которая находится за пределами верхней/нижней границы, модуль отказывает в этой операции и выдает сообщение об ошибке.

Значения MIN/MAX для области блоков данных должны указываться как расчетные номера DB.

Пример применения для FC 05, 06, 16

Ограничения доступа для записи в SIMATIC (FC 05, 06, 16)			
Блок данных параметров		Введенный параметр	Значение
Блоки данных (DB: расчетный номер DB)	MIN 600	1 ... 65535	Первый DB, в который разрешена запись
	MAX 699	1 ... 65535	Последний DB, в который разрешена запись MAX=0: запись запрещена во все DB
Биты памяти M (номер байта памяти)	MIN 1000	0 ... 65535	Первый байт памяти, в который разрешена запись
	MAX 1127	1 ... 65535	Последний байт памяти, в который разрешена запись MAX=0: запись запрещена во все биты памяти
Выходы Q (номер байта выхода)	MIN 256	0 ... 65535	Первый выходной байт, в который разрешена запись
	MAX 319	1 ... 65535	Последний выходной байт, в который разрешена запись MAX=0: запись запрещена во все выходы

К блокам данных SIMATIC DB 600 ... DB 699 можно обращаться с помощью функций записи FC 06, 16.

К байтам памяти SIMATIC MB 1000 ... MB 1127 можно обращаться с помощью функций FC 05, 15.

К выходным байтам SIMATIC QB 256 ... QB 319 можно обращаться с помощью функций записи FC 05, 15.

3.6 Диагностика

3.6.1 Диагностическая информация, предоставляемая светодиодами состояния

На передней панели модуля последовательного интерфейса ET 200S SI Modbus/USS находятся следующие светодиоды состояния:

- **TX** (зеленый): светится, когда модуль ET 200S SI Modbus/USS передает данные через свой порт
- **RX** (зеленый): светится, когда модуль ET 200S SI Modbus/USS принимает данные через свой порт
- **SF** (красный): указывает, что произошла одна из следующих ошибок:
 - Аппаратная неисправность
 - Неисправность внутренней микропрограммы
 - Ошибка (ошибки) параметров
 - Оборван провод или отсоединился кабель между модулем ET 200S SI Modbus/USS и коммуникационным партнером: может быть обнаружено только при использовании интерфейса RS-422, для которого выбрано начальное состояние линии приема = R(A) 5V / R(B) 0V.
 - Ошибки связи (четность, ошибки кадров, переполнение буфера)

3.6.2 Структура диагностических сообщений функциональных блоков

В каждом функциональном блоке имеется параметр STATUS, служащий для диагностики ошибок. Независимо от используемого функционального блока, за сообщениями в слове STATUS закреплены одни и те же номера. Структура параметра STATUS показана на рисунке 3-10.

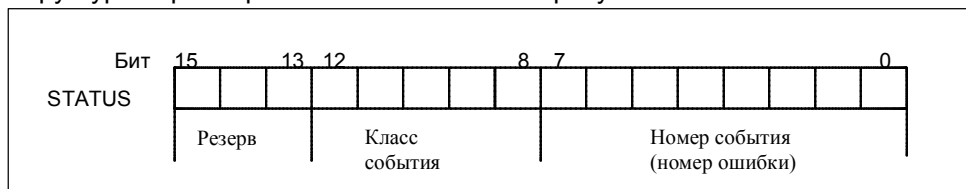


Рисунок 3–10 Структура параметра STATUS

Пример: на рисунке 3-10 показано содержимое параметра STATUS для события “Задание прервано в результате “холодного” перезапуска, “горячего” перезапуска или простого перезапуска” (класс события 1Е_н, номер события 0D_н).

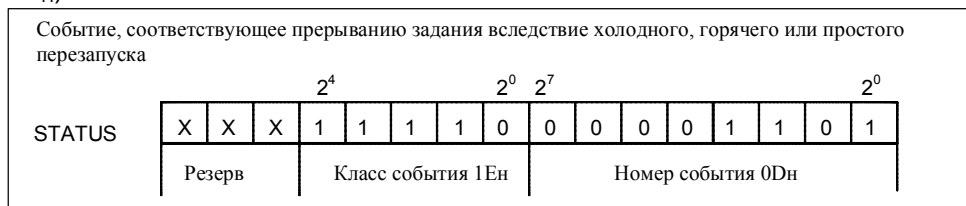


Рисунок 3–11 Пример: параметр STATUS для класса события 1Е_н, событие 0D_н

Диагностические сообщения функциональных блоков

В таблицах 6-1 и 6-2 описаны классы событий, а также приведены номера событий и рекомендуемые меры по устранению каждого из условий возникновения ошибки.

Вызов переменной SFCERR

Дополнительные сведения об ошибках 14 (1E 0E_н) и 15 (1E 0F_н) для класса события 30 можно получить из переменной SFCERR.

Переменную SFCERR можно загрузить из экземплярного DB, принадлежащего соответствующему функциональному блоку.

Сообщения об ошибках, содержащиеся в переменной SFCERR, описаны в разделе, посвященном системным функциям "DPRD_DAT" и SFC15 "DPWR_DAT" в руководстве *System Software for S7 300/400, System and Standard Functions*.

Класс события 5 (05 _H): "Ошибка при выполнении задания CPU"		
Номер события	Событие	Меры по устранению
(05) 02 _H	Задание не может быть выполнено в данном режиме модуля ET 200S SI (например, не выполнено параметрирование интерфейса устройства).	Обработайте диагностическое прерывание и устраните ошибку соответствующим образом.
(05) 0E _H	Недопустимая длина кадра сообщения.	Длина кадра сообщения > 200 байтов. Модулем ET 200S SI будет приниматься оставшаяся часть кадра сообщения (> 256 байтов), а первый его фрагмент, таким образом, будет утерян. Выберите меньшую длину кадра сообщения.
(05) 51 _H	В процессе связи между модулем ET 200S SI и программируемым логическим контроллером произошла ошибка выполнения кадра. Ошибка произошла, когда принятый кадр сообщения модуля ET 200S SI передавался программируемому логическому контроллеру.	Модуль и программируемый логический контроллер прервали передачу. Повторите задание на прием; модуль ET 200S SI передаст принятое сообщение вновь.

Класс события 8 (08 _H): "Ошибка приема"			
Номер события	Номер события (десятичный)	Событие	Меры по устранению
(08) 06 _H	6	Превышено время ожидания символа: после приема символа следующий символ не был принят в пределах времени ожидания символа	Коммуникационный партнер реагирует слишком медленно или неисправен. Проверьте исправность коммуникационного партнера, используя для этого, в случае необходимости, прибор для проверки интерфейсов (FOXPG), включенный в линию передачи.
08 0A _H	10	Переполнение буфера приема ведущего устройства при приеме ответного сообщения.	Проверьте настройки протокола для ведомого устройства.
(08) 0C _H	12	Обнаружена ошибка передачи (ошибка четности, ошибка стоп-бита, ошибка переполнения).	Возмущения в линии передачи приводят к повтору кадров сообщений, снижая тем самым пропускную способность канала. При этом возрастает опасность пропуска ошибок. Неисправность такого рода можно устранить, если изменить настройки системы или саму линию связи. Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов.
(08) 0D _H	13	ОБРЫВ: обрыв в линии приема, подключенной к партнеру.	Переустановите соединение или включите партнерское устройство.
(08) 10 _H	16	Ошибка четности: если включен светодиод SF (красный), в соединительном кабеле между двумя коммуникационными партнерами произошел обрыв.	Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов. Устраните ошибку, изменив системные настройки или саму линию.

Класс события 8 (08 _H): "Ошибка приема"			
Номер события	Номер события (десятичный)	Событие	Меры по устранению
(08) 11 _H	17	Ошибка кадра символа: если включен светодиод SF (красный), в соединительном кабеле между двумя коммуникационными партнерами произошел обрыв.	Проверьте соединительный кабель коммуникационного партнера. Также проверьте, совпадают ли следующие настройки обоих устройств: скорость передачи, четность и количество стоп-битов. Устраните ошибку, изменив системные настройки или саму линию.
(08) 12 _H	18	После сброса модулем CTS в "0" были приняты другие символы.	Настройте параметры коммуникационного партнера или производите чтение данных быстрее.
08 30 _H	48	<p>Ведущее устройство: после передачи сообщения запроса истекло время ожидания ответа, в течение которого не было обнаружено начало ответного сообщения.</p> <p>Ведомое устройство: широковещение (групповая передача) для данного кода функции не допускается.</p>	<p>Проверьте, нет ли обрыва в линии (может потребоваться анализатор интерфейса). Проверьте, установлены ли для модуля и коммуникационного партнера одинаковые параметры: скорость передачи, количество битов данных, четность, количество стоп-битов.</p> <p>Проверьте, установлено ли достаточно большое значение времени ожидания ответа в параметре R_{tP}_PARAM.</p> <p>Проверьте, существует ли адресуемое ведомое устройство.</p> <p>Широковещение в ведущей системе Modbus допускается только для отдельных функций, специально для этого предназначенных.</p>

Класс события 8 (08н): Ошибка приема			
Номер события	Номер события (десятичный)	Событие	Способ устранения
08 31 _н	49	<p>Ведущее устройство: первый символ в ответном сообщении ведомого устройства отличается от адреса ведомого устройства, переданного в запросе (для рабочего режима Normal (обычный)).</p> <p>Ведомое устройство: принят недопустимый код функции.</p>	<p>Ответ поступил от другого ведомого устройства. Проверьте линию передачи на обрывы (может понадобится анализатор интерфейса).</p> <p>Для данного драйвера нельзя использовать этот код функции.</p>
08 32 _н	50	Превышено максимальное количество битов или регистров, или количество битов не делится на 16 при обращении к областям таймеров и счетчиков памяти SIMATIC.	Максимальное количество битов составляет 2040, максимальное количество регистров – 127. При обращении к таймерам и счетчикам SIMATIC осуществляется только пословный доступ (по 16 бит).
08 33 _н	51	Количество битов или регистров для функций FC 15/16 и элемент сообщения “количество _байтов” не совпадают.	Исправьте количество битов/регистров или количество _байтов.
08 34 _н	52	Для команд “установить бит/сбросить бит” использован недопустимый код.	Для FC05 можно использовать коды 0000Hex или FF00Hex.
08 35 _н	53	Для функции FC 08 “Проверка связи” распознан недопустимый диагностический подкод (≠ 0000Hex).	Для FC08 можно использовать только подкод 0000Hex.
08 36 _н	54	Внутреннее сгенерированное значение контрольной суммы CRC 16 не совпадает с принятым значением CRC.	Проверьте, как генерируется контрольная сумма CRC ведущей системой Modbus.
08 37 _н	55	Ошибка последовательности сообщений: ведущая система Modbus передала новое сообщение запроса, от ведомого устройства поступило предыдущее ответное сообщение.	Увеличьте время ожидания ответного сообщения от ведомого устройства для ведущей системы Modbus.

Класс события 14 (0E _H) Общие ошибки выполнения <Назначение параметров>			
Номер события	Номер события (десятичный)	Событие	Способ устранения
0E20 _H	32	Для данного канала связи количество битов данных должно быть установлено равным 8. Драйвер не готов к работе.	Исправьте параметры драйвера.
0E21 _H	33	Множитель, выбранный для времени задержки символа, лежит за пределами 1...10. Драйвер работает со значением, принимаемым по умолчанию (1).	Исправьте параметры драйвера.
0E22 _H	34	Для драйвера выбран недопустимый рабочий режим. Необходимо указать либо "Normal operation" (нормальная работа) или "Interference Suppression" (подавление помех). Драйвер не готов к работе.	Исправьте параметры драйвера.
0E23 _H	35	Ведущее устройство: для контрольного времени ожидания ответа выбрано недопустимое значение. Допустимыми являются значения в диапазоне 50 ... 655000 мс. Драйвер не готов к работе. Ведомое устройство: выбрано недопустимое значение для адреса ведомого устройства. Нельзя указывать нулевой адрес ведомого устройства. Драйвер не готов к работе.	Исправьте параметры драйвера. Исправьте параметры драйвера.
0E2E _H	46	При чтении файла параметров интерфейса произошла ошибка. Драйвер не готов к работе.	Перезапустите ведущее устройство (Mains_ON).

Класс события 14 (0E _H) Общие ошибки выполнения <Выполнения задания S_SEND>			
Номер события	Номер события (десятичный)	Событие	Способ устранения
0E 40 _H	64	Введен слишком маленький параметр LEN (длина) для функционального блока S_SEND.	Минимальная длина составляет 2 байта.
0E 41 _H	65	Введен слишком маленький параметр LEN (длина) для функционального блока S_SEND. Для запрошенной функции необходимо указать большее значение.	Минимальная длина для данной функции составляет 6 байтов.
0E 42 _H	66	Передан недопустимый код функции.	Допустимые коды функций перечислены в разделе 3.3.
0E 43 _H	67	Для данного кода функции не допускается указывать нулевой адрес ведомого	Нулевой адрес ведомого устройства

		устройства (= ширококестание).	можно использовать только для некоторых функций.
--	--	--------------------------------	--

Класс события 14 (0E _H) Общие ошибки выполнения <Выполнения задания S_SEND>			
Номер события	Номер события (десятичный)	Событие	Способ устранения
0E 44 _H	68	Переданный параметр "количество битов" выходит за диапазон 1 ... 2040.	Параметр "количество битов" должен находиться в диапазоне 1 ... 2040.
0E 45 _H	69	Переданный параметр "количество регистров" выходит за диапазон 1 ... 127.	Параметр "количество регистров" должен находиться в диапазоне 1 ... 127.
0E 46 _H	70	Коды функций 15 или 16: параметры "количество битов" и/или "количество регистров" выходят за диапазон 1...2040 и/или 1...127.	"Количество битов" и/или "количество регистров" должны находиться в диапазоне 1 ... 2040 и/или 1 ... 127.
0E 47 _H	71	Коды функций 15 или 16: параметр LEN (длина) для S_SEND не совпадает с переданным параметром "количество битов" и/или "количество регистров". Значение LEN слишком мало.	Повышайте параметр LEN (длина) для SEND, пока на модуль не будет передан достаточный объем данных пользователя. На модуль следует передавать больший объем данных пользователя, определяемый параметрами "количество битов" и/или "количество регистров".
0E 48 _H	72	Код функции 5: в исходном блоке данных SEND указан недопустимый код для команд "установить бит" (FF00H) или "сбросить бит" (0000H).	Допускаются только коды "установить бит" (FF00H) или "стереть бит" (0000H).
0E 49 _H	73	Код функции 8: в исходном блоке данных SEND указан неправильный код для параметра "диагностический код".	Допускается только "диагностический код" 0000H.
0E 4A _H	74	Указанная для данной функции длина превышает максимально допустимую.	Максимальную длину для каждого функций смотрите в данном руководстве.

Класс события 14 (0E_H) Общие ошибки выполнения <Диагностика приема>			
Номер события	Номер события (десятичный)	Событие	Способ устранения
0E 50 _H	80	На ведущее устройство поступил ответ, которому не предшествовала передача.	Ведомое устройство или другое ведущее устройство в сети. Проверьте, не оборвалась ли линия передачи (может понадобиться анализатор интерфейса).
0E 51 _H	81	Неправильный код функции: код функции, принятый в ответном сообщении, отличается от переданного кода функции.	Проверьте ведомое устройство
0E 52 _H	82	Недобор байтов: количество принятых символов оказалось меньше количества, указанного в параметре "количество байтов" ответного сообщения, либо меньше ожидаемого для данного кода функции.	Проверьте ведомое устройство
0E 53 _H	83	Перебор байтов: количество принятых символов оказалось больше количества, указанного в параметре "количество байтов" ответного сообщения, либо больше ожидаемого для данного кода функции.	Проверьте ведомое устройство
0E 54 _H	84	Неправильное значение количества байтов: слишком маленькое значение параметра "количество байтов", принятого в ответном сообщении.	Проверьте ведомое устройство
0E 55 _H	85	Неправильное значение количества байтов: неправильное значение параметра "количество байтов", принятого в ответном сообщении.	Проверьте ведомое устройство
0E 56 _H	86	Ошибка эха: данные в ответном сообщении (количество битов, ...), возвращенные ведомым устройством, отличаются от данных, переданных в сообщении запроса.	Проверьте ведомое устройство
0E 57 _H	87	Неправильная контрольная сумма CRC: ошибка проверки контрольной суммы CRC 16 ответного сообщения ведомого устройства.	Проверьте ведомое устройство

Класс события 14; (0E_H)			
Общие ошибки выполнения <Сообщение с кодом ошибки приема>			
Номер события	Номер события (десятичный)	Событие	Способ устранения
0E 61 _H	97	Ответное сообщение с кодом ошибки 01: недопустимая функция	Смотрите руководство на используемое ведомое устройство
0E 62 _H	98	Ответное сообщение с кодом ошибки 02: недопустимый адрес данных	Смотрите руководство на используемое ведомое устройство
0E 63 _H	99	Ответное сообщение с кодом ошибки 03: недопустимое значение данных	Смотрите руководство на используемое ведомое устройство
0E 64 _H	100	Ответное сообщение с кодом ошибки 04: сбой соответствующего устройства	Смотрите руководство на используемое ведомое устройство
0E 65 _H	101	Ответное сообщение с кодом ошибки 05: подтверждение	Смотрите руководство на используемое ведомое устройство
0E 66 _H	102	Ответное сообщение с кодом ошибки 06: занято, в запросе отказано	Смотрите руководство на используемое ведомое устройство
0E 67 _H	103	Ответное сообщение с кодом ошибки 07: отрицательное подтверждение	Смотрите руководство на используемое ведомое устройство

Класс события 30 (1E_H): Ошибки связи при обмене данными между модулем SI и CPU			
Номер события	Номер события (десятичный)	Событие	Способ устранения
(1E) 0D _H	13	"Задание прервано из-за холодного перезапуска, горячего перезапуска или сброса"	
(1E) 0E _H	14	Статическая ошибка при вызове DP_RDDAT SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 0F _H	15	Статическая ошибка при вызове DP_WRDAT SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 10 _H	16	Статическая ошибка при вызове RD_LGADR SFC. Можно произвести оценку возвратного значения (RET_VAL) SFC в переменной SFCERR в экземплярном DB.	Загрузите переменную SFCERR из экземплярного DB.
(1E) 20 _H	32	Параметр превышает допустимый диапазон.	Измените вход функционального блока таким образом, чтобы он попал в допустимый диапазон.

(1E) 41 _H	65	В параметре LEN функциональных блоков задано недопустимое количество байтов.	Укажите длину в диапазоне 1 ... 200 байт.
----------------------	----	--	---

3.6.3 Диагностика ведомых устройств PROFIBUS

Диагностика ведомых устройств соответствует стандарту PROFIBUS EN 50170, Том 2. В зависимости от конкретного DP-ведущего, диагностические данные могут быть прочитаны либо с помощью STEP 5, либо с помощью STEP 7 для всех DP-ведомых, соответствующих указанному стандарту.

Диагностические данные ведомых устройств PROFIBUS состоят из диагностических данных модуля, сведений о состоянии модуля и диагностических данных канала. Подробные сведения о диагностических данных ведомых устройств DP можно найти в главе "Ввод в эксплуатацию и диагностика" в руководстве *ET 200S Distributed I/O Device*.

Диагностические данные канала:

Диагностические данные канала следуют за состоянием модуля и содержат сведения об ошибках канала модулей. В таблице 3-28 перечислены типы ошибок канала.

Таблица 3–28 Типы ошибок канала модуля последовательного интерфейса ET 200S Modbus/US\$		
Тип ошибки	Значение	Рекомендуемые действия
00110: Обрыв провода	Оборвался или отсоединился провод.	Проверьте подключение проводов к клеммам. Проверьте кабель коммуникационного партнера.
00111: Переполнение	Переполнение буфера; превышение длины сообщения.	Функциональный блок S_RCV следует вызывать чаще.
01000: Нулевая длина	Передано сообщение нулевой длины.	Проверьте, почему партнер по связи передает кадры, не содержащие данных пользователя.
01001: Ошибка	Произошла внутренняя ошибка модуля.	Замените модуль.
10000: Ошибка назначения параметра	Модулю не были назначены параметры.	Исправьте назначение параметров.
10110: Ошибка сообщения	Ошибка кадра, ошибка четности.	Проверьте настройки связи.

3.6.4 Функции диагностики ведомого устройства Modbus

Функциональный блок Modbus-коммуникаций имеет два выходных параметра, служащих для индикации произошедших ошибок:

- Параметр **ERROR_NR**
- Параметр **ERROR_INFO**

ERROR_NR, ERROR_INFO

Ошибки индицируются выходом ERROR_NR. Дополнительные сведения, касающиеся ошибок на выходе ERROR_NR, отображаются на выходе ERROR_INFO.

Удаление ошибок

Стирание ошибок происходит по переднему фронту на входе START. В случае необходимости, отображаемые ошибки могут быть удалены пользователем в любое другое время.

Коды ошибок FB

- **ERROR_No 1 ... 9 (Номер ошибки 1...9)**

Ошибка при инициализации FB и CP

Номера ошибок 1...9 указывают на ошибку при инициализации. Параметр START_ERROR имеет значение 1.

Modbus-коммуникации с ведущей системой невозможны.

- **ERROR_No 10 ... 19**

Ошибка при выполнении кода функции

Номера ошибок 10...19 указывают на ошибку при выполнении функции.

Модуль передал недопустимое задание (запрос) функциональному блоку коммуникаций.

Об ошибке также сообщается драйверу.

Следующие задания (запросы) продолжают выполняться.

- **ERROR_No 90 ... 99**

Другие ошибки

Произошла ошибка выполнения.

Драйверу об ошибке не сообщается.

Последующие задания продолжают выполняться.

3.6.5 Ошибки

Ошибки при инициализации			
Номер ошибки (десятичный)	ERROR_INFO	Описание	Способ устранения
0	0	Ошибок нет	
1	SFC51– >RET_VAL	Ошибка при чтении SZL с помощью SFC51.	Проанализируйте RET_VAL в ERROR_INFO, устраните причину.
2	S_SEND– >STATUS, S_RCV– >STATUS	Превышение времени при инициализации модуля или ошибка при инициализации модуля (ошибка в задании S_SEND).	Проверьте, были ли назначены параметры протокола “Modbus Slave” (ведомое устройство Modbus) для данного интерфейса. Проверьте, правильное ли значение указано для идентификатора “ID” в функциональном блоке коммуникаций. Проанализируйте ERROR_INFO.

Ошибки при выполнении функций			
Номер ошибки (десятичный)	ERROR_INFO	Описание	Способ устранения
11	Start Address (начальный адрес)	Драйвер передал недопустимый начальный адрес в FB-коммуникаций..	Проверьте адрес Modbus в ведущей системе Modbus.
12	Amount of Registers (количество регистров)	Драйвер передал недопустимое количество регистров в FB-коммуникаций: количество регистров = 0.	Проверьте количество регистров в ведущей системе Modbus, если требуется, перезапустите модуль (Mains_ON).
13	Amount of Registers (количество регистров)	Драйвер передал недопустимое количество регистров в FB-коммуникаций: количество регистров > 128.	Проверьте количество регистров в ведущей системе Modbus, если требуется, перезапустите модуль (Mains_ON).
14	Memory bits M –End Address (биты памяти M – конечный адрес)	Попытка обращения к области памяти SIMATIC (биты памяти) за пределы допустимого диапазона. Внимание: длина в CPU SIMATIC зависит от типа CPU.	Уменьшите начальный адрес Modbus и/или длину адресуемой области в ведущей системе Modbus.

Номер ошибки (десятичный)	ERROR_INFO	Описание	Способ устранения
15	Outputs Q – End Address (выходы Q – конечный адрес) Inputs I – End Address (входы I – конечный адрес)	Попытка обращения к области памяти SIMATIC (выходы) за пределы допустимого диапазона. Внимание: Длина в CPU SIMATIC зависит от типа CPU.	Уменьшите начальный адрес Modbus и/или длину адресуемой области ведущей системы Modbus.
16	Timers T – End Address (таймеры T – конечный адрес)	Попытка обращения к области памяти SIMATIC (таймеры) за пределы допустимого диапазона. Внимание: Длина в CPU SIMATIC зависит от типа CPU.	Уменьшите начальный адрес Modbus и/или длину адресуемой области ведущей системы Modbus.
17	Counters C – End Address (счетчики C – конечный адрес)	Попытка обращения к области памяти SIMATIC (счетчики) за пределы допустимого диапазона. Внимание: Длина в CPU SIMATIC зависит от типа CPU.	Уменьшите начальный адрес Modbus и/или длину адресуемой области ведущей системы Modbus.
18	0	Драйвер передал недопустимую область памяти SIMATIC в коммуникационной FB.	Если требуется, перезапустите модуль (Mains_ON).
19		Ошибка обращения к входам/выходам SIMATIC.	Проверьте, существуют ли требуемые входы/выходы, и отсутствуют ли связанные с ними ошибки.
20	DB#	DB не существует	Добавьте блок данных проект.
21	DB#	Недопустимая длина DB	Увеличьте длину DB
22	DB#	Номер DB меньше минимального допустимого значения для DB	Измените предельное минимальное значение для DB
23	DB#	Номер DB превышает максимально-допустимое значение для DB	Измените предельное максимальное значение для DB
24	M memory address (адрес памяти M)	Память M ниже минимального значения	Измените минимальные пределы адресов памяти M в DB преобразования

25	M memory address (адрес памяти M)	Память M выше максимального значения	Измените максимальные пределы адресов памяти M в DB преобразования
26	Q memory address (адрес памяти Q)	Память Q ниже минимального значения	Измените минимальные пределы адресов памяти Q в DB преобразования
27	Q memory address (адрес памяти Q)	Память Q выше максимального значения	Измените максимальные пределы адресов памяти Q в DB преобразования

Прочие ошибки			
Номер ошибки (десятичный)	ERROR_INFO	Описание	Способ устранения
90	S_SEND→STATUS	Ошибка при передаче сообщения подтверждения драйверу с помощью S_SEND.	Проанализируйте информацию в STATUS.
94	S_RCV→STATUS	Ошибка при чтении SYSTAT с помощью S_RCV (STATUS).	Проанализируйте информацию в STATUS.

3.7 USS-Master (Ведущее устройство USS)

3.7.1 Протокол USS

Протокол USS – это простой протокол последовательной передачи данных, созданный специально для применения в приложениях, связанных с управлением приводами. Подробные спецификации протокола, физического интерфейса, структуры шины, а также описание передаваемых данных, связанных с приводными задачами, содержатся в спецификации “Универсальный протокол последовательной передачи данных USS” (часть E20125–D0001–S302–A1–7600).

Протоколом USS определяется метод доступа по принципу “ведущий-ведомый” для обмена данными по последовательной шине. К шине может быть подключено одно ведущее устройство и не более 31 ведомого устройства. Отдельное ведомое устройство выбирается ведущим устройством по символу адреса, содержащемуся в сообщении. Ведомое устройство не может передавать данные по своей инициативе, а лишь по инициативе ведущего устройства, поэтому непосредственный обмен данными между отдельными ведомыми устройствами невозможен. Для обмена данными используется полудуплексный режим связи. Функция ведущего устройства не может быть передана другому устройству. USS является системой с одним ведущим устройством.

Структура сообщения

Каждое сообщение начинается с начального символа (STX), за которым следует длина (LGE) и байт адреса (ADR). После этого следуют поля данных. Признаком завершения сообщения является символ проверки блока (BCC).

STX	LGE	ADR	1	2	...	n	BCC
-----	-----	-----	---	---	-----	---	-----

При передаче слова (16 бит) в блоке неинтерпретируемых данных первым передается старший байт, а за ним младший. Соответственно, при передаче двойного слова старшее слово передается первым, а младшее – последним. Данные (задания) в полях данных протоколом не устанавливаются (не интерпретируются).

Кодирование данных

Информация кодируется следующим образом:

- STX: 1 байт, начало текста 02H
- LGE: 1 байт, содержит длину сообщения в формате двоичного числа
- ADR: 1 байт, состоит из адреса ведомого устройства и типа сообщения.
- данные в полях представлены в двоичной форме: содержимое каждого байта зависит от конкретной задачи

- BCC: 1 байт, символ проверки блока

Процедура передачи данных

Ведущее устройство осуществляет циклическую передачу телеграмм. Ведущее устройство обращается ко всем ведомым устройствам по очереди, передавая им телеграмму с заданием. Адресуемые узлы возвращают в ответ ответную телеграмму. Согласно процедуре “ведущий/ведомый”, ведомое устройство должно вернуть ответную телеграмму ведущему устройству после получения телеграммы с заданием, до того как ведущее устройство обратится к следующему ведомому устройству.

Общая структура блока неинтерпретируемых данных

Блок неинтерпретируемых данных подразделяется на две области: Параметры (PKW) и Процессные данные (PZD).

STX	LGE	ADR	Параметры (PKW)	Процессные данные (PZD)	BCC
-----	-----	-----	-----------------	-------------------------	-----

Область параметров (PKW)

Область PKW отвечает за обмен параметрами между двумя коммуникационными партнерами (например, регулятором с разомкнутым контуром управления и приводом). Это может быть, например, чтение или запись параметров, чтение описаний параметров и связанной с ними текстовой информации. Область PKW используется также для операторского управления и отображения, а также для целей обслуживания и диагностики.

Область процессных данных (PZD)

Область PZD содержит сигналы, необходимые для целей автоматизации:

- Слова управления и задания, поступающие от ведущего устройства на ведомое устройство.
- Слова состояний и текущие значения, считываемые ведущим устройством из ведомых устройств.

Содержимое области параметров и области процессных данных определяется приводами (ведомыми устройствами). Сведения о данной области следует искать в документации на конкретный привод.

3.7.2 Конфигурирование и параметрирование

С помощью приложения конфигурирования аппаратных средств для драйвера должны быть установлены следующие параметры и рабочие режимы:

- Диагностическое сообщение

- Скорость передачи

3.7.3 Обзор функций

Функциональные блоки осуществляют циклический обмен неинтерпретируемыми данными с приводами ведомых устройств (до 31 ведомого устройства) в соответствии с последовательностью опроса, определяемой списком опроса (блок данных настройки параметров). В любой момент времени активным является только один запрос к ведомому устройству. Неинтерпретируемые данные для каждого ведомого устройства хранятся пользователем в блоке данных (блок неинтерпретируемых данных) и берутся оттуда. Они передаются коммуникационному процессору и считываются из него в соответствии со списком опроса, определяемым программой, через другую область данных (блок данных коммуникационного процессора).

Данная процедура предусматривает вызов двух функций (одна для блока передачи и одна – для блока приема). Другая функция поддерживает создание и предварительное назначение блоков данных, требуемых для коммуникаций.

Свойства:

- Создание областей данных для коммуникаций зависит от конфигурации шины
- Предварительное создание списка опроса
- Структура кадра соответствует спецификации USS
- Обмен неинтерпретируемыми данными можно спараметризовать в соответствии с требуемой структурой неинтерпретируемых данных
- Выполнение и мониторинг запросов PKW
- Обработка отчетов об обмене параметрами
- Мониторинг всей системы и обработка ошибок

Для передачи неинтерпретируемых данных могут использоваться различные структуры.

В зависимости от выбранной структуры, неинтерпретируемые данные могут состоять из области PZD для процессных данных и области PKW для параметрирования.

С помощью области PKW ведущее устройство может считывать и записывать параметры, а ведомое устройство может сообщать об изменении параметров (сообщения об изменении параметров).

Область PZD содержит сигналы, необходимые для управления процессом, например, слова управления и задания, поступающие с ведущего устройства на ведомое, а также слова состояний и текущие значения, считываемые ведущим устройством из ведомого.

Функции должны вызываться в следующем порядке: S_USST, S_SEND, S_RCV, S_USSR. Это очень важно, поскольку выходы функций S_SEND и S_RCV действительны только для текущего цикла ПЛК.

На рисунке 3–12 показан маршрут прохождения данных между программой управления пользователя и ведомым устройством USS.

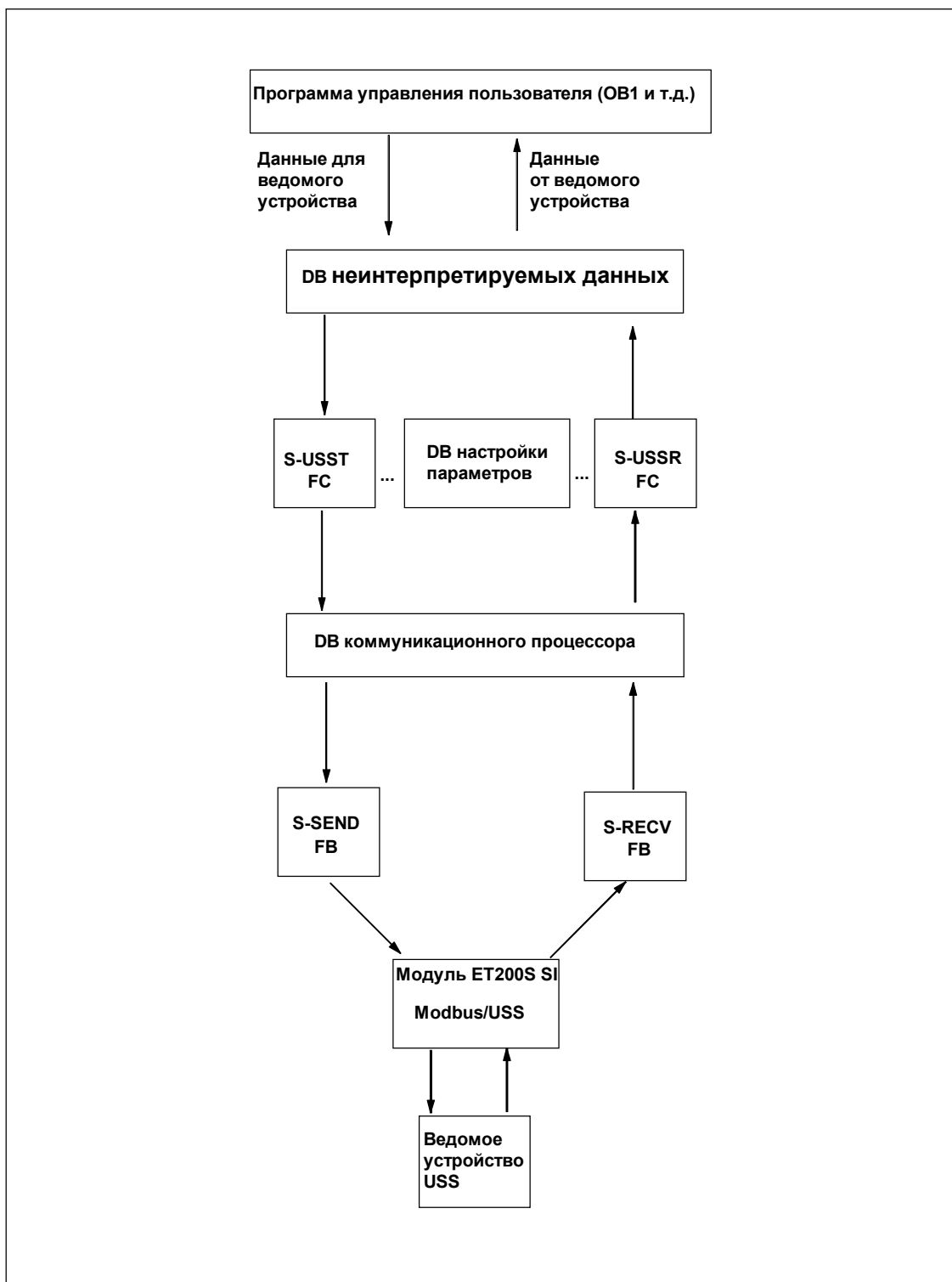


Рисунок 3–12 Маршрут прохождения данных между программой управления пользователя и ведомым устройством USS

3.7.4 FC17 S_USST: Передача данных ведомому устройству

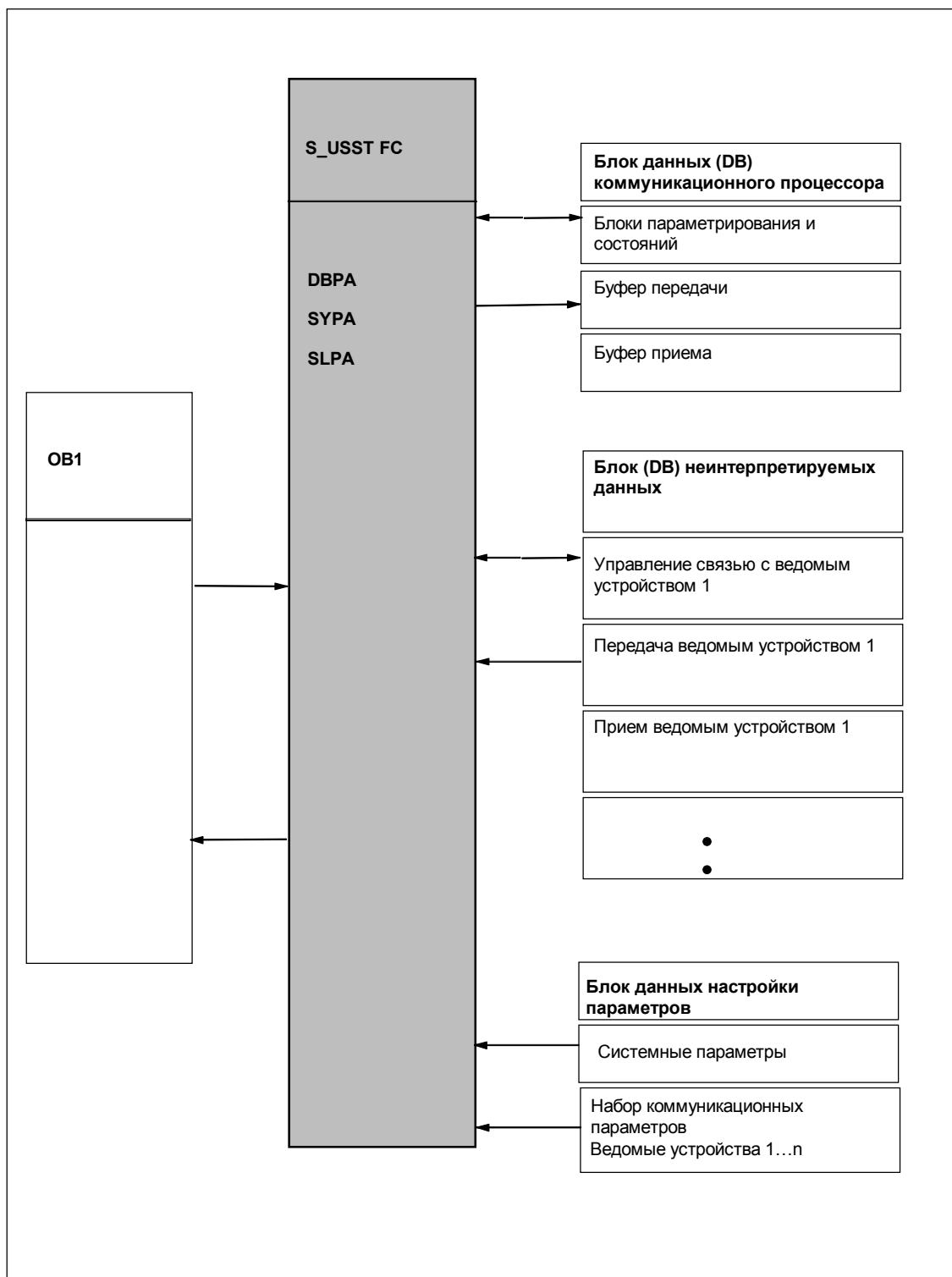
FC S_USST управляет передачей неинтерпретируемых данных (PZD и любые данные PKW) ведомым устройствам, в зависимости от используемой структуры неинтерпретируемых данных.

Функциональный блок берет набор параметров текущего ведомого устройства из списка опроса (DB настройки параметров), а данные, предназначенные для передачи этому ведомому устройству, из блока неинтерпретируемых данных. Он анализирует слово управления коммуникациями для текущего ведомого устройства (инициирование запроса PKW или подтверждение сообщения об изменении параметров), формирует данные для передачи по USS и передает их в буфер передачи блока данных коммуникационного процессора. Наконец, он инициирует передачу неинтерпретируемых данных ведомому устройству с помощью функционального блока S_SEND.

В случае обнаружения функцией ошибки параметров в блоке данных настройки параметров, сообщение об ошибке записывается в два последних байта блока неинтерпретируемых данных.

Вызов функции FC17 осуществляется один раз за цикл ПЛК.

На следующем рисунке показана структура программы S_USST.





Примечание

Параметры EN и ENO отображаются только при графическом представлении (LAD или FBD). Для обработки данных параметров компилятор использует бинарный результат BR.

В случае завершения работы блока без ошибок бинарный результат устанавливается в состояние 1, если произошла ошибка, BR сбрасывается в 0.

Параметры FC17 S_USST

В таблице 3 – 29 перечислены параметры FC S_USST.

Таблица 3–29 Параметры FC S_USST

Название	Тип	Тип данных	Описание	Комментарий
DBPA	Вход	INT	Номер блока для DB настройки параметров	Зависит от CPU (нулевое значение не допускается)
SYPA	Вход	INT	Начальный адрес системных параметров в блоке данных (DB) настройки параметров	0 <= SYPA <= 8174
SLPA	Вход	INT	Начальный адрес параметров ведомого устройства в DB настройки параметров	0 <= SLPA <= 8184

3.7.5 FC18 S_USSR: Прием данных от ведомого устройства

FC S_USSR управляет приемом неинтерпретируемых данных (PZD и любые данные PKW) от ведомых устройств в зависимости от используемой структуры неинтерпретируемых данных.

Функциональный блок берет набор параметров текущего ведомого устройства из списка опроса (блок данных настройки параметров) и анализирует слово состояния передаваемого блока.

Если текущий запрос был завершен без ошибок (бит 9 = 0 в слове состояния связи блока неинтерпретируемых данных), принятые данные передаются из буфера приема блока данных коммуникационного процессора в блок неинтерпретируемых данных и анализируются, после чего слово состояния связи в блоке неинтерпретируемых данных обновляется.

Если текущий запрос был завершен с ошибкой (бит 9 = 1 в слове состояния связи блока неинтерпретируемых данных), данные текущего ведомого устройства из буфера приема блока данных коммуникационного процессора не берутся. FC18 сообщает об этом в слове состояния связи блока неинтерпретируемых данных и записывает причину ошибки в слово ошибок связи.

В случае обнаружения блоком ошибки параметров в блоке данных настройки параметров, сообщение об ошибке записывается в 1 байт блока неинтерпретируемых данных.

FC18 вызывается один раз в каждом цикле ПЛК.

На рисунке ниже показана структура программы для S_USSR.

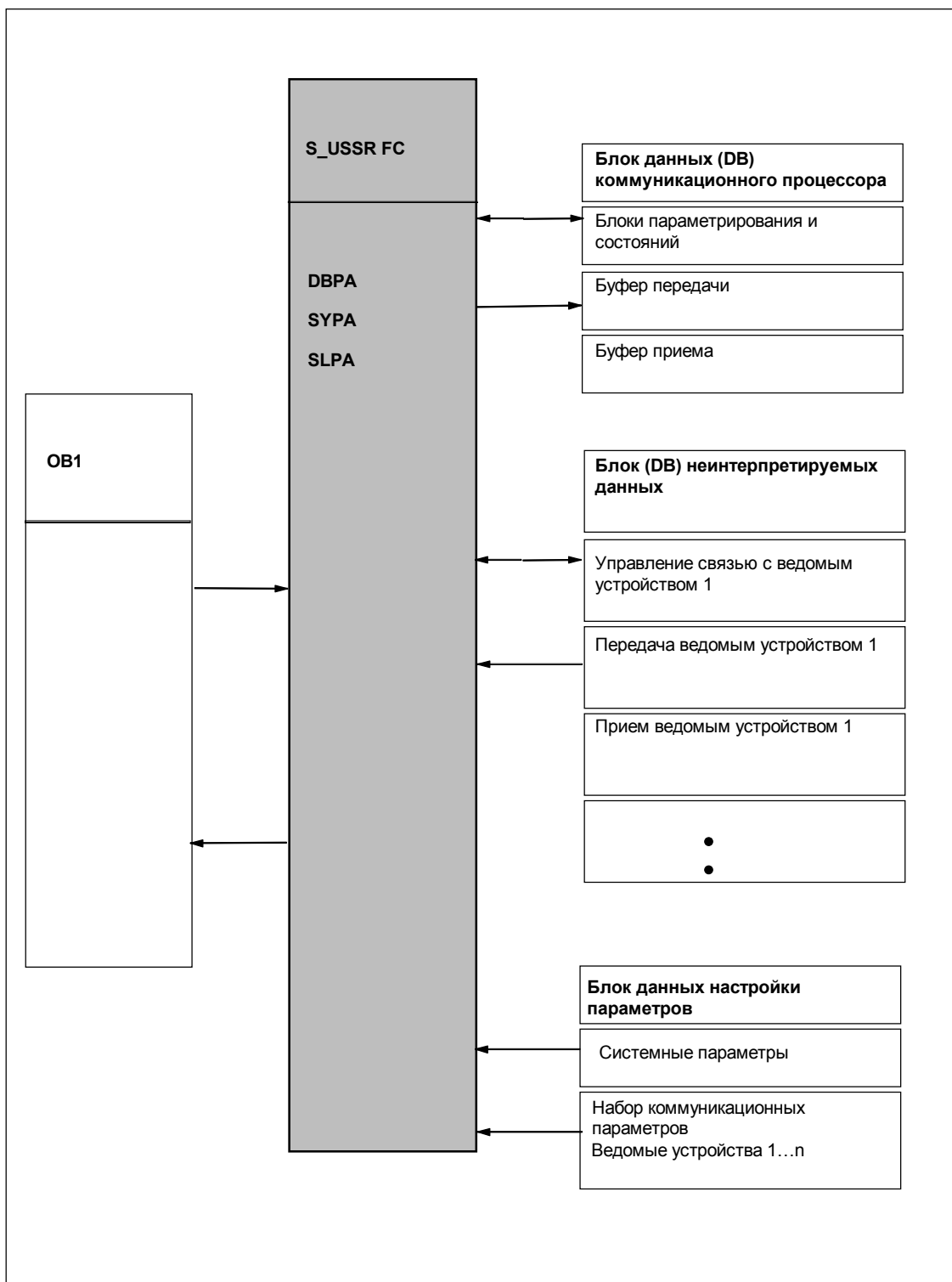
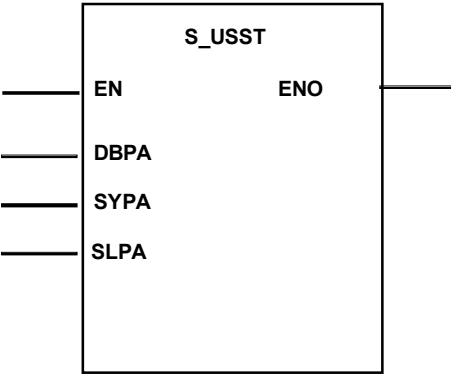


Рисунок 3–13 Структура программы

Представление в формате STL	Представление в формате LAD
<p>CALL S_USST</p> <p>DBPA =</p> <p>SYPA =</p> <p>SLPA =</p>	

Примечание

Параметры EN и ENO отображаются только при графическом представлении (LAD или FBD). Для обработки данных параметров компилятор использует бинарный результат BR.

В случае завершения работы блока без ошибок бинарный результат устанавливается в состояние 1, если произошла ошибка, BR сбрасывается в 0.

Параметры FC18 S_USSR

В таблице 3–30 перечислены параметры FC S_USSR.

Таблица 3–30 Параметры FC S_USSR

Название	Тип	Тип данных	Описание	Комментарий
DBPA	Вход	INT	Номер блока для DB настройки параметров	Зависит от CPU (нулевое значение не допускается)
SYPA	Вход	INT	Начальный адрес системных параметров в блоке данных (DB) настройки параметров	$0 \leq SYPA \leq 8174$
SLPA	Вход	INT	Начальный адрес параметров ведомого устройства в DB настройки параметров	$0 \leq SLPA \leq 8184$

Параметры FC U_USST совпадают с параметрами для FC S_USSR. Две этих функции обращаются к тому же набору параметров (системные параметры и параметры ведомых устройств) в блоке данных настройки параметров, и, следовательно, должны быть спараметрированы одинаково.

3.7.6 FC19 S_USSI: Инициализация

FC S_USSI является необязательной функцией.

При инициировании запуска системы S7 FC генерирует блоки данных коммуникационного процессора, блоки неинтерпретируемых данных и настройки параметров, необходимых для коммуникаций. Также предварительно назначается DBPA. FC S_USSI можно использовать для генерации и предварительного назначения указанных областей данных только в том случае, если все устройства имеют одну и ту же структуру неинтерпретируемых данных.

После вызова FC выполняет проверку соответствия своих параметров: количество ведомых устройств, структуру неинтерпретируемых данных, количество начальных узлов и количество повторов PKW. В случае обнаружения ошибки генерация и предварительное назначение блоков данных не выполняются. CPU переходит в режим STOP, а пользователю передается сигнал ошибки в байте ошибок FC S_USSI. После устранения ошибки параметров любые ранее сгенерированные блоки данных должны быть удалены до повторного запуска.

После проверки соответствия блок проверяет, существуют ли уже блоки данных, которые должны быть сгенерированы:

- Если генерируемые блоки данных еще не существуют, они генерируются и выполняется предварительное назначение DBPA.

- Если блоки данных, которые должны быть сгенерированы, уже существуют, проверяется длина каждого блока данных. Если блок данных имеет достаточную длину, вновь выполняется предварительное назначение блока данных настройки параметров, а содержимое блока неинтерпретируемых данных и блока данных коммуникационного процессора стирается. Если блок данных имеет недостаточную длину, CPU переходит в состояние STOP. Пользователь может определить, какой блок данных явился причиной ошибки, с помощью байта состояния в FC S_USSI. Для исправления ошибки должны быть полностью удалены три блока данных. После этого блоки данных генерируются снова при следующем повторном запуске, и выполняется предварительное назначение блока данных настройки параметров.

S_USSI должна вызываться только один раз при запуске системы (OB100).

STL Representation	LAD Representation
CALL S_USSI SANZ = TNU1 = PKW = PZD = DBND = DBPA = DBCP = WDH = ANZ =	<p>The LAD representation shows a function block labeled 'S_USSI'. It has an input 'EN' at the top left and an output 'ENO' at the top right. Below 'EN', there are ten input lines labeled 'SANZ', 'TNU1', 'PKW', 'PZD', 'DBND', 'DBPA', 'DBCP', 'WDH', and 'ANZ' from top to bottom. The block is represented by a rectangle with these labels and lines.</p>

Примечание

Параметры EN и ENO отображаются только при графическом представлении (LAD или FBD). Для обработки данных параметров компилятор использует бинарный результат BR.

В случае завершения работы блока без ошибок бинарный результат устанавливается в состояние 1, если произошла ошибка BR сбрасывается в 0

Параметры FC19 S_USSI

В таблице 3–31 представлены параметры FC S_USSI.

Таблица 3–31 Параметры FC S_USSI				
Название	Тип	Тип данных	Описание	Комментарий
SANZ	Вход	INT	Количество ведомых устройств с точно такой же структурой неинтерпретируемых данных (системные параметры DBPA)	1 <= SANZ <= 31
TNU1	Вход	INT	Номер начального узла (станции)	0 <= TNU1 <= 31
PKW	Вход	INT	Количество PKW	Количество слов интерфейса PKW 0, 3 или 4
PZD	Вход	INT	Количество PZD	Количество слов интерфейса PZD 0 <= PZD <= 16
DBND	Вход	INT	Номер блока неинтерпретируемых данных	Зависит от CPU (нулевое значение не допускается)
DBPA	Вход	INT	Номер блока данных настройки параметров	Зависит от CPU (нулевое значение не допускается)
DBCP	Вход	INT	Номер блока данных коммуникационного процессора	Зависит от CPU (нулевое значение не допускается)
WDH	Вход	INT	Допустимое количество повторов запроса PKW	0 <= WDH <= 32767
ANZ	Выход	BYTE	Байт ошибки	0: ошибок нет 1: слишком большое количество ведомых устройств 2: недопустимая структура неинтерпретируемых данных 3: блок данных настройки параметров слишком короткий 4: блок неинтерпретируемых данных слишком короткий 5: ошибка номера станции 6: блок данных коммуникационного процессора слишком короткий 7: не исп. 8: счетчик повторов: неправильное значение

3.7.7 Блок неинтерпретируемых данных

Эти блоки данных могут генерироваться или предварительно назначаться с помощью S_USSI во время запуска CPU (только DBPA) или вводиться вручную.

Блок (DB) неинтерпретируемых данных играет роль интерфейса между коммуникационной частью и программой пользователя. Пользователь должен заботиться о том, чтобы данный блок был пустым и имел достаточную длину. Ответные данные от ведомого устройства берутся из соответствующего буфера приема (после анализа бита 9 слова управления связью). Слово состояния позволяет осуществлять контроль связи, а слово управления позволяет инициировать запрос на требуемую установку параметров.

Коммуникационный интерфейс содержит следующие данные для каждого ведомого устройства:

- Коммуникационные данные конкретного ведомого устройства (управление связью, контроль (слежение), 6 слов данных)
- Буфер для текущего запроса PKW (только, если область PKW существует)
- Буфер передачи для неинтерпретируемых данных (не более 20 слов данных)
- Приемный буфер для неинтерпретируемых данных (не более 20 слов данных)

Длины буферов передачи и приема зависят от выбранной структуры неинтерпретируемых данных. Если интерфейс PKW не существует, буфер для текущего запроса PKW не используется.

Суммарная длина, необходимая для неинтерпретируемых данных, зависит от количества ведомых устройств и используемой структуры неинтерпретируемых данных:

Количество слов данных на одно ведомое устройство

$$= 2 \times (\text{PKW} + \text{PZD}) + \text{PKW} + 6,$$

где PKW равно 0, 3 или 4 и $0 \leq \text{PZD} \leq 16$

Например, для привода с областью PKW, состоящей из трех слов, и с областью PZD из 2 слов требуется 19 слов данных в блоке неинтерпретируемых данных.

Для 31 ведомого устройства блок неинтерпретируемых данных состоит из 1550 слов данных. DBW0 зарезервировано для других целей.

Распределение данных ведомого устройства в блоке неинтерпретируемых данных с областью PKW из 4 слов и с областью PZD, состоящей из 0...16 слов

DBWn	Слово управления связью (KSTW)	Управление связью
DBWn+2	Внутренние	Контроль связи
DBWn+4	Слово состояния связи	
DBWn+6	Слово ошибки связи	
DBW n+8	Внутренние	Счетчик повторов PKW
DBW n+10	Pafe 1–байт, Pafe 2–байт	Ошибка параметра
DBW n+12	Идентификатор параметра (ID) PKE	Буфер для текущего запроса PKW
DBW n+14	Index (Указатель) IND	
DBW n+16	Значение параметра 1 PWE1	
DBW n+18	Значение параметра 2 PWE2	
DBW n+20	Идентификатор параметра PKE	Область PKW
DBW n+22	Index (Указатель) IND	
DBW n+24	Значение параметра 1 PWE1	
DBW n+26	Значение параметра 2 PWE2	
DBW n+28	Слово управления (STW) PZD1	Буфер передачи
DBW n+30	Главное задание (HSW) PZD2	
DBW n+32	Задание/дополнительное слово управления PZD3	
DBW n+34	Задание/дополнительное слово управления PZD4	
...	...	(макс. 16 слов PZD)
DBW n+58	Задание/дополнительное слово управления PZD16	Область PZD
DBW n+60	Идентификатор параметра PKE	
DBW n+62	Index IND	

DBW n+64	Значение параметра 1 PWE1	Буфер приема
DBW n+66	Значение параметра 2 PWE2	
DBW n+68	Слово состояния (ZSW) PZD1	
DBW n+70	Главное текущее значение PZD2	
DBW n+72	Текущее значение/дополнительное слово состояния PZD3	
DBW n+74	Текущее значение/дополнительное слово состояния PZD4	
...	...	
DBW n+98	Текущее значение/дополнительное слово состояния PZD16	
	•	
(n = 2,4,6...)	•	
		Область PZD (макс. 16 слов PZD)

Примечание

Если области PKW не существует, в этом случае буфер для текущего запроса PKW и область PKW в буфере отсутствуют.

Слово управления связью KSTW (DBW n)

Биты в слове управления связью служат для координации программы пользователя и FC S_USST.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Бит 0: Запрос PKW принят

Бит 0 устанавливается пользователем, если в буфере приема находится новый запрос PKW, и он готов к обработке. Данный бит сбрасывается FC, если запрос PKW был принят.

Бит 1: Сообщение об изменении параметра принято

Бит 1 устанавливается пользователем, если было принято сообщение об изменении параметра. Данный бит сбрасывается FC для подтверждения приема. Ведомое устройство возобновляет прерванную обработку текущего запроса после подтверждения приема или передает следующее сообщение об изменении параметра.

Слово состояния связи (DBW n+4)

Биты в слове состояния связи устанавливаются с помощью FC S_USST и S_USSR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Бит 0: Запрос PKW обрабатывается

FC S_USST устанавливает бит 0, если запрос PKW был принят и идентификатор (ID) параметра (PKE) содержит допустимый ID запроса. FC S_USSR сбрасывает данный бит, если запрос PKW был выполнен (с или без ошибок) или если имеется ошибка интерфейса PKW.

Бит 1: Запрос PKW завершился без ошибок

FC S_USSR устанавливает данный бит, если запрос PKW был завершен без ошибок. Ответ находится в буфере приема. Данный бит сбрасывается FC S_USST, если инициируется новый запрос PKW.

Примечание

Запросы PKW на ведомые устройства выполняются в последовательности, определяемой списком опроса (DBPA). В любой момент времени активным является только один запрос на одно ведомое устройство. Если в список опроса введено несколько ведомых устройств, ответные данные на запрос PKW становятся доступными только по переднему фронту бита 1 (или бита 2).

Бит 2: Запрос PKW завершен без ошибок.

FC S_USSR устанавливает бит 2 для ответа с ID7 в PKE. Номер ошибки содержится в PWE ответа ведомого устройства. Он сбрасывается FC S_USST, если инициируется новый запрос PKW.

Примечание

Последний переданный пользователем запрос PKW сохраняется в интерфейсе передачи после того, как он был обработан. Передача на ведомое устройство повторяется до тех пор, пока не будет введен новый запрос. Для этого могут понадобиться дополнительные ответы в программе пользователя в случае, если запрос состояний PKW завершился с ошибкой (бит 2) и произошла ошибка интерфейса PKW (бит 4).

Бит 3: Недопустимый идентификатор (ID) запроса PKW.

FC S_USST устанавливает бит 3, если в PKE обнаружен ID запроса = 15, или если для запроса ID 4 введен индекс 255. Данный бит сбрасывается FC S_USST при инициировании нового запроса PKW с допустимым идентификатором запроса PKE.

Бит 4: Интерфейс PKW с ошибкой (переполнение счетчика).

FC S_USSR устанавливает бит 4, если запрос PKW не был подтвержден ведомым устройством за сконфигурированное количество повторных попыток запроса (см. параметр WDH в блоке данных настройки параметров), или в случае ответа ID 8 в PKE. Данный бит сбрасывается FC S_USSR, если инициируется новый запрос PKW, который выполняется без ошибок.

Бит 5: Данные ответы содержат сообщение об изменении параметров. FC S_USSR устанавливает бит 5, если от ведомого устройства поступил ответ об изменении параметра (ответ ID 9 ... 12 и инвертирован бит переключения 11). Данный бит сбрасывается FC S_USST, если пользователь подтвердил прием сообщения об изменении параметров (слово управления связью, бит 1).

Бит 6: Сбой в работе ведомого устройства.

Бит 6 устанавливается и сбрасывается FC S_USSR. FC выполняет анализ слова состояния (бит 3) ведомого устройства.

Бит 7: Имеется предупреждение от ведомого устройства.

Бит 7 устанавливается и сбрасывается в FC S_USSR. FC анализирует слово состояния (бит 7) ведомого устройства.

Бит 8: Запрошено управление ПЛК.

FC S_USSR устанавливает и сбрасывает бит 8. FC анализирует слово состояния (бит 9) и слово управления (бит 10).

Бит 9: Групповая ошибка.

Бит 9 устанавливается и сбрасывается в FC S_USSR. FC анализирует сигналы обратной связи стандартных блоков S_SEND и S_RCV и выполняет проверку кадра, принятого для ADR, STX, BCC и LGE. С помощью данного бита FC также сигнализирует о нарушении контрольного времени кадра.

Примечание

Принятые данные в блоке неинтерпретируемых данных действительны только в том случае, когда бит 9 равен 0..

Структура слова ошибки связи (DBW n+6)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Бит 0: Ошибка адресации (ADR)

Бит 3: Не распознано начало кадра (отсутствует STX в качестве первого символа)

Бит 4: Ошибка символа проверки блока (BCC)

Бит 6: Неправильная длина кадра (LGE)

Биты 0, 3, 4 и 6 устанавливаются FC S_USSR, если при проверке принятого кадра обнаружена ошибка (ADR, STX, BCC, LGE).

Бит 7: Превышение контрольного времени кадра.

Бит 7 устанавливается FC S_USSR, если время, затраченное на передачу кадра ведущим устройством ведомому и поступления от ведомого устройства ответа, превышает допустимое предельное время, рассчитанное программой (контрольное время кадра).

Оставшиеся биты не используются.

Байт Pafe 1

Сообщение об ошибке от FC S_USSR, ошибки параметрирования в блоке данных настройки параметров

Значение 0: Ошибок нет

Значение 1: Неправильные данные для PKW / PZD

Байт Pafe 2

Сообщение об ошибке от FC S_USST, ошибки параметрирования в блоке данных настройки параметров

Значение 0: Ошибок нет

Значение 1: Неправильные данные для PKW / PZD

Идентификатор параметра PKE в буфере передачи

Пользователь должен назначить номер параметра (биты 0 ... 10) и идентификатор (ID) запроса (биты 12 ... 15). Бит переключения для сообщения об изменении параметров (бит 11) маскируется функциями S_USSR и S_USST.

3.7.8 Блок данных настройки параметров

Блок данных настройки параметров содержит параметры программы, необходимые для управления связью. Пользователь должен создать данный блок и соответствующим образом настроить конфигурацию системы связи (с помощью S_USSI или вручную). Ведомые устройства, присутствующие в шине, опрашиваются в порядке, определяемом списком опроса, содержащимся в DBPA.

Ведомое устройство может быть указано в блоке данных настройки параметров несколько раз, что повышает его приоритет над остальными устройствами.

Размер блока данных настройки параметров зависит от количества ведомых устройств (n), обращение к которым будет выполняться в пределах одного цикла шины. Количество слов данных в блоке данных настройки параметров $= (n \times 4) + 5$.

Для организации связи с каждым ведомым устройством требуется 4 слова данных, и еще 4 слова данных назначается для параметров системы. Слово DBW0 зарезервировано для других целей.

DBW 0	Не исп.	Параметры системы
DBW 2	DBCP	
DBW 4	SANZ	
DBW 6	SLAV	
DBW 8	WDH	
DBW 10	Количество PKW, количество PZD	Связь
DBW 12	TNU	
DBW 14	DBND	
DBW 16	KSTW	Набор параметров связи для ведомого устройства 1
DBW 18	Количество PKW, количество PZD	
DBW 20	TNU	
DBW 22	DBND	Связь
DBW 24	KSTW	
		Набор параметров связи для ведомого устройства 2
	Количество PKW, количество PZD	
	TNU	
	DBND	Связь
	KSTW	
	Количество PKW, количество PZD	Набор параметров связи для ведомого устройства n
	TNU	
	DBND	
DBW (n x 8 + 8)	KSTW	

Параметры системы

DBCP:	Номер блока данных коммуникационного процессора
SANZ:	Количество наборов параметров ведомых устройств в блоке данных настройки параметров. Если к некоторым ведомым устройствам обращение должно выполняться чаще по сравнению с другими в пределах цикла шины, параметры для этих устройств должны быть введены несколько раз в блоке данных настройки параметров. При этом должен быть соответствующим образом скорректирован системный параметр SANZ.
SLAV:	Порядковый номер текущего ведомого устройства. Требуется для FC S_USST и FC S_USSR при вычислении текущего набора параметров. При инициализации слово данных должно быть установлено в 1. Это делается FC S_USSI, если последняя используется.
WDH:	Допустимое количество повторов запроса PKW (диапазон назначения: 0...32767). Если текущий запрос PKW не был выполнен за определенное количество попыток, генерируется сигнал об ошибке интерфейса PKW.

Набор параметров связи с ведомым устройством

Кол-во PKW,	
Кол-во PZD:	<p>Определение структуры неинтерпретируемых данных.</p> <p>Левый байт: Количество слов в области PKW (0, 3, 4)</p> <p>Правый байт: Количество слов в области PZD (0...16)</p> <p>Любые отклонения от данных параметров объявляются как ошибки параметрирования FC S_USST и FC S_USSR, и вводятся в байт Pafe 1, Pafe 2 блока неинтерпретируемых данных.</p>
TNU:	Номер узла, соответствующий адресу привода в шине (0...31).
DBND:	Номер блока неинтерпретируемых данных.
KSTW:	Адрес слова управления связью KSTW для ведомого устройства в блоке неинтерпретируемых данных.

3.7.9 Блок данных коммуникационного процессора

Этот блок данных управляет обменом данными между CPU и модулем последовательного интерфейса ET 200S Modbus/USS. Пользователь должен обеспечить достаточную длину данного блока. Длина блока данных коммуникационного процессора должна быть, по крайней мере, 50 слов (DBW 0...98).

DBW 0	Состояние связи		TRANSMIT (передача) и RECEIVE (прием)
DBW 2	Максимальное количество циклов в режиме ожидания приема	Счетчик циклов для расчета превышения времени в режиме ожидания приема	FC17
DBW 4	Измерение длительности паузы		FC17
DBW 6	Длительность последнего цикла (OB1_MIN_CYCLE)		FC17, OB1
DBW 8	Длина переданной телеграммы (LEN)		TRANSMIT
DBB10	Не используется		
DBB 11 : DBB 54	Буфер передачи		Передача кадра модулю (длина зависит от структуры неинтерпретируемых данных текущего ведомого устройства)
DBB 55 : DBB 98	Буфер приема		Буфер приема для модуля (длина зависит от структуры неинтерпретируемых данных текущего ведомого устройства)

Состояние связи DBW0

Бит 0: Вход REQ блока S_SEND. Данный бит сбрасывается, когда установлен бит 8.

Бит 1: Вход R блока S_SEND. Данный бит сбрасывается циклически S_USST.

Бит 2: Выход DONE блока S_SEND.

Бит 3: Выход ERROR блока S_SEND.

Бит 4: Вход EN_R блока S_RCV. Данный бит устанавливается циклически S_USSR.

Бит 5: Вход R блока S_RCV. Данный бит сбрасывается циклически S_USSR.

Бит 6: Выход NDR блока S_RCV.

Бит 7: Выход ERROR блока S_RCV.

Бит 8: Запрос обрабатывается (сохраненный бит DONE блока S_SEND). Данный бит устанавливается и сбрасывается S_USST.

Длительность последнего цикла DBW6

Данный параметр используется S_USST для измерения времени ожидания ответа ведомого устройства. Программа пользователя должна копировать длительность цикла программы ПЛК (и т. д.) в данный параметр перед каждым вызовом S_USST.

3.8 Процедура запуска модуля ET 200S SI Modbus/USS и его рабочие режимы

3.8.1 Загрузка конфигурации и параметров

Управление данными

После закрытия программы конфигурирования аппаратных средств данные будут автоматически сохранены в проекте Step 7.

Загрузка конфигурации и параметров

Конфигурацию и назначенные параметры можно загрузить в CPU с помощью программатора в режиме установленной связи (on-line). Для этого используется команда PLC→ Download (ПЛК→Загрузить).

Во время запуска CPU, а также при каждом переходе из режима STOP в режим RUN параметры модуля передаются ему автоматически, как только он становится доступен через объединительную шину S7–300.

Код драйвера сохраняется в энергонезависимой памяти модуля с помощью специального документа назначения параметров. Это означает, что модуль нельзя заменить без использования программатора.

Дополнительные сведения

Подробное описание перечисленных ниже таблиц следует смотреть в руководстве пользователя Step 7:

- Сохранение конфигурации и параметров
- Загрузка конфигурации и параметров в CPU
- Чтение, изменение, копирование и вывод на печать конфигурации и параметров

3.8.2 Рабочие режимы модуля ET 200S SI Modbus/USS

Модуль последовательного интерфейса ET 200S SI Modbus/USS может работать в следующих режимах:

- **STOP:** Когда модуль находится в режиме STOP, ни один из драйверов протокола не активен, и все запросы на передачу и прием, поступающие от CPU, получают негативное подтверждение. Модуль остается в режиме STOP, пока причина, вызывающая данный режим не будет устранена (например: обрыв провода или неправильный параметр).

- **Перенастройка параметров:** При перенастройке параметров модуля инициализируется драйвер протокола. В процессе перенастройки светится светодиод SF.

Передача и прием невозможны, а передаваемые и принимаемые кадры сообщений, сохраненные в модуль, будут утеряны при перезапуске драйвера. Обмен данными между модулем и CPU перезапускается (активные кадры сообщений отменяются).

По завершении перенастройки параметров модуль возвращается в режим RUN и снова готов к передаче и приему.

- **RUN:** Модуль выполняет (обрабатывает) запросы на передачу, поступающие от CPU, и предоставляет CPU возможность чтения кадров сообщений, принятых коммуникационным партнером.

3.8.3 Процедура запуска модуля ET 200S SI Modbus/USS

Процедура запуска состоит из двух этапов:

- **Инициализация:** сразу же после подачи питания на модуль инициализируется последовательный интерфейс, после чего модуль ожидает данные из CPU о назначении параметров.
- **Параметрирование:** в процессе параметрирования на модуль ET 200S SI Modbus/USS поступают параметры, которые были назначены текущему слоту в STEP 7.

3.8.4 Действия модуля ET 200S SI Modbus/USS при изменении режимов работы CPU

По завершении процедуры запуска модуля ET 200S SI Modbus/USS начинается обмен данными между CPU и модулем, который происходит с использованием функциональных блоков.

- **CPU–STOP:** В режиме CPU–STOP обмен данными PROFIBUS невозможен. Любой обмен данными между модулем ET 200S SI Modbus/USS и CPU, активной в настоящий момент, включая и передачу и прием кадров сообщений, прерывается, а соединение устанавливается вновь.
- **Запуск CPU (Start–up):** Во время запуска CPU передает модулю параметры.

Настроев параметры надлежащим образом, можно добиться того, чтобы буфер приема модуля стирался автоматически при запуске CPU.

- **CPU–RUN:** Когда CPU находится в режиме RUN, передача и прием не запрещены. В первых циклах FB, следующих сразу же за перезапуском CPU, выполняется синхронизация модуля и соответствующих FB. Пока этот механизм не будет завершен, никакие новые блоки S_SEND или S_RCV выполняться не будут.

Что следует помнить при передаче кадров сообщений

Кадры сообщений могут передаваться только в режиме RUN.

Если CPU переключается в режим STOP в тот момент, когда происходит передача данных из CPU на модуль, блок S_SEND сообщает об ошибке (05) 02_h после перезапуска. Чтобы избежать этого, программа пользователя может вызвать блок S_SEND со входа RESET из OB-запуска.

Примечание

Модуль ET 200S SI Modbus/USS не выполняет передачу данных коммуникационному партнеру до тех пор, пока он не принял все данные от модуля.

Что следует помнить при приеме кадров сообщений

С помощью STEP 7 можно спараметрировать параметр “delete SI receive buffer at start-up = yes/no” (“стирать буфер приема SI при запуске = да/нет”).

- Если выбрать Да, буфер приема ET 200S SI Modbus/USS будет автоматически очищаться всякий раз, когда режим CPU изменяется из STOP в RUN.
- Если выбрать Нет, кадр сообщения сохраняется в буфере приема ET 200S SI Modbus/USS.

Если CPU переходит в режим STOP в процессе передачи данных из ET 200S SI Modbus/USS в CPU, блок S_RCV сообщает об ошибке (05) 02_n после перезапуска. Чтобы этого избежать в программе пользователя можно вызвать блок S_SEND со входа RESET из OB-запуска. Если параметр “Delete ET 200S Serial Interface Modbus/USS Module receive buffer at start-up = no” (“Стирать буфер приема ET 200S SI Modbus/USS при запуске = нет”), кадр сообщения будет передан модулем ET 200S SI Modbus/USS на CPU повторно.

Более подробные сведения об обмене данными между CPU и ET 200S SI Modbus/USS можно посмотреть в руководстве “Модуль последовательного интерфейса ET 200S”.

Времена выполнения

Время, необходимое для завершения процедуры обмена “ведущий/ведомый” (включая время обновления данных), можно определить следующим образом:

Общее время выполнения (t_8) = Время выполнения запроса ведущего устройства (t_1) + Время передачи запроса ведущего устройства (t_2) + Время выполнения запроса ведомого устройства (t_3) + 1 цикл CPU (время на выполнение кодов функции) (t_4) + Время на выполнение ответа ведомого устройства (t_5) + Время на передачу ведомого устройства (t_6) + Время на выполнение ответа ведущего устройства (t_7)

Время выполнения запроса/ответа

Формула для расчета времени приема или передачи совпадает для ведущего и ведомого устройств и имеет следующий вид:

Если CPU превышает > (цикл входов/выходов + 10 мс), тогда Время выполнения = 1 цикл CPU на 7 байтов, иначе Время выполнения = (2 цикла CPU + 3 цикла входов/выходов + 10 мс) на 7 байтов

Время передачи/приема для запроса/ответа

Время, необходимое для передачи запроса или приема ответа, определяется следующим образом:

Время передачи/приема = 10 мс + кол-во символов сообщений/скорость передачи

Пример расчета суммарного времени выполнения:

Чтение	Скорость передачи	Цикл входов/выходов	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
10 слов	9600	2мс	40	15	40 мс	40 мс	160	31.7	160	488.7

	бит/с		MC	MC			MC	MC	MC	MC
--	-------	--	----	----	--	--	----	----	----	----

3.9 Технические характеристики

Характеристики протоколов и интерфейсов

Таблица 3–32 Общие характеристики модуля ET 200S Modbus/USS Module	
Общие характеристики	
Элементы отображения	Светодиоды: зеленый, TX (передача) зеленый, RX (прием) красный, SF (системная ошибка)
Поддерживаемые драйверы протоколов	Драйверы Modbus Драйверы USS
Скорости передачи для протокола Modbus	110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 бит/с (полудуплексный режим)
Скорости передачи для драйвера USS	110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 бит/с (полудуплексный режим)
Кадры символов (11 бит)	Количество битов на символ: 8 Количество старт/стоп битов: 1 или 2 Четность: нет, чет, нечет, любая
Требования к размеру области хранения для стандартных блоков (функциональные блоки)	Передача и прием: приблизительно, 4300 байт
Технические характеристики интерфейса RS–232C	
Интерфейс	RS–232C, 8 выводов
Сигналы RS–232C	TXD, RXD, RTS, CTS, DTR, DSR, DCD, PE Все указанные выводы электрически развязаны от внутреннего источника питания модуля ET 200S.
Максимальное расстояние передачи	15 м
Технические характеристики интерфейса RS–422/485	
Интерфейс	RS–422, 5 выводов RS–485, 3 вывода
Сигналы RS–422 Сигналы RS–485	TXD (A), RXD (A), TXD (B), RXD (B), PE R/T (A), R/T (B), PE Все указанные выводы электрически развязаны от внутреннего источника питания модуля ET 200S.
Максимальное расстояние передачи	1200 м

Технические характеристики

Габариты и вес	
Габариты Ш x В x Г (мм)	15 x 81 x 52
Вес	Приблиз., 50 г
Данные для отдельных модулей	
RS-232C	
• Количество входов	4
• Количество выходов	3
RS-422	
• Кол-во входных пар	1
• Кол-во выходных пар	1
RS-485	
• Кол-во входных/ выходных пар	1
Длина кабеля	
• Экранированного (RS-232C)	Макс. 15 м
• Экранированного(RS-422/485)	Макс. 1200 м
Степень защиты ¹	IEC 801-5
Напряжение, токи, потенциалы	
Номинальное напряжение питания электронных модулей (L+)	24 В DC
• Защита от неправильной полярности	Да
Развязка	
• Между каналами и задней шиной	Да
• Между каналами и цепями питания электронных устройств	Да
• Между каналами	Нет
• Между каналами и PROFIBUS-DP	Да
Испытания на пробой изоляции между	
• Каналами и задней шиной и между каналами и линией L+ (напряжение нагрузки)	500 В DC
• Линией L+ (напряжение нагрузки) и задней шиной	500 В AC
Источник тока	
• От задней шины	Макс. 10 мА
• От линии L+ (напряжение нагрузки)	Макс. 120 мА, Тип. 50 мА
Мощность, рассеиваемая модулем	Тип. 1,2 Вт

Состояния, прерывание, диагностика	
Отображение состояний	Зеленый светодиод (TX) Зеленый светодиод (RX)
Диагностические функции	
• Отображение групповой ошибки	Красный светодиод "SF"
• Возможность отображения диагностической информации	Возможно
Выходы	
Выходы, уровни RS-232C	± Макс. 10 В
• Для емкостной нагрузки	Макс. 2500 пФ Да
• Защита от короткого замыкания	Приблиз., 60 мА
• Ток короткого замыкания	Макс. 25 В
• Напряжение между выходами или входами и РЕ (земля)	
Выходы, RS-422/485	
Сопротивление нагрузки	Мин. 50 кОм
• Защита от короткого замыкания	Да
• Ток короткого замыкания	Приблиз., 60 мА

- ¹ При подключении цепей напряжения пользователя требуется внешнее защитное оборудование:
- Молниеотвод, адаптер для установки на стандартную рейку
 - Молниеотвод, защитный модуль типа KTAD-24B

Предметный указатель

Ф

- FB2 S_RCV, 3–23
 - временная диаграмма, 3–26
 - присвоения в области данных, 3–24
- FB3 S_SEND, 3–18
 - временная диаграмма, 3–22
 - вызов, 3–20
 - параметры, 3–21
 - присвоения в области данных, 3–20

В

- Ведомые устройства, адрес, 3–14
 - код функции 01 – Чтение логической ячейки (выхода), 3–49
 - адрес первого бита, 3–49
 - номер_бита, 3–50
 - обращение к битам памяти и выходам, 3–49
 - обращение к таймерам и счетчикам, 3–49
 - пример применения, 3–50
 - состояние, 3–49
 - код функции 02 – Чтение входного состояния, 3–52
 - адрес первого бита, 3–52
 - номер_бита, 3–50
 - пример применения, 3–53
 - код функции 03 – Чтение выходных регистров, 3–55
 - адрес первого регистра, 3–55
 - количество регистров, 3–55
 - пример применения, 3–56
 - формула расчета первого регистра, 3–55
 - код функции 04 – Чтение входных регистров, 3–58
 - адрес первого регистра, 3–58
 - количество регистров, 3–58
 - пример применения, 3–59
 - код функции 05 – Установка значения логической ячейки, 3–61
 - адрес_ячейки, 3–61
 - ДАННЫЕ = "ВКЛ/ВЫКЛ", 3–61
 - обращение к битам памяти и выходам, 3–61
 - обращение к таймерам и счетчикам, 3–61
 - пример применения, 3–62
 - код функции 06 – Установка значения отдельного регистра, 3–63
 - адрес первого регистра, 3–63
 - значение в поле ДАННЫЕ, 3–63
 - пример применения, 3–63
 - код функции 08 – Проверка связи, 3–65
 - пример применения, 3–65
 - код функции 15 – Установка значений нескольких логических ячеек, 3–66
 - адрес первой ячейки, 3–66
 - ДАННЫЕ, 3–67
 - количество, 3–66
 - обращение к битам памяти и выходам, 3–66
 - обращение к таймерам и счетчикам, 3–66
 - пример применения для преобразования адресов Modbus, 3–67
 - код функции 16 – Установка значений нескольких регистров, 3–68
 - адрес первого регистра, 3–69
 - ДАННЫЕ (старший, младший байты), 3–69
 - количество, 3–69
 - пример применения, 3–69
- Ведущее устройство USS, 3–99
 - FC17 S_USST: Передача данных ведомому устройству, 3–103
 - параметры, 3–105
 - FC18 S_USSR: Прием данных от ведомого устройства, 3–106
 - параметры, 3–109
 - FC19 S_USSI: Инициализация, 3–109
 - параметры, 3–111
 - блок данных коммуникационного процессора, 3–120
 - блок данных состояния связи DBW0, 3–120
 - длительность последнего цикла DBW0, 3–120
 - блок данных настройки параметров, 3–118
 - настройка параметров связи ведомого устройства, 3–119
 - параметры системы, 3–119
 - блок сетевых данных, 3–112
 - второй байт Pafe, 3–117
 - назначение данных ведомого устройства, 3–113
 - параметр ID PKE в буфере передачи, 3–117
 - первый байт Pafe, 3–117
 - слово состояния связи (DBW n+4), 3–115
 - слово управления связью, KSTW(DBWn), 3–114
 - структура слова ошибок связи (DBW n+6), 3–117
 - конфигурирование и установка параметров, 3–100
 - обзор функций, 3–101
 - порядок вызова функции, 3–101
 - протокол USS, 3–99
 - блок сетевых данных, 3–100
 - кодирование данных, 3–99
 - процедуры передачи данных, 3–100
 - структура сообщения, 3–99

Г – Д

- Групповое сообщение, широковещательное сообщение, 3–14
- Действия при запуске и режимы работы
 - действия модуля последовательного интерфейса ET 200S MODBUS/ USS при переключении режимов работы CPU, 3–123
 - загрузка назначенных данных, 3–121
 - режим работы, 3–122
- Действия при запуске, 3–122

Диагностика, 3–84
 вызов переменной SFCERR, 3–85
 диагностика ведомого устройства PROFIBUS, 3–94
 диагностика с помощью светодиодов
 состояния, 3–84
 диагностические сообщения функциональных
 блоков, 3–85
 класс события 14 (0EH) "Загружаемый драйвер –
 Общие ошибки выполнения", 3–90
 класс события 14 (0EH) "Загружаемый драйвер –
 Общие ошибки выполнения", 3–90
 класс события 30 (1EH): "Ошибка обмена
 данными между SI и CPU", 3–93
 структура диагностических сообщений
 функциональных блоков, 3–85
 типы ошибок канала модуля последовательного
 интерфейса ET 200S, 3–94
 Диагностика функционального блока связи
 ERROR_NR, ERROR_INFO, 3–94
 диагностика с помощью параметров ERROR_NR,
 ERROR_INFO, 3–95
 ERROR_No 1...9, 3–95
 ERROR_No 10...19, 3–95
 ERROR_No 90...99, 3–95
 ошибки во время "инициализации", 3–96
 удаление ошибок, 3–95
 функции диагностики, 3–94
 Дополнительные сигналы RS–232C, 2–48
 Драйвер ASCII, дополнительные сигналы
 RS–232C, 2–48
 Драйвер ведомого устройства
 компоненты, 3–41
 канал связи MODBUS Slave, 3–41
 коммуникационный функциональный блок
 ведомого устройства MODBUS, 3–41
 структуры данных, 3–41
 параметры
 адрес ведомого устройства, 3–47
 биты данных, 3–47
 конфигурирование, 3–45
 множитель времени задержки символа, 3–47
 настройка параметров для драйвера
 ведомого устройства, 3–45
 обычный режим, 3–47
 подавление помех, 3–47
 скорость передачи, 3–47
 стоп-биты, 3–47
 четность, 3–47
 Драйвер ведомого устройства Modbus, 3–41
 Дуплексный режим, 3–22

3 – И

Завершение сообщения, 3–16
 Интерфейс связи RS–232C, назначение клемм, 3–5
 Интерфейс связи RS–422, назначение клемм, 3–6
 Интерфейс связи RS–485, схема назначения
 клемм, 3–6
 Интерфейсы
 RS–232C, 3–10
 RS–422/485, 3–13

Используемые интерфейсы и протоколы, 3–18
 Исходный блок данных SEND
 код функции ведущего устройства 01, 3–31
 код функции ведущего устройства 02, 3–32
 код функции ведущего устройства 03, 3–33
 код функции ведущего устройства 04, 3–34
 код функции ведущего устройства 05, 3–34
 код функции ведущего устройства 06, 3–35
 код функции ведущего устройства 07, 3–36
 код функции ведущего устройства 08, 3–37
 код функции ведущего устройства 11, 3–37
 код функции ведущего устройства 12, 3–38
 код функции ведущего устройства 15, 3–39
 код функции ведущего устройства 16, 3–40

К

Коды функций ведущего устройства, 3–15
 код функции ведущего устройства 01 – Чтение
 выходного сигнала, 3–31
 конечный блок данных RCV, 3–32
 код функции ведущего устройства 02 – Чтение
 дискретного входа, 3–32
 исходный блок данных SEND, 3–32
 конечный блок данных RCV, 3–32
 код функции ведущего устройства 04 – Чтение
 входных регистров, 3–33
 исходный блок данных SEND, 3–34
 код функции ведущего устройства 05 –
 Изменение состояния логической ячейки, 3–34
 код функции ведущего устройства 06 – Запись
 нового значения в регистр, 3–35
 конечный блок данных RCV, 3–35
 код функции ведущего устройства 07 – Чтение
 состояния ошибок, 3–36
 конечный блок данных RCV, 3–36
 код функции ведущего устройства 08 – Проверка
 связи, 3–36
 конечный блок данных RCV, 3–37
 код функции ведущего устройства 11 – Счетчик
 событий связи, 3–37
 конечный блок данных RCV, 3–37
 код функции ведущего устройства 12 – Протокол
 событий связи, 3–38
 конечный блок данных RCV, 3–38
 код функции ведущего устройства 15 –
 Изменение состояния нескольких логических
 ячеек, 3–39
 код функции ведущего устройства 16 – Запись
 новых значений в несколько регистров, 3–40
 коды функций ведущего устройства 01 – Чтение
 выходного сигнала
 исходный блок данных SEND, 3–31
 коды функций ведущего устройства 03 – Чтение
 выходных регистров, 3–33
 исходный блок данных SEND, 3–33
 конечный блок данных RCV, 3–33
 коды функций ведущего устройства 04 – Чтение
 входных регистров
 конечный блок данных RCV, 3–34

коды функций ведущего устройства 05 -
Изменение состояния логической ячейки
 исходный блок данных SEND, 3–34
 конечный блок данных RCV, 3–35
коды функций ведущего устройства 06 – Запись
нового значения в регистр
 исходный блок данных SEND, 3–35
коды функций ведущего устройства 07 – Чтение
состояния ошибок
 исходный блок данных SEND, 3–36
коды функций ведущего устройства 08 – Проверка
связи
 исходный блок данных SEND, 3–37
коды функций ведущего устройства 11 – Чтение
счетчика событий связи
 исходный блок данных SEND, 3–37
коды функций ведущего устройства 12 – Чтение
протокола событий связи
 исходный блок данных SEND, 3–38
коды функций ведущего устройства 15 -
Изменение состояния нескольких логических
ячеек
 исходный блок данных SEND, 3–39
коды функций ведущего устройства 16 – Запись
новых значений в несколько регистров, исходный
блок данных SEND, 3–40
Коды функций, 3–15
Конечный блок данных RCV
 код функции ведущего устройства 01, 3–32
 код функции ведущего устройства 02, 3–32
 код функции ведущего устройства 03, 3–33
 код функции ведущего устройства 04, 3–34
 код функции ведущего устройства 05, 3–35
 код функции ведущего устройства 06, 3–35
 код функции ведущего устройства 07, 3–36
 код функции ведущего устройства 08, 3–37
 код функции ведущего устройства 11, 3–37
 код функции ведущего устройства 12, 3–38
Конфигурирование Modbus модуля, 3–27
Конфигурирование и настройка параметров
 предустановка параметров линии приема, 3–29
Краткое описание процедуры ввода в эксплуатацию,
последовательный интерфейс, 2–6

М – Н

Модуль последовательного интерфейса, технические
характеристики, 2–99, 2–100, 3–126
Назначение выводов 15-контактного разъема
соединительного кабеля RS–422, 3–9
Назначение выводов 25-контактного разъема
соединительного кабеля RS–232C, 3–8
Назначение выводов 9-контактного разъема
соединительного кабеля RS–232C, 3–7
Назначение контактов 15-контактного разъема
соединительного кабеля RS–485, 3–10

Настройка параметров драйвера ведущего
устройства, 3–27
 биты данных, 3–29
 время отклика, 3–30
 множитель времени задержки символа, 3–30
 обычный режим, 3–30
 подавление помех, 3–30
 полудуплекс (RS 485), двухпроводная
 схема, 3–29
 полудуплекс, исходное состояние линии
 приема, 3–29
 скорость передачи, 3–29
 стоп-биты, 3–29
 четность, 3–29

О – П

Обзор изделий, 3–2
 заказной номер, 3–2
Передача данных модулю ET 200S в качестве
ведущего устройства Modbus, 3–18
Поле данных (ДАННЫЕ), 3–15
 адрес первого регистра, 3–15
 адрес первой ячейки, 3–15
 количество байтов, 3–15
 количество регистров, 3–15
 количество ячеек, 3–15
Полудуплексный режим, 2–22
Примеры применения ведомого устройства
 код функции 05, 3–62
 код функции 06 – Установка значения
 регистра, 3–63
 код функции 08, 3–65
 код функции 15, Преобразование адресов
 Modbus, 3–67
 код функции 16, 3–69
 код функции ведомого устройства 01, 3–50
 код функции ведомого устройства 02, 3–53
 код функции ведомого устройства 03, 3–56
 код функции ведомого устройства 04, 3–59
Проверка CRC, 3–16
Протокол 3964(R), 2–32
Протокол USS, общая структура блока сетевых
данных,
 область данных процесса (PZD), 3–100
 область параметров (PKW), 3–100
Протокол передачи данных, 3–14

Р

Режим работы канала связи для драйвера ведомого
устройства
 области данных в CPU SIMATIC,
 преобразование адреса, 3–43
 обращение с использованием бит-
 ориентированных кодов-функций, 3–71
 код функции 02, 3–71
 коды функций 01, –05, –15, 3–74

- обращение с использованием регистр-ориентированных кодов-функций, 3–71
 - код функции 04, 3–72
 - коды функций 03, –06, –16, 3–71
 - номер слова в DB, 3–72
 - расчетный номер блока данных, 3–72
- ограничение для функций записи, 3–82
 - обзор FC 05, 06, 15, 16, 3–82
 - области памяти SIMATIC "Мин./Макс.", 3–82
 - пример для FC 05, 06, 16, 3–83
- преобразование адресов MODBUS для бит-ориентированных функций, 3–74
 - адрес MODBUS, "От/до", 3–75, 3–77
 - обзор FC 01, 05, 15, 3–74
 - обзор FC 02, 3–77
 - область памяти SIMATIC, "Начиная с...", 3–75
 - область памяти SIMATIC, "Начиная с...", 3–77
 - пример для FC 01, 05, 15, 3–75
 - пример для FC 02, 3–78
- преобразование адресов MODBUS для функций, ориентированных на регистры, 3–79
 - "Начиная с блока данных", 3–80
 - обзор FC 03, 06, 16, 3–79
 - обзор FC 04, 3–80
 - пример для FC 04, 3–81
 - пример, 3–80
- разрешить/запретить доступ для записи, 3–73
 - коды функций 05, 06, 15, 16, 3–73
 - разрешить доступ для записи, 3–73

С

Сигналы

- RS–232C, 3–11
- автоматическое использование вспомогательных сигналов, 3–12
- временная диаграмма управления вспомогательными сигналами, 3–13
- Сообщения с кодами ошибок, 3–17
- Структура сообщения, 3–14
- Схемы назначения клемм, 3–5

Т – У

- Технические характеристики, 3–125
 - протоколы и характеристики интерфейсов, 3–125
- Технические характеристики, модуль последовательного интерфейса, 2–99
- Указания по выполнению проводных соединений и подключению цепей, 3–5