

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.

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.

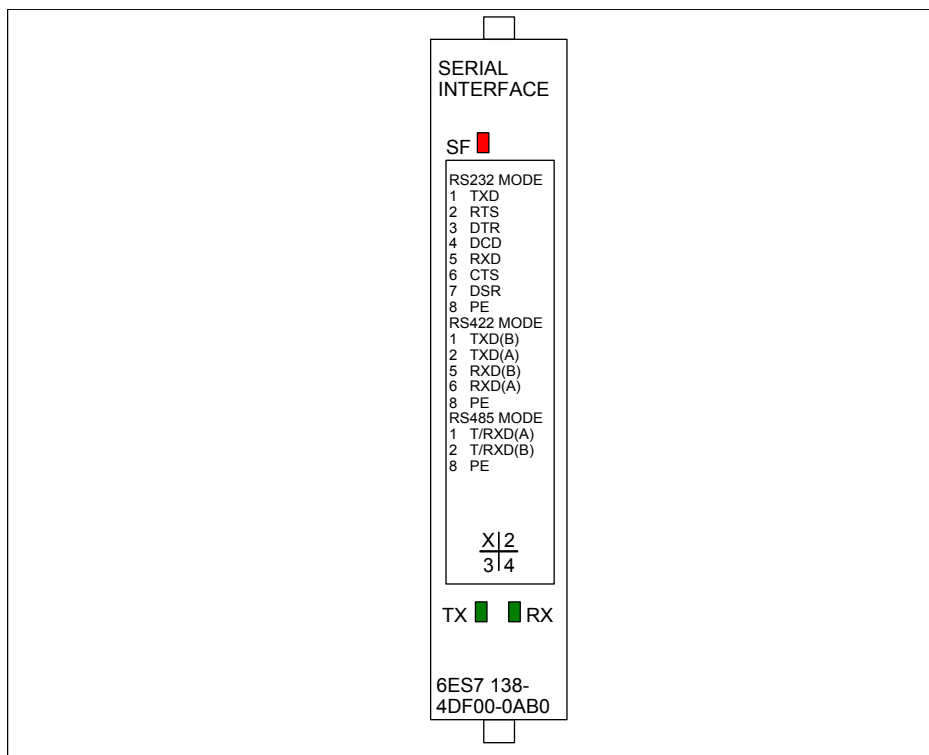


Рисунок 2–1 Маркировка на передней панели модуля последовательного интерфейса ET 200S

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 TM-E15S24-01 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").

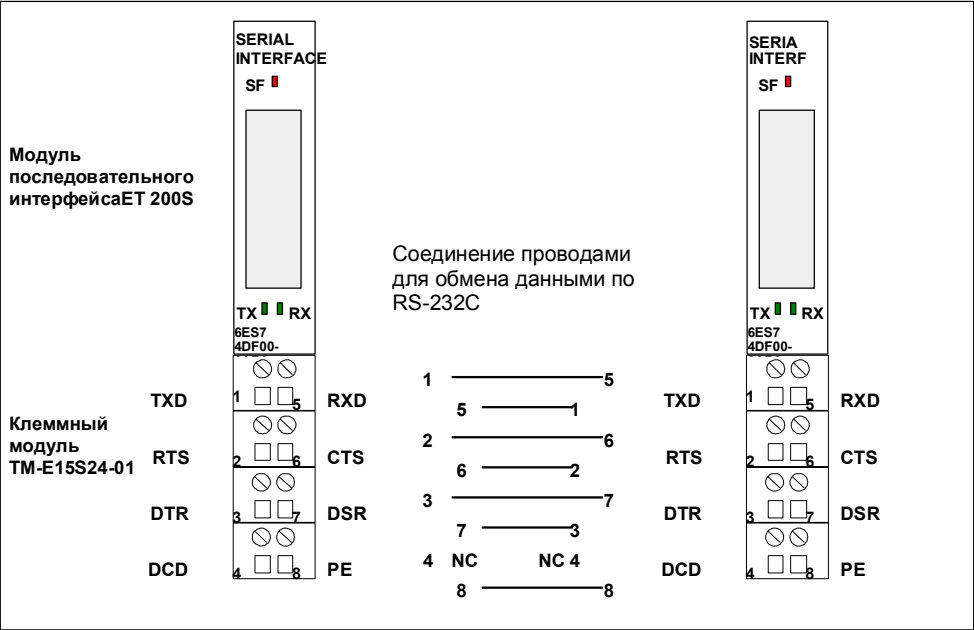


Рисунок 2-2 Назначение клемм, используемых в примере

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).

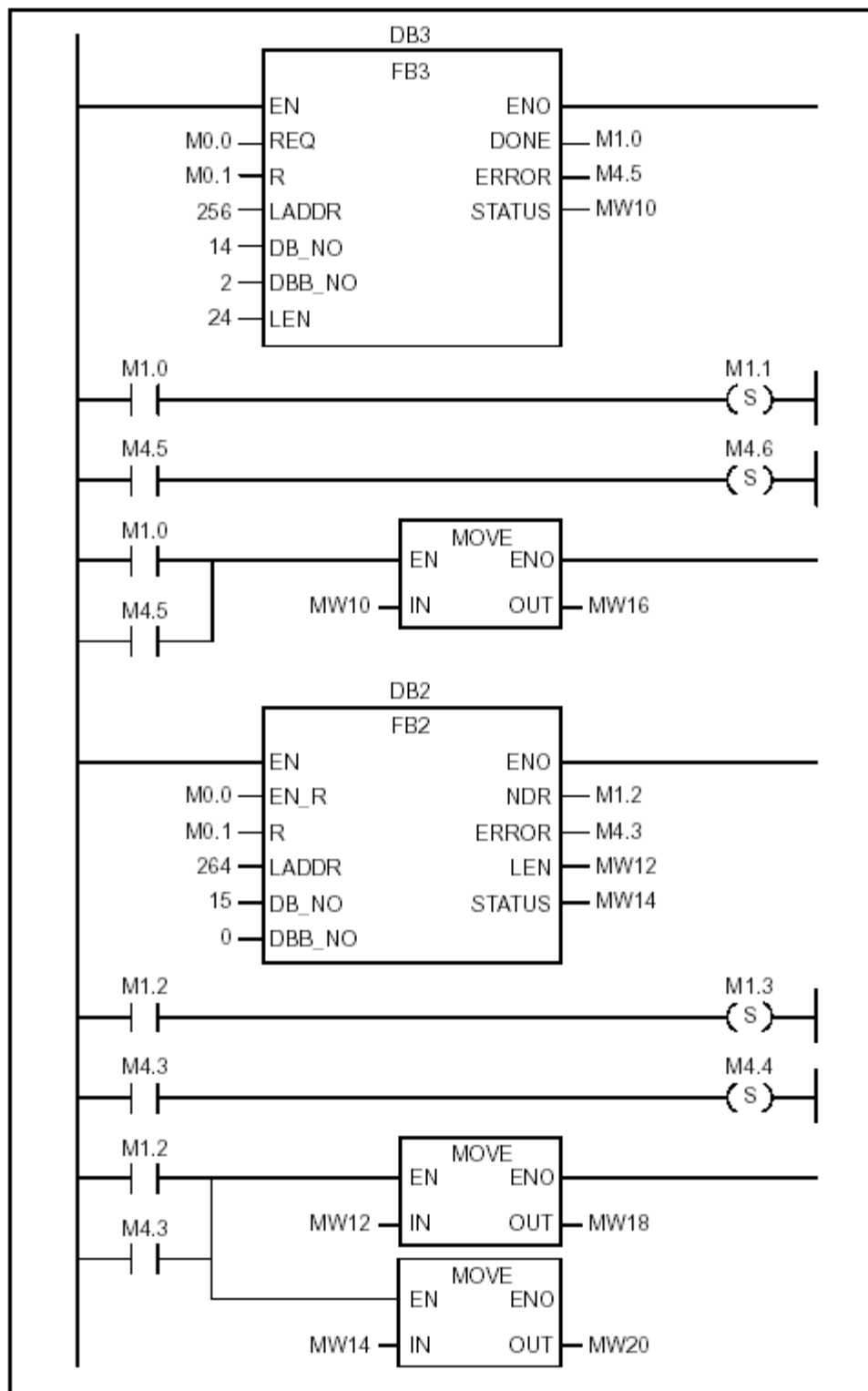


Рисунок 2–3 Пример программы для рассматриваемого случая применения

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 состоянии 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).

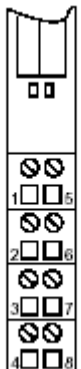
2.3 Схемы с указанием назначения контактов

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").

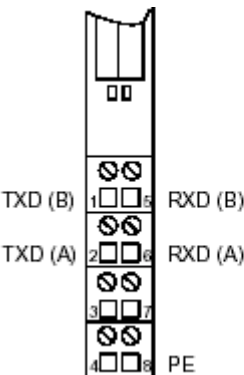
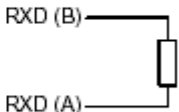
Назначение клемм для связи по RS-232C

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
		Режим: полудуплексный и 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

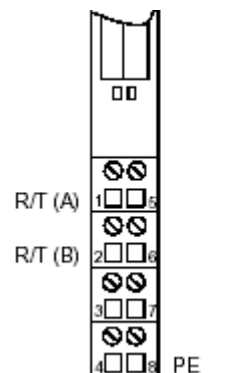
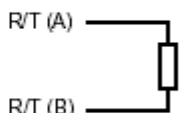
Назначение клемм для связи по RS-422

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
 TXD (B) 1 □ □ 5 TXD (A) 2 □ □ 6 RXD (B) 3 □ □ 7 RXD (A) 4 □ □ 8 PE	Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом. 	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
 R/T (A) 1 □ □ 5 R/T (B) 2 □ □ 6 RXD (B) 3 □ □ 7 RXD (A) 4 □ □ 8 PE	Примечание: Чтобы обмен данными происходил без ошибок, при использовании кабелей длиной свыше 50м следует включить нагрузочное сопротивление, приблизительно, 330 Ом. 	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.

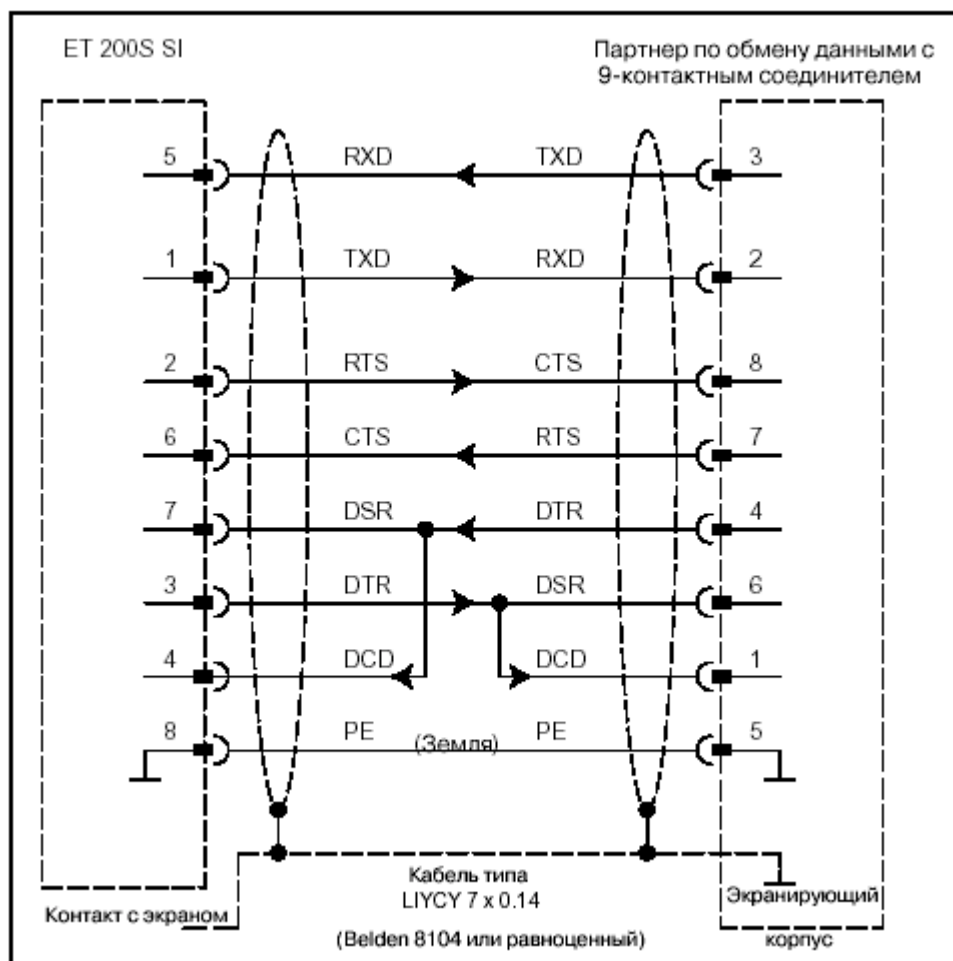


Рисунок 2-4 Разводка контактов соединительного кабеля RS-232C для 9-контактного разъема

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.

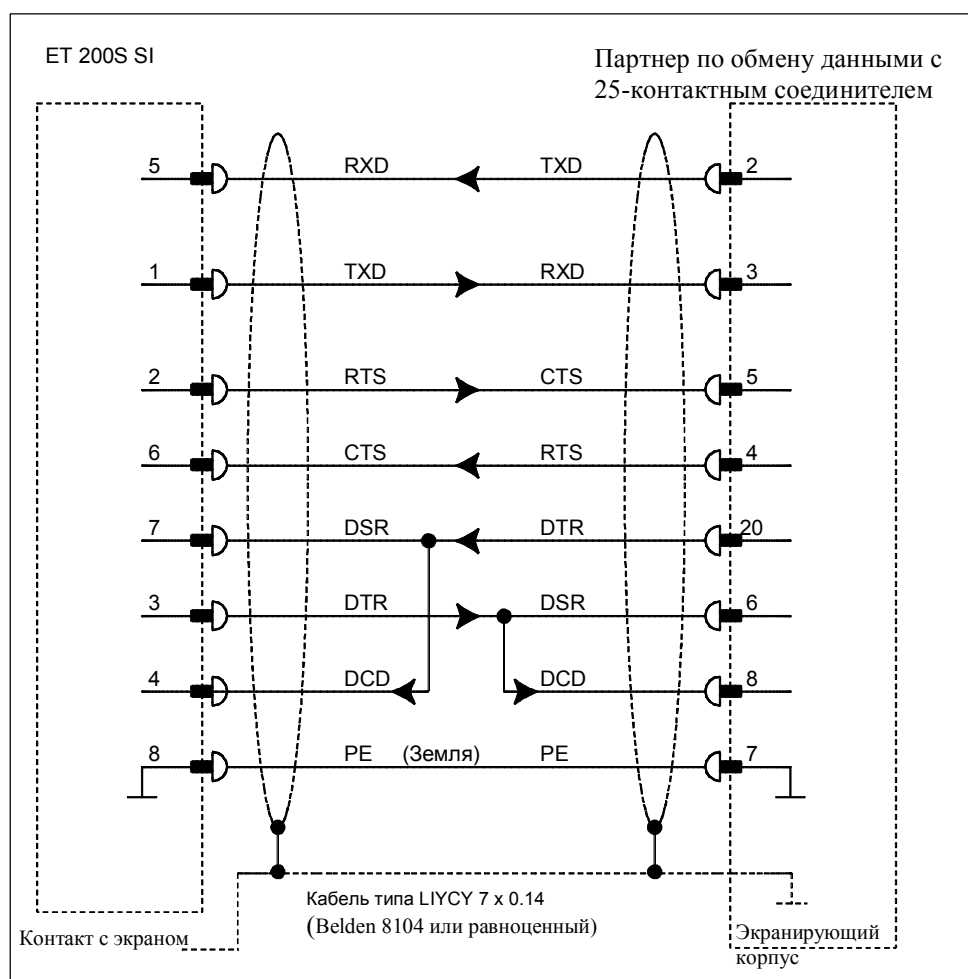


Рисунок 2-4 Разводка контактов соединительного кабеля RS-232C с 25-контактным разъемом

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.

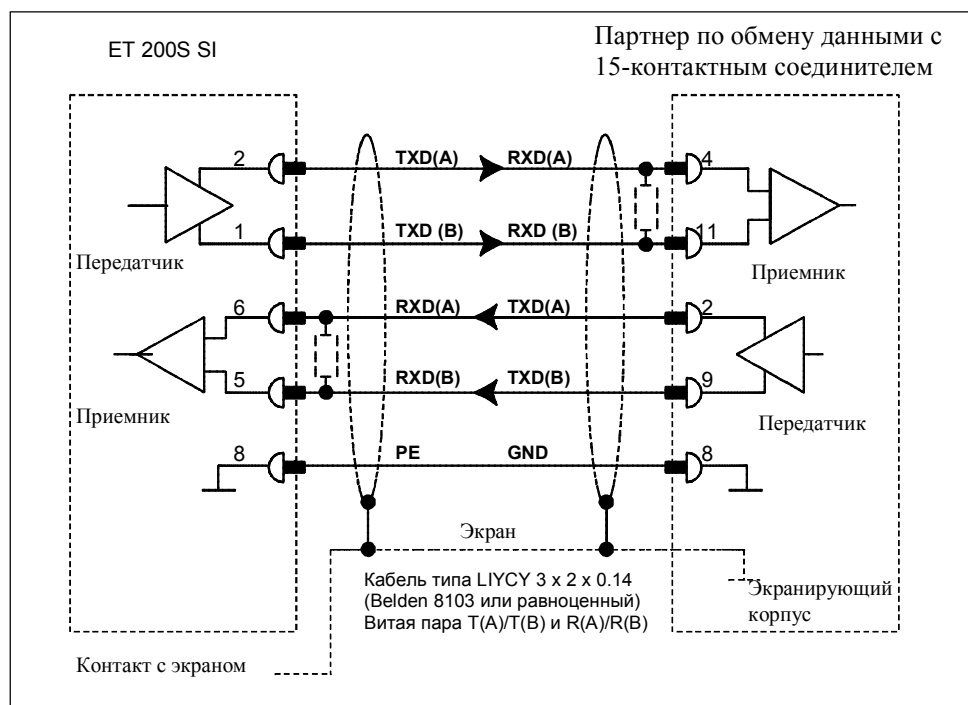


Рисунок 2-6 Разводка контактов соединительного кабеля RS-422 с 15-контактным разъемом

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.

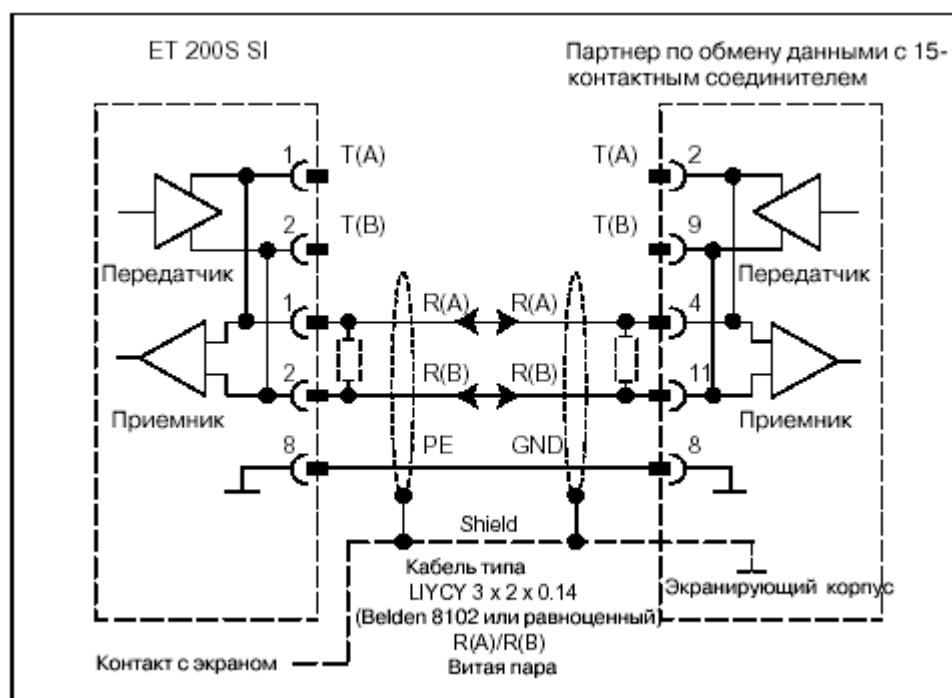


Рисунок 2-7 Разводка контактов соединительного кабеля RS-485C с 15-контактным разъемом

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.

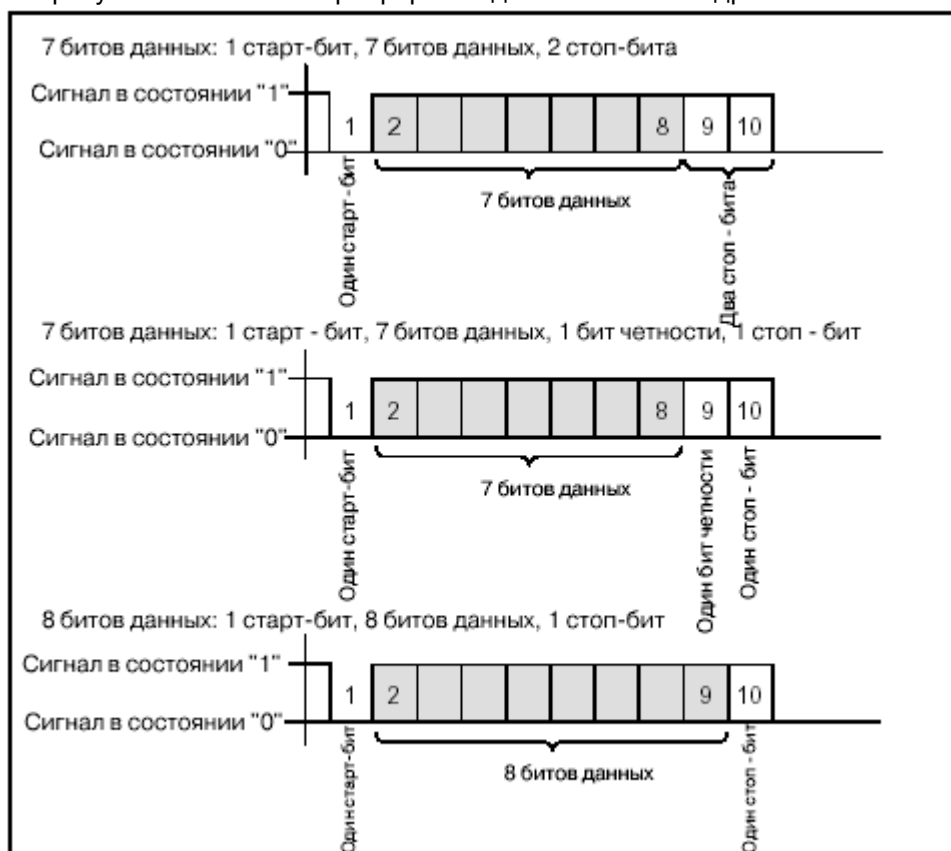


Рисунок 2–8 Кадр символа длиной 10 битов

Кадр символа длиной 11 битов

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

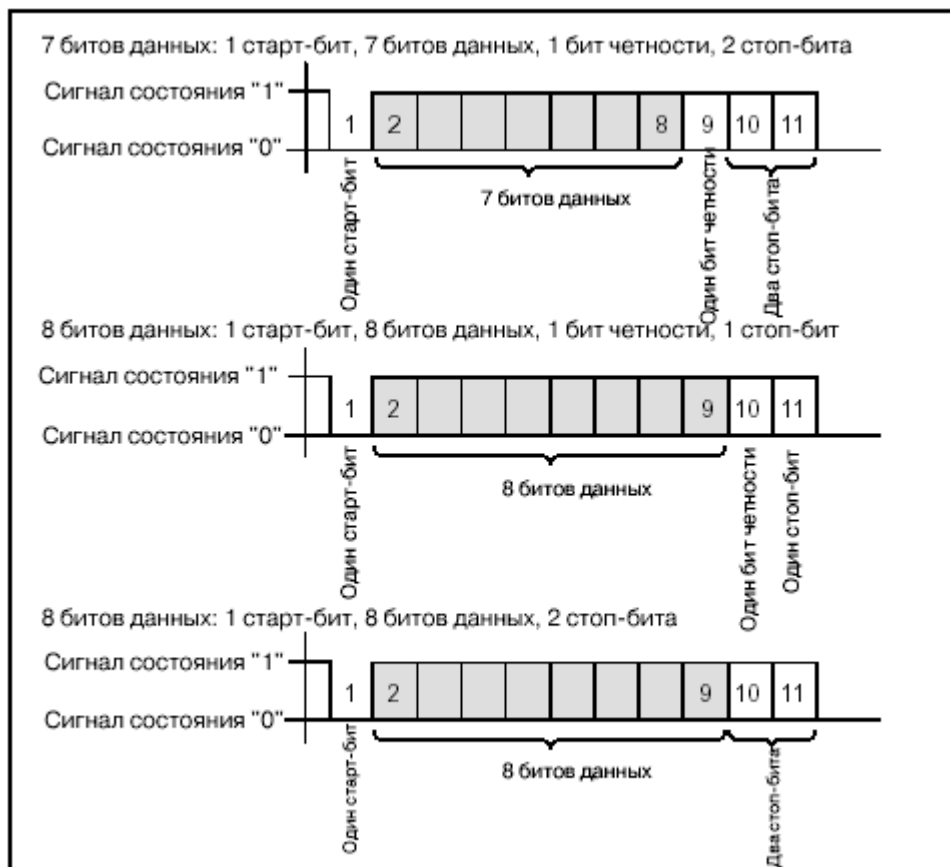


Рисунок 2-9 Кадр символа длиной 11 битов

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.

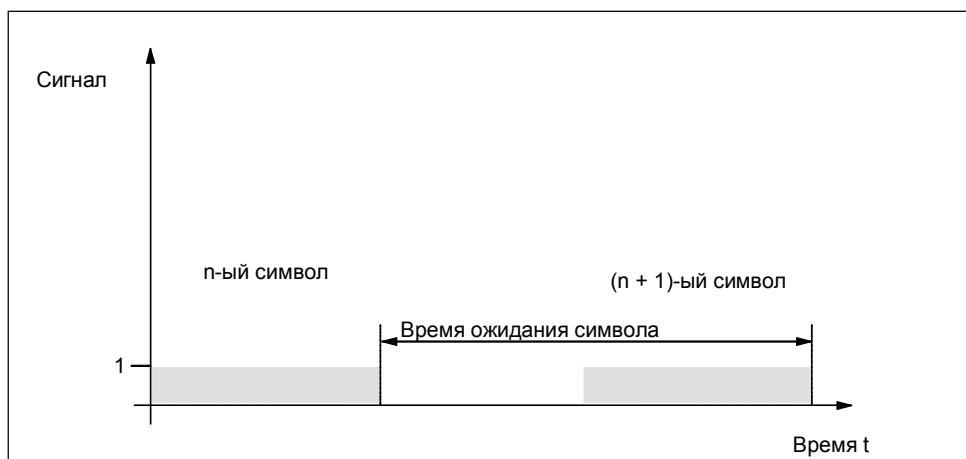


Рисунок 2–10 Время ожидания символа

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.

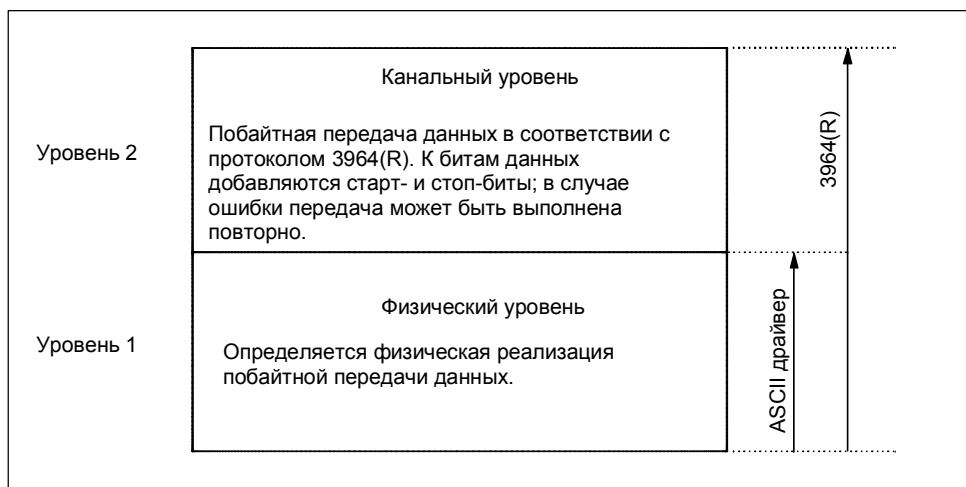


Рисунок 2–11 Место протоколов, поддерживаемых модулем ET 200 S SI, в эталонной модели ISO

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).

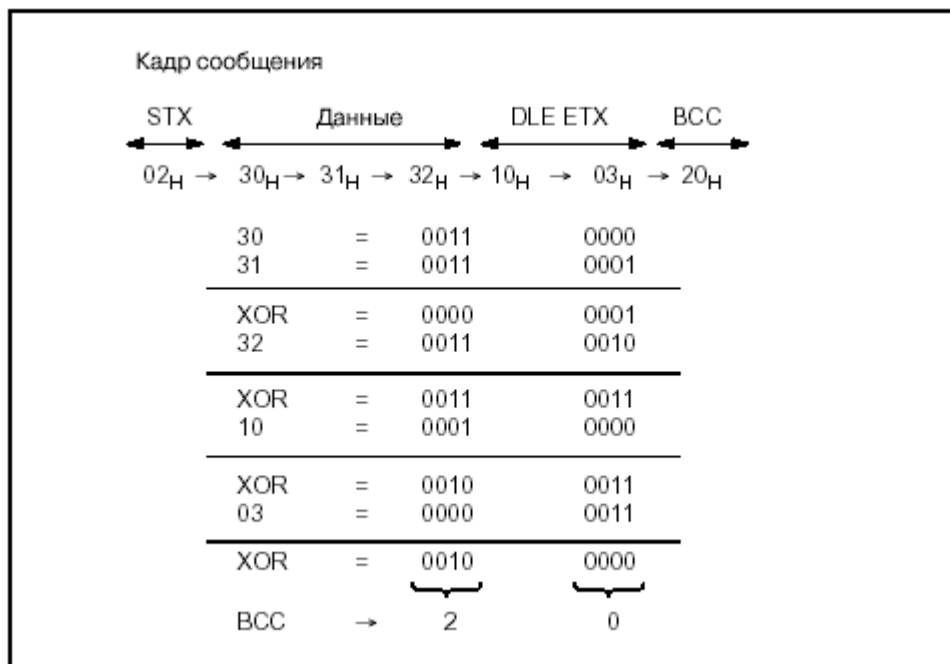


Рисунок 2–12 Контрольная сумма блока

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).

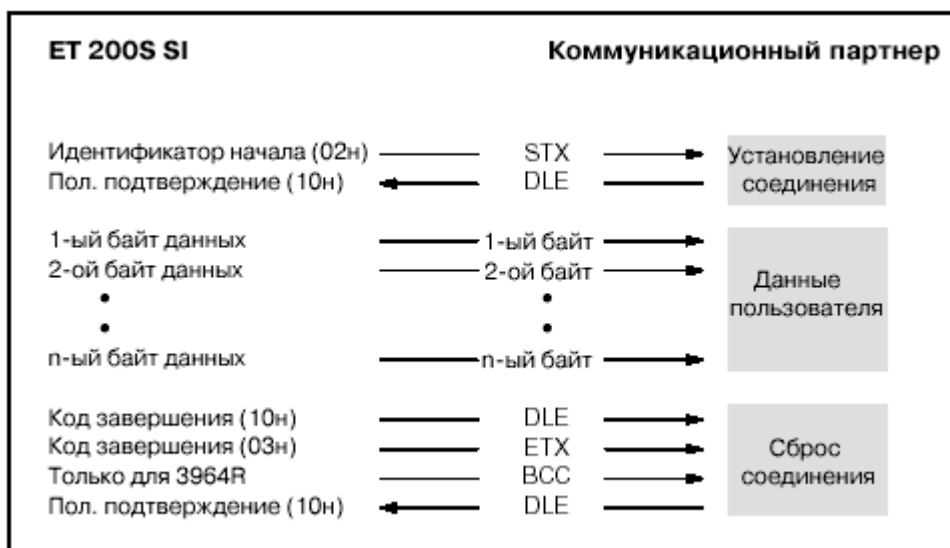


Рисунок 2–13 Последовательность передачи данных по протоколу 3964(R)

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 неуспешных 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).

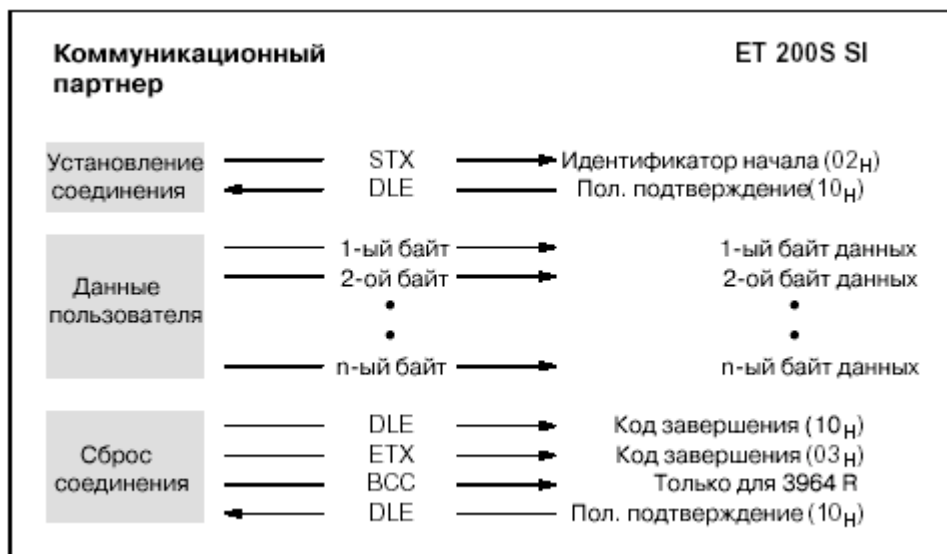


Рисунок 2–14 Поток данных при выполнении приема данных по протоколу 3964(R)

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).

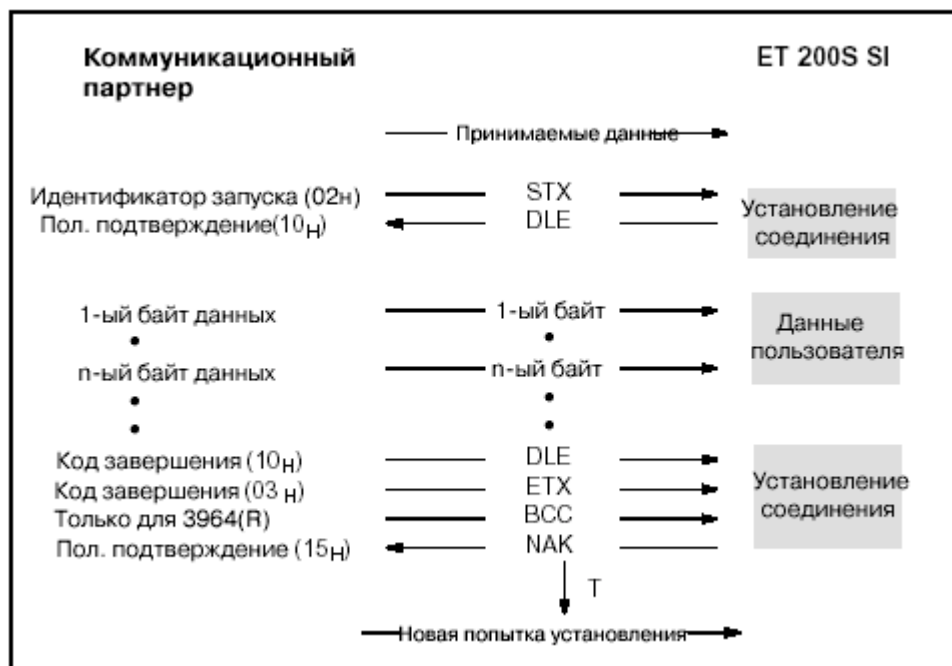


Рисунок 2–15 Последовательность передачи при приеме поврежденных данных

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.

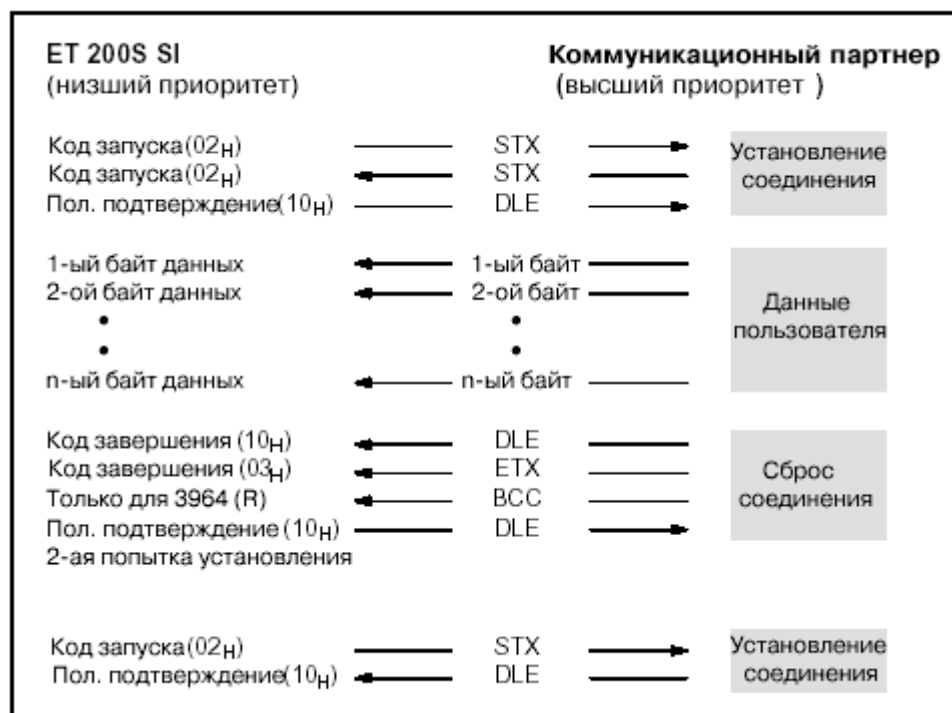


Рисунок 2–16 Последовательность передачи при конфликте инициализации

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_н (если используются кадры символа с 8-ю битами данных) или 00...7F_н (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_н, а в качестве XOFF – DC3 = 13_н.

Передача данных

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

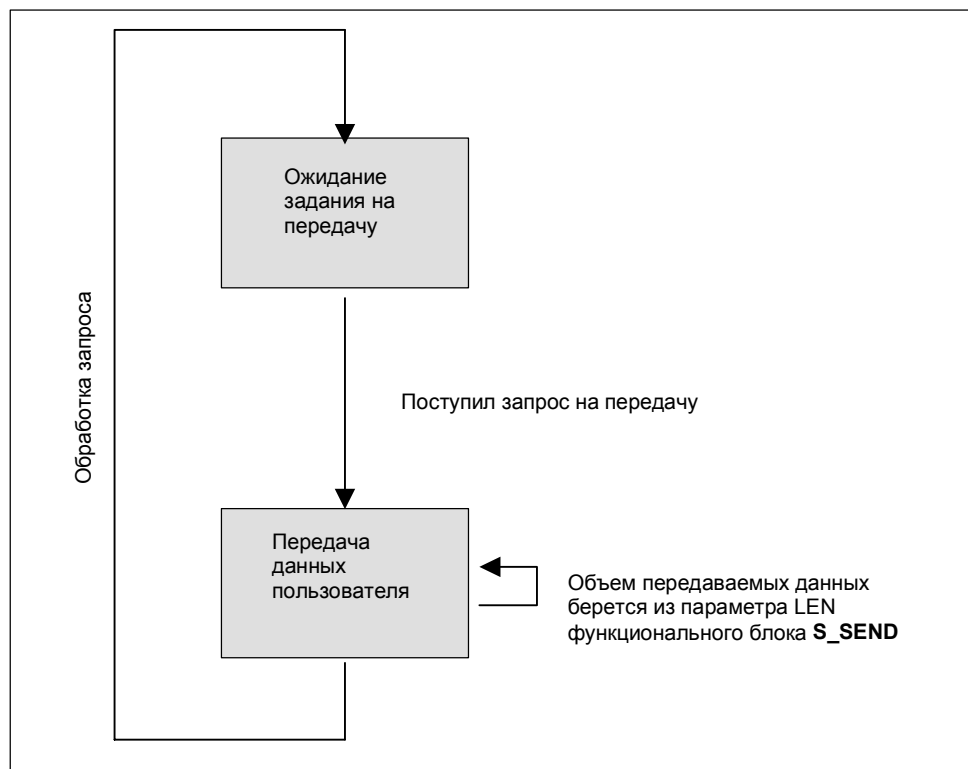


Рисунок 2–17 Последовательность действий при выполнении передачи

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".

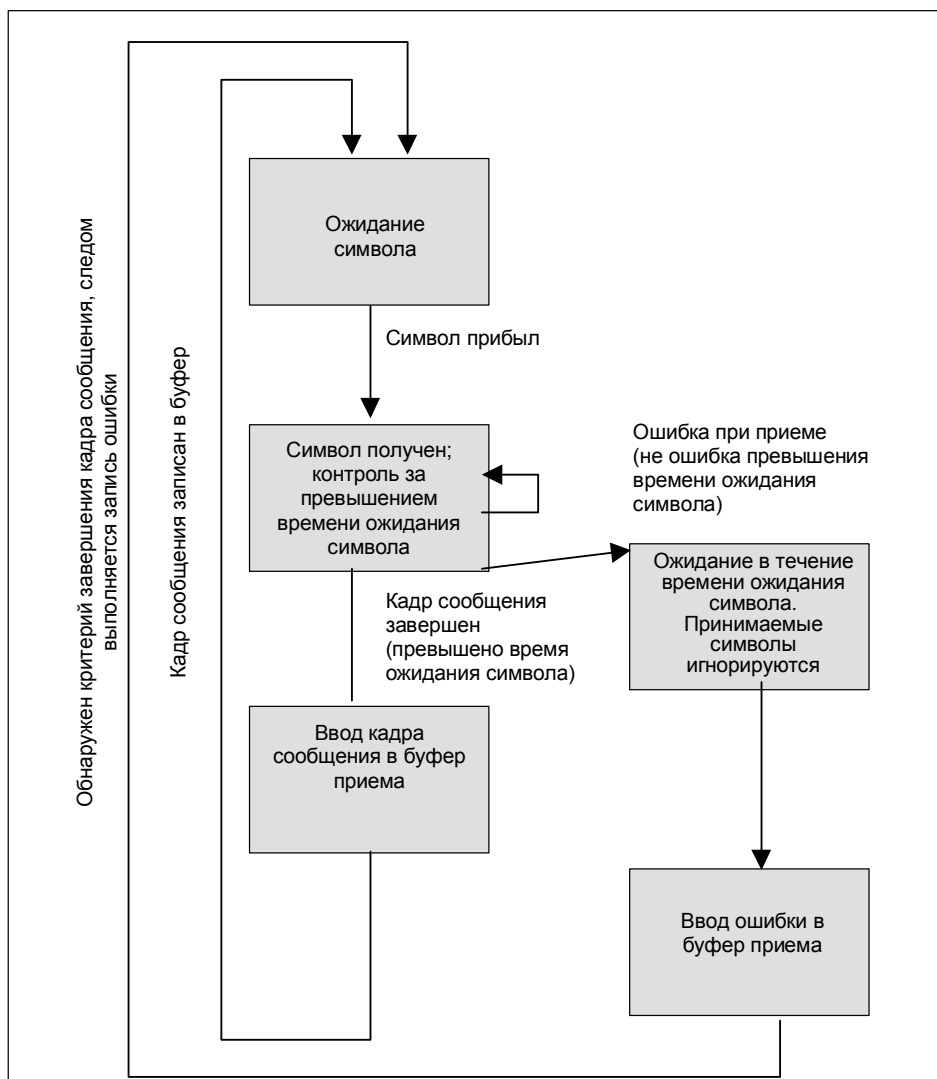


Рисунок 2–18 Последовательность действий при приеме данных с использованием критерия завершения "Превышение времени ожидания символа".

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".

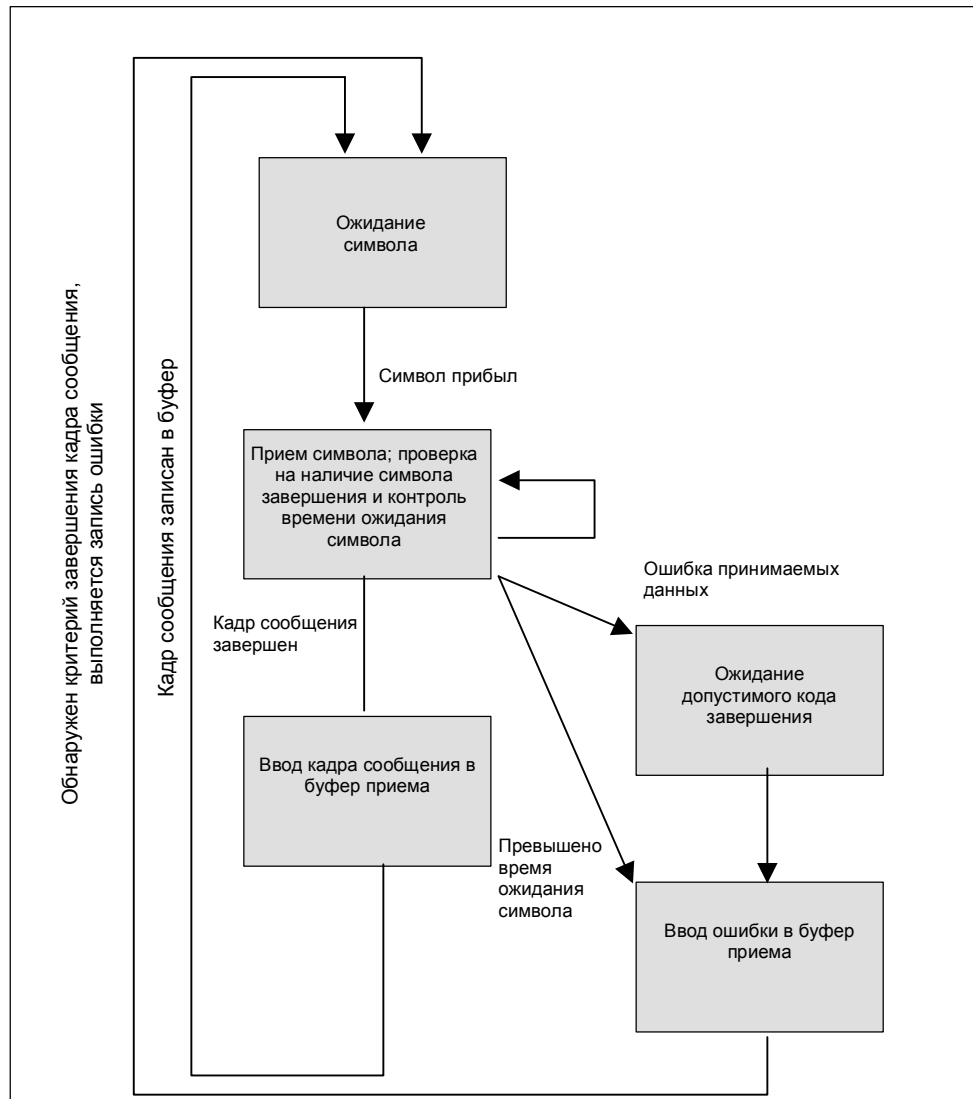


Рисунок 2–19 Последовательность действий при приеме данных с использованием критерия завершения "Символ текста"

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".

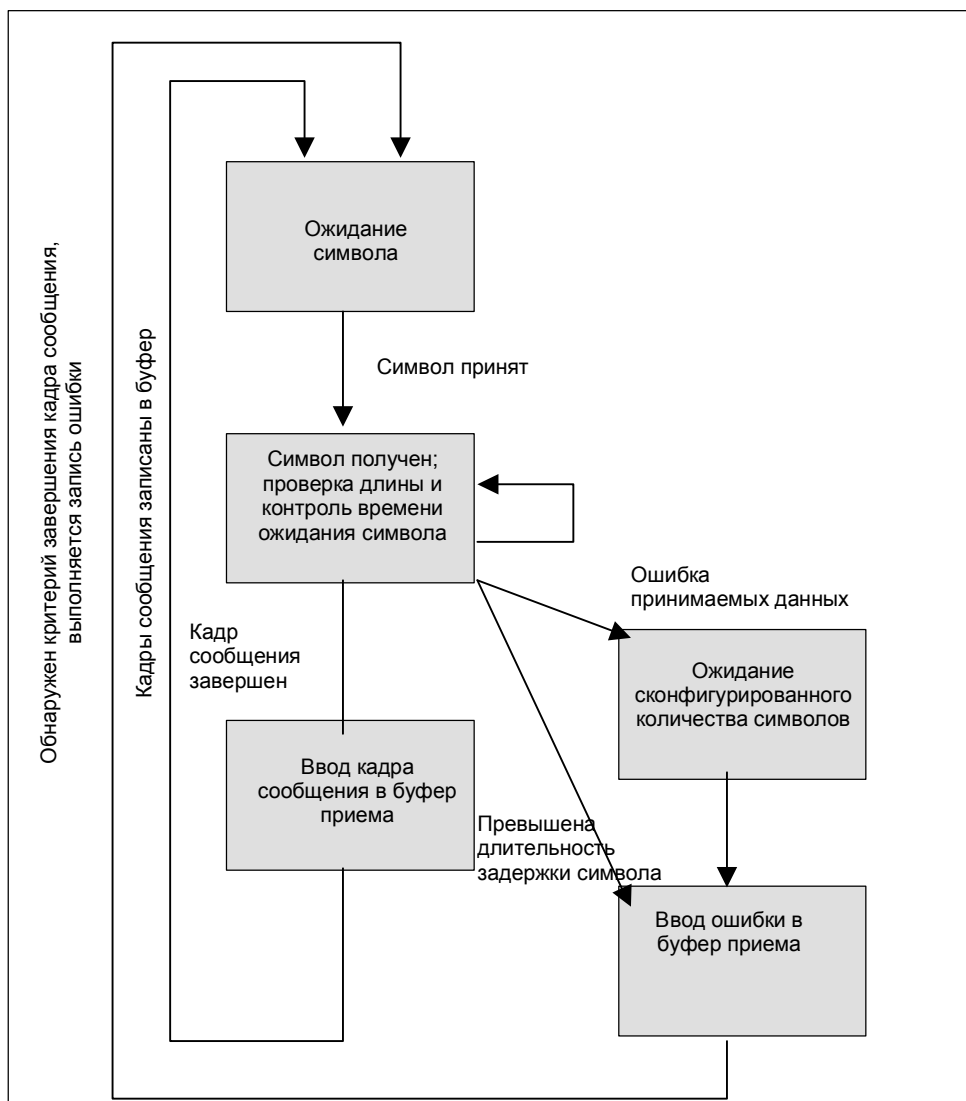


Рисунок 2–20 Последовательность действий при приеме данных с использованием критерия завершения "Фиксированная длина кадра сообщения".

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.

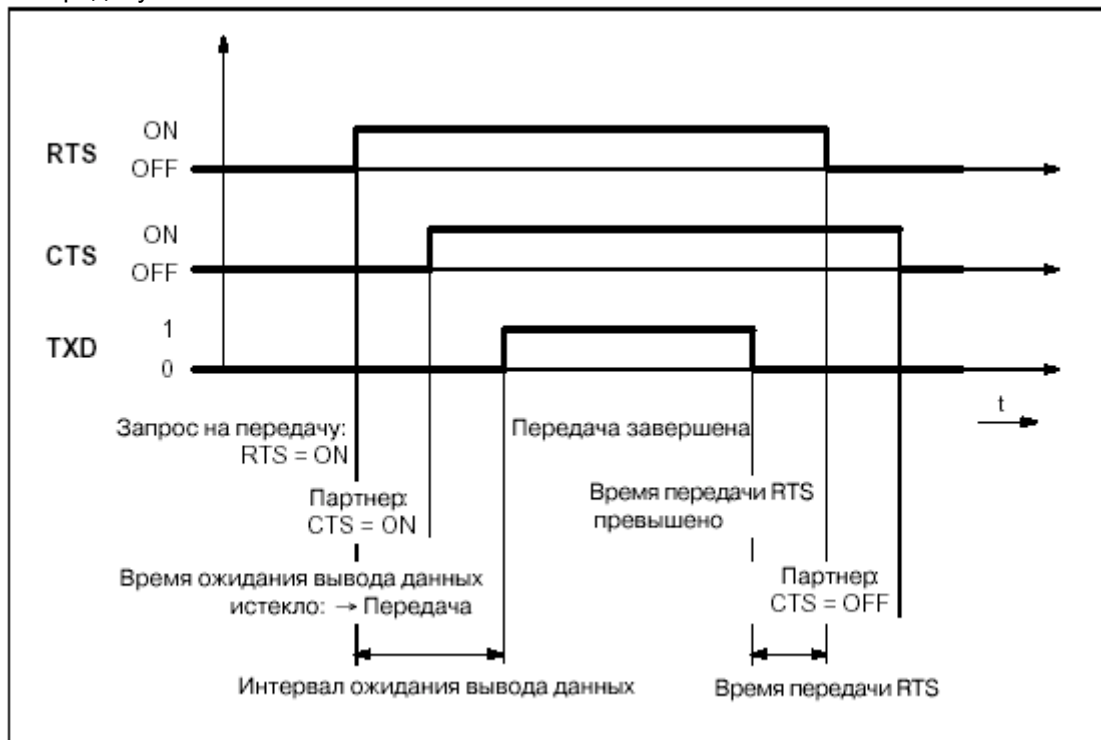


Рисунок 2-21 Временная диаграмма при автоматическом функционировании дополнительных сигналов RS-232C

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	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. Любой: состояние бита четности не играет роли. Четность не проверяется при приеме данных, но при передаче данных всегда устанавливается = "0".	None – нет Odd – нечет Even – чет Any – любой	Even – чет

Таблица 2–10 Параметры ASCII драйвера (продолжение)

Признак завершения принимаемого кадра сообщения	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. 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"). Превышение времени ожидания символа: признаком завершения кадра сообщения является истечение времени ожидания символа. По приему символа завершения текста: признаком завершения кадра сообщения является поступление заданного символа (-ов) завершения текста. По приему установленного количества символов: признаком завершения кадра сообщения является поступление кадра сообщения сконфигурированной длины. Все принимаемые кадры сообщений имеют одинаковую длину.	Превышение времени ожидания символа По приему символа завершения текста По приему установленного количества символов	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	Для приема данных с символами завершения текста можно выбрать до двух символов завершения текста. Выбранные символы завершения текста ограничивают длину кадра сообщения.	7 битов данных:³ 1 ... 7F_H 8 битов данных:³ 1 ... FF_H	3
Символ завершения текста 2 ²	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).	7 битов данных:³ 0 ... 7F_H 8 битов данных:³ 0 ... FF_H	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	Укажите, должен ли вместе с данными передаваться символ контроля блока (BCC), позволяющий повысить целостность передаваемых данных. 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. 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.	No block check (Без символа контроля блока) Block check (С символом контроля блока)	Block check (С символом контроля блока)
Скорость передачи	Выберите скорость передачи данных (бит/с).	110 300 600 1200 2400 4800 9600 19200	9600
Количество битов данных	Выберите количество битов, из которых будет состоять символ.	7 8	8
Стоп-биты	Выберите количество стоп-битов, которое будет добавляться к каждому передаваемому символу в процессе передачи для индикации 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. Любой: состояние бита четности не играет роли. Четность не проверяется при приеме данных, но при передаче данных всегда устанавливается = "0".	None – Нет Odd – Нечет Even – Чет Any - Любой	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 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 P_PRINT (см. стр. 2-58). Задания, иницированные 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.

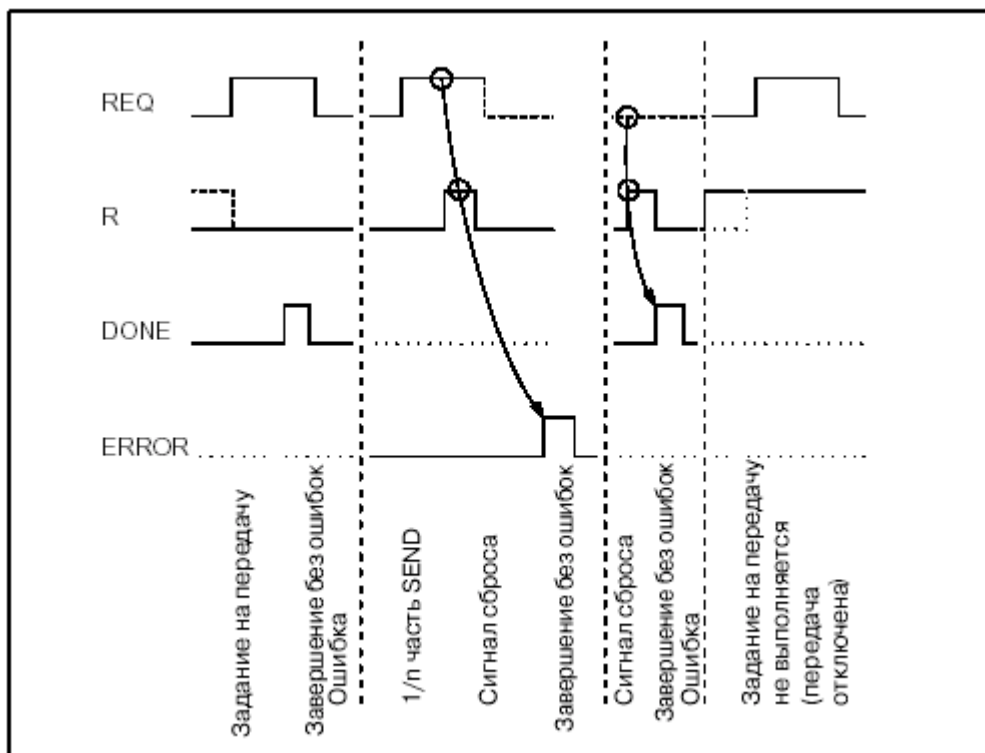


Рисунок 2-22 Временная диаграмма для FB3 S_SEND

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.

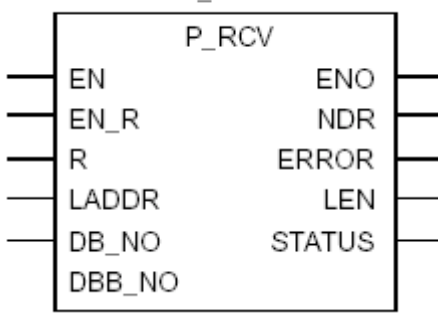
Выход NDR указывает, что задание было завершено без ошибок (все данные прочитаны). Выход ERROR указывает на наличие ошибки. В случае возникновения ошибки в выходном состоянии STATUS содержится соответствующий номер ошибки. Если приемный буфер заполнен более чем на 2/3, на выход STATUS поступает предупреждение 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.

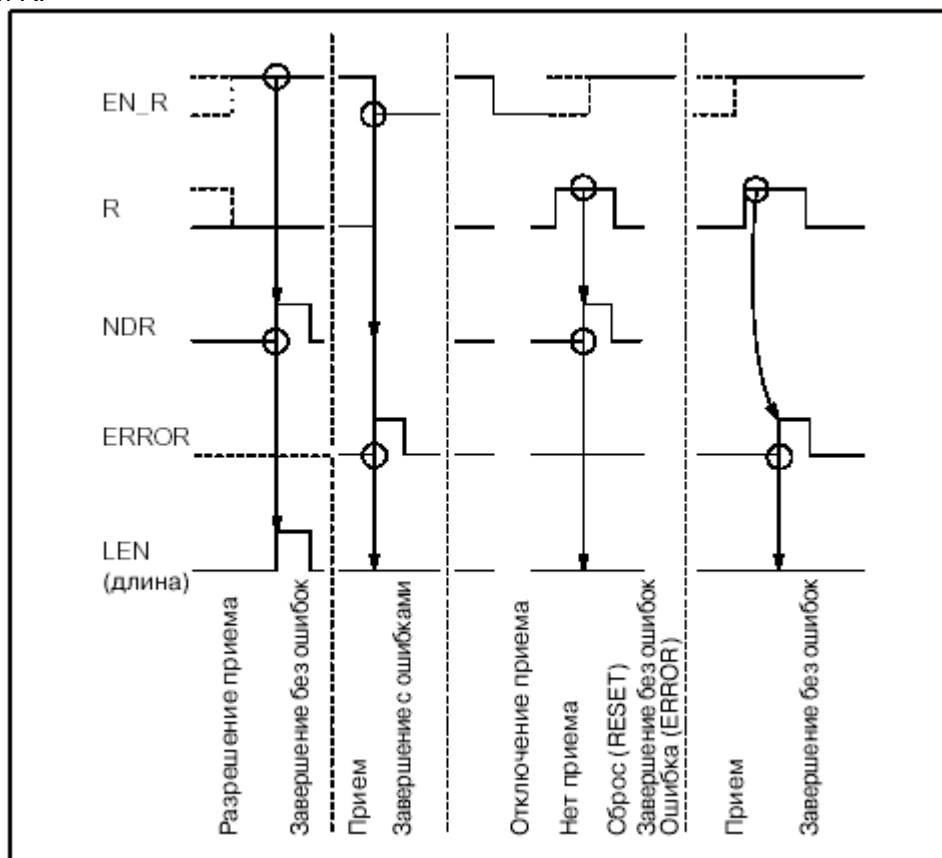


Рисунок 2-23 Временная диаграмма для FB2 S_RCV

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.

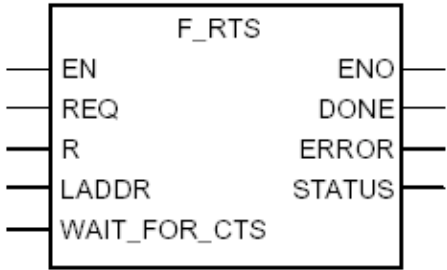
Параметры FB6 S_XON

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

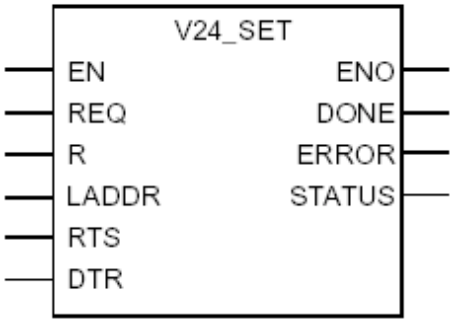
В таблице 2-18 приводятся параметры функционального блока S_VSTAT (FB4).

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.

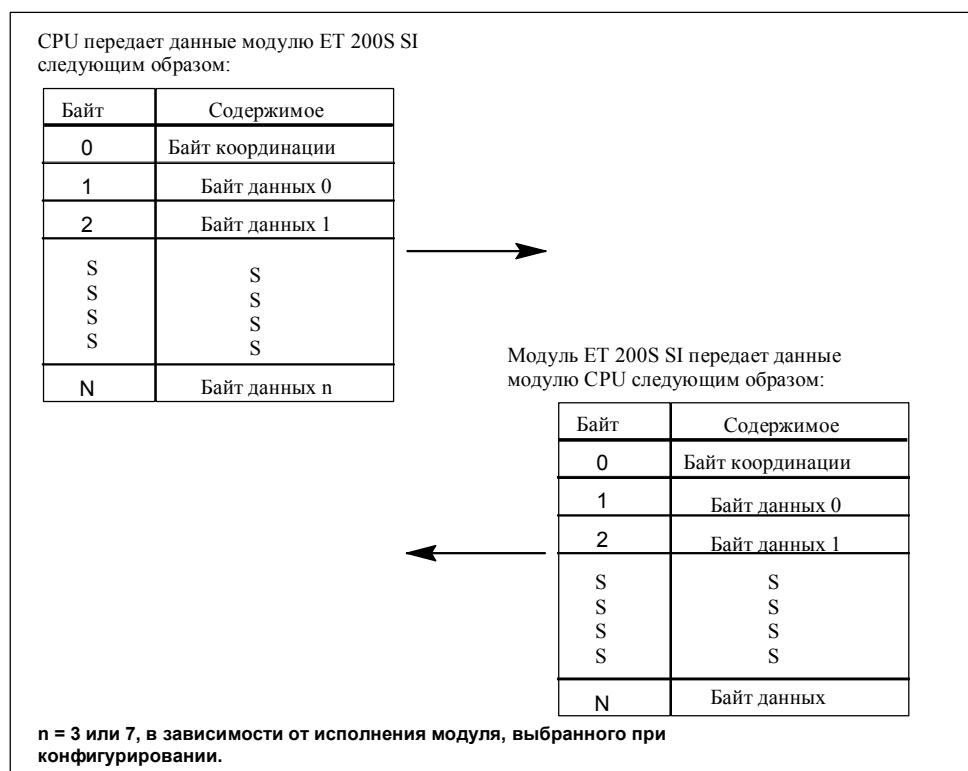


Рисунок 2–24 Обмен данными между CPU и модулем ET200S SI

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	Бит 7 6 5 4 3 2 1 0 Рез. Код задания Ошибка Номер исполнения
Бит 7	Резерв
Код задания	Устанавливается модулем CPU для инициирования задания
Порядковый номер исполнения	Задание на передачу: увеличивается модулем CPU на 1, когда CPU передает следующий сегмент модулю ET 200S SI...или 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).
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	Бит 7 6 5 4 3 2 1 0 Рез. Код задания Ошибка Номер исполнения
Бит 7	Резерв
Код задания	Принимается модулем ET 200S SI в качестве подтверждения того, что задание было принято.
Порядковый номер исполнения	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. Задание на прием: увеличивается модулем на 1, когда модуль передает другой сегмент модулю CPU (значение находится в пределах 1...7).
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:				Байт 01234567 <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>								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																										
					Байт 01234567 <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>								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:				Байт 01234567 <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>								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																										
	CPU повторяет задание на передачу :				Байт 01234567 <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>								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:				Байт 01234567 <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>								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:	← <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>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> 0000_H = принятых сообщений нет 0001_H = имеется принятое сообщение 0B01_H = буфер приема переполнен на 2/3	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 :																														
	Байт <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> →	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)	Программа пользователя получает от модуля следующий ответ (модуль подтверждает прием, возвращает в ответ первый сегмент и увеличивает на 1 порядковый номер исполнения): <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>	21H	0006H	'a'	'b'	'c'	'd'	'e'	→	Квит. зад.	Длина	Данные																																		
21H	0006H	'a'	'b'	'c'	'd'	'e'	→																																							
Квит. зад.	Длина	Данные																																												
	CPU записывает задание, чтобы подтвердить первый сегмент: <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>	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)	Программа пользователя читает 2-ой сегмент от модуля <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> <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>	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)	Модуль возвращается в состояние ожидания по завершении первой операции приема <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>	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																															
		<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																														
2-ой	Программа пользователя получает от модуля следующий ответ:	Байт 0 1 2 3 4 5 6 7																													
		<table><tr><td>31H</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>Сигналы</td><td colspan="6">Не имеет значения</td></tr></table>								31H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Сигналы	Не имеет значения											
		31H	nnnnH	xxH	xxH	xxH	xxH	xxH	xxH																						
Квит. зад.	Сигналы	Не имеет значения																													
<table><tr><td colspan="4">Старш. байт</td><td colspan="4">Младш. байт</td></tr><tr><td>00</td><td>0</td><td>0</td><td>0</td><td>DCD</td><td>CTS</td><td>RTS</td><td>DSR</td></tr><tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>								Старш. байт				Младш. байт				00	0	0	0	DCD	CTS	RTS	DSR		7	6	5	4	3	2	1
Старш. байт				Младш. байт																											
00	0	0	0	DCD	CTS	RTS	DSR																								
	7	6	5	4	3	2	1																								
	CPU записывает подтверждение и принимает порядковый номер исполнения:	Байт 0 1 2 3 4 5 6 7																													
		<table><tr><td>31H</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>								31H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	Задание	Не имеет значения												
31H	xxH	xxH	xxH	xxH	xxH	xxH	xxH																								
Задание	Не имеет значения																														
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>ppppH</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	ppppH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		00H	ppppH	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>ppppH</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	ppppH	xxH	xxH	xxH	xxH	xxH	xxH	Квит. зад.	Состояние	Не имеет значения					
		40H	ppppH	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>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>								00H	xxH	xxH	xxH	xxH	xxH	xxH	xxH	Задание	Не имеет значения						
		00H	xxH	xxH	xxH	xxH	xxH	xxH	xxH																
Задание	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 _н	
2 и 3	Длина	0004 _н	0004 _н
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 _н	
2 и 3	Длина	0002 _н	0002 _н
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 _н	
2 и 3	Длина	0004 _н	0004 _н
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>0004 н</td></tr><tr><td>Задание</td><td>Поток данных</td><td>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 (предложение)												
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).

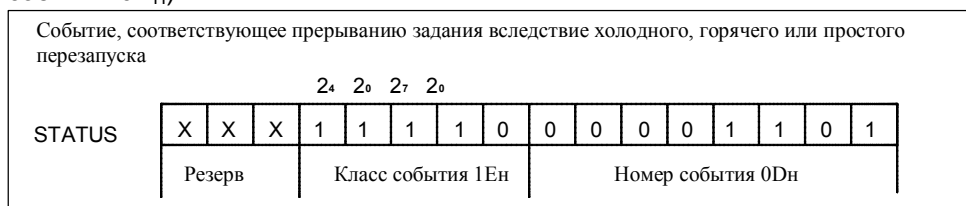


Рисунок 2–26 Пример: параметр STATUS для класса события 1Е_н, событие 0D_н

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: - 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: - 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: - 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: - После приема символа следующий символ не был принят в пределах времени ожидания символа, или Только для 3964(R): - 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: - Обнаружена ошибка передачи (ошибка четности, ошибка стоп-бита, ошибка переполнения). Только для 3964(R): - 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 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 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	
Сопrotивление 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