

## 6 Двухточечное соединение

### 6.1 Обзор

#### 6.1.1 Описание продукта

С помощью последовательного интерфейса вы можете через двухточечное соединение обмениваться данными между системами автоматизации, компьютерами или простыми устройствами. Обмен данными между участвующими устройствами происходит на основе последовательной асинхронной передачи.

Встроенный последовательный интерфейс CPU 313/314C-2PtP предоставляет доступ к обмену данными через интерфейс X27 (RS422/485).

В качестве протоколов в вашем распоряжении имеются:

- CPU 313C-2PtP: ASCII, 3964(R)
- CPU 314C-2PtP: ASCII, 3964(R) и RK 512

Вид обмена данными устанавливается через экранные формы для параметризации.

Вы можете передать не более 1024 байтов. В качестве скоростей передачи возможны при полнодуплексной связи 19,2 кБод, при полудуплексной связи 38,4 кБод.

#### 6.1.2 Партнеры по обмену данными

Последовательный интерфейс CPU делает возможным двухточечное соединение с различными модулями фирмы Siemens и продуктами других фирм. Ниже приведены некоторые примеры:

- SIMATIC S5 через 3964(R)/RK 512 с соответствующим интерфейсным модулем на стороне S5
- BDE-терминалы Siemens семейства ES 2 через драйвер 3964(R)
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) и станция сбора и обработки данных ES 030K через драйвер 3964(R)
- SIMOVERT и SIMOREG (протокол USS) через драйвер ASCII (ET 200S SI RS 422/485) с соответствующей адаптацией протокола с помощью программы STEP 7
- Персональные компьютеры через процедуру 3964(R) (для этого существуют инструментальные средства разработки для программирования на PC: PRODAVE DOS 64R (6ES5 897-2UD11) для MS-DOS, PRODAVE WIN 64R (6ES5 897-VD01) для Windows или драйвер ASCII)

- Считыватель штрихового кода через драйвер 3964(R) или драйвер ASCII
- Программируемые контроллеры других фирм через RK512, драйвер 3964(R) или драйвер ASCII
- Другие устройства с простыми структурами протокола через соответствующую адаптацию протокола с помощью драйвера ASCII
- Другие устройства, которые тоже снабжены драйвером 3964(R)/RK 512

### 6.1.3 Компоненты для двухточечного соединения

В **CPU** встроены протоколы для последовательного соединения. Через этот последовательный интерфейс подключается ваш партнер по обмену данными.

В качестве **соединительного кабеля** применяется экранированный кабель. В разделе 6.10.7 описаны соединительные кабели для различных партнеров по обмену данными.

В качестве **партнеров по обмену данными** можно подключать устройства, которые снабжены интерфейсом RS422/485 и владеют соответствующим протоколом.

С помощью **PG/PC**

- Выполните параметризацию CPU, используя экранные формы параметризации для технологических функций CPU.
- Запрограммируйте CPU с помощью системных функциональных блоков, которые вы можете включить непосредственно в программу пользователя.
- Введите CPU в действие и протестируйте его CPU с помощью стандартного пользовательского интерфейса STEP7 (Функции наблюдения и таблица переменных).

### 6.1.4 Свойства интерфейса X27 (RS 422/485)

#### Определение

Интерфейс X27 (RS 422/485) - это интерфейс, использующий разность потенциалов, который служит для последовательной передачи данных в соответствии со стандартом X27.

В режиме RS422 передача данных осуществляется через четыре провода (четырехпроводный режим). В распоряжении имеются по два провода (разностный сигнал) для направления передачи и направления приема. Благодаря этому передачу и прием можно вести одновременно (полнодуплексный режим).

В режиме RS485 передача данных осуществляется через два провода (двухпроводный режим). Эти два провода (разностный сигнал) используются попеременно для направления передачи и направления приема. Из-за этого можно вести только передачу или только прием (полудуплексный режим). После процесса передачи немедленно производится переключение на прием (передатчик высокоомный).

Выбор режима осуществляется через экранные формы для параметризации.

## Свойства

Интерфейс X27 (RS 422/485) обладает следующими свойствами и удовлетворяет следующим требованиям:

- Вид: Интерфейс, использующий разность потенциалов
- Фронтштекер: 15-контактная миниатюрная D-образная розетка с винтовым фиксатором
- Макс. скорость: 38,4 кБод (полудуплексный режим)
- Стандарт: DIN 66259, части 1 и 3, EIA-RS 422/485, CCITT V.11

### 6.1.5 Последовательная передача символа

Для обмена данными между двумя или несколькими партнерами имеются в распоряжении различные возможности объединения в сеть. Двухточечное соединение между двумя партнерами по обмену данными является простейшим случаем обмена информацией. Передача данных при двухточечном соединении осуществляется последовательно.

#### Последовательная передача данных

При последовательной передаче данных отдельные биты байта подлежащей передаче информации передаются друг за другом в определенной последовательности. Передача данных партнеру по обмену данными осуществляется через последовательный интерфейс автоматически. Для этого CPU оснащен тремя различными драйверами.

- драйвер ASCII
- процедура 3964(R)
- RK 512

#### Полудуплексный/полнодуплексный режим

При передаче данных различают:

- Полудуплексный режим (драйверы ASCII, процедура 3964(R), RK 512)  
данные между партнерами передаются попеременно в обоих направлениях. Полудуплексный режим означает, что в каждый данный момент времени ведется только передача или только прием. Исключением могут быть только отдельные управляющие символы для управления потоком данных (напр., XON/XOFF), которые могут приниматься и передаваться также в ходе одного режима приема/передачи.
- Полнодуплексный режим (драйвер ASCII)  
Обмен данными между партнерами производится одновременно, т.е. в один и тот же момент времени могут вестись передача и прием. Каждый партнер по обмен данными должен иметь возможность одновременно управлять направлением передачи и направлением приема.

При настройке RS 485 (2-проводный режим) возможна работа только в полудуплексном режиме с драйвером ASCII без управления потоком данных.

## Асинхронная передача данных

Последовательная передача данных производится асинхронно. Так называемый синхронизм (фиксированный временной растр при передаче фиксированной последовательности символов) поддерживается только во время передачи одного символа. Каждому передаваемому символу предшествует синхронизирующий импульс, называемый также стартовым битом. Передача символа завершается стоповым битом.

## Соглашения

Кроме стартового и стопового бита, для последовательной передачи данных между двумя партнерами необходимы дополнительные соглашения. К ним относятся

- скорость передачи,
- время задержки символа (CDT) и, при необходимости, квитирования,
- четность,
- количество битов данных и
- количество стоповых битов.

## Кадр для передачи символа

Данные через последовательный интерфейс передаются в виде кадра. Для каждого кадра имеются в распоряжении два формата данных. Кадр с 7 битами данных без бита четности не поддерживается. Желаемый формат передачи данных можно установить с помощью экранной формы для параметризации.

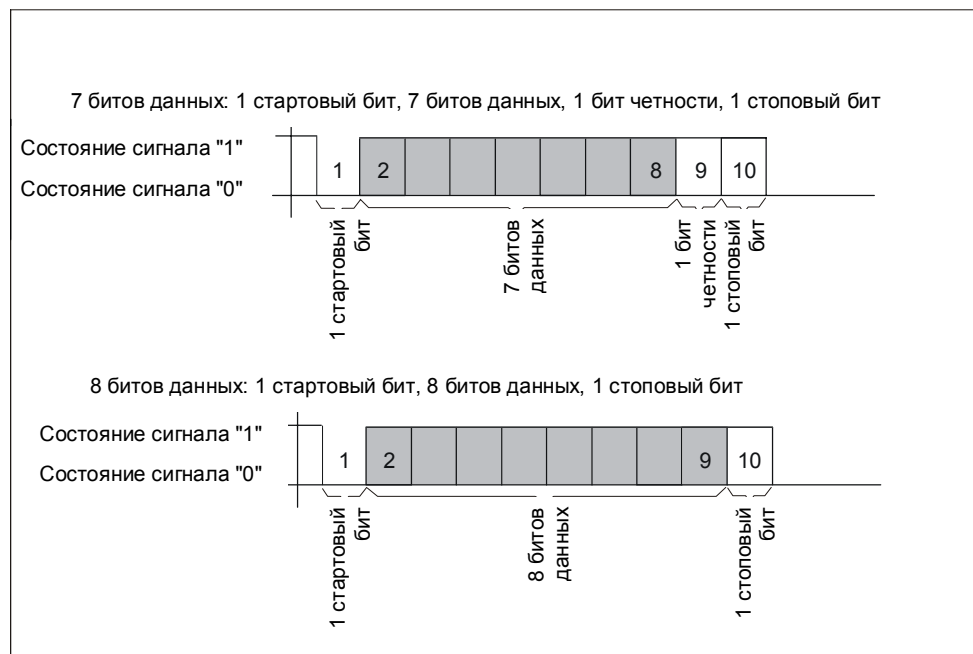
---

### Замечание

Кадр с 7 битами данных без бита четности не поддерживается.

---

Например, на следующем рисунке представлены два формата данных 10-битового кадра для передачи символа:



### Время задержки символа (CDT)

На следующем рисунке представлен максимально допустимый временной интервал между двумя принимаемыми символами внутри кадра сообщения = время задержки символа:



## **6.2 Подключение**

### **6.2.1 Правила подключения**

#### **Соединительный кабель**

- Кабели должны быть экранированными.
- Экраны кабелей должны быть с обеих сторон присоединены к зажимам.

#### **Зажим для экрана**

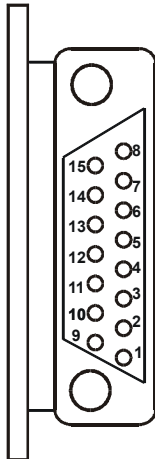
С помощью зажима для экрана можно соединить все экранированные кабели с землей путем прямого соединения с профильной шиной.

#### **Дальнейшие указания**

Дальнейшие указания вы найдете в руководстве "Данные CPU " и в руководстве по установке вашего CPU.

## 6.2.2 Присоединение последовательного кабеля

В следующей таблице вы найдете назначение контактов 15-контактной миниатюрной D-образной розетки на передней панели CPU.

Розетка RS 422/485 (вид спереди)	Кон- такт	Обозначение	Вход/выход	Описание
	1	-	-	-
	2	T (A) -	Выход	Передаваемые данные (четырёхпроводный режим)
	3	-	-	-
	4	R (A) - R (A)/T (A) -	Вход Вход/выход	Принимаемые данные (четырёхпроводный режим) Принимаемые/передаваемые данные (двухпроводный режим)
	5	-	-	-
	6	-	-	-
	7	-	-	-
	8	GND	-	Рабочее заземление (потенциально свободное)
	9	T (B) +	Выход	Передаваемые данные (четырёхпроводный режим)
	10	-	-	-
	11	R (B) + R (B)/T (B) +	Вход Вход/выход	Принимаемые данные (четырёхпроводный режим) Принимаемые/передаваемые данные (двухпроводный режим)
	12	-	-	-
	13	-	-	-
	14	-	-	-
	15	-	-	-

Если вы сами готовите соединительные кабели, учтите, что можно применять только экранированные корпуса штекеров. Экран кабеля должен быть с двух сторон на большой площади соединен с корпусом штекера и экранирующим проводом.



### Осторожно

Никогда не соединяйте экран кабеля с контактом GND, так как это может привести к повреждению интерфейса.

GND (контакт 8) в любом случае должен быть подключен на обеих сторонах, так как иначе возможно повреждение интерфейса.

Распределение контактов различных соединительных кабелей для компонентов Simatic-S7 и Simatic-S5 вы найдете в разделе 6.10.7.

## 6.3 Параметризация

С помощью параметризации вы настраиваете последовательную связь на свое конкретное приложение. Параметризация производится через два различных вида параметров:

### 1. Параметры модуля

Здесь речь идет об основных настройках, которые можно определить один раз, а затем, во время работы, нельзя изменить. Описание этих параметров вы найдете в данном разделе.

- Параметризация производится помощью экранных форм для параметризации.
- Сохранение происходит в памяти системных данных CPU.

---

#### Замечание

Изменение этих параметров, когда CPU находится в состоянии RUN, невозможно.

---

### 2. Параметры SFB

Параметры, которые должны изменяться во время работы, находятся в экземплярном DB системного функционального блока (SFB). Описание параметров SFB вы найдете в разделе 6.5.

- Параметризация выполняется в режиме offline в редакторе DB или online в программе пользователя.
- Сохранение производится в рабочей памяти CPU.
- Изменения этих параметров можно производить в режиме RUN CPU из программы пользователя.

### Экранные формы для параметризации

С помощью экранных форм для параметризации можно устанавливать параметры протокола:

Экранные формы для параметризации не требуют больших пояснений. Описание параметров вы найдете в соответствующих разделах и во встроенной помощи к экранным формам для параметризации.



## Процесс параметризации

Вызов экранных форм для параметризации предполагает, что вы создали проект, в котором вы можете сохранить параметризацию.

1. Запустите Администратор SIMATIC (SIMATIC Manager) и вызовите в своем проекте Конфигуратор аппаратуры.
2. Дважды щелкните на submodule "PtP [точка-точка]" своего CPU. Вы попадете в диалоговое окно "Properties [Свойства]".
3. Выполните параметризацию submodule "PtP", завершите экранную форму параметризации с помощью **OK**.
4. Сохраните свой проект в HW Config командой меню **Station > Save and compile [Станция > Сохранить и скомпилировать]**.
5. Загрузите данные параметризации в CPU в состоянии STOP с помощью **PLC > Download to module... [ПЛК > Загрузить в модуль...]**. Теперь данные находятся в системной памяти CPU.
6. Выполните запуск CPU.

## Встроенная помощь

Для экранных форм параметризации имеется встроенная помощь, которая оказывает вам поддержку при параметризации. У вас есть следующие возможности для вызова встроенной помощи:

- нажатием клавиши **F1** в соответствующих областях
- щелчком на кнопке **Help [Помощь]** в отдельных экранных формах для параметризации

### 6.3.1 Основные параметры

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Interrupt Selection</b> [Выбор прерывания]	Здесь вы выбираете, должно ли запускаться диагностическое прерывание. Диагностическое прерывание описано в разделе 6.7.3.	<ul style="list-style-type: none"> <li>None [Нет]</li> <li>Diagnostics [Диагностическое]</li> </ul>	None [Нет]
<b>Reaction to CPU Stop</b> [Реакция на переход CPU в STOP]	<p>Этот параметр влияет на сохранение принимаемых кадров сообщений в буфере приема.</p> <p>В обоих случаях процесс передачи прекращается.</p> <p>Ранее сохраненные кадры сообщений остаются сохраненными во всех случаях.</p> <p>Более подробную информацию вы можете найти в следующей таблице.</p>	<ul style="list-style-type: none"> <li>Continue [Продолжить]</li> <li>STOP</li> </ul>	Continue [Продолжить]

Реакция на переход CPU в STOP зависит от того, происходит ли работа с управлением или без управления потоком данных.

Управление потоком данных	Реакция на переход CPU в STOP	Поступающий в данный момент кадр сообщения	Новые кадры сообщений
Отсутствует	Дальнейшая работа	Сохраняется. При полном буфере отбрасывается.	Сохраняются до заполнения буфера, затем отбрасываются.
	STOP	Отбрасывается.	Отбрасываются.
XON/XOFF	Дальнейшая работа	Сохраняется. При полном буфере активизируется контроль потока.	Сохраняются. При полном буфере активизируется управление потоком данных.
	STOP	Прием дальнейших данных предотвращается активизированным управлением потоком данных.	Прием дальнейших данных предотвращается активизированным управлением потоком данных.

### 6.3.2 Данные параметризации драйвера ASCII

Параметры драйвера ASCII задаются с помощью экранной формы для параметризации. Ниже вы найдете подробное описание этих параметров.

#### Замечание

Драйвер ASCII может использоваться в четырехпроводном (RS 422) и двухпроводном (RS 485) режиме.

#### Передача

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Baud rate</b> [Скорость передачи]	Скорость передачи данных в битах в секунду (бодах)  * 38400 бит/с только в полудуплексном режиме	<ul style="list-style-type: none"> <li>• 300</li> <li>• 600</li> <li>• 1200</li> <li>• 2400</li> <li>• 4800</li> <li>• 9600</li> <li>• 19200</li> <li>• 38400*</li> </ul>	9600
<b>Start bit</b> [Стартовый бит]	Стартовый бит ставится впереди при передаче каждого символа.	1 (не может быть изменено)	1
<b>Data bits</b> [Биты данных]	Количество битов, на которые отображается символ.	<ul style="list-style-type: none"> <li>• 7</li> <li>• 8</li> </ul>	8
<b>Stop bits</b> [Стопové биты]	Стопové биты ставятся в конце при передаче каждого символа и обозначают конец символа .	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> </ul>	1
<b>Parity</b> [Четность]	Последовательность информационных битов может быть расширена еще на один бит, бит четности, который своим значением («0» или «1») дополняет общее значение всех битов до согласованного состояния. Благодаря этому повышается надежность передачи данных. Настройка «None» означает, что бит четности не посылается. При установке 7 битов данных настройка «None» невозможна.	<ul style="list-style-type: none"> <li>• None [Нет]</li> <li>• Odd [Нечетное]</li> <li>• Even [Четное]</li> </ul>	Even [Четное]
<b>Data flow control</b> [Управление потоком данных]	Определение, каким способом осуществляется управление потоком данных. Управление потоком данных возможен только в полнодуплексном (RS 422) четырехпроводном режиме двухточечного соединения. Включением программного управления потоком с помощью XON/XOFF удастся избежать потери данных при передаче у устройств, работающих с различной скоростью.	<ul style="list-style-type: none"> <li>• None [Нет]</li> <li>• XON/XOFF</li> </ul>	None [Нет]

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>XON character</b> [Символ XON]	Код для символа XON Как только CPU переводится в режим с управлением потоком данных, он посылает символ XON. Как только кадр сообщения извлечен, и приемный буфер снова готов к приему, CPU посылает символ XON.	<ul style="list-style-type: none"> <li>при 7 битах данных: от 0 до 7Fh (16-рич.)</li> <li>при 8 битах данных: от 0 до FFh (16-рич.)</li> </ul>	11h = DC1
<b>XOFF character</b> [Символ XOFF]	Код для символа XOFF При достижении указанного при параметризации количества кадров сообщений или состояния, когда до переполнения приемного буфера остается 50 символов (величина приемного буфера – 2048 байт), CPU посылает символ XOFF. Если партнер по обмену данными несмотря на это продолжает передачу, при переполнении приемного буфера генерируется сообщение об ошибке. Принимаемые данные последнего кадра сообщения отбрасываются.	<ul style="list-style-type: none"> <li>при 7 битах данных: от 0 до 7Fh (16-рич.)</li> <li>при 8 битах данных: от 0 до FFh (16-рич.)</li> </ul>	13h = DC3
<b>Wait for XON after XOFF</b> [Ожидание XON после XOFF]	Время, в течение которого CPU при передаче должен ожидать символа XON. Если CPU принимает символ XOFF, он прерывает процесс передачи. Если по истечении определенного, указанного при параметризации времени приема символа XON не происходит, процесс передачи прекращается, и на выходе STATUS системного функционального блока генерируется соответствующее сообщение об ошибке (0708h).	от 20 до 65530 мс шагами по 10 мс	20000 мс

## Конечный символ

Параметр	Описание	Диапазон значений	Значение по умолчанию																		
End-of-message recognition for received frames [Распознавание конца принимаемого кадра сообщения]	<p>Задание критерия, определяющего конец кадра сообщения.</p> <ul style="list-style-type: none"><li>По истечении времени задержки символа (CDT): Кадр сообщения не имеет ни фиксированной длины, ни определенного конечного символа, конец кадра сообщения определяется по паузе в линии (истечение времени задержки символа).</li><li>Прием фиксированного числа символов: Длина принимаемого кадра сообщения всегда одна и та же.</li><li>Прием конечного символа (-ов): В конце кадра сообщения находятся один или два определенных конечных символа.</li></ul>	<ul style="list-style-type: none"><li>On expiration of character delay time [По истечении времени задержки символа]</li><li>On receipt of fixed number of characters [После приема фиксированного числа символов]</li><li>On receipt of end-of-text character [После приема конечного символа]</li></ul>	По истечении времени задержки символа																		
Character Delay Time [Время задержки символа (CDT)]	Время задержки символа определяет максимально допустимый интервал между 2 следующими друг за другом принимаемыми символами.	от 1 до 65535 мс Самое малое время задержки зависит от скорости передачи	4 мс																		
Monitoring time for missing end code [Время контроля отсутствия кода конца]	<p>Время задержки символа (CDT) применяется как время контроля на отсутствие распознавания конца. Действует при следующих настройках для распознавания конца</p> <ul style="list-style-type: none"><li>после приема фиксированного числа символов</li><li>после приема конечного символа (-ов)</li></ul>	<table><thead><tr><th>Бод</th><th>Время задержки [мс]</th></tr></thead><tbody><tr><td>300</td><td>130</td></tr><tr><td>600</td><td>65</td></tr><tr><td>1200</td><td>32</td></tr><tr><td>2400</td><td>16</td></tr><tr><td>4800</td><td>8</td></tr><tr><td>9600</td><td>4</td></tr><tr><td>19200</td><td>2</td></tr><tr><td>38400</td><td>1</td></tr></tbody></table>	Бод	Время задержки [мс]	300	130	600	65	1200	32	2400	16	4800	8	9600	4	19200	2	38400	1	
Бод	Время задержки [мс]																				
300	130																				
600	65																				
1200	32																				
2400	16																				
4800	8																				
9600	4																				
19200	2																				
38400	1																				
Send pause between message frames for the length of monitoring time [Пауза в передаче между кадрами сообщений, равная времени контроля]	При критерии конца "После приема фиксированного числа символов" во время передачи между двумя кадрами сообщений выдерживается пауза, равная времени контроля (на отсутствие распознавания конца сообщения), чтобы партнер мог войти в синхронизм (распознавание приема кадра сообщения).	<ul style="list-style-type: none"><li>Yes [Да]</li><li>No [Нет]</li></ul>	Yes [Да]																		

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Message frame length on receiving</b> [Длина кадра сообщения при приеме]	При критерии конца "После приема фиксированного числа символов" определяется количество байтов, из которых состоит кадр сообщения.	от 1 до 1024 [байтов]	1024
<b>End-of-text Character</b> [Символ конца текста]	Можно работать с одним или с двумя конечными символами. По выбору, после признака конца можно дополнительно принять один или два символа. Эти символы можно, напр., использовать для передачи символа контроля блока (BCC). Расчет значения BCC у передатчика и его анализ у приемника вы должны выполнять сами в программе пользователя.	<ul style="list-style-type: none"> <li>• 1 End-of-text Character [1 конечный символ]</li> <li>• 1 End-of-text Character with 1 BCC [1 конечный символ с 1 BCC]</li> <li>• 1 End-of-text Character with 2 BCC [1 конечный символ с 2 BCC]</li> <li>• 1st and 2nd end-of-text Character [1-й и 2-й конечный символ]</li> <li>• 1st and 2nd end-of-text Character with 1 BCC [1-й и 2-й конечный символ с 1 BCC]</li> <li>• 1st and 2nd end-of-text Character with 2 BCC [1-й и 2-й конечный символ с 2 BCC]</li> </ul>	1 End-of-text Character [1 конечный символ]
<b>End-of-text Character 1</b> [Символ конца текста 1]	Код первого конечного символа.	<ul style="list-style-type: none"> <li>• при 7 битах данных: от 0 до 7Fh (16-рич.)</li> <li>• при 8 битах данных: от 0 до FFh (16-рич.)</li> </ul>	03h = ETX
<b>End-of-text Character 1</b> [Символ конца текста 2]	Код второго конечного символа, если выбран.	<ul style="list-style-type: none"> <li>• при 7 битах данных: от 0 до 7Fh (16-рич.)</li> <li>• при 8 битах данных: от 0 до FFh (16-рич.)</li> </ul>	0

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Transmitting with end-of-text character</b> [Передача с символом конца текста]	<p>При критерии конца "После приема конечного символа (-ов)" вы можете вести передачу с конечными символами.</p> <ul style="list-style-type: none"> <li>Передача до символа конца текста включительно: Символ конца текста должен содержаться в подлежащих передаче данных. Данные передаются только до признака конца текста включительно, даже если на SFB указана большая длина данных.</li> <li>Передача до длины, указанной при параметризации блока: Данные передаются до длины, указанной при параметризации SFB. Последним должен быть символ конца текста.</li> <li>Передача до длины, указанной при параметризации блока, с автоматическим добавлением символа конца текста: Данные передаются до длины, указанной при параметризации SFB. Кроме того, автоматически добавляется (добавляются) символ (символы) конца текста; т.е. признаки конца не должны содержаться в передаваемых данных. В зависимости от количества признаков конца текста партнеру посылается на 1 или 2 символа больше, чем указывается на SFB (максимум 1024 байта).</li> </ul>	<ul style="list-style-type: none"> <li>Transmission including the end-of-text character [Передача до символа конца текста включительно]</li> <li>Transmission up to the length specified in the block parameters [Передача до длины, указанной при параметризации блока]</li> <li>Transmission up to the length specified in the block parameters and automatically appending the end-of-text character [Передача до длины, указанной при параметризации блока, с автоматическим добавлением символа конца текста]</li> </ul>	Transmission including the end-of-text character [Передача до символа конца текста включительно]

## Прием данных

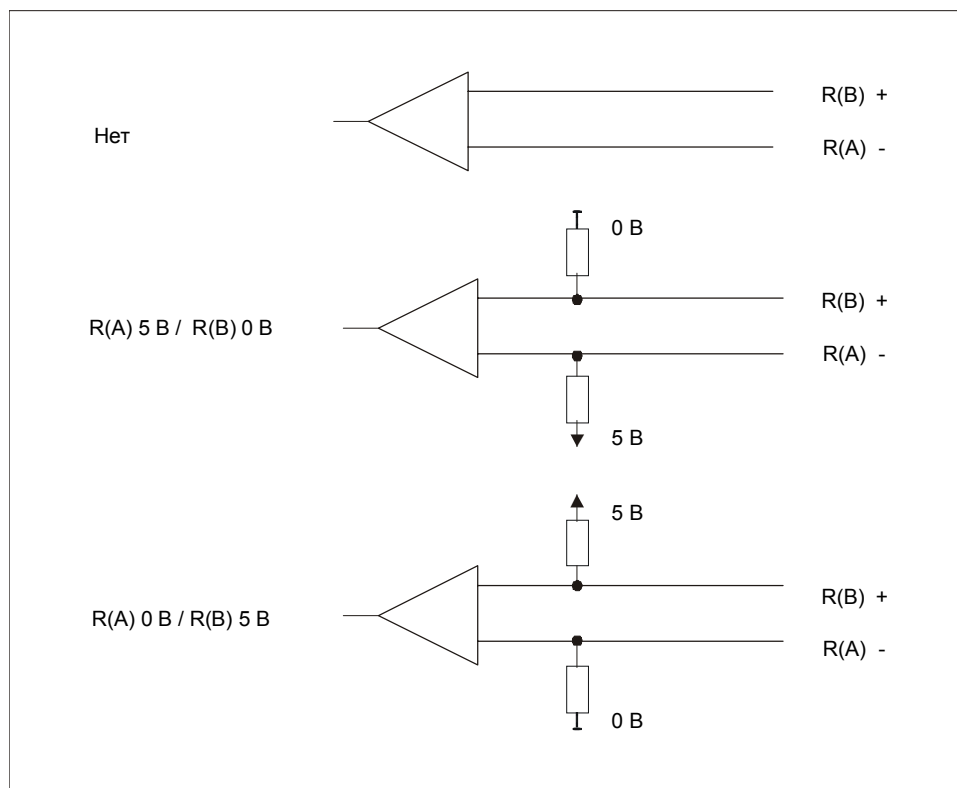
Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Clearing the receive buffer on startup [Очистка приемного буфера при запуске]</b>	При включении питания или при переходе CPU из STOP в RUN приемный буфер очищается.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	No [Нет]
<b>Prevent overwriting [Предотвращение замены]</b>	С помощью этого параметра вы можете воспрепятствовать тому, чтобы при полном приемном буфере данные в нем заменялись.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	Yes [Да]
<b>Utilizing the whole buffer [Использование всего буфера]</b>	Вы можете использовать весь буфер или указать число принимаемых кадров сообщений, которые должны сохраняться в буфере.  Если вы используете весь буфер из 2048 байтов, то количество сохраняемых в нем поступающих кадров сообщений зависит только от их длины.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	Yes [Да]
<b>Maximum number of received message frames in the buffer [Максимальное количество сохраняемых в буфере принимаемых кадров сообщений]</b>	С установкой "Do not use the whole buffer [Не использовать весь буфер]" вы можете указать число кадров сообщений, которые должны сохраняться в приемном буфере.  Если здесь вы укажете при параметризации «1», а параметр «Prevent overwriting [Предотвращение замены]» деактивизируете и будете считывать принимаемые данные в программе пользователя циклически, то в целевой блок данных всегда будет передаваться текущий кадр сообщения.	от 1 до 10	10



## Назначение сигналов для интерфейса X27 (RS 422/485)

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Operating mode [Режим]</b>	<p>Определение, должен ли интерфейс X27 (RS 422/485) эксплуатироваться в полнодуплексном (RS 422) или полудуплексном (RS 485) режиме.</p> <ul style="list-style-type: none"> <li>Полнодуплексный (RS 422) четырехпроводный режим "точка-точка" Четырехпроводный режим для двухточечных соединений</li> <li>Полнодуплексный (RS 422) четырехпроводный режим "многоточечный мастер" Четырехпроводный режим для многоточечного (Multipoint) соединения, когда CPU является master-устройством.</li> <li>Полудуплексный (RS 485) двухпроводный режим Двухпроводный режим для двухточечного или многоточечного (Multipoint) соединения. CPU может быть master- или slave-устройством.</li> </ul>	<ul style="list-style-type: none"> <li>Full Duplex (RS 422) four-wire PtP communication [Полнодуплексная (RS 422) четырехпроводная связь "точка-точка"]</li> <li>Full Duplex (RS 422) four-wire operation, Multipoint Master [Полнодуплексный (RS 422) четырехпроводный режим "многоточечный мастер"]</li> <li>Halbduplex (RS 485) Zweidrahtbetrieb [Полудуплексный (RS 485) двухпроводный режим]</li> </ul>	Vollduplex (RS 422) Vierdrahtbetrieb Punkt zu Punkt [Полнодуплексная (RS 422) четырехпроводная связь точка-точка]
<b>Default for the receive line [Настройка приемной линии по умолчанию]</b>	<ul style="list-style-type: none"> <li>Нет: Настройка имеет смысл только для специальных драйверов, способных к работе с шиной.</li> <li>Сигнал R(A) 5 вольт/ Сигнал R(B) 0 вольт: При этой настройке по умолчанию возможно распознавание обрыва. (Не может устанавливаться в полнодуплексном (RS422) четырехпроводном режиме "многоточечный мастер" и в полудуплексном (RS485) двухпроводном режиме)</li> <li>Сигнал R(A) 0 вольт/ Сигнал R(B) 5 вольт: Эта настройка по умолчанию соответствует исходному состоянию (нет активных передатчиков). При этой настройке по умолчанию распознавание обрыва невозможно.</li> </ul>	<ul style="list-style-type: none"> <li>None [Нет]</li> <li>Signal R(A) 5 Volt/ Signal R(B) 0 Volt (Breakerkennung [распознавание обрыва])</li> <li>Signal R(A) 0 Volt/ Signal R(B) 5 Volt</li> </ul>	Зависит от установленного режима

На следующем рисунке показано подключение приемника к интерфейсу X27 (RS 422/485):



CPU может использоваться в режиме RS422 или RS485 в различных топологиях.

Различают соединения с

- двумя абонентами (**двухточечное**) и
- несколькими абонентами (**многоточечное**).

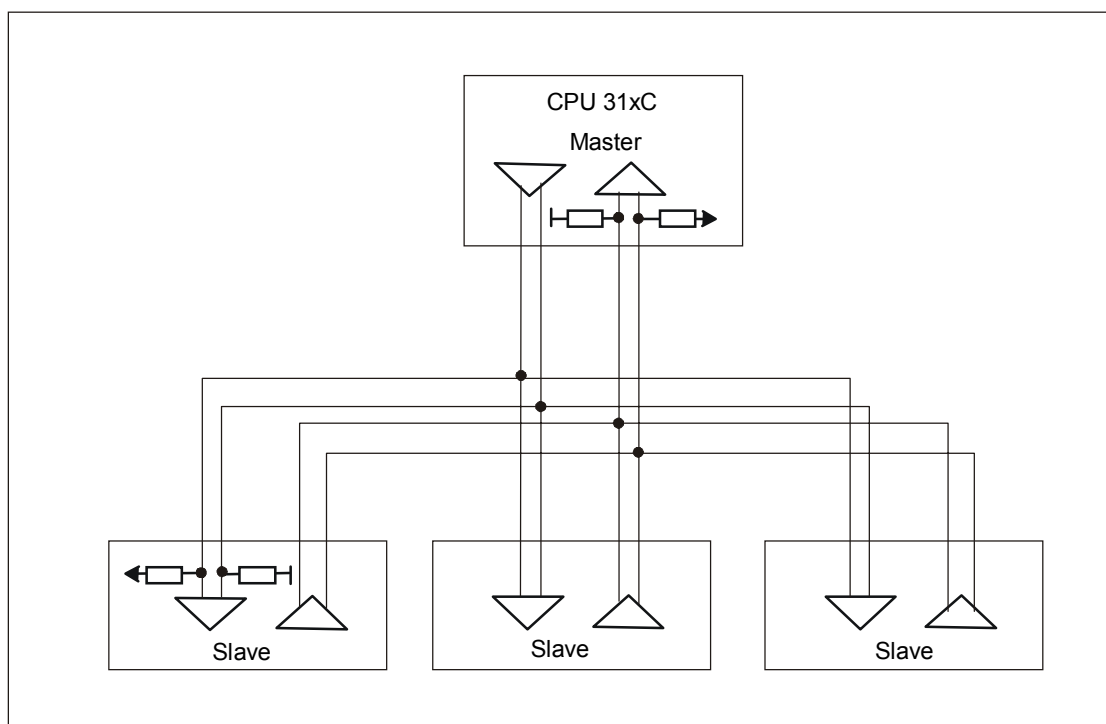
При этом он может использоваться как

- **Master** или
- **Slave** (только в режиме RS485).

При **топологии Master/Slave** в программе пользователя должна программироваться соответствующий кадр сообщения. Пример: Master посылает всем slave-устройствам кадр сообщения с информацией об адресе. Все slave-устройства слушают эту информацию и сравнивают адрес со своим собственным. При совпадении соответствующий slave передает свой ответ.

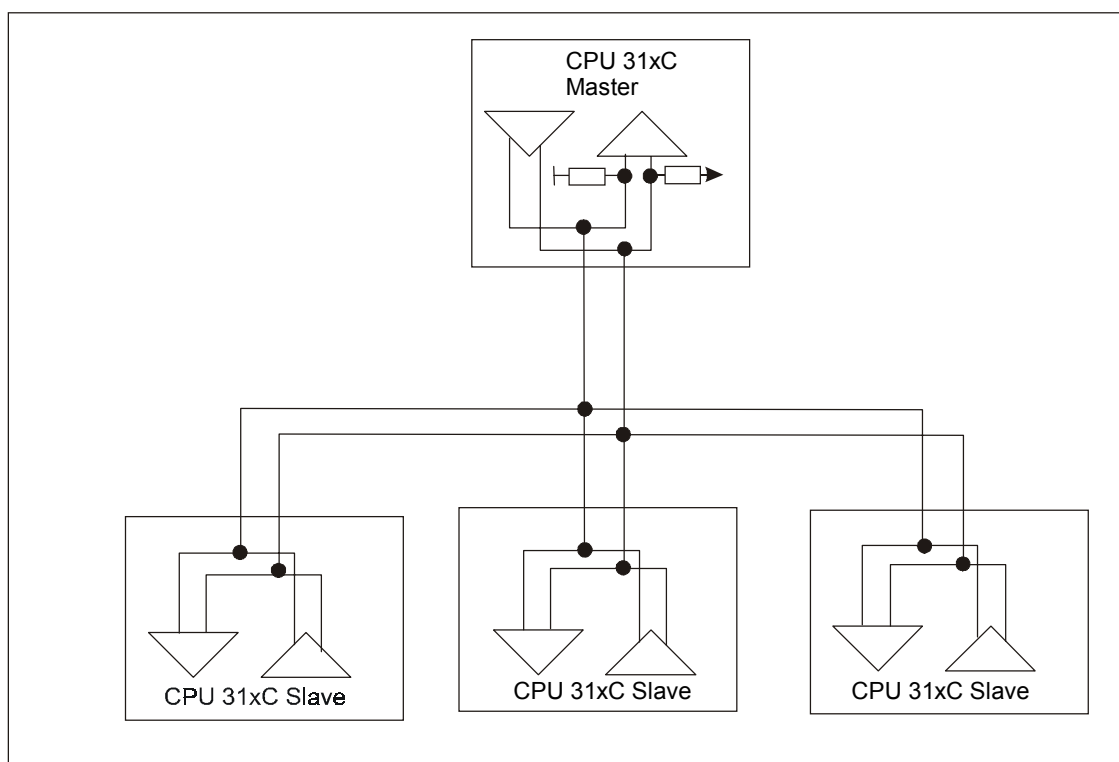
Передатчики всех slave-устройств должны быть в состоянии включаться как высокоомные.

- При топологии Master/Slave в режиме RS422
  - CPU может использоваться только в режиме master-устройства,
  - передатчик master-устройства соединяется с приемниками всех slave-устройств,
  - передатчики slave-устройств соединяются с приемником master-устройства,
  - настройка по умолчанию производится только у приемника master-устройства и у приемника одного slave-устройства. Все остальные slave-устройства работают без настройки по умолчанию.



- **При топологии в режиме RS485**

- общая пара проводов используется в качестве линии приема и передачи всех абонентов,
- настройка по умолчанию производится только у приемника одного абонента. Все остальные модули работают без настройки по умолчанию.



Настройки, необходимые для различных топологий, производятся с помощью экранной формы для параметризации "Interface [Интерфейс]".

---

**Замечание**

Если вы используете драйвер ASCII в многоточечном режиме RS422 или в режиме RS485, то вы должны позаботиться в программе пользователя о том, чтобы передачу вел только один абонент. Если передачу ведут одновременно несколько абонентов, то кадр сообщения искажается.

---

### 6.3.3 Данные параметризации для процедуры 3964(R)

Параметры процедуры 3964(R) задаются с помощью экранной формы для параметризации. Ниже вы найдете подробное описание этих параметров.

#### Замечание

Процедура 3964(R) может использоваться только в четырехпроводном режиме (RS 422).

#### Передача

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Baud rate</b> [Скорость передачи]	Скорость передачи данных в битах в секунду (бодах)	<ul style="list-style-type: none"> <li>• 300</li> <li>• 600</li> <li>• 1200</li> <li>• 2400</li> <li>• 4800</li> <li>• 9600</li> <li>• 19200</li> <li>• 38400</li> </ul>	9600
<b>Start bit</b> [Стартовый бит]	Стартовый бит ставится впереди при передаче каждого символа.	1 (не может быть изменено)	1
<b>Data bits</b> [Биты данных]	Количество битов, на которые отображается символ.	<ul style="list-style-type: none"> <li>• 7</li> <li>• 8</li> </ul>	8
<b>Stop bits</b> [Стоповые биты]	Стоповые биты устанавливаются в конце кадра при передаче каждого символа и обозначают конец символа .	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> </ul>	1
<b>Parity</b> [Четность]	Последовательность информационных битов может быть расширена еще на один бит, бит четности, который своим значением («0» или «1») дополняет общее значение всех битов до согласованного состояния. Благодаря этому повышается надежность передачи данных. Настройка «None» означает, что бит четности не посылается. <b>При установке 7 битов данных настройка «None» невозможна.</b>	<ul style="list-style-type: none"> <li>• None [Нет]</li> <li>• Odd [Нечетное]</li> <li>• Even [Четное]</li> </ul>	Even [Четное]
<b>Priorität</b> [Приоритет]	Партнер имеет высокий приоритет, если его желание выполнить передачу имеет преимущество перед желанием выполнить передачу другого партнера. Партнер имеет низкий приоритет, если его желание выполнить передачу должно уступать желанию выполнить передачу другого партнера. При использовании процедуры 3964(R) вы должны устанавливать разные приоритеты у партнеров по обмену данными, т.е., один партнер получает высокий приоритет, а другой партнер получает низкий приоритет.	<ul style="list-style-type: none"> <li>• Low [Низкий]</li> <li>• High [Высокий]</li> </ul>	High [Высокий]

Параметр	Описание	Значение по умолчанию
<b>Message frame parameter 3964(R) with standard values and block check</b> [Параметры кадра сообщения 3964(R) со стандартными значениями и контролем блока]	Параметры протокола получают значения по умолчанию. Если CPU распознает последовательность символов DLE ETX BCC, то он прекращает прием. Он сравнивает принятый символ контроля блока BCC с внутренне сформированным контролем четности длины. Если символ контроля блока верен, и не возникло другой ошибки приема, то он посылает символ DLE (в случае ошибки партнеру по обмену данными посылается символ NAK).	3964(R) со стандартными значениями и контролем блока: Задержка символа = 220 мс Задержка квитирования = 2000 мс Число попыток установления связи = 6 Число попыток передачи = 6
<b>The message frame parameter 3964(R) is programmable with block check</b> [Параметры кадра сообщения 3964(R) могут устанавливаться с контролем блока]	Параметры протокола могут устанавливаться свободно. Если CPU распознает последовательность символов DLE ETX BCC, то он прекращает прием. Он сравнивает принятый символ контроля блока BCC с внутренне сформированным контролем четности длины. Если символ контроля блока верен, и не возникло другой ошибки приема, то он посылает символ DLE (в случае ошибки партнеру по обмену данными посылается символ NAK).	
<b>Message frame parameter 3964 with standard values and without block check</b> [Параметры кадра сообщения 3964 со стандартными значениями без контроля блока]	Параметры протокола получают значения по умолчанию. Если CPU распознает последовательность символов DLE ETX, то он завершает прием и передает партнеру по обмену данными DLE для правильно принятого блока (или NAK для блока, принятого с ошибками).	
<b>The message frame parameter 3964 can be programmed not to contain a block check</b> [Параметры кадра сообщения 3964 могут устанавливаться без контроля блока]	Параметры протокола могут устанавливаться свободно. Если CPU распознает последовательность символов DLE ETX, то он завершает прием и передает партнеру по обмену данными DLE для правильно принятого блока (или NAK для блока, принятого с ошибками).	

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Character delay time [Время задержки импульса]</b>	Время задержки импульса определяет максимально допустимый временной интервал между принимаемыми символами внутри кадра сообщения.	от 20 мс до 65530 мс шагами по 10 мс Самое малое время задержки зависит от скорости передачи: 300 бит/с      60 мс 600 бит/с      40 мс 1200 бит/с     30 мс от 2400 до 38400 бит/с    20 мс	220 мс
<b>Acknowledgment delay time [Время задержки квитирования]</b>	Время задержки квитирования определяет максимально допустимый интервал времени до квитирования партнера при установлении связи (время между STX и квитированием DLE партнера) или до установления связи (время между DLE ETX (BCC) и квитированием DLE партнера).	от 20 мс до 65530 мс шагами по 10 мс Наименьшее время задержки квитирования зависит от скорости передачи: 300 бит/с      60 мс 600 бит/с      40 мс 1200 бит/с     30 мс от 2400 до 38400 бит/с    20 мс	2000 мс (550 мс в случае 3964 без контроля блока)
<b>Attempts to connect [Попытки установления связи]</b>	Параметр определяет максимальное число попыток CPU установить связь.	от 1 до 255	6
<b>Attempts to transmit [Попытки передачи]</b>	Параметр определяет максимальное число попыток передать кадр сообщения (включая первую), в случае ошибок.	от 1 до 255	6

## Прием данных

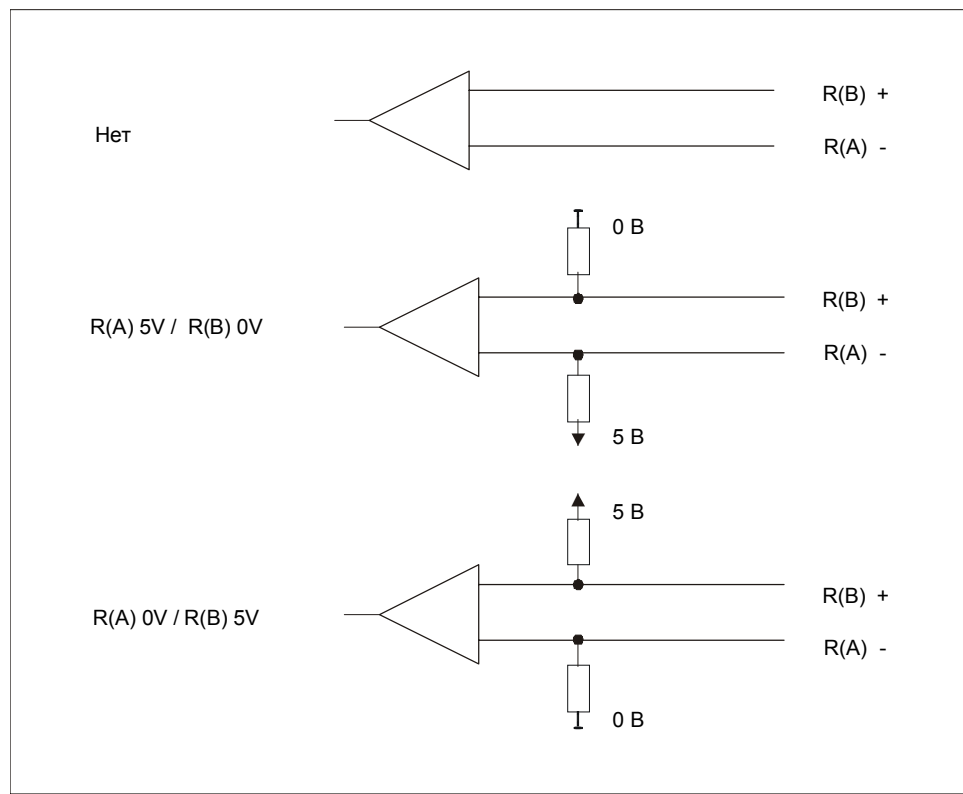
Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Clearing the receive buffer on startup [Очистка приемного буфера при запуске]</b>	При включении питания или при переходе CPU из STOP в RUN приемный буфер очищается.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	No [Нет]
<b>Prevent overwriting [Предотвращение замены]</b>	С помощью этого параметра вы можете воспрепятствовать тому, чтобы при полном приемном буфере данные в нем заменялись.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	Yes [Да]
<b>Utilizing the whole buffer [Использование всего буфера]</b>	Вы можете использовать весь буфер или указать число принимаемых кадров сообщений, которые должны сохраняться в буфере.  Если вы используете весь буфер из 2048 байтов, то количество сохраняемых в нем поступающих кадров сообщений зависит только от их длины.	<ul style="list-style-type: none"> <li>• Yes [Да]</li> <li>• No [Нет]</li> </ul>	Yes [Да]
<b>Maximum number of received message frames in the buffer [Максимальное количество сохраняемых в буфере принимаемых кадров сообщений]</b>	С установкой "Do not use the whole buffer [не использовать весь буфер]" вы можете указать число кадров сообщений, которые должны сохраняться в приемном буфере.  Если здесь вы укажете при параметризации «1», а параметр "Prevent overwriting [Предотвращение замены]" деактивируете и будете считывать принимаемые данные в программе пользователя циклически, то в целевой блок данных всегда будет передаваться текущий кадр сообщения.	от 1 до 10	10

## Назначение сигналов для интерфейса X27 (RS 422/485)

Параметр	Описание	Диапазон значений	Значение по умолчанию
<b>Default for the receive line [Настройка приемной линии по умолчанию]</b>	<ul style="list-style-type: none"> <li>• Нет: Настройка имеет смысл только для драйверов, способных к работе с шиной.</li> <li>• R(A) 5 Volt/ R(B) 0 Volt: При этой настройке по умолчанию возможно распознавание обрыва.</li> <li>• R(A) 0 Volt/ R(B) 5 Volt: При этой настройке по умолчанию распознавание обрыва невозможно.</li> </ul>	<ul style="list-style-type: none"> <li>• None [Нет]</li> <li>• R(A) 5V/R(B) 0V</li> <li>• R(A) 0V/R(B) 5V</li> </ul>	R(A) 5V/ R(B) 0V



На следующем рисунке показано подключение приемника к интерфейсу X27 (RS 422):



#### 6.3.4 Данные параметризации для компьютерного интерфейса RK 512

Эти параметры идентичны параметрам процедуры 3964(R), так как процедура 3964(R) является подмножеством компьютерного интерфейса RK 512.

Исключение:

- Число битов данных на символ у компьютерного интерфейса RK 512 фиксировано и равно 8.
- Нет приемного буфера (отпадают параметры для приема данных).

Параметры для цели или источника данных вы должны задать в используемых системных функциональных блоках (SFB).

### 6.4 Включение в программу пользователя

Последовательным соединением вы управляете через свою пользовательскую программу. Для этого вы вызываете системные функциональные блоки (SFB). Эти SFB находятся в стандартной библиотеке (Standard Library) в разделе "System Function Blocks [Системные функциональные блоки]".

Следующие разделы дают вам возможность разрабатывать программу пользователя в соответствии с вашим приложением.

#### Вызов SFB

SFB вызывается с соответствующим экземплярным DB.

Пример: CALL SFB 60, DB20

#### Экземплярный DB

В экземплярном DB хранятся все необходимые для SFB параметры.

---

#### Замечание

- Вы должны в своей пользовательской программе каждый тип SFB (SEND; FETCH, RCV, ...) всегда вызывать с одним и тем же экземплярным DB, так как в экземплярном DB хранятся состояния, необходимые для внутреннего исполнения SFB.
  - Доступ на запись к данным экземплярного DB недопустим.
-

## Структура программы

Обработка SFB происходит асинхронно. Для полной обработки SFB должен вызываться столько раз, пока он не будет завершен с ошибками или без них.

---

### Замечание

Если вы в своей программе запрограммировали SFB, то вам нельзя еще раз вызывать тот же SFB в какой-либо части программы с другим классом приоритета, так как SFB не может прерывать сам себя.

Пример: Недопустимо вызывать SFB в OB 1 и тот же SFB в OB прерываний.

## Классификация параметров SFB

Параметры SFB по их функции можно разделить на следующие четыре класса:

- **Управляющие параметры** служат для активизации блока.
- **Параметры передачи** указывают на те области данных, которые должны быть переданы удаленному партнеру.
- **Параметры приема** указывают на те области данных, в которые заносятся данные, принятые от удаленного партнера.
- **Параметры состояния** служат для контроля того, выполнил ли блок свою задачу без ошибок, или для анализа возникших ошибок. Параметры состояния устанавливаются только на один вызов.

## 6.5 Коммуникационные функции

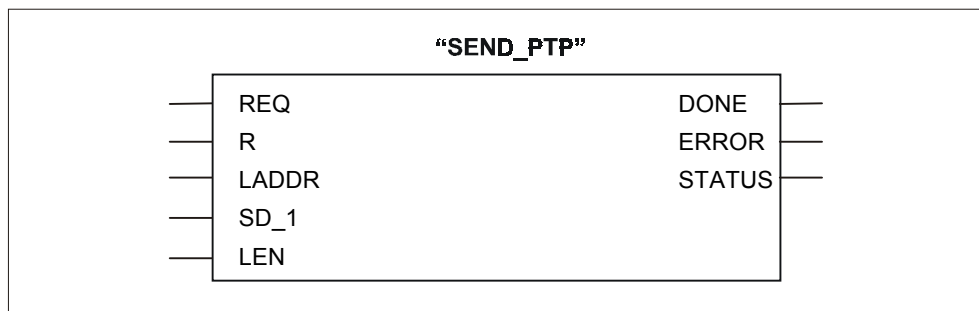
### 6.5.1 Коммуникационные функции для ASCII/3964(R)

В этом разделе описываются функции, которые имеются в вашем распоряжении для протоколов ASCII и 3964.

Блок		Описание
SFB 60	SEND_PTP	Передать партнеру по обмену данными всю область или часть области блока данных.
SFB 61	RCV_PTP	Принять данные от партнера и сохранить в блоке данных.
SFB 62	RES_RCVB	Сбросить приемный буфер CPU.

#### 6.5.1.1 Передача данных с помощью SFB 60 "SEND\_PTP"

С помощью этого SFB передаются данные из блока данных.



Активизация процесса передачи производится после вызова блока и положительного фронта на управляющем входе **REQ**. Область подлежащих передаче данных задается через **SD\_1** (номер DB и начальный адрес), длина блока передаваемых данных через **LEN**.

Чтобы SFB мог выполнить задание, вы должны его вызвать с **R(Reset) = FALSE**. При положительном фронте на управляющем входе **R** текущий процесс передачи прекращается, а SFB переводится в исходное состояние. Прерванное задание завершается с сообщением об ошибке (выход **STATUS**).

С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то **DONE** устанавливается в **TRUE**, если задание было завершено с ошибками, то в **TRUE** устанавливается **ERROR**.

Если задание было выполнено с **DONE = TRUE**, это означает:

- При использовании драйвера ASCII: данные были отправлены партнеру по обмену данными. Не гарантируется, что данные были также и приняты партнером.
- При использовании процедуры 3964(R): данные были отправлены партнеру по обмену данными и были им положительно квитированы. Не гарантируется, что данные были также переданы CPU партнера.

В параметре **STATUS** в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8). Параметры **DONE** или **ERROR/STATUS** выводятся также при сбросе (RESET) SFB (**R = TRUE**). При возникновении ошибки двоичный результат **BIE** сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние **TRUE**.

#### Замечание

SFB не проверяет параметры, при неверной параметризации CPU может перейти в состояние **STOP**.

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>REQ</b>	IN	BOOL	Управляющий параметр "Request [Запрос]": Активизирует обмен данными при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр "Reset [Сброс]": Задание прерывается. Передача блокируется.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>DONE</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): <ul style="list-style-type: none"> <li>• FALSE: Задание еще не было запущено или еще выполняется.</li> <li>• TRUE: Задание было выполнено без ошибок.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>STATUS</b>	OUT	WORD	<p>Параметр состояния (Этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.):</p> <p>STATUS в зависимости от бита ERROR имеет следующее значение:</p> <ul style="list-style-type: none"> <li>• ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>- 0000h: ни предупреждения, ни ошибки</li> <li>- 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>• ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0
<b>SD_1</b>	IN_OUT	ANY	<p>Параметр передачи:</p> <p>Здесь вы указываете:</p> <ul style="list-style-type: none"> <li>• Номер DB, из которого передаются данные.</li> <li>• Номер байта данных, начиная с которого должны передаваться данные.</li> </ul> <p>Напр.: DB10 от байта 2 -&gt; DB10.DBB2</p>	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	<p>Здесь вы указываете длину подлежащих передаче данных в байтах.</p> <p>(Длина здесь указывается косвенно.)</p>	от 1 до 1024	1

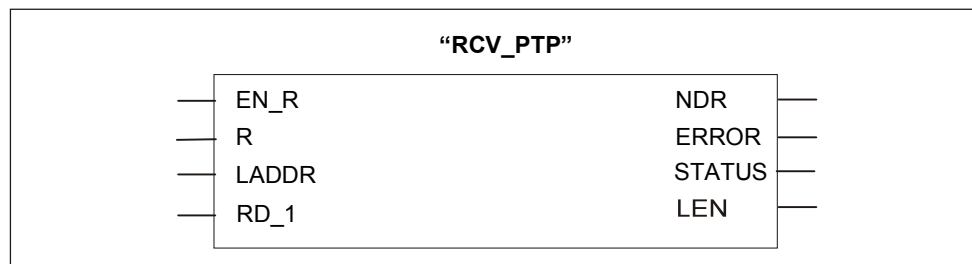
### Согласованность данных

Согласованность данных ограничена 206 байтами. Для согласованной передачи количества данных, большего 206 байтов, необходимо учесть следующее:

Описывайте снова используемую в данный момент часть области передачи SD\_1 только тогда, когда процесс передачи закончен. Это имеет место, когда параметр состояния DONE принимает значение TRUE.

### 6.5.1.2 Прием данных с помощью SFB 61 "RCV\_PTP"

С помощью этого SFB производится прием данных и их сохранение в блоке данных.



Блок готовится к приему после вызова значением TRUE на управляющем входе **EN\_R**. Текущую передачу можно прекратить с помощью состояния сигнала FALSE на параметре EN\_R. Прерванное задание завершается с сообщением об ошибке (выход **STATUS**). Прием выключается, пока на параметре EN\_R сохраняется состояние FALSE.

Область приема задается через **RD\_1** (номер DB и начальный адрес), длина блока передаваемых данных через **LEN**.

Чтобы SFB мог выполнить задание, вы должны его вызвать с **R(Reset) = FALSE**. При положительном фронте на управляющем входе R текущая передача прекращается, и SFB переводится в исходное состояние. Прерванное задание на прием завершается с сообщением об ошибке (выход **STATUS**).

С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то параметр **NDR** устанавливается в TRUE, если задание завершилось с ошибкой, то в TRUE устанавливается параметр **ERROR**.

В параметре **STATUS** в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8).

NDR или ERROR/STATUS выводятся также при сбросе (RESET) SFB (R = TRUE) (параметр LEN = 16#00).

При возникновении ошибки двоичный результат BIE сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние TRUE.

#### Замечание

SFB не проверяет параметры, при неверной параметризации CPU может перейти в состояние STOP.

Параметр	Объявление	Тип данных	Описание	Диапазон значений	Значение по умолчанию
<b>EN_R</b>	IN	BOOL	Управляющий параметр "Enable to receive [Разрешение на прием]": деблокировка приема	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр «Reset [Сброс]»: Выполнение задания прекращено	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>NDR</b>	OUT	BOOL	Параметр состояния "New data ready [Новые данные готовы]": Задание готово без ошибок, данные приняты <ul style="list-style-type: none"> <li>FALSE: Задание еще не было запущено или еще выполняется.</li> <li>TRUE: Задание было успешно завершено.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Параметр состояния (этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.): STATUS в зависимости от бита ERROR имеет следующее значение: <ul style="list-style-type: none"> <li>ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>0000h: ни предупреждения, ни ошибки</li> <li>&lt;&gt; 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0
<b>RD_1</b>	IN_OUT	ANY	Параметр приема: Здесь вы указываете: <ul style="list-style-type: none"> <li>Номер DB, в котором сохраняются принимаемые данные.</li> <li>Номер байта данных, начиная с которого сохраняются принимаемые данные.</li> </ul> Напр.: DB20, начиная с байта 5 -> DB20.DBB5	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Вывод длины данных (число байтов)	от 0 до 1024	0



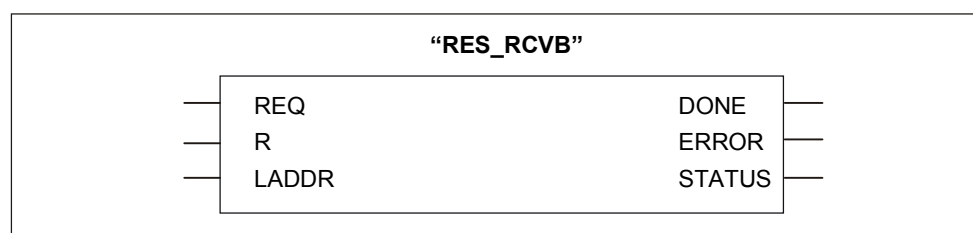
## Согласованность данных

Согласованность данных ограничена 206 байтами. Для согласованной передачи количества данных, большего 206 байтов, необходимо учесть следующее:

Снова обращайтесь к принимающему DB только тогда, когда данные были полностью приняты (NDR = TRUE). После этого заблокируйте принимающий DB до тех пор (EN\_R = FALSE), пока вы не обработаете данные.

### 6.5.1.3 Стирание приемного буфера с помощью SFB 62 "RES\_RCVB"

С помощью этого SFB полностью стирается весь приемный буфер CPU. Все сохраненные кадры сообщений отбрасываются. Кадр сообщения, поступающий в момент вызова "RES\_RCVB", сохраняется.



Активизация задания производится после вызова блока и положительного фронта на управляющем входе **REQ**. Задание может выполняться в течение нескольких вызовов (программных циклов).

Чтобы SFB мог выполнить задание, вы должны его вызвать с **R**(Reset) = FALSE. При положительном фронте на управляющем входе R процесс стирания прекращается, а SFB переводится в исходное состояние. Прерванное задание завершается с сообщением об ошибке (выход STATUS).

С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то **DONE** устанавливается в TRUE, если задание было завершено с ошибками, то в TRUE устанавливается **ERROR**.

В параметре **STATUS** в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8).

Параметры DONE или ERROR/STATUS выводятся также при сбросе (RESET) SFB (R = TRUE).

При возникновении ошибки двоичный результат BIE сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние TRUE.

#### Замечание

SFB не проверяет параметры, при неверной параметризации CPU может перейти в состояние STOP.

Параметр	Объявление	Тип данных	Описание	Диапазон значений	Значение по умолчанию
<b>REQ</b>	IN	BOOL	Управляющий параметр «Request [Запрос]»: Активизирует задание при положительном фронте.	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр «Reset [Сброс]»: Выполнение задания прекращено.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>DONE</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): <ul style="list-style-type: none"> <li>FALSE: Задание еще не было запущено или еще выполняется.</li> <li>TRUE: Задание было выполнено без ошибок.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Параметр состояния (этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.): STATUS в зависимости от бита ERROR имеет следующее значение: <ul style="list-style-type: none"> <li>ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>0000h: ни предупреждения, ни ошибки</li> <li>&lt;&gt; 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0

## 6.5.2 Коммуникационные функции для компьютерного интерфейса RK 512

В этом разделе описываются функции, которые имеются в вашем распоряжении для протокола RK 512.

Блок		Описание
SFB 63	SEND_RK	Передать партнеру по обмену данными всю область или часть области блока данных.
SFB 64	FETCH_RK	Извлечь всю область или часть области блока данных удаленного партнера по обмену данными.
SFB 65	SERVE_RK	<ul style="list-style-type: none"> <li>Принять данные от партнера и сохранить в блоке данных.</li> <li>Подготовить данные для партнера.</li> </ul>

### Одновременно обрабатываемые задания

В программе пользователя нельзя одновременно активизировать задание SEND и задание FETCH. Т.е., когда, напр., задание SEND еще не завершено, задание FETCH не может быть запущено.

### SYNC\_DB

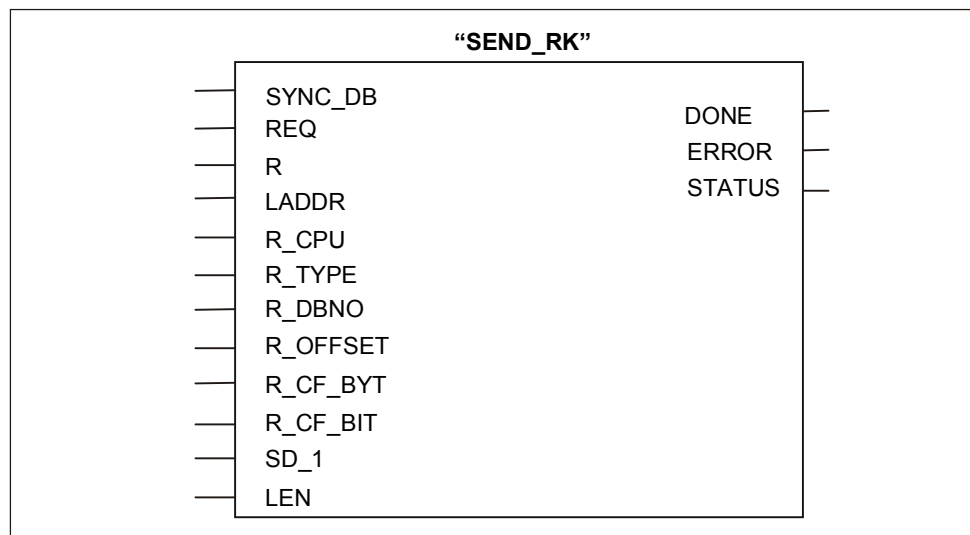
Для инициализации при запуске и синхронизации SFB друг с другом все SFB, используемые вами для компьютерного интерфейса RK 512, нуждаются в общей области данных. Номер DB определяется через параметр SYNC\_DB. Номер DB должен быть одинаков для всех используемых в вашей пользовательской программе SFB. Этот DB должен иметь длину не менее 240 байт.

### Связующий меркер

Известная в SIMATIC S5 функция меркера связи поддерживается системным функциональным блоком "SERVE\_RK" (SFB 65), для координации асинхронной перезаписи при приеме или подготовки данных и их обработки на CPU.

### 6.5.2.1 Передача данных с помощью SFB 63 "SEND\_RK"

С помощью этого SFB передаются данные из блока данных.



Активизация процесса передачи производится после вызова блока и положительного фронта на управляющем входе **REQ**.

Область подлежащих передаче данных задается через **SD\_1** (номер DB и начальный адрес), длина блока передаваемых данных через **LEN**.

На этом SFB задается также принимающая область у партнера. Эта информация вносится CPU в заголовок кадра сообщения (см. также раздел 6.9.3) и передается партнеру.

Цель указывается через номер CPU **R\_CPU** (имеет значение только при многопроцессорном обмене данными), тип данных **R\_TYPE** (блоки данных (DB) и расширенные блоки данных (DX)), номер блока данных **R\_DBNO** и смещение **R\_OFFSET**, где должен быть записан первый байт.

С помощью **R\_CF\_BYT** и **R\_CF\_BIT** определяются байт и бит меркера связи на CPU партнера.

С помощью параметра **SYNC\_DB** определяется DB, в котором хранятся общие данные всех используемых вами SFB для инициализации при запуске и синхронизации. Номер DB для всех используемых в программе пользователя SFB.

Чтобы SFB мог выполнить задание, вы должны его вызвать с **R(Reset) = FALSE**. При положительном фронте на управляющем входе R текущий процесс передачи прекращается, а SFB переводится в исходное состояние. Прерванное задание завершается с сообщением об ошибке (выход **STATUS**).

С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то **DONE** устанавливается в TRUE, если задание было завершено с ошибками, то в TRUE устанавливается **ERROR**.

Если задание было пройдено с DONE = TRUE, то данные были отправлены партнеру, им положительно квитированы и переданы на CPU партнера.

В параметре STATUS в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8).

Параметры DONE или ERROR/STATUS выводятся также при сбросе (RESET) SFB (R = TRUE).

При возникновении ошибки двоичный результат BIE сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние TRUE.

#### Замечание

SFB не проверяет параметры, при неверной параметризации CPU может перейти в состояние STOP.

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации SFB RK (Минимальная длина 240 байт).	Зависит от CPU, ноль не разрешен.	0
<b>REQ</b>	IN	BOOL	Управляющий параметр «Request [Запрос]»: Активизирует обмен данными при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр «Reset [Сброс]»: Задание прерывается. Передача блокируется.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>R_CPU</b>	IN	INT	Номер CPU партнера (только в многопроцессорном режиме)	от 0 до 4	1
<b>R_TYPE</b>	IN	CHAR	Тип адреса на CPU партнера (разрешены только большие буквы): • 'D' = блок данных • 'X' = расширенный блок данных	'D', 'X'	'D'
<b>R_DBNO</b>	IN	INT	Номер блока данных на CPU партнера	от 0 до 255	0
<b>R_OFFSET</b>	IN	INT	Номер байта данных на CPU партнера	от 0 до 510 (только четные значения)	0
<b>R_CF_BYT</b>	IN	INT	Байт меркеров связи на CPU партнера (255 означает: без меркера связи)	от 0 до 255	255

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>R_CF_BIT</b>	IN	INT	Бит меркера связи на CPU партнера	от 0 до 7	0
<b>DONE</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): <ul style="list-style-type: none"> <li>FALSE: Задание еще не было запущено или еще выполняется.</li> <li>TRUE: Задание было выполнено без ошибок.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Параметр состояния (этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.): STATUS в зависимости от бита ERROR имеет следующее значение: <ul style="list-style-type: none"> <li>ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>0000h: ни предупреждения, ни ошибки</li> <li>&lt;&gt; 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0
<b>SD_1</b>	IN_OUT	ANY	Параметр передачи: Здесь вы указываете: <ul style="list-style-type: none"> <li>Номер DB, из которого передаются данные.</li> <li>Номер байта данных, начиная с которого должны передаваться данные.</li> </ul> Напр.: DB10 от байта 2 -> DB10.DBB2	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Здесь вы указываете длину подлежащих передаче данных в байтах. (Длина здесь указывается косвенно.)	от 1 до 1024	1

### Согласованность данных

Согласованность данных ограничена 128 байтами. Для согласованной передачи количества данных, превышающего 128 байт, необходимо учесть следующее:

Описывайте снова используемую в данный момент часть области передачи SD\_1 только тогда, когда процесс передачи закончен. Это имеет место, когда параметр состояния DONE принимает значение TRUE.

### Особенности при передаче данных

Обратите внимание на следующие особенности при передаче данных:

- С помощью RK 512 можно передавать только четное количество данных. Если вы в качестве длины (LEN) укажете нечетное число данных, то в конце данных передается дополнительный байт со значением "0".
- В случае RK 512 можно задавать только четное смещение. Если вы укажете нечетное смещение, то данные у партнера сохраняются с ближайшего меньшего четного смещения.

Пример: Если смещение равно 7, то данные сохраняются с байта 6.

### Данные в заголовке кадра сообщения

В следующей таблице представлены данные в заголовке кадра сообщения RK 512 (см. также раздел 6.9.3).

Источник на вашей системе автоматизации S7 (локальный CPU)	для цели, CPU партнера	Заголовок кадра сообщения		
		Байт 3/4: Вид команды	Байт 5/6: Z-DBNR/Z-Offset	Байты 7/8: Количество в
Блок данных	Блок данных	AD	DB/DW	словах
Блок данных	Расширенный блок данных	AD	DB/DW	словах

Объяснение сокращений:

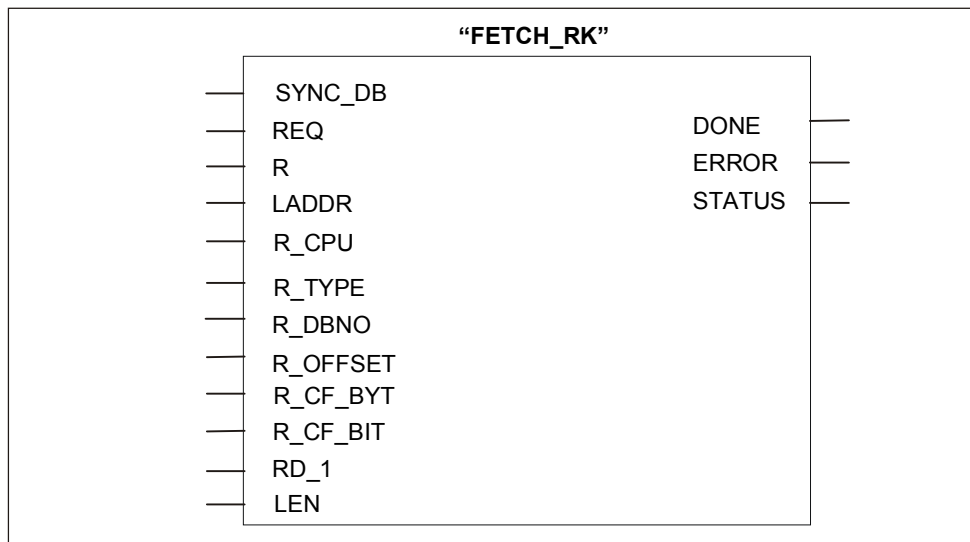
Z-DBNR: Номер целевого блока данных

Z-Offset: Начальный адрес цели

DW: Смещение в словах

### 6.5.2.2 Извлечение данных с помощью SFB 64 "FETCH\_RK"

С помощью этого SFB данные извлекаются у партнера и сохраняются в блоке данных.



Активизация процесса передачи производится после вызова блока и положительного фронта на управляющем входе **REQ**.

Область, в которой сохраняются извлеченные данные, задается через **RD\_1** (номер DB и начальный адрес), а длина блока передаваемых данных через **LEN**.

На SFB задается также, из какой области у партнера извлекаются данные. Эта информация вносится CPU в заголовок кадра сообщения RK512 (см. также раздел 6.9.3) и передается партнеру.

Область у партнера задается через номер CPU **R\_CPU** (имеет значение только при многопроцессорном обмене данными), тип данных **R\_TYPE** (блоки данных, расширенные блоки данных, меркеры, входы, выходы, счетчики и таймеры), номер блока данных **R\_DBNO** (имеет значение только у блоков данных и расширенных блоков данных) и смещение **R\_OFFSET**, откуда должен быть извлечен первый байт.

С помощью **R\_CF\_BYT** и **R\_CF\_BIT** определяется байт и бит меркера связи на CPU партнера.

С помощью параметра **SYNC\_DB** определяется DB, в котором хранятся общие данные всех используемых вами SFB для инициализации при запуске и синхронизации. Номер DB должен быть одинаков для всех применяемых в вашей пользовательской программе SFB.

Чтобы SFB мог выполнить задание, вы должны его вызвать с **R(Reset) = FALSE**. При положительном фронте на управляющем входе R текущая передача прекращается, и SFB переводится в исходное состояние. Прерванное задание завершается с сообщением об ошибке (выход STATUS).



С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то **DONE** устанавливается в TRUE, если задание было завершено с ошибками, то в TRUE устанавливается **ERROR**.

В параметре **STATUS** в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8).

Параметры DONE или ERROR/STATUS выводятся также при сбросе (RESET) SFB (R = TRUE).

При возникновении ошибки двоичный результат BIE сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние TRUE.

#### Замечание

SFB не проверяет параметры, при неверной параметризации CPU может перейти в состояние STOP.

Если данные извлекаются вашим CPU, то вы должны запрограммировать на своем CPU SFB "SERVE\_RK"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации SFB RK (Минимальная длина 240 байт).	Зависит от CPU, ноль не разрешен.	0
<b>REQ</b>	IN	BOOL	Управляющий параметр «Request [Запрос]»: Активизирует обмен данными при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр «Reset [Сброс]»: Выполнение задания прекращено.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>R_CPU</b>	IN	INT	Номер CPU партнера (только в многопроцессорном режиме)	от 0 до 4	1
<b>R_TYPE</b>	IN	CHAR	Тип адреса на CPU партнера (разрешены только большие буквы): <ul style="list-style-type: none"> <li>'D' = блок данных</li> <li>'X' = расширенный блок данных</li> <li>'M' = меркеры</li> <li>'E' = входы</li> <li>'A' = выходы</li> <li>'Z' = счетчики</li> <li>'T' = таймеры</li> </ul>	'D', 'X', 'M', 'E', 'A', 'Z', 'T'	'D'
<b>R_DBNO</b>	IN	INT	Номер блока данных на CPU партнера	от 0 до 255	0

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>R_OFFSET</b>	IN	INT	Номер байта данных на CPU партнера	См. таблицу: „Параметры на FB для источника данных (CPU партнера)“	0
<b>R_CF_BYT</b>	IN	INT	Байт меркеров связи на CPU партнера (255 означает: без меркера связи)	от 0 до 255	255
<b>R_CF_BIT</b>	IN	INT	Бит меркера связи на CPU партнера	от 0 до 7	0
<b>DONE</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): <ul style="list-style-type: none"> <li>FALSE: Задание еще не было запущено или еще выполняется.</li> <li>TRUE: Задание было выполнено без ошибок.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (этот параметр устанавливается только на протяжении вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Параметр состояния (этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.): STATUS в зависимости от бита ERROR имеет следующее значение: <ul style="list-style-type: none"> <li>ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>0000h: ни предупреждения, ни ошибки</li> <li>&lt;&gt; 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0
<b>RD_1</b>	IN_OUT	ANY	Параметр приема: Здесь вы указываете: <ul style="list-style-type: none"> <li>номер DB, в котором сохраняются извлеченные данные.</li> <li>номер байта данных, с которого сохраняются извлеченные данные.</li> </ul> Напр.: DB10 от байта 2 -> DB10.DBB2	Зависит от CPU	0

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
LEN	IN_OUT	INT	Здесь указывается длина подлежащего извлечению кадра сообщения в байтах. (Длина здесь указывается косвенно.) Для каждого таймера и счетчика в качестве длины нужно указывать два байта.	от 1 до 1024	1

### Согласованность данных

Согласованность данных ограничена 128 байтами. Для согласованной передачи количества данных, превышающего 128 байт, необходимо учесть следующее:

Используемую в данный момент часть принимающей области RD\_1 можно заменять только тогда, когда процесс передачи завершен. Это имеет место, когда параметр состояния DONE принимает значение TRUE.

### Особенности в случае (расширенных) блоков данных

Обратите внимание на следующие особенности извлечения данных у блоков данных и расширенных блоков данных:

- С помощью RK 512 можно извлечь только четное количество данных. Если в качестве длины (LEN) вы указали нечетное число, то всегда передается еще один байт. В целевой DB, однако, вносится правильное количество данных.
- В случае RK 512 можно указывать только четное смещение. Если вы задали нечетное смещение, то данные у партнера извлекаются из ближайшего меньшего четного смещения.

Пример: Смещение равно 7, данные извлекаются, начиная с байта 6.

### Особенности для таймеров и счетчиков

Если от партнера по обмену данными вы извлекаете таймеры или счетчики, то вы должны учесть, что для каждого таймера или счетчика извлекаются по 2 байта. Если вы, напр., хотите извлечь 10 счетчиков, то вы должны ввести в качестве длины 20.

## Параметры на SFB для источника данных (CPU партнера)

В следующей таблице представлены типы данных, которые можно передавать.

Источник на CPU партнера	R_TYPE	R_DBNO	R_OFFSET** (в байтах)
Блок данных	'D'	0 - 255	0 - 510*
Расширенный блок данных	'X'	0 - 255	0 - 510*
Меркеры	'M'	не имеет значения	0 - 255
Входы	'E'	не имеет значения	0 - 255
Выходы	'A'	не имеет значения	0 - 255
Счетчики	'Z'	не имеет значения	0 - 255
Таймеры	'T'	не имеет значения	0 - 255

\* Имеют смысл только четные значения!

\*\* Это значение задается через CPU партнера.

## Данные в заголовке кадра сообщения

В следующей таблице представлены данные в заголовке кадра сообщения RK 512 (см. также раздел 6.9.3).

Источник на CPU партнера	для цели, ваша система автоматизации S7 (локальный CPU)	Заголовок кадра сообщения		
		Байт 3/4: Вид команды	Байт 5/6: Q-DBNR/Q-Offset	Байт 7/8: Количество в
Блок данных	Блок данных	ED	DB/DW	словах
Расширенный блок данных	Блок данных	EX	DB/DW	словах
Меркеры	Блок данных	EM	Байтовый адрес	байтах
Входы	Блок данных	EE	Байтовый адрес	байтах
Выходы	Блок данных	EA	Байтовый адрес	байтах
Счетчики	Блок данных	EZ	Номер счетчика	словах
Таймеры	Блок данных	ET	Номер таймера	словах

Объяснение сокращений:

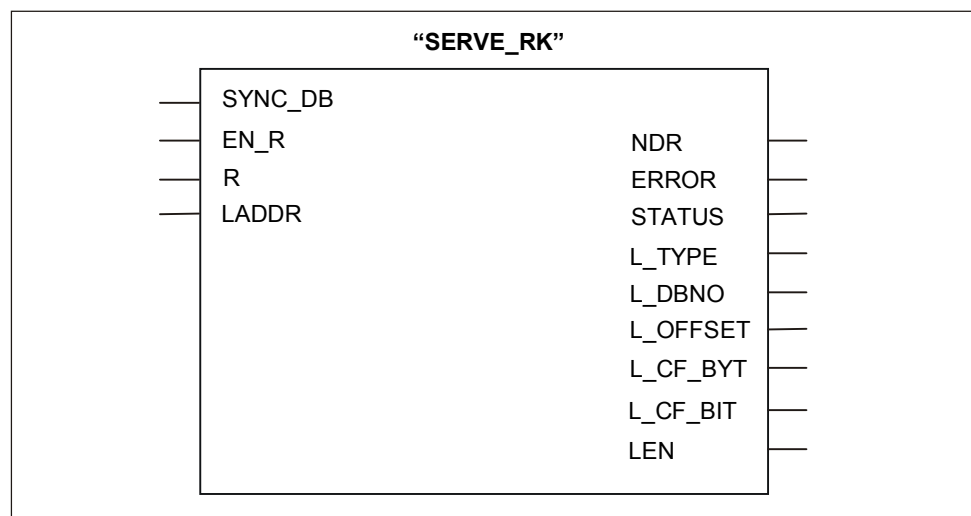
Q-DBNR: Номер блока данных источника

Q-Offset: Начальный адрес источника

### 6.5.2.3 Прием и подготовка данных с помощью SFB 65 "SERVE\_RK"

SFB применяется для

- **приема данных:** Данные сохраняются в области данных, указанной партнером в заголовке кадра сообщения RK 512 (см. также раздел 6.9.3). Вызов SFB необходим, когда партнер по обмену данными выполняет задание "Передача данных" (задание SEND).
- **подготовки данных:** Данные извлекаются из области данных, указанной партнером в заголовке кадра сообщения RK 512 (см. также раздел 6.9.3). Вызов SFB необходим, когда партнер по обмену данными выполняет задание "Извлечение данных" (задание FETCH).



SFB готов к действию после вызова со значением TRUE на управляющем входе EN\_R. Текущую передачу можно прекратить с помощью состояния сигнала FALSE на параметре **EN\_R**. Прерванное задание завершается с сообщением об ошибке (выход STATUS). Прием выключается, пока на параметре EN\_R сохраняется состояние FALSE.

С помощью параметра **SYNC\_DB** определяется DB, в котором хранятся общие данные всех используемых вами SFB для инициализации при запуске и синхронизации. Номер DB должен быть одинаков для всех применяемых в вашей пользовательской программе SFB.

Чтобы SFB мог обработать задание, вы должны его вызвать с **R(Reset) = FALSE**. При положительном фронте на управляющем входе R текущая передача прекращается, и SFB переводится в исходное состояние. Прерванное задание завершается с сообщением об ошибке (выход STATUS).

С помощью **LADDR** вы указываете адрес входов/выходов вашего субмодуля, который вы определили в «HW Konfig».

Если задание было завершено без ошибок, то параметр **NDR** устанавливается в TRUE, если задание завершилось с ошибкой, то в TRUE устанавливается параметр **ERROR**.

Блок с **NDR = TRUE** отображает в течение вызова в параметрах **L\_TYPE**, **L\_DBNO** и **L\_OFFSET**, где данные были сохранены или откуда они были извлечены. Кроме того, в течение вызова отображаются параметры **L\_CF\_BYT** и **L\_CF\_BIT** и длина **LEN** соответствующего задания.

В параметре **STATUS** в случае ошибки или предупреждения отображается соответствующий номер события (см. раздел 6.10.8).

**NDR** или **ERROR/STATUS** выводятся также при сбросе (RESET) **SFB** (**R = TRUE**) (параметр **LEN = 16#00**).

При возникновении ошибки двоичный результат **BIE** сбрасывается. Если блок завершается без ошибок, то двоичный результат имеет состояние **TRUE**.

#### Замечание

**SFB** не проверяет параметры, при неверной параметризации **CPU** может перейти в состояние **STOP**.

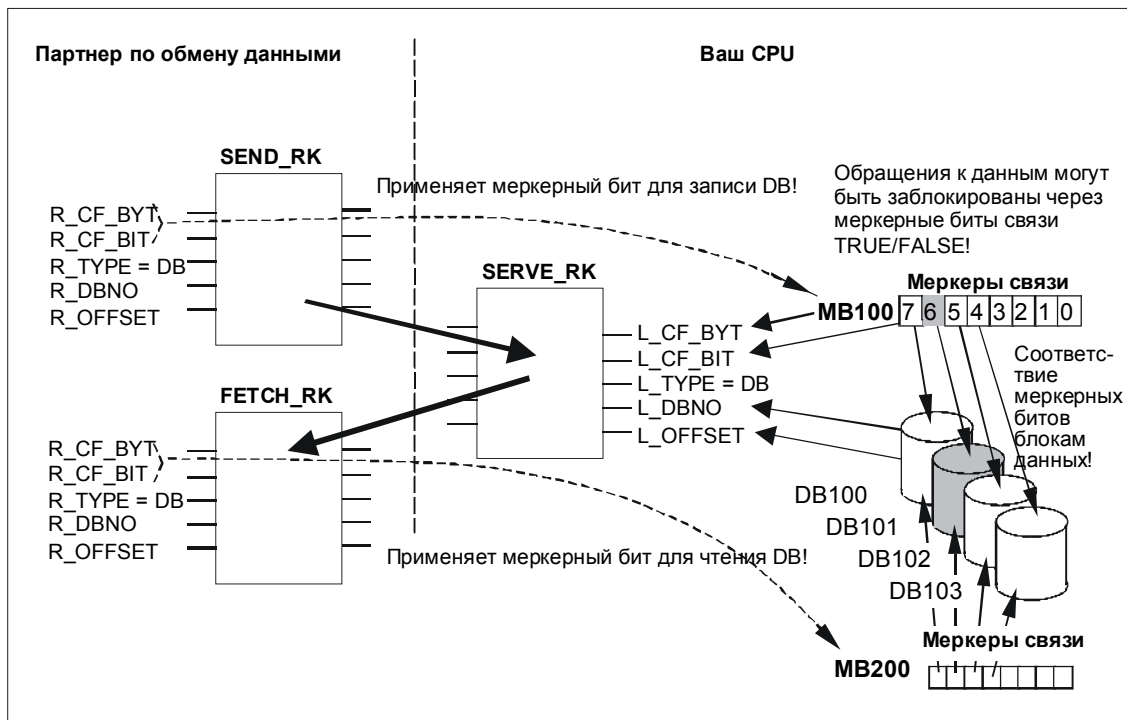
Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации <b>SFB RK</b> (Минимальная длина 240 байт).	Зависит от <b>CPU</b> , но не разрешен.	0
<b>EN_R</b>	IN	BOOL	Управляющий параметр «Enable to receive [Разрешение на прием]»: Деблокировка задания	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Управляющий параметр «Reset [Сброс]»: Выполнение задания прекращено.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от <b>CPU</b>	3FFh
<b>NDR</b>	OUT	BOOL	Параметр состояния «New data ready [Новые данные готовы]» (Этот параметр устанавливается только на время вызова): Задание завершено без ошибок <ul style="list-style-type: none"> <li>FALSE: Задание еще не было запущено или еще выполняется.</li> <li>TRUE: Задание было успешно завершено.</li> </ul>	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Параметр состояния (Этот параметр устанавливается только на время вызова): Задание завершено с ошибкой	TRUE/FALSE	FALSE

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>STATUS</b>	OUT	WORD	<p>Параметр состояния (этот параметр устанавливается только на протяжении вызова. Поэтому для отображения параметра STATUS его следует скопировать в свободную область данных.):</p> <p>STATUS в зависимости от бита ERROR имеет следующее значение:</p> <ul style="list-style-type: none"> <li>• ERROR = FALSE: STATUS имеет значение: <ul style="list-style-type: none"> <li>- 0000h: ни предупреждения, ни ошибки</li> <li>- &lt;&gt; 0000h: предупреждение, STATUS дает подробную справку</li> </ul> </li> <li>• ERROR = TRUE: Имеет место ошибка. STATUS дает подробную справку о виде ошибки (номер ошибки см. раздел 6.10.8).</li> </ul>	от 0 до FFFFh	0
<b>LEN</b>	IN_OUT	INT	Длина кадра сообщения, количество в байтах (Этот параметр устанавливается только на время вызова.)	от 0 до 1024	0
<b>L_TYPE</b>	OUT	CHAR	<p>(Параметры вида L_... устанавливаются только на время вызова.)</p> <p>Данные приняты:</p> <p>Тип целевой области на локальном CPU (разрешены только большие буквы):</p> <p>'D' = блок данных</p> <p>Подготовка данных:</p> <p>Тип области-источника на локальном CPU (разрешены только большие буквы):</p> <ul style="list-style-type: none"> <li>• 'D' = блок данных</li> <li>• 'M' = меркеры</li> <li>• 'E' = входы</li> <li>• 'A' = выходы</li> <li>• 'Z' = счетчики</li> <li>• 'T' = таймеры</li> </ul>	<p>'D'</p> <p>'D', 'M', 'E', 'A', 'Z', 'T'</p>	''
<b>L_DBNO</b>	OUT	INT	Номер блока данных на локальном CPU	Зависит от CPU	0
<b>L_OFFSET</b>	OUT	INT	Номер байта данных на локальном CPU	0-510	0
<b>L_CF_BYT</b>	OUT	INT	Байт меркеров связи на локальном CPU (255: означает: без меркера связи)	от 0 до 255	0
<b>L_CF_BIT</b>	OUT	INT	Бит меркера связи на локальном CPU	от 0 до 7	0

## Применение меркеров связи

Через меркер связи вы можете блокировать и деблокировать задания SEND и FETCH своего партнера по обмену данными. Так вы можете воспрепятствовать тому, чтобы данные, которые еще не были обработаны, переписывались или считывались.

Вы можете определить меркер связи для каждого задания.



### Пример: SEND\_RK с меркером связи:

В этом примере партнер посылает данные в DB 101 на вашем CPU.

1. Установите на своем CPU меркер связи 100.6 на FALSE.
2. Укажите у партнера по обмену данными на задании SEND меркер связи 100.6 (параметры R\_CF\_BYT, R\_CF\_BIT).

Меркер связи передается на ваш CPU в заголовке кадра сообщения RK 512 (структура заголовка кадра сообщения описана в разделе 6.9.3).

Перед обработкой задания ваш CPU проверяет меркер связи, указанный в заголовке кадра сообщения RK 512. Задание обрабатывается только тогда, когда меркер связи на вашем CPU имеет значение FALSE. Если меркер связи имеет значение TRUE, то партнеру по обмену данными посылается в ответном кадре сообщение об ошибке "32h".

После того как данные переданы в DB101, SFB SERVE устанавливает меркер связи 100.6 на вашем CPU на значение TRUE, а байт и бит меркера связи выводятся на SFB SERVE в течение вызова (если NDR = TRUE).



3. В программе пользователя вы можете путем анализа меркера связи (меркер связи 100.6 = TRUE) узнать, что задание завершено, и переданные данные могут обрабатываться.
4. После того как вы обработали в своей пользовательской программе, вы должны снова установить меркер связи 100.6 на FALSE. Только тогда ваш партнер по соединению снова сможет выполнить задание без ошибок.

### Согласованность данных

Согласованность данных ограничена 128 байтами. Для согласованной передачи количества данных, превышающего 128 байт, необходимо учесть следующее:

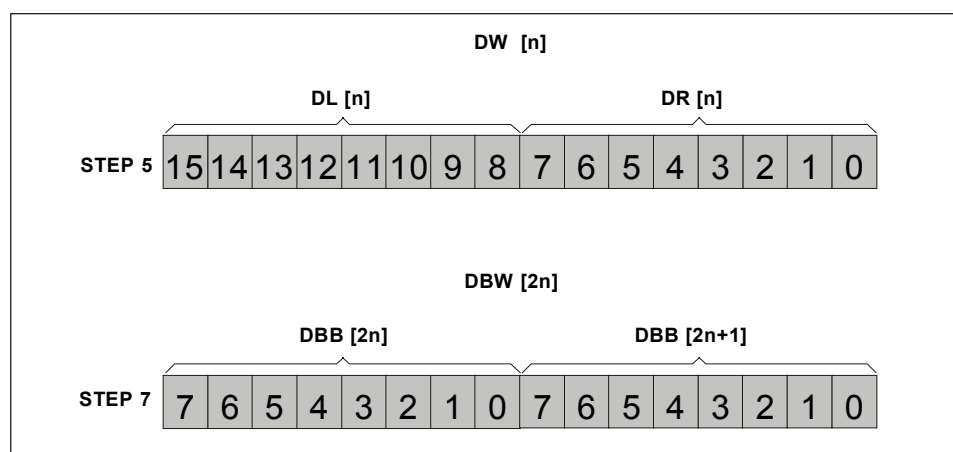
Используйте функцию меркера связи. Снова обращайтесь к данным только тогда, когда данные переданы полностью (анализ меркера связи, определенного для этого задания; меркер связи в ходе выполнения задания установлен на SFB, когда NDR = TRUE). Снова устанавливайте меркер связи на FALSE только тогда, когда вы обработали данные.

### 6.5.3 Указания по программированию системных функциональных блоков

Этот раздел ориентирован на всех, кто переходит от SIMATIC S5 на SIMATIC S7. Далее описано, на что вы должны обратить внимание при программировании функциональных блоков в STEP 7.

#### 6.5.3.1 Адресация

Адресация операндов в блоках данных в STEP 7 осуществляется байтами (в отличие от STEP 5, где адресация производится словами). Поэтому вы должны соответствующим образом пересчитать адреса операндов данных.



Адрес слова данных в STEP 7 относительно STEP 5 удваивается. Больше нет деления на правый и левый байт данных. Нумерация битов во всех случаях идет от 0 до 7.

## Примеры

Из операндов данных в STEP 5 (левый столбец таблицы) получаются операнды данных в STEP 7 (правый столбец таблицы).

STEP 5	STEP 7
DW 10	DBW 20
DL 10	DBB 20
DR 10	DBB 21
D 10.0	DBX 21.0
D 10.8	DBX 20.0
D 255.7	DBX 511.7

### 6.5.3.2 Присваивание значений параметрам блока

#### Прямая и косвенная параметризация

Косвенная параметризация, как в STEP 5 (передача параметров в открытый в данный момент блок данных), у блоков для STEP 7 невозможна.

Ко всем параметрам блока могут быть приложены как константы, так и переменные, так что в STEP 7 больше нет необходимости делать различие между прямой и косвенной параметризацией.

Исключение составляет параметр "LEN" у SFB 60, 63 и 64, который может получать значения только косвенно.

#### Пример "прямой параметризации"

Вызов SFB 60 "SEND\_PTP" в соответствии с "прямой параметризацией":

##### AWL

##### Netzwerk 1:

CALL SFB 60, DB10		
REQ	:= M 0.6	//запуск SEND
R	:= M 5.0	//запуск RESET
LADDR	:= +336	//адрес входов/выходов
DONE	:= M 26.0	//завершение без ошибок
ERROR	:= M 26.1	// завершение с ошибками
STATUS	:= MW 27	//слово состояния
SD 1	:= P#DB11.DBX0.0	//блок данных DB 11,
		//начиная с байта данных DBB 0
LEN	:= DB10.DBW20	//длина параметризуется косвенно

#### Пример "символической адресации фактических операндов"

Вызов SFB 60 "SEND\_PTP" с символической адресацией фактических операндов:

##### AWL

##### Netzwerk 1:

CALL SFB 60, DB10		
REQ	:= SEND_REQ	// запуск SEND
R	:= SEND_R	// запуск RESET
LADDR	:= BGADR	// адрес входов/выходов
DONE	:= SEND_DONE	// завершение без ошибок
ERROR	:= SEND_ERROR	// завершение с ошибками
STATUS	:= SEND_STATUS	// слово состояния
SD_1	:= QUELLZEIGER	// указатель типа ANY на целевую область
LEN	:= CPU_DB.SEND_LAE	// длина TG

## 6.6 Ввод в действие

### 6.6.1 Ввод в действие интерфейса на физическом уровне

Если после завершения проектирования не создается связь с устройством партнера, то необходимо проверить соединение. Для этого действуйте следующим образом:

Шаг	Что делать?
1	<p>Обнаружение источника ошибок:</p> <ul style="list-style-type: none"> <li>• Не перепутана ли полярность на передающей или принимающей линии?</li> <li>• Правильно ли выполнено предварительное распределение контактов? Возможно, предварительно было выполнено несколько распределений контактов с различной полярностью. Частично распределение контактов в устройстве фиксировано.</li> <li>• Отсутствуют или имеют неправильные номиналы оконечные сопротивления?</li> <li>• Перепутаны старший и младший байты в контрольном слове (напр., CRC)?</li> </ul>
2	<p>Последовательность действий:</p> <ul style="list-style-type: none"> <li>• Сначала проверьте присоединение проводов с помощью руководства: <ul style="list-style-type: none"> <li>- распределение контактов/полярность (см. раздел 6.2.2)</li> <li>- настройки по умолчанию (см. раздел 6.3)</li> </ul> </li> <li>• Затем протестируйте на экспериментальной структуре</li> </ul>
3	<p>Создание по возможности простой экспериментальной структуры:</p> <ul style="list-style-type: none"> <li>• Соедините друг с другом только двух абонентов</li> <li>• Если возможно, установите режим RS485 (2-проводная линия)</li> <li>• Используйте короткий соединительный кабель</li> <li>• Из-за малого расстояния оконечные сопротивления могут не понадобиться</li> <li>• Сначала выполните передачу в одном направлении, затем в другом</li> </ul>
4	<p>Испытание:</p> <ul style="list-style-type: none"> <li>• Случай 1: Полярность безусловно правильна <ul style="list-style-type: none"> <li>- Варьируйте распределение контактов (все возможности)</li> <li>- Проверьте контрольное слово (напр., CRC)</li> </ul> </li> <li>• Случай 2: Распределение контактов безусловно правильно <ul style="list-style-type: none"> <li>- Поменяйте местами присоединения (Внимание: у RS422 поменять местами обе пары проводов)</li> <li>- Проверьте контрольное слово (напр., CRC)</li> </ul> </li> <li>• Случай 3: Неизвестны ни правильная полярность, ни правильное назначение контактов <ul style="list-style-type: none"> <li>- Поменяйте местами присоединения (Внимание: у RS422 поменяйте местами провода в обеих парах)</li> <li>- В случае неудачи изменяйте распределение контактов (все возможности) с соответствующими попытками установления связи</li> <li>- В случае неудачи обратно поменяйте местами присоединения и изменяйте распределение контактов (все возможности)</li> <li>- Проверьте контрольное слово (напр., CRC)</li> </ul> </li> <li>• Не забудьте, пожалуйста, при окончательном монтаже установки вернуть на место возможно удаленные вами ранее оконечные сопротивления.</li> </ul>
5	<p>Дополнительные советы:</p> <ul style="list-style-type: none"> <li>• Включите в соединительный кабель испытатель интерфейса, если он есть, (при необходимости, преобразователь RS422/485 → V.24).</li> <li>• Проверьте измерительным прибором уровни сигналов (уровни измеряйте относительно GND (контакт 8)).</li> <li>• Некоторые устройства сообщают об отсутствии приема, когда данные принимаются, но контрольное слово CRC неверно.</li> <li>• В случае необходимости замените CPU, чтобы исключить электрический дефект.</li> </ul>

## 6.7 Обработка ошибок и прерывания

Диагностические функции позволяют быстро локализовать возникшие ошибки. В вашем распоряжении имеются следующие диагностические возможности:

- сообщения об ошибках в системном функциональном блоке (SFB)
- у RK 512: номера ошибок в ответном кадре сообщения
- диагностическое прерывание

### 6.7.1 Сообщения об ошибках в системном функциональном блоке (SFB)

При возникновении ошибки, параметр ERROR устанавливается на TRUE. В параметре STATUS отображается причина ошибки. Возможные номера ошибок приведены в разделе 6.10.8.

#### Замечание

Сообщение об ошибке выдается только тогда, когда одновременно устанавливается бит ERROR (завершение задания с ошибкой). В любом другом случае слово состояния STATUS пусто. Поэтому для отображения параметра STATUS вам следует скопировать STATUS при установленном бите ERROR в свободную область данных.

### 6.7.2 Номера ошибок в ответном кадре сообщения

Если вы работаете с компьютерным интерфейсом RK 512, и при использовании кадра сообщения SEND или FETCH у партнера по обмену данными возникает ошибка, то партнер передает ответный кадр сообщения с номером ошибки в 4-ом байте.

В следующей таблице вы найдете соответствие номеров ошибок в ответном кадре сообщения (REATEL) классам и номерам событий в слове STATUS партнера по обмену данными. Номера ошибок в ответном кадре сообщения выводятся как шестнадцатеричные значения.

REATEL	Сообщение об ошибке (Класс/номер события)
0Ah	0905h
0Ch	0301h, 0609h, 060Ah, 0902h
10h	0301h, 0601h, 0604h
12h	0904h
14h	0903h
16h	0602h, 0603h, 090Ah
2Ah	090Dh
32h	060Fh, 0909h
34h	090Ch
36h	060Eh, 0908h

### 6.7.3 Диагностическое прерывание

При обрыве провода последовательного соединения с партнером по обмену данными (080Dh) вы можете запустить диагностическое прерывание. Диагностическое прерывание отображается как при наступающих, так и при уходящих ошибках.

С помощью диагностического прерывания вы можете в своей пользовательской программе немедленно реагировать на ошибки.

#### Процесс

1. Разблокируйте диагностическое прерывание в экранной форме для параметризации "Basic parameters [Основные параметры]".
2. Вставьте в свою пользовательскую программу OB диагностических прерываний (OB 82)..

#### Реакция при ошибке с диагностическим прерыванием

- Диагностическое прерывание не оказывает влияния на исполняемую в данный момент функцию.
- Операционная система CPU вызывает в программе пользователя OB 82.

---

#### Замечание

Если запускается прерывание, а соответствующий OB не загружен, то CPU переходит в STOP.

---

- CPU включает светодиод SF.
- Ошибка отображается в диагностическом буфере CPU как "поступающая" и "уходящая".

#### Анализ диагностического прерывания в программе пользователя

После запуска диагностического прерывания вы можете в OB 82 проанализировать, какое диагностическое прерывание имеет место.

- Если в OB 82, байт 6 +7 (OB 82\_MDL\_ADDR), внесен адрес вашего субмодуля, то диагностическое прерывание было запущено двухточечным соединением вашего CPU.
- Если в очереди стоит хотя бы еще одна ошибка, то в OB 82, байт 8 установлен бит 0 (модуль неисправен).
- Если обо всех стоящих в очереди ошибках сообщается, что они "уходящие", то в OB 82, байт 8 бит 0 сбрасывается.
- При обрыве провода в последовательном соединении в байтах 8 и 10 одновременно устанавливаются биты ошибок "Модуль неисправен", "Обрыв провода", "Внешняя ошибка" и "Коммуникационная ошибка".

ОВ82, байт 8	Описание:
Бит 0	Модуль неисправен
Бит 1	-
Бит 2	Внешняя ошибка
Бит 3	-
Бит 4	-
Бит 5	Обрыв провода
Бит 6	-
Бит 7	-

ОВ82, байт 10	Описание:
Бит 0	-
Бит 1	Коммуникационная ошибка
Бит 2	-
Бит 3	-
Бит 4	-
Бит 5	-
Бит 6	Потеряно аппаратное прерывание
Бит 7	-

## 6.8 Примеры

Примеры (программа и описание) находятся на прилагаемом к вашей документации компакт-диске, или вы можете получить их через Интернет. Проект состоит из нескольких откомментированных программ S7 различной сложности и назначения.

Инсталляция примеров описана на компакт-диске в файле readme.wri. После инсталляции примеры находятся в каталоге  
**...\\STEP7\\EXAMPLES\\ZDt26\_01\_TF\_\_\_\_31xC\_PtP.**

## 6.9 Описание протокола

### 6.9.1 Передача данных с помощью драйвера ASCII

Драйвер ASCII управляет передачей данных при двухточечном соединении между CPU и партнером по обмену данными.

Структура кадра сообщения сохраняется благодаря тому, что пользователь S7 передает на двухточечный интерфейс весь отправляемый кадр сообщения. Для направления приема должен быть при параметризации указан критерий окончания кадра сообщения. Структура передаваемого кадра сообщения может отличаться от структуры принимаемого кадра сообщения.

С помощью драйвера ASCII могут передаваться и приниматься данные любой структуры (все печатные символы ASCII, а также все остальные символы от 00 до FFh (при кадре символа с 8 битами данных) или от 00 до 7Fh (при кадре символа с 7 битами данных)).

Возможен как режим RS422, так и RS485.

#### Режим RS422

При использовании режима RS422 передача данных осуществляется через четыре провода (четырёхпроводный режим). Имеются в распоряжении по два провода (разностный сигнал) для направления передачи и для направления приема. Поэтому возможны одновременно передача и прием (полнодуплексный режим).

#### Режим RS485

При использовании режима RS485 передача данных осуществляется через два провода (двухпроводный режим). Эти два провода (разностный сигнал) по очереди предоставляются в распоряжение направлению передачи и направлению приема. Поэтому возможна только передача или только прием (полудуплексный режим). По окончании процесса передачи немедленно производится переключение на прием (передатчик высокоомный). Время переключения составляет не более 1 мс.

### Передача данных с помощью драйвера ASCII

При передаче количество байтов подлежащих передаче полезных данных указывается при вызове SFB в качестве параметра "LEN".

Если вы работаете с **критерием конца кадра сообщения "Истечение времени задержки символа"**, то драйвер ASCII и при передаче выдерживает паузу между двумя кадрами сообщений. Вы можете вызвать SFB в любой момент времени, но драйвер ASCII начинает вывод только тогда, когда после последнего отправленного кадра сообщения прошло время, большее, чем указанное при параметризации время задержки символа (CDT).

Если вы работаете с **критерием конца кадра сообщения "Фиксированное число символов"**, то в направлении передачи передается количество данных, указанное вами у SFB SEND\_PTP в параметре "LEN". В направлении приема, т.е. в принимающий DB, вносится количество данных, которое вы указали у приемника в экранной форме для параметризации через параметр



"Fixed character length [фиксированное число символов]". Чтобы обеспечить безупречный обмен данными, оба параметра следует выбирать одинаковыми. При передаче между двумя кадрами сообщений выдерживается пауза, равная по длине времени контроля, при отсутствии признака конца, чтобы партнер мог войти в синхронизм (распознавание начала кадра сообщения).

Если синхронизация осуществляется другими механизмами, то соблюдение паузы при передаче может быть отключено с помощью пользовательского интерфейса для проектирования.

Если вы работаете с **критерием конца кадра сообщения "Символ конца текста"**, то у вас есть три возможности для выбора:

1. Передача до символа конца текста включительно:  
Символ конца текста должен содержаться в посылаемых данных. Данные передаются только до конечного символа включительно, даже если на SFB указана большая длина данных.
2. Передача до длины, указанной при параметризации на SFB:  
Данные передаются до длины, указанной при параметризации на SFB. Последний символ должен быть конечным.
3. Передача до длины, указанной при параметризации на SFB и автоматическое присоединение конечного символа (-ов).  
Передача до длины, указанной при параметризации на SFB. Кроме того, автоматически присоединяется конечный символ (символы); т.е. конечные символы не должны содержаться в передаваемых данных. В зависимости от количества концевых символов партнеру передается на 1 или 2 символа больше, чем задано в SFB (максимум 1024 байта).

---

#### **Замечание**

При параметризации управления потоком данных с помощью XON/XOFF полезные данные не должны содержать ни одного из указанных при параметризации символов XON или XOFF. Настройками по умолчанию являются DC1 = 11h для XON и DC3 = 13h для XOFF.

---

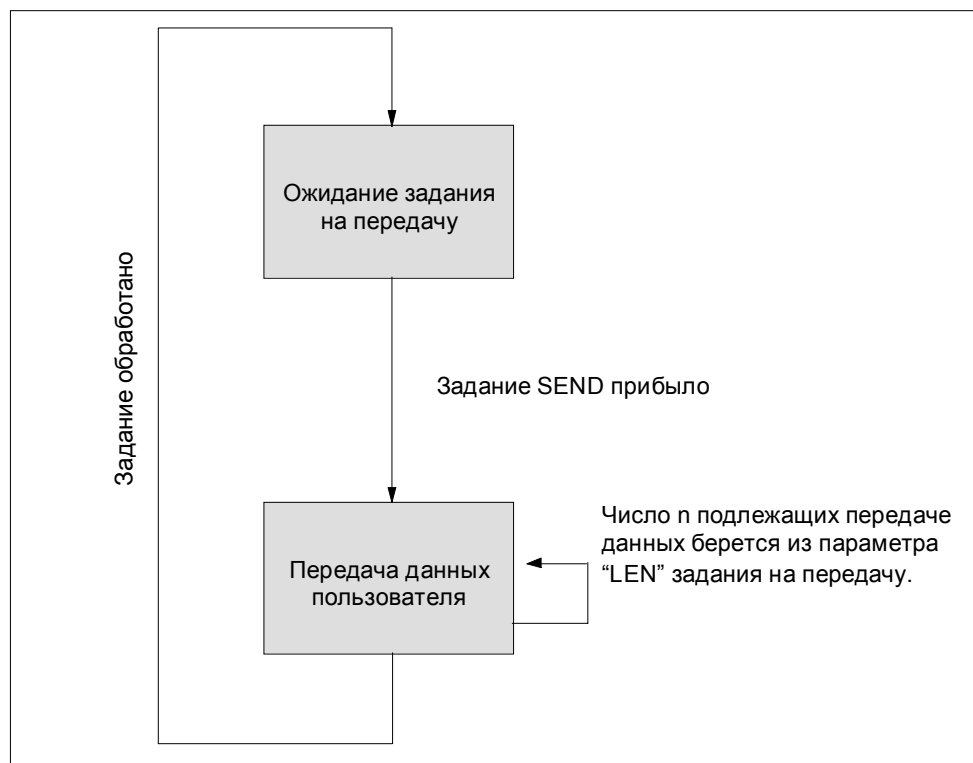
### **Передача символа контроля блока**

Если вы хотите защитить данные с помощью одного или двух символов контроля блока (BCC), то при использовании критерия конца кадра сообщения "End-of-text Character [Символ конца текста]" вы должны применить настройку "Sending up to the length declared in the SFB parameters [Передача до длины, указанной в параметрах SFB]". Тогда вы можете после конечного символа дополнительно передать один или два символа контроля блока.

Расчет символа контроля блока вы должны выполнять сами в программе пользователя.

## Передача данных

На следующем рисунке представлены процессы, происходящие при передаче:



## Данные, принятые с помощью драйвера ASCII

При передаче данных с помощью драйвера ASCII вы можете при приеме данных выбирать среди трех различных критериев окончания передачи. Критерий окончания передачи определяет, когда кадр сообщения полностью принят. Можно установить следующие критерии:

- Истечение времени задержки символа (CDT):  
Кадр сообщения не имеет ни фиксированной длины, ни определенного конечного символа, конец кадра сообщения определяется паузой на линии (истечение времени задержки символа).
- Прием определенного числа символов:  
Длина принимаемого кадра сообщения всегда одна и та же.
- Прием конечного символа (-ов):  
В конце кадра сообщения стоят один или два определенных конечных символа.

## Независимость от кода

Независимость процедуры от кода определяется выбором устанавливаемого при параметризации критерия окончания передачи и управления потоком данных:

- С одним или двумя конечными символами:  
нет независимости от кода
- Критерием окончания является время задержки символа (CDT) или фиксированное число символов:  
независимость от кода имеет место
- При применении контроля потока XON/XOFF кодонезависимая работа невозможна.

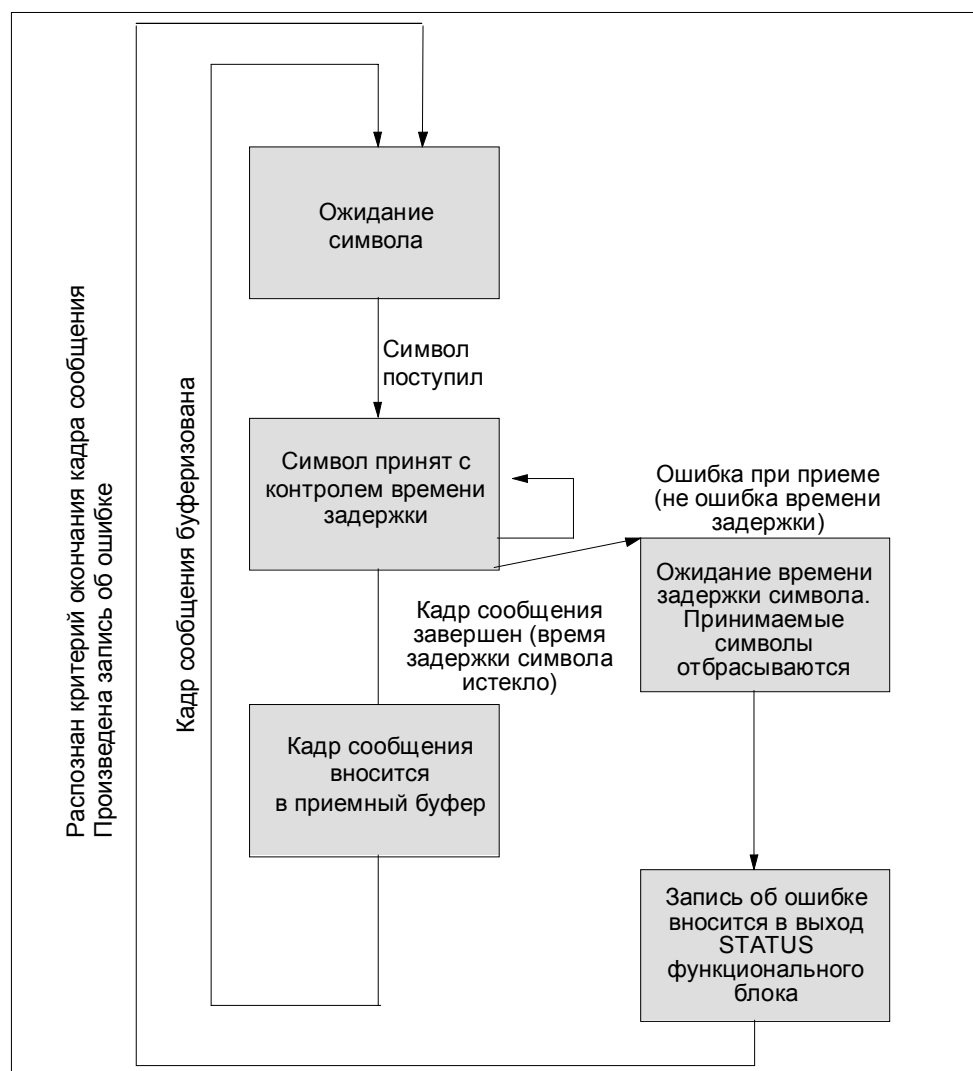
"Независимость от кода" означает, что в данных пользователя могут встречаться любые комбинации символов, не оказывая влияния на распознавание конца передачи.

**Критерий окончания "Истечение времени задержки символа (CDT)"**

При приеме данных окончание кадра сообщения распознается, когда истекает время задержки символа. CPU принимает поступившие данные.

Время задержки символа в этом случае должно быть установлено таким образом, чтобы оно наверняка истекало между двумя следующими друг за другом кадрами сообщений. Но оно должно быть достаточно большим, чтобы при паузах в передаче партнера по соединению внутри кадра сообщения ошибочно не распознавался ее конец.

На следующем рисунке представлены процессы при приеме с критерием окончания "Истечение времени задержки символа":



**Критерий окончания "Фиксированное количество символов"**

При приеме данных конец кадра сообщения распознается, когда принято количество символов, указанное при параметризации. CPU принимает поступившие данные.

Истечение времени задержки символа (CDT) до достижения указанного при параметризации числа символов ведет к завершению приема. Время задержки символа используется в этом случае как время контроля. Выдается сообщение об ошибке, а фрагмент кадра сообщения отбрасывается.

Если количество принимаемых символов не совпадает с фиксированным количеством, установленным при параметризации, то обратите внимание на следующее:

- Количество принимаемых символов больше количества, установленного при параметризации:

Все символы, принимаемые после достижения количества символов, установленного при параметризации,

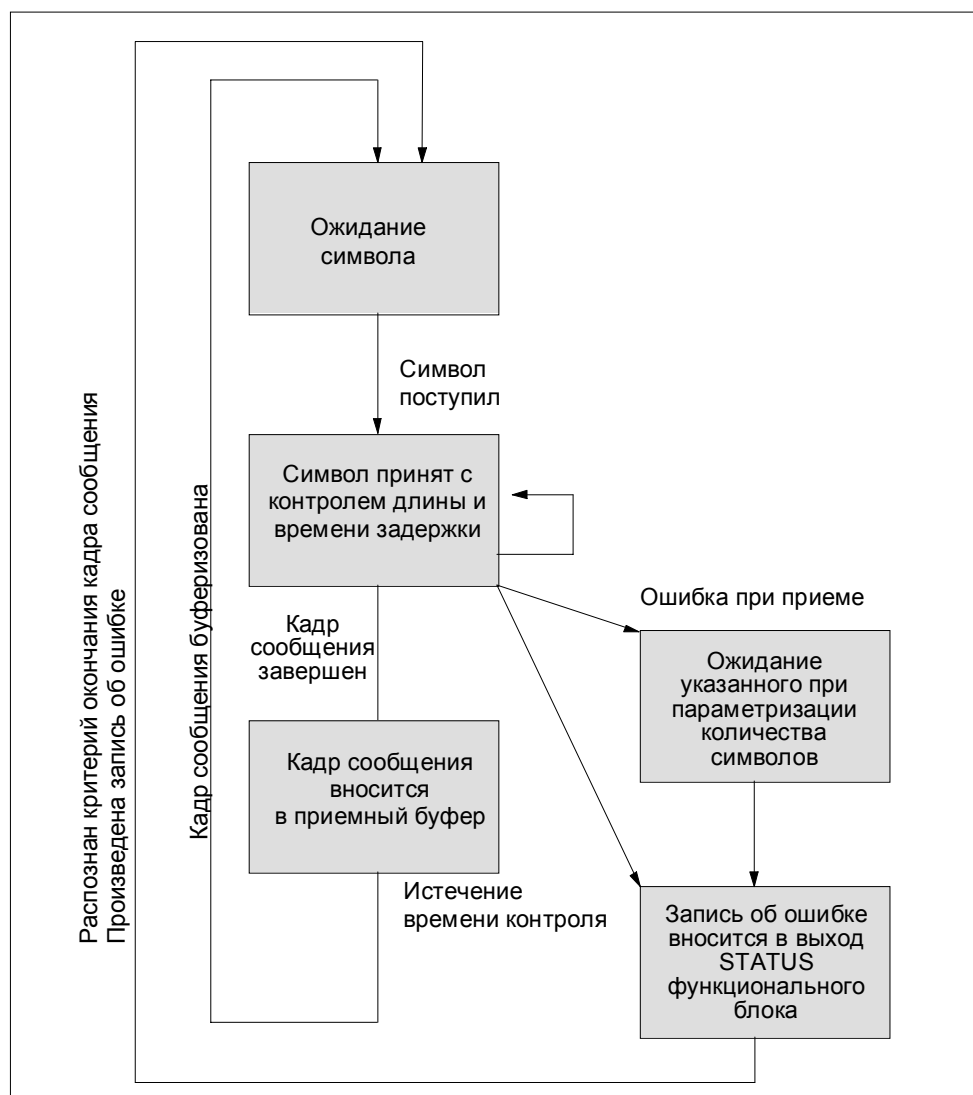
- отбрасываются, если к концу кадра сообщения истекает время контроля.
- сливаются со следующим кадром сообщения, если новый кадр сообщения принимается до истечения времени контроля.

- Количество принимаемых символов меньше количества, установленного при параметризации:

Кадр сообщения

- отбрасывается, если к концу кадра сообщения истекает время контроля.
- сливается со следующим кадром сообщения, если новый кадр сообщения принимается до истечения времени контроля.

На следующем рисунке представлены процессы при приеме с критерием окончания "Фиксированное количество символов":



**Критерий окончания "Символ конца текста"**

При приеме данных конец кадра сообщения распознается, когда принимается указанный (-ые) при параметризации символ (-ы) конца текста. У вас есть следующие возможности для выбора:

- один конечный символ
- два конечных символа

CPU принимает поступившие данные, включая символы конца текста.

Если в принимаемых данных отсутствует конечный символ, то время задержки символа (CDT) истекает во время приема, что ведет к завершению кадра сообщения. Время задержки символа используется в этом случае как время контроля. Выдается сообщение об ошибке, а фрагмент кадра сообщения отбрасывается.

При работе с конечными символами передача не является кодонезависимой, и наличие идентификатора (-ов) конца в полезных данных пользователя должно быть исключено.

Если в принимаемом кадре сообщения последний символ не является конечным, то обратите внимание на следующее:

- Символ конца текста в кадре сообщения стоит на произвольном месте:

Все символы до конечного включительно вносятся в принимающий DB. Символы, стоящие после символа конца текста,

- отбрасываются, если к концу кадра сообщения истекает время контроля.
- сливаются со следующим кадром сообщения, если новый кадр сообщения принимается до истечения времени контроля.

- Символ конца текста не содержится в кадре сообщения :

Кадр сообщения

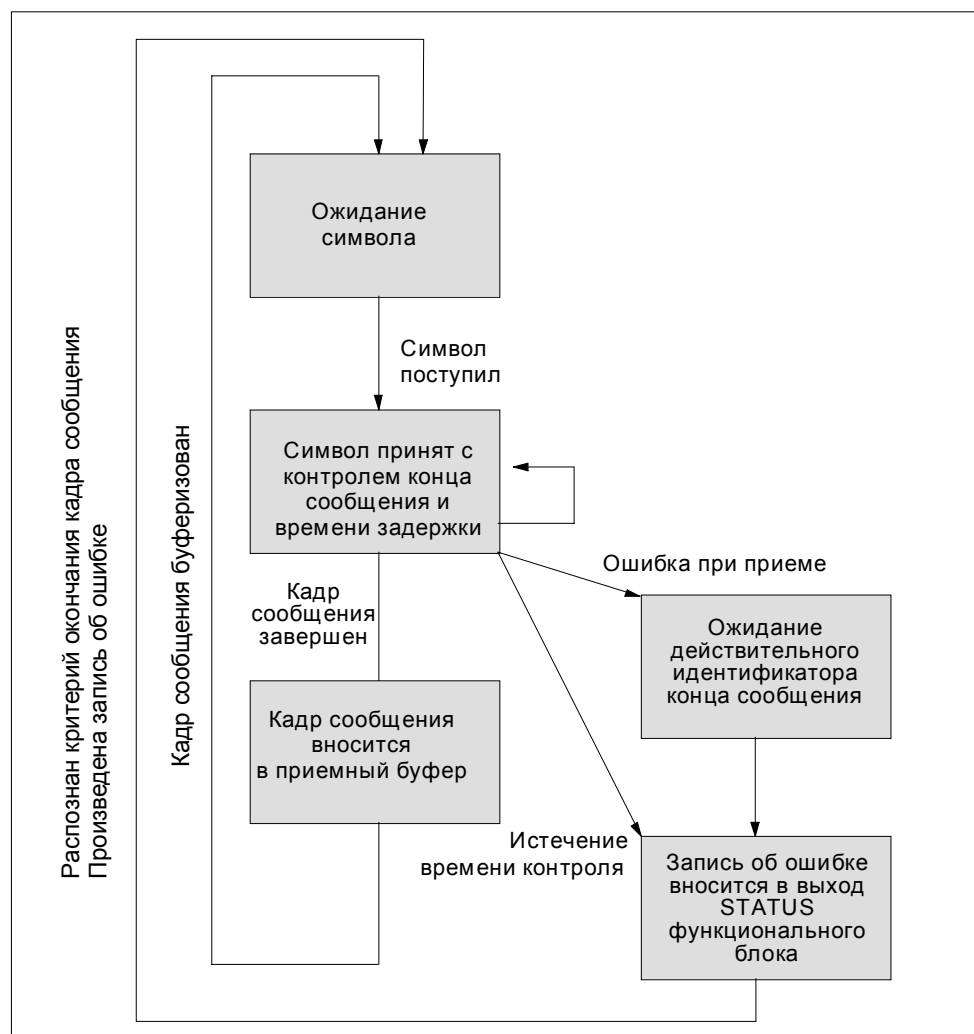
- отбрасывается, если к концу кадра сообщения истекает время контроля.
- сливается со следующим кадром сообщения, если новый кадр сообщения принимается до истечения времени контроля.

## Прием с символами контроля блока

Кроме символа конца текста, через экранной формы для параметризации вы можете выбрать, хотите ли вы работать с одним или двумя символами контроля блока (BCC). Тогда в принимающий DB дополнительно вносятся символы (1 или 2), следующие за конечным символом.

Анализ символа контроля блока вы должны сами выполнить в программе пользователя.

На следующем рисунке представлены процессы при приеме с критерием окончания "Символ конца текста":





## Приемный буфер на CPU

Приемный буфер имеет величину 2048 байтов. При параметризации вы можете указать, нужно ли препятствовать перезаписи данных в приемном буфере. Кроме того, вы можете указать диапазон значений (от 1 до 10) для количества буферизуемых кадров сообщений или объем всего приемного буфера.

Приемный буфер можно стирать при запуске. Эта настройка осуществляется по выбору через экранную форму параметризации или вызовом SFB RES\_RCV (см. раздел 6.5.1.3).

Приемный буфер является кольцевым буфером:

- Если в приемный буфер вносятся несколько кадров сообщений, то имеет место следующее: В целевой блок данных в качестве первого всегда передается самый старый кадр сообщения.
- Если вы хотите всегда передавать в целевой блок данных только самый новый кадр сообщения, то при параметризации в качестве количества буферизуемых кадров сообщений нужно указать значение "1" и деактивизировать защиту от перезаписи.

### Замечание

Если в программе пользователя на некоторое время приостановлено постоянное считывание принимаемых данных, то при новом запросе принимаемых данных может получиться, что в целевой блок данных сначала вносится старый кадр сообщения и только потом самый новый кадр сообщения.

Старыми кадрами сообщений являются те кадры, которые во время перерыва находились в пути между CPU и партнером или уже были приняты SFB.

## Управление потоком данных/процедуры квитирования

Процедуры квитирования управляют потоком данных между двумя партнерами по обмену данными. Благодаря процедурам квитирования удастся избежать потерь данных при передаче в случае устройств, работающих с различной скоростью. CPU поддерживает квитирование с помощью XON/XOFF.

Реализация управления потоком данных осуществляется следующим образом:

1. Как только CPU путем параметризации переведен в режим с управлением потоком данных, он передает символ XON.
2. При достижении указанного при параметризации количества кадров сообщений или 50 символов до переполнения приемного буфера (размер приемного буфера 2048 байт) CPU передает символ XOFF. Если партнер по обмену данными несмотря на это продолжает передачу, то при переполнении приемного буфера генерируется сообщение об ошибке. Принятые данные последнего кадра сообщения отбрасываются.
3. Как только кадр сообщения извлечен из приемного буфера и буфер готов к приему, CPU передает символ XON.
4. Если CPU принимает символ XOFF, он прерывает процесс передачи. Если по истечении определенного времени, которое может быть указано при параметризации, не принимается XON, то процесс передачи прекращается и на выходе STATUS системного функционального блока генерируется соответствующее сообщение об ошибке (0708h).

## 6.9.2 Передача данных с помощью процедуры 3964(R)

Процедура 3964(R) управляет передачей данных при двухточечном соединении между CPU и партнером по обмену данными.

### Управляющие символы

Процедура 3964(R) при передаче данных добавляет к полезным данным управляющие символы. С помощью этих управляющих символов партнер по обмену данными может контролировать, прибыли ли к нему данные полностью и без ошибок.

Процедура 3964(R) анализирует следующие управляющие символы:

- STX: Start of Text (начало текста); начало подлежащей передаче последовательности символов
- DLE: Data Link Escape (переключение передачи данных) или положительное ответное сообщение
- ETX: End of Text (конец текста); конец подлежащей передаче последовательности символов
- BCC: Block Check Character (только у 3964(R)); символ контроля блока
- NAK: Negative Acknowledge (отрицательное ответное сообщение)

---

### Замечание

Если символ DLE передается в качестве информационного, то он передается дважды (удвоение DLE), чтобы отличить его от управляющего символа DLE, используемого при установлении и прекращении связи в передающей линии. Приемник аннулирует удвоение DLE.

---

### Приоритет

При использовании процедуры 3964(R) одному из партнеров по обмену данными должен быть присвоен более высокий приоритет, а другому – более низкий приоритет. Если оба партнера начинают устанавливать соединение одновременно, то партнер с более низким приоритетом откладывает свое задание на передачу.

## Контрольная сумма блока

У протокола передачи 3964(R) надежность передачи данных увеличивается с помощью дополнительно передаваемого символа контроля блока (BCC = Block Check Character).

Кодовая посылка:						
STX	Данные			DLE ETX		BCC
02H	30H	31H	32H	10H	03H	20H
	30	=	0011	0000		
	31	=	0011	0001		
	XOR	=	0000	0001		
	32	=	0011	0010		
	XOR	=	0011	0011		
	10	=	0001	0000		
	XOR	=	0010	0011		
	03	=	0000	0011		
	XOR	=	0010	0000		
BCC	2		0			

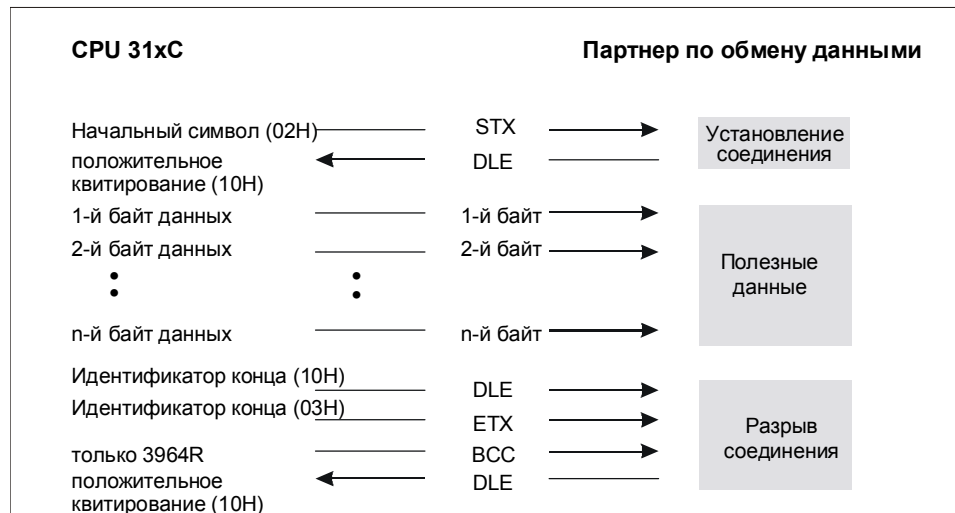
Контрольная сумма блока реализует контроль четности длины (логического сопряжения всех байтов данных с использованием исключающего ИЛИ) передаваемого или принимаемого блока. Ее формирование начинается с первого байта полезных данных (1-й байт кадра сообщения) после установления соединения и заканчивается после символа DLE ETX при разрыве соединения.

### Замечание

При удвоении DLE символ DLE дважды включается в формирование BCC.

## Передача данных с помощью 3964(R)

На следующем рисунке представлен процесс передачи данных с помощью процедуры 3964(R):



### Установление соединения при передаче

Для установления соединения процедура 3964(R) передает управляющий символ STX. Если партнер по обмену данными до истечения времени задержки квитирования (ADT) отвечает символом DLE, то процедура переходит в режим передачи.

Если партнер по обмену данными отвечает с помощью NAK или любого другого символа (кроме DLE или STX), или время задержки квитирования проходит без реакции, то процедура повторяет попытку установления соединения. После указанного при параметризации числа неудачных попыток установления соединения процедура прекращает устанавливать соединение и передает партнеру по обмену данными символ NAK. CPU сообщает об ошибке на SFB SEND\_PTP (выходной параметр STATUS).

### Передача данных

Если установление соединения происходит успешно, то данные, подлежащие передаче, передаются партнеру по обмену данными с выбранными параметрами передачи. Партнер проверяет временной интервал поступающих символов. Интервал между двумя символами не может превышать время задержки символа (CDT).

Если партнер по обмену данными во время текущей передачи передает символ NAK, то процедура завершает передачу блока и повторяет ее вышеописанным способом, начиная с установления соединения. При другом символе процедура сначала ожидает истечения времени задержки символа (CDT), а затем передает NAK, чтобы привести партнера в исходное состояние. После этого процедура снова начинает передачу с установления соединения STX.

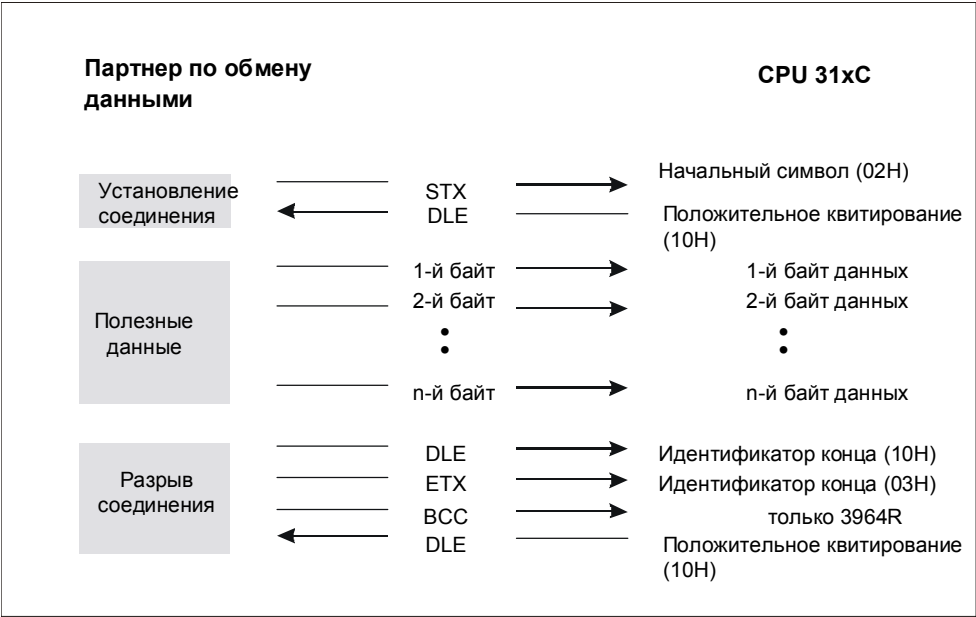
Установка соединения при передаче

После передачи содержимого буфера процедура добавляет символы DLE, ETX и, только у 3964(R), контрольную сумму блока BCC в качестве идентификатора конца и ждет символа квитирования. Если партнер по обмену данными передает в течение времени задержки квитирования символ DLE, то данные были приняты без ошибок. Если партнер по обмену данными отвечает передачей NAK, любого другого символа (кроме DLE) или искаженного символа, или время задержки квитирования прошло без реакции, то процедура снова начинает передачу с установления соединения STX.

По истечении указанного при параметризации числа попыток передачи данных, процедура прекращает попытки и передает партнеру по обмену данными NAK. Ошибка отображается на SFB SEND\_PTP (выходной параметр STATUS).

Данные, принятые с помощью 3964(R)

На следующем рисунке представлен процесс передачи данных при приеме с помощью процедуры 3964(R):



Замечание

Процедура 3964(R), как только она готова к работе, передает однократно символ NAK партнеру, чтобы привести его в исходное состояние.

## Установление соединения при приеме

В исходном состоянии, когда задание на передачу не должно обрабатываться, процедура ожидает установления соединения партнером по обмену данными.

Если при установлении соединения с помощью STX нет в распоряжении пустого приемного буфера, то запускается время ожидания в 400 мс. Если по истечении этого времени пустого приемного буфера еще нет, то на выходе STATUS функционального блока отображается ошибка. Процедура передает символ NAK и снова переходит в исходное состояние. В противном случае, процедура передает символ DLE и принимает данные.

Если процедура в исходном состоянии принимает любой символ (кроме STX или NAK), то она ждет истечения времени задержки символа (CDT), а затем передает символ NAK. Ошибка отображается на выходе STATUS функционального блока.

## Данные приняты

После удачного установления соединения поступающие полезные данные сохраняются в приемном буфере. Если принимаются два следующих друг за другом символа DLE, то в приемный буфер принимается только один DLE.

После каждого принятого символа в течение времени задержки символа (CDT) происходит ожидание следующего символа. Если время задержки символа проходит без приема, то партнеру по обмену данными передается символ NAK. Системная программа передает сообщение об ошибке на SFB RCV\_PTP (выходной параметр STATUS).

Если во время приема возникают ошибки передачи (потерянный символ, ошибка кадра, ошибка четности и т.д.), то прием продолжается до разрыва соединения, а затем партнеру по обмену данными передается символ NAK. После этого ожидается повторение. Если данные не смогли быть приняты без ошибок и после указанного в статическом наборе параметров числа попыток передачи или повторение не было начато партнером в течение времени ожидания данных (соответствует времени задержки квитирования), то процедура прекращает прием. CPU сообщает о первой неудачной передаче и об окончательном ее прекращении на SFB RCV\_PTP (выходной параметр STATUS).

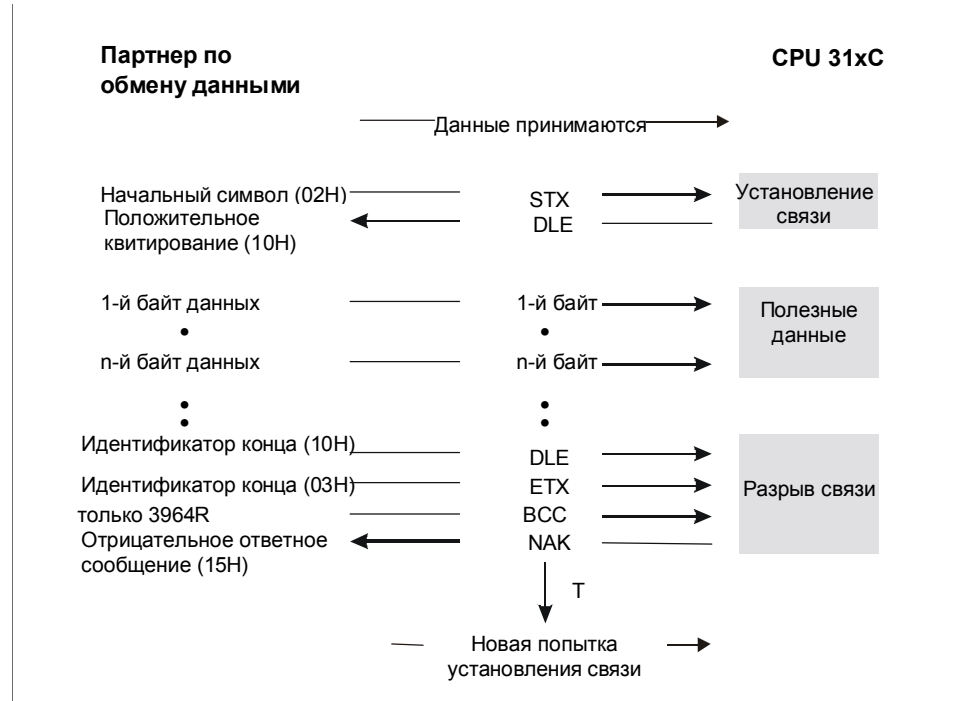
## Разрыв соединения при приеме

Если процедура 3964 распознает последовательность символов DLE ETX, то она завершает прием и передает DLE партнеру по обмену данными для безошибочно принятого блока. В случае ошибки приема партнеру по обмену данными передается NAK. Затем ожидается повторение.

Если процедура 3964(R) распознает последовательность символов DLE ETX BCC, то она завершает прием. Она сравнивает принятый символ контроля блока BCC с внутренне сформированным контролем четности длины. Если символ контроля блока правилен и нет других ошибок приема, то процедура 3964(R) передает DLE и возвращается в исходное состояние. При ошибочном BCC или другой ошибке приема партнеру по обмену данными передается NAK. Затем ожидается повторение.

## Обработка данных, содержащих ошибки

На следующем рисунке представлен процесс обработки данных, содержащих ошибки, процедурой 3964(R):

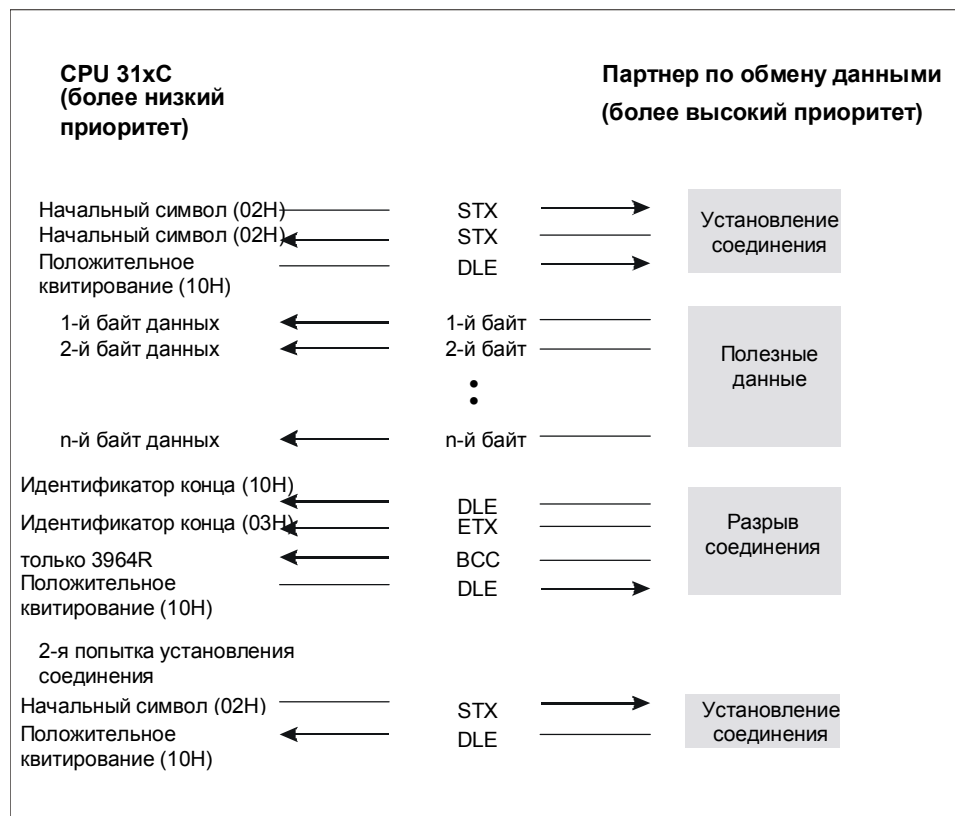


После приема DLE, ETC, BCC CPU сравнивает BCC партнера по обмену данными с собственным внутренне сформированным значением. Если BCC верен и нет других ошибок приема, то CPU отвечает передачей DLE.

В противном случае он отвечает передачей NAK и ждет новой попытки в течение времени ожидания данных. Если по истечении указанного при параметризации числа попыток передачи данные не были приняты или в течение времени ожидания данных новая попытка не предпринималась, CPU прекращает прием.

## Конфликт при инициализации

На следующем рисунке представлен процесс передачи данных при конфликте во время инициализации:



Если устройство отвечает на желание партнера по обмену данными вести передачу (символ STX) в течение времени квитирования (ADT) не символом квитирования DLE или NAK, а символом STX, то имеет место конфликт инициализации. Оба устройства хотели бы выполнить имеющееся у них задание на передачу. Устройство с более низким приоритетом откладывает свое задание на передачу и отвечает символом DLE. Устройство с более высоким приоритетом передает свои данные ранее описанным способом. После разрыва соединения устройство с более низким приоритетом может выполнить свое задание на передачу.

Чтобы разрешить конфликт инициализации, вы должны при параметризации установить для партнеров по обмену данными разные приоритеты.



## Ошибка процедуры

Процедура распознает как ошибки, вызванные ошибочным поведением партнера по обмену данными, так и ошибки, обусловленные помехами на линии.

В обоих случаях сначала делается попытка правильно передать или принять данные при повторении. Если блок данных не удается передать или принять без ошибок за максимально возможное количество попыток передачи (или получается новое ошибочное состояние), то процедура прекращает передачу или прием. Она сообщает номер первой распознанной ошибки и возвращается в исходное состояние. Эти сообщения об ошибках отображаются на выходе STATUS системного функционального блока.

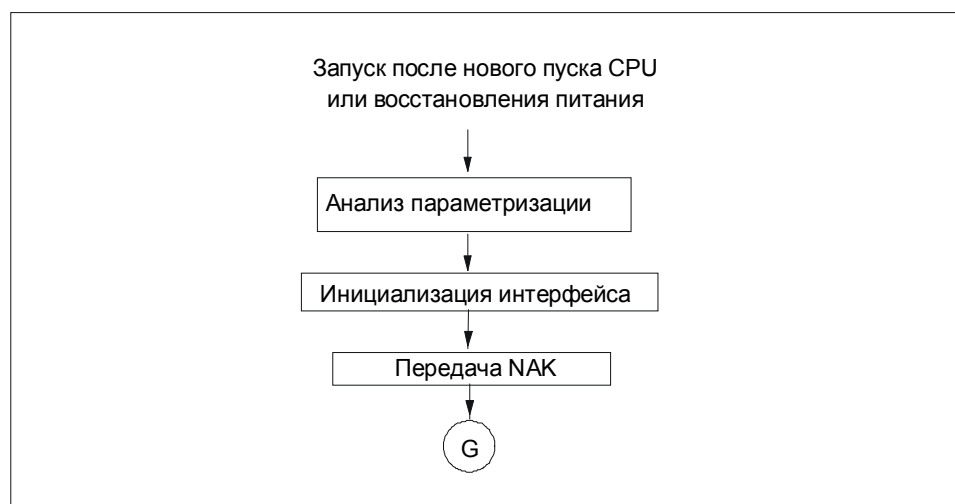
Если один номер ошибки неоднократно появляется на выходе STATUS системного функционального блока для повторных попыток передачи и приема, то можно сделать вывод о случайных помехах при обмене данными. Однако множество попыток передачи их компенсирует. В этом случае мы рекомендуем исследовать путь передачи на воздействие помех, так как при многих попытках скорость передачи полезных данных и надежность передачи снижаются. Однако причина неисправности может находиться также и в ошибочном поведении партнера по обмену данными.

При обрыве (BREAK) в принимающей линии на выходе STATUS системного функционального блока отображается сообщение об ошибке. Повторение в этом случае не производится. Состояние BREAK автоматически сбрасывается, как только восстанавливается связь на линии.

Для всех распознанных ошибок передачи (потеря символа, ошибка кадра или четности) при приеме данных сообщается единый номер. Но об ошибке сообщается только тогда, когда сначала безуспешно выполняются повторения.

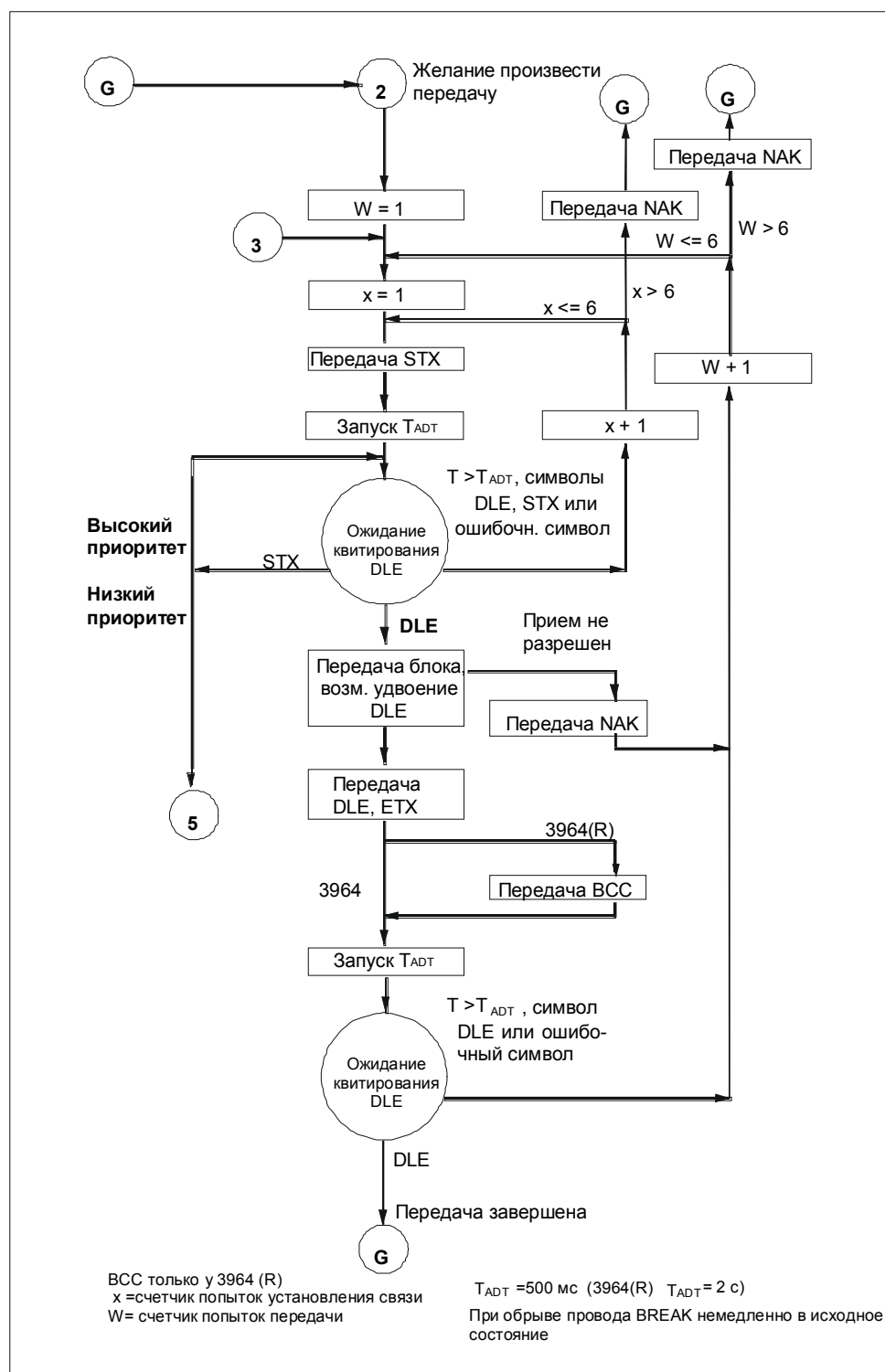
## Запуск процедуры 3964(R)

На следующем рисунке представлены процессы при запуске процедуры 3964(R):



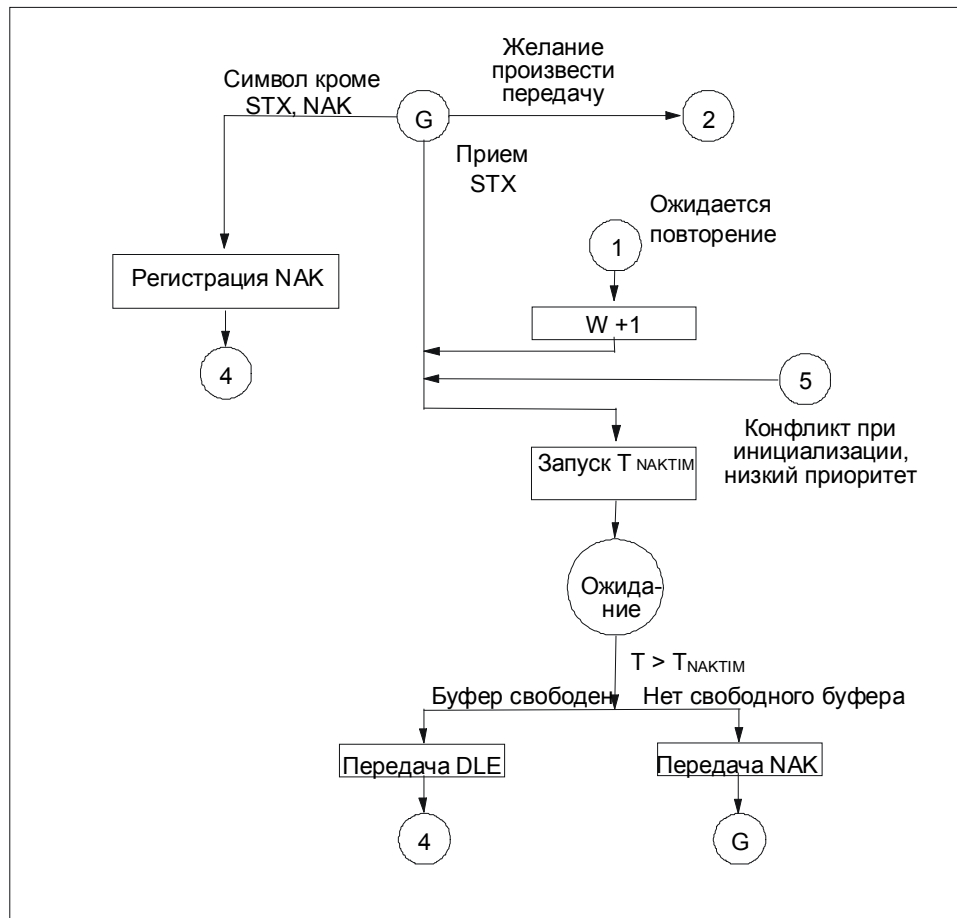
## Передача с помощью процедуры 3964(R)

На следующем рисунке представлены процессы при передаче с помощью процедуры 3964(R):



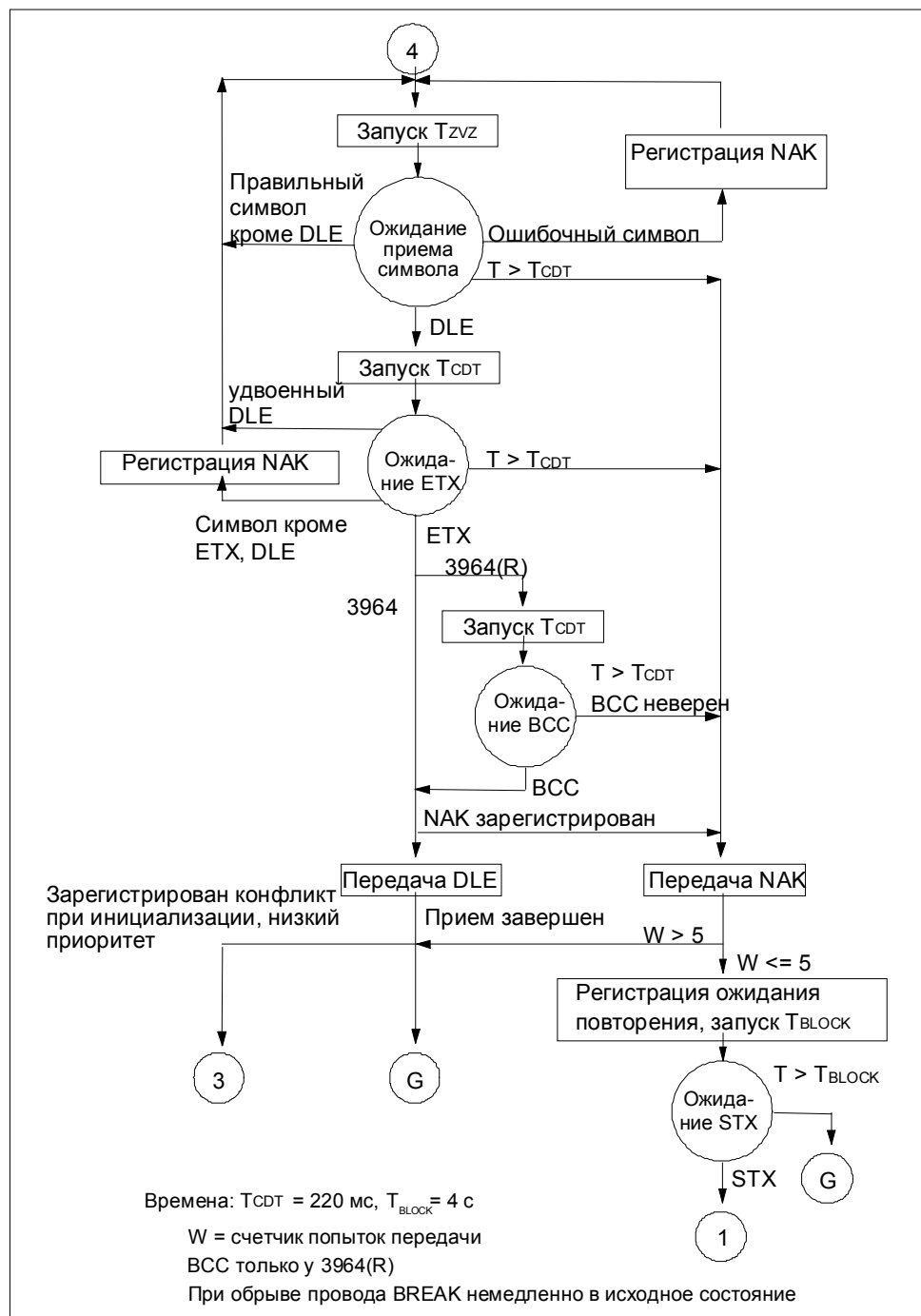
**Процедура 3964(R), прием (часть 1)**

На следующем рисунке представлены процессы при приеме с помощью процедуры 3964(R):



## Процедура 3964(R), прием (часть 2)

На следующем рисунке представлены процессы при приеме данных с помощью процедуры 3964(R):



## Приемный буфер на CPU

Приемный буфер имеет величину 2048 байтов. При параметризации вы можете указать, нужно ли препятствовать перезаписи данных в приемном буфере. Кроме того, вы можете указать диапазон значений (от 1 до 10) для количества буферизуемых принимаемых кадров сообщений или объем всего приемного буфера.

Приемный буфер можно стирать при запуске. Эта настройка осуществляется по выбору через экранную форму параметризации или вызовом SFB RES\_RCV (см. раздел 6.5.1.3).

Приемный буфер является кольцевым буфером:

- Если в приемный буфер вносятся несколько кадров сообщений, то имеет место следующее: В целевой блок данных в качестве первого всегда передается самый старый кадр сообщения.
- Если вы хотите всегда передавать в целевой блок данных только самый новый кадр сообщения, то при параметризации в качестве количества буферизуемых кадров сообщений нужно указать значение "1" и деактивизировать защиту от перезаписи.

---

### Замечание

Если в программе пользователя на некоторое время приостановлено постоянное считывание принимаемых данных, то при новом запросе принимаемых данных может получиться, что в целевой блок данных сначала вносится старый кадр сообщения и только потом самый новый кадр сообщения.

Старыми кадрами сообщений являются те кадры, которые во время перерыва находились в пути между CPU и партнером или уже были приняты SF.

---

## 6.9.3 Передача данных с помощью компьютерного интерфейса RK 512

Компьютерный интерфейс RK 512 управляет передачей данных при двухточечном соединении между CPU и партнером по обмену данными.

В отличие от процедуры 3964(R) компьютерный интерфейс RK 512 обеспечивает более высокую надежность передачи данных и лучшие возможности адресации.

### Ответный кадр сообщения

Компьютерный интерфейс RK 512 отвечает CPU на каждый правильно принятый командный кадр ответным кадром сообщения. Благодаря этому отправитель может проверить, прибыли ли его данные на CPU без ошибок или имеются ли запрошенные им данные CPU.

### Командный кадр

Командные кадры – это кадры сообщений SEND или кадры сообщений FETCH. Как запустить кадр сообщения SEND или FETCH, вы найдете в разделе 6.5.

**Кадр сообщения SEND**

В случае кадра сообщения SEND CPU передает командный кадр сообщения с полезными данными, а партнер по обмену данными передает ответный кадр сообщения без полезных данных.

**Кадр сообщения FETCH**

В случае кадра сообщения FETCH CPU передает командный кадр сообщения без полезных данных, а партнер по обмену данными передает ответный кадр сообщения с полезными данными.

**Дополнительный кадр сообщения**

Если объем данных превышает 128 байт, то в случае кадров сообщений SEND и FETCH автоматически передается дополнительный кадр сообщения.

**Заголовок кадра сообщения**

Каждый кадр сообщения у RK 512 начинается заголовком кадра сообщения. Он может содержать идентификаторы кадра сообщения, указания о цели и источнике данных и номер ошибки.

**Структура заголовка кадра сообщения**

В следующей таблице вы найдете структуру заголовка командного кадра сообщения.

Байт	Описание
1	Идентификатор кадра сообщения у командных кадров сообщений (00h), у дополнительных кадров сообщений (FFh)
2	Идентификатор кадра сообщения (00h)
3	<ul style="list-style-type: none"><li>'A' (41h): Задание SEND с целевым DB</li><li>'O' (4Fh): Задание SEND с целевым DX</li><li>'E' (45h): Задание FETCH</li></ul>
4	Подлежащие передаче данные исчерпаны (при передаче возможно только 'D'): <ul style="list-style-type: none"><li>'D' (44h): Блок данных 'X' (58h) = расширенный блок данных</li><li>'E' (45h): Входные байты 'A' (41h) = выходные байты</li><li>'M' (4Dh): Меркерные байты 'T' (54h) = таймеры</li><li>'Z' (5Ah): Счетчики</li></ul>
5	Цель данных у задания SEND или источник данных у задания FETCH, напр., байт 5 = номер DB, байт 6 = номер DW <sup>1</sup>
6	
7	Старший байт длины: Длина подлежащих передаче данных в зависимости от типа в байтах или словах
8	Младший байт длины: Длина подлежащих передаче данных в зависимости от типа в байтах или словах
9	Номер байта меркера связи; если вы не указали меркера связи, то здесь стоит FFh.

Байт	Описание
10	<ul style="list-style-type: none"> <li>Биты 0 – 3: Номер бита меркера связи, если вы не указали меркера связи, то в протокол здесь вносится Fh.</li> <li>Биты 4 – 7: Номер CPU (число от 1 до 4); если вы не указали номер CPU (число 0), но указали меркер связи, то здесь стоит 0h; если вы не указали ни номера CPU, ни меркера связи, то здесь стоит Fh.</li> </ul>

<sup>1</sup>: Адресация RK 512 описывает источник и цель данных в границах слов. Пересчет в байтовые адреса в SIMATIC S7 осуществляется автоматически.

Буквы в байтах 3 и 4 являются символами ASCII.

Заголовок дополнительного кадра сообщения состоит только из байтов с 1 по 4.

### Ответный кадр сообщения

После того как был передан командный кадр сообщения, RK 512 в течение времени контроля ожидает ответный кадр сообщения партнера по обмену данными. Время контроля составляет 20 с.

### Структура и содержание ответного кадра сообщения

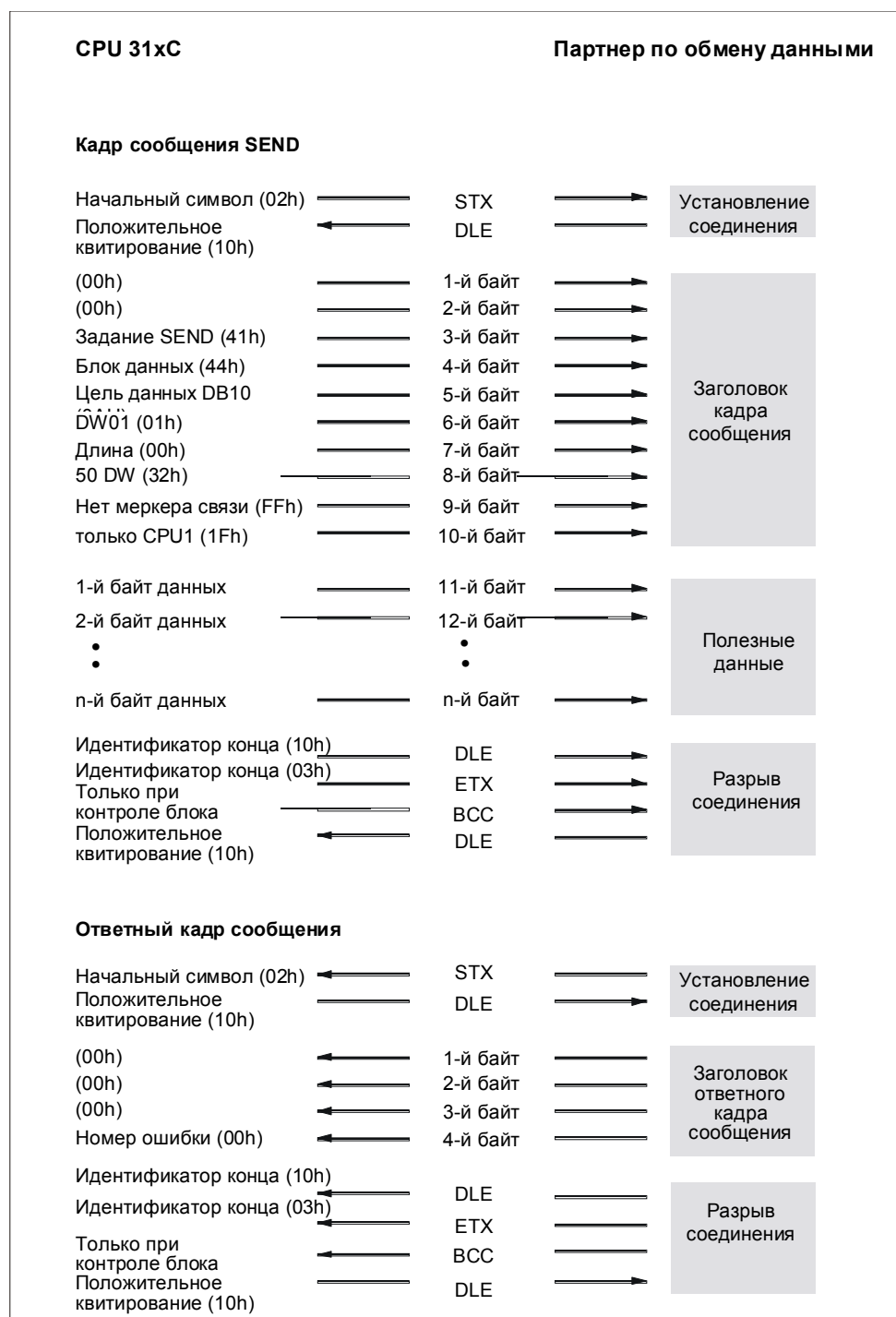
Ответный кадр сообщения состоит из 4 байтов и содержит информацию о процессе выполнения задания.

Байт	Описание
1	Идентификатор кадра сообщения у ответных кадров сообщений (00h), у дополнительных кадров сообщений (FFh)
2	Идентификатор кадра сообщения (00h)
3	Занят значением 00h
4	Номер ошибки партнера по обмену данными (см. раздел 6.10.8) в ответном кадре сообщения: <ul style="list-style-type: none"> <li>00h, если при передаче нет ошибок</li> <li>&gt; 00h – номер ошибки</li> </ul>

\* Номер ошибки в ответном кадре сообщения автоматически воздействует на номер события на выходе STATUS системного функционального блока.

## Передача данных с помощью RK 512

На следующем рисунке представлен процесс передачи данных с ответным кадром сообщения у компьютерного интерфейса RK 512:





## Передача данных

Задание SEND выполняется в следующей последовательности:

- **Активный партнер**  
Передаёт кадр сообщения SEND. Он содержит заголовок кадра сообщения и данные.
- **Пассивный партнер**  
Принимает кадр сообщения, проверяет заголовок кадра сообщения, а также данные и квитирует ответным кадром сообщения после сохранения данных в целевом блоке данных.
- **Активный партнер**  
Принимает ответный кадр сообщения.  
Если количество полезных данных превышает 128 байт, он передаёт дополнительный кадр сообщения SEND.
- **Пассивный партнер**  
Принимает дополнительный кадр сообщения SEND, проверяет заголовок кадра сообщения, а также данные и квитирует дополнительным ответным кадром сообщения после сохранения данных в целевом блоке данных.

---

### Замечание

Если кадр сообщения SEND был принят CPU с ошибками или возникла ошибка в заголовке кадра сообщения, то партнер по обмену данными вносит номер ошибки в 4-й байт ответного кадра сообщения. При ошибках протокола запись в ответный кадр сообщения не производится.

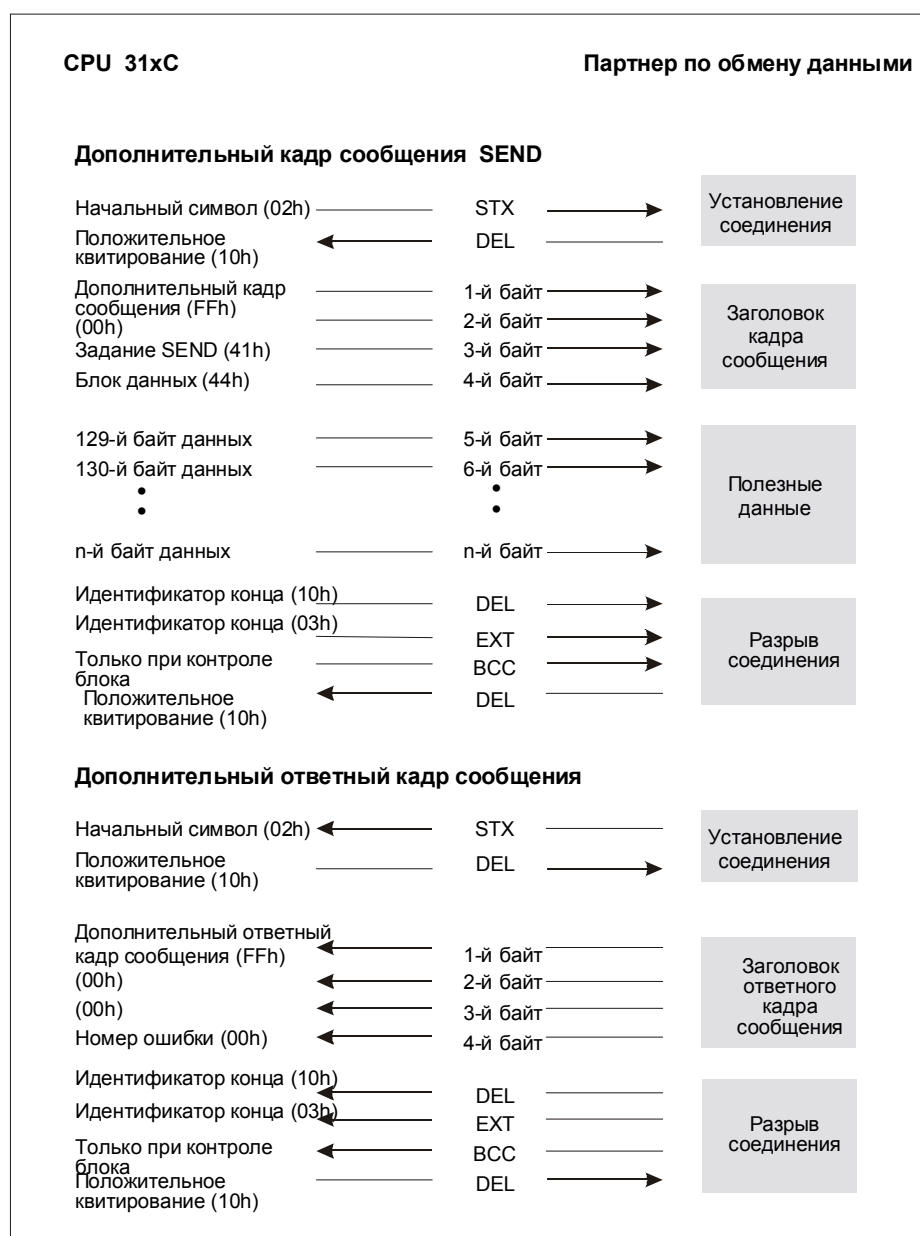
---

## Дополнительный кадр сообщения SEND

Дополнительный кадр сообщения SEND запускается, если количество данных превышает 128 байт. Процесс аналогичен процессу передачи кадра сообщения SEND.

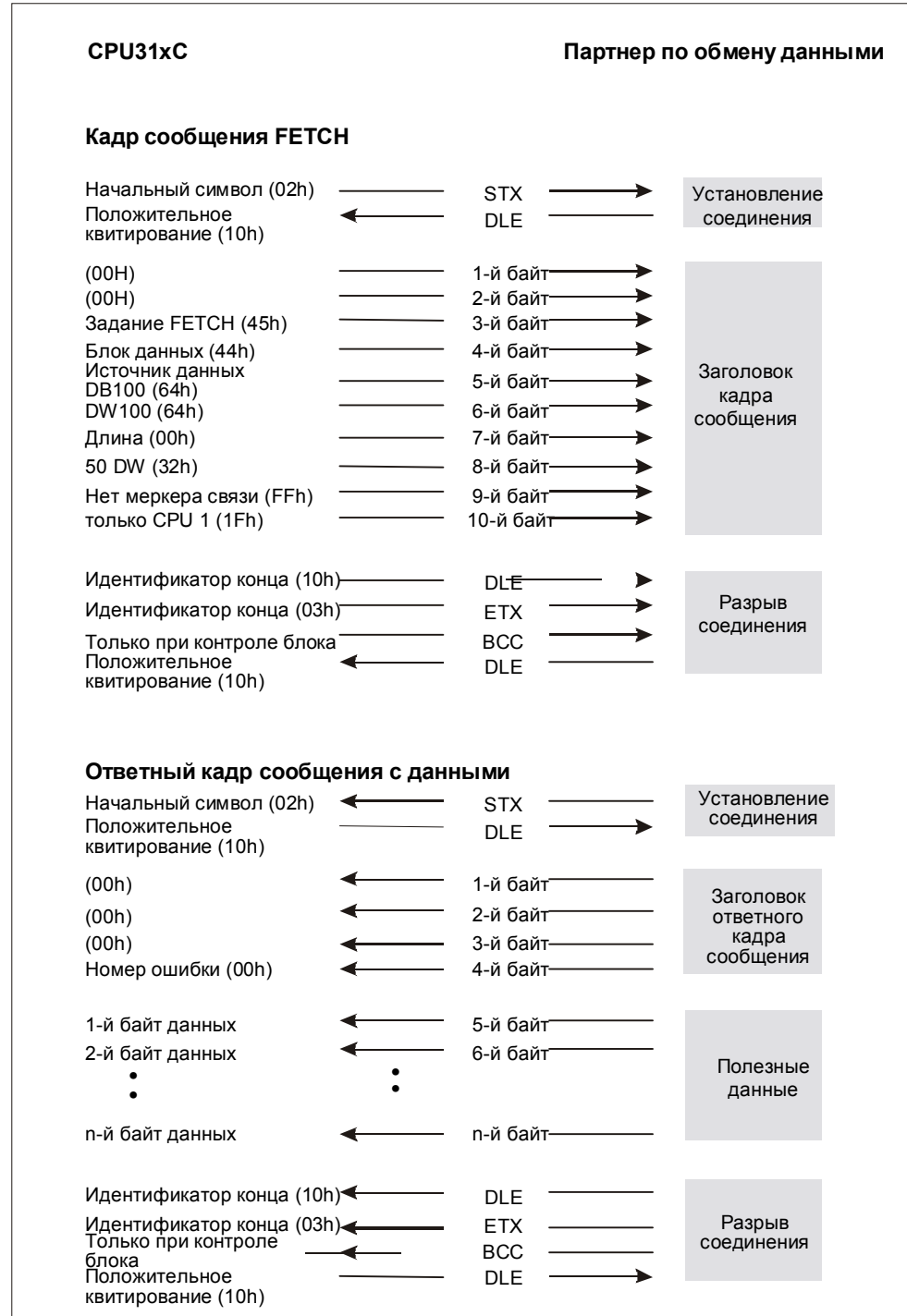
Если посылается более 128 байт, то они передаются автоматически в одном или нескольких дополнительных кадрах сообщений.

На следующем рисунке представлен процесс передачи данных при передаче дополнительного кадра сообщения SEND с дополнительным ответным кадром сообщения:



## Извлечение данных с помощью RK 512

На следующем рисунке представлен процесс извлечения данных с ответным кадром сообщения у компьютерного интерфейса RK 512:



## Извлечение данных

Задание FETCH выполняется в следующей последовательности:

1. **Активный партнер:**  
Передает кадр сообщения FETCH. Она содержит заголовок кадра сообщения.
2. **Пассивный партнер:**  
Принимает кадр сообщения, проверяет заголовок кадра сообщения, извлекает данные из CPU и квитирует ответным кадром сообщения, который содержит данные.
3. **Активный партнер:**  
Принимает ответный кадр сообщения.
4. Если количество полезных данных превышает 128 байт, он передает дополнительный кадр сообщения FETCH. Он содержит с 1 по 4 байты заголовка кадра сообщения.
5. **Пассивный партнер:**  
Принимает дополнительный кадр сообщения FETCH, проверяет заголовок кадра сообщения, извлекает данные из CPU и квитирует дополнительным ответным кадром сообщения с дальнейшими данными.

При номере ошибки (не равном 0) в 4-ом байте ответный кадр сообщения не содержит данных.

Если затребовано больше 128 байтов, то они автоматически извлекаются в одном или нескольких дополнительных кадрах сообщений.

---

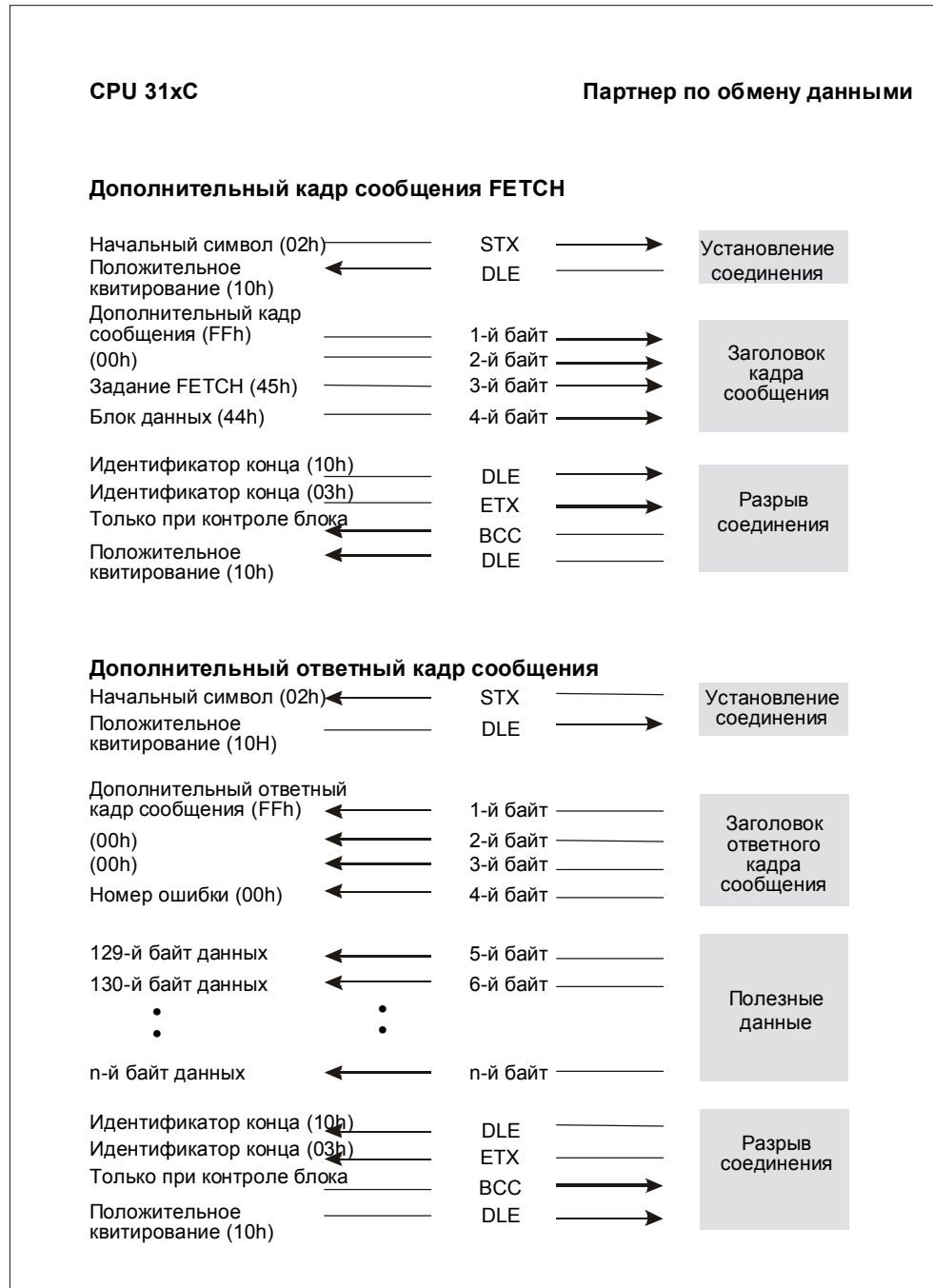
### Замечание

Если кадр сообщения FETCH был принят CPU с ошибками или возникла ошибка в заголовке кадра сообщения, то партнер по обмену данными вносит номер ошибки в 4-й байт ответного кадра сообщения. При ошибках протокола запись в ответный кадр сообщения не производится.

---

## Дополнительный кадр сообщения FETCH

На следующем рисунке представлен процесс передачи данных при извлечении данных с дополнительным ответным кадром сообщения:

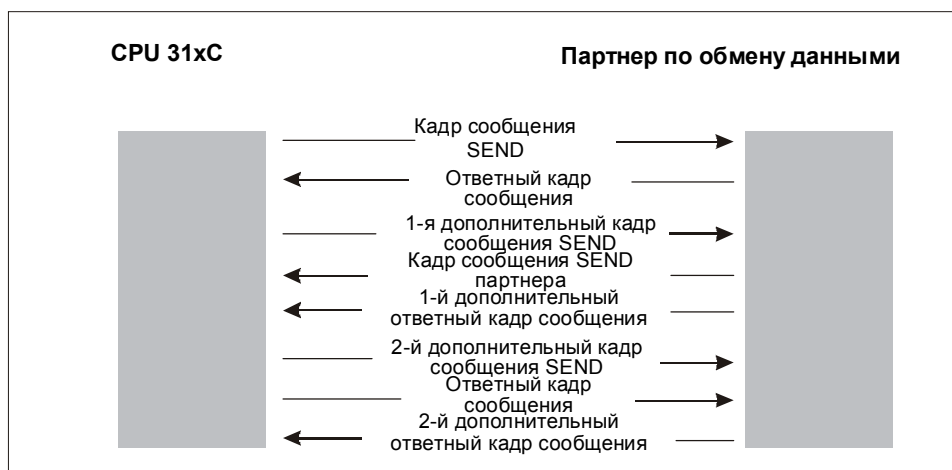


### Квазиполнодуплексный режим

Квазиполнодуплексный режим означает: Партнеры могут в любой момент времени передавать командные и ответные кадры сообщений, за исключением тех случаев, когда другой партнер ведет передачу. Максимальная глубина вложения для командных и ответных кадров сообщений составляет "1". Таким образом, следующий командный кадр сообщения может быть обработан только тогда, когда на предыдущий был отправлен ответный кадр сообщения.

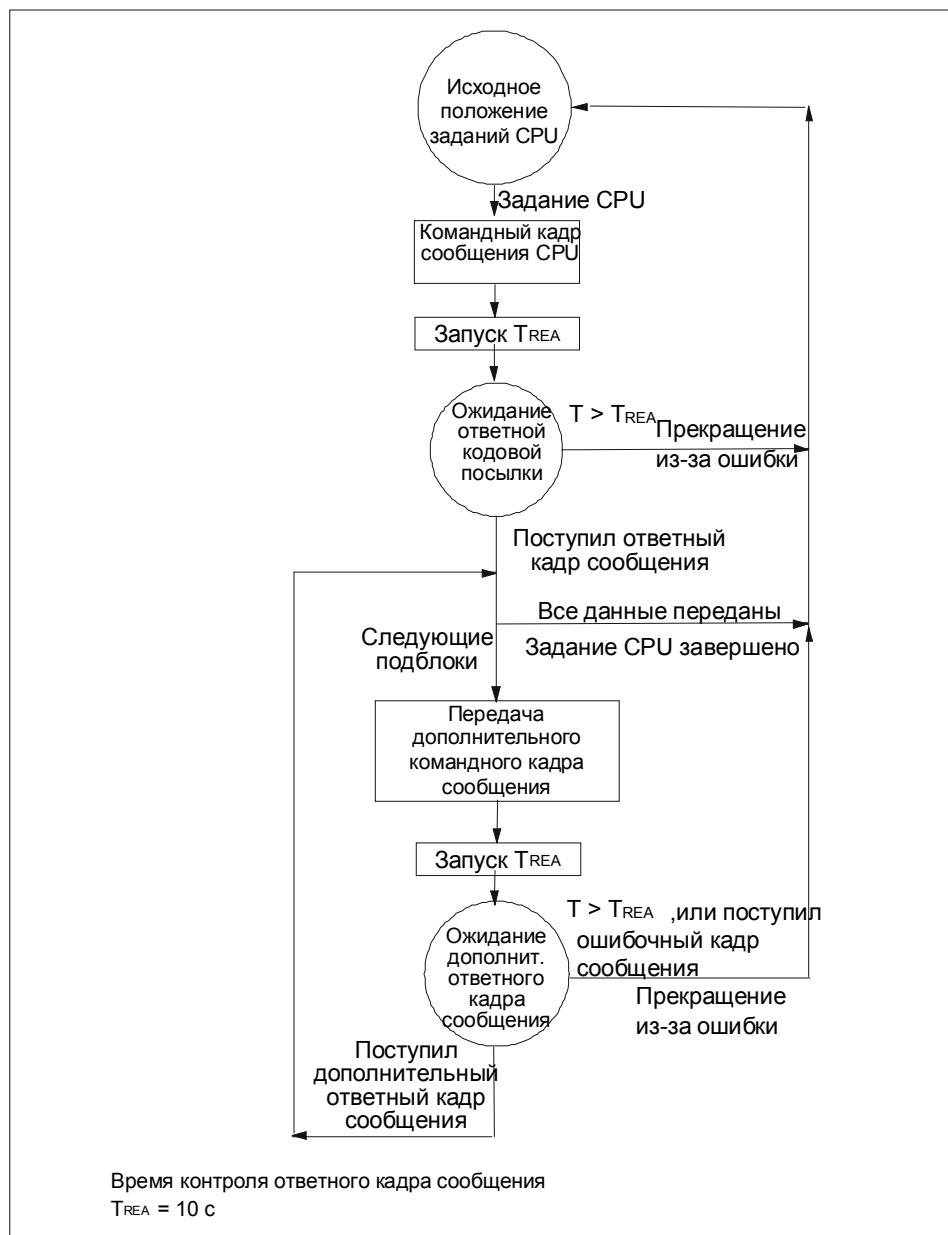
При определенных обстоятельствах, если оба партнера хотят вести передачу, перед ответным кадром сообщения может быть передан кадр сообщения SEND партнера.

На следующем рисунке дополнительный ответный кадр сообщения на первый кадр сообщения SEND передается только после кадра сообщения SEND партнера:



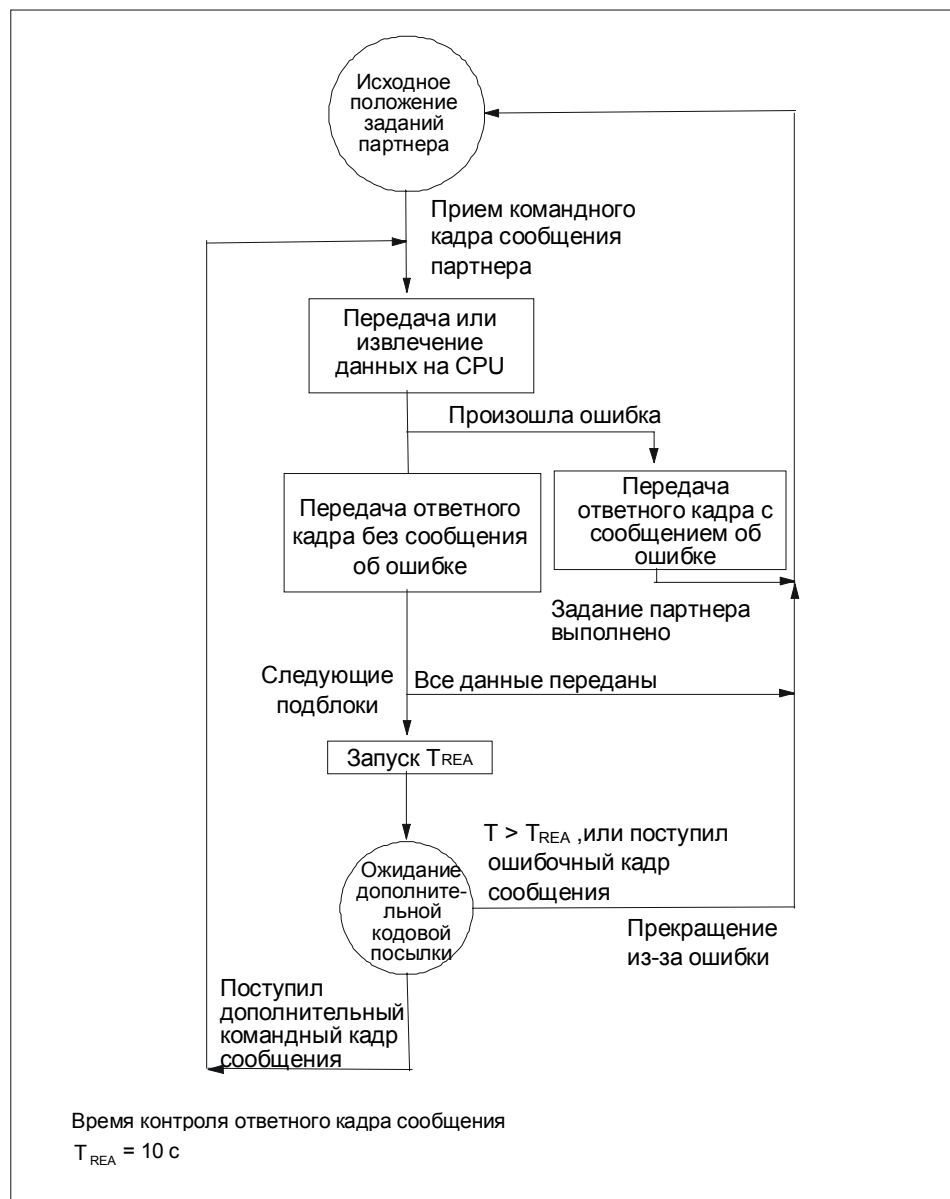
## Задания CPU RK 512

На следующем рисунке представлены процессы для компьютерного интерфейса RK 512 вследствие заданий CPU:



## Задания партнера RK 512

На следующем рисунке представлены процессы для компьютерного интерфейса RK 512 вследствие заданий партнера:





## 6.10 Технические данные

### 6.10.1 Общие технические данные

В следующей таблице вы найдете общие технические данные.

Дальнейшие общие технические данные для SIMATIC S7 300 вы можете найти в справочном руководстве **Системы автоматизации S7 300, Данные модулей**, глава 1 "Общие технические данные" и в руководстве по монтажу *Система автоматизации S7300, Монтаж*.

- Электромагнитная совместимость
- Условия транспортировки и хранения
- Механические и климатические условия окружающей среды
- Указания по испытаниям изоляции, классу защиты и степени защиты
- Сертификаты

Технические данные	
Имеющиеся драйверы протоколов	Драйвер ASCII Процедура 3964(R) RK 512
Скорость передачи с протоколами 3964(R) и RK 512	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Бод
Скорость передачи с драйвером ASCII	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (полудуплексный режим)
Кадр символа	<ul style="list-style-type: none"> <li>• Число битов на символ (7 или 8), у RK 512 только 8 символов.</li> <li>• Число стартовых/стоповых битов (1 или 2)</li> <li>• Контроль четности (отсутствует, четность, нечетность); при 7 битах на символ может быть установлен только контроль на "четность" или "нечетность".</li> </ul>

### Технические данные интерфейса X27 (RS 422/485)

В следующей таблице вы найдете технические данные интерфейса X27 (RS 422/ 485).

Технические данные	
Интерфейс	RS 422 или RS 485, 15-контактная миниатюрная D-образная розетка
Сигналы RS 422	TXD (A), RXD (A), TXD (B), RXD (B), GND R/T (A), R/T (B), GND
Сигналы RS 485	Все потенциально развязаны относительно внутреннего для S7 питания (задняя шина) и внешнего источника питания 24 В пост. тока
Макс. расстояние передачи	1200 м
Макс. скорость передачи	38400 Бод

## 6.10.2 Технические данные драйвера ASCII

В следующей таблице вы найдете технические данные драйвера ASCII.

<b>Драйвер ASCII</b>	
Макс. длина кадра сообщения	1024 байта
Параметры	параметрируется: <ul style="list-style-type: none"> <li>• скорость передачи: 300, 600, 1200, 2400, 4800, 9600, 19200 Бод, 38400 Бод (полудуплексный режим)</li> <li>• кадр символа: 10, 11 или 12 бит</li> <li>• время задержки символа (CDT): от 1 мс до 65535 мс шагами по 1 мс</li> <li>• управление потоком данных: нет, XON/XOFF</li> <li>• символ XON/XOFF (только в случае, когда "Flow control [Управление потоком данных]" = "XON/XOFF")</li> <li>• ожидание XON после XOFF: от 20 мс до 65530 мс шагами по 10 мс</li> <li>• число буферизуемых кадров сообщений: от 1 до 10, использование всего буфера</li> <li>• запрет переписывания: да/нет</li> <li>• распознавание конца принимаемого кадра сообщения:             <ul style="list-style-type: none"> <li>- по истечении времени задержки символа (CDT)</li> <li>- после приема конечного символа (-ов)</li> <li>- после приема фиксированного числа символов</li> </ul> </li> </ul>
<b>Драйвер ASCII с распознаванием конца кадра сообщения по истечении времени задержки символа (CDT)</b>	
Параметры	Другие параметры устанавливать не нужно. Конец кадра сообщения распознается как результат истечения установленного при параметризации времени задержки символа.
<b>Драйвер ASCII с распознаванием конца кадра сообщения с помощью устанавливаемых при параметризации конечных символов</b>	
Параметры	Кроме того, параметрируется: <ul style="list-style-type: none"> <li>• число конечных символов: 1, 2</li> <li>• шестнадцатеричный код первого/второго конечного символа</li> <li>• число символов BCC: 1, 2</li> </ul>
<b>Драйвер ASCII с распознаванием конца кадра сообщения с помощью установленного при проектировании числа символов</b>	
Параметры	Кроме того, параметрируется: <ul style="list-style-type: none"> <li>• число символов: от 1 до 1024 байта</li> </ul>

### 6.10.3 Технические данные процедуры 3964(R)

В следующей таблице вы найдете технические данные процедуры 3964(R):

<b>Процедура 3964(R) со стандартными значениями</b>	
Макс. длина кадра сообщения	1024 байта
Параметры	параметрируется: <ul style="list-style-type: none"> <li>• с символом контроля блока или без него</li> <li>• приоритет: низкий/высокий</li> <li>• скорость передачи: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Бод</li> <li>• кадр символа: 10, 11 или 12 бит</li> <li>• настройка линии приема: нет, R(A)5V/R(B)0V, R(A)0V/R(B)5V</li> <li>• число буферизуемых кадров сообщений: от 1 до 10, использование всего буфера</li> </ul>
<b>Процедура 3964(R) с возможностью параметризации</b>	
Макс. длина кадра сообщения	1024 байта
Параметры	параметрируется: <ul style="list-style-type: none"> <li>• с символом контроля блока или без него</li> <li>• Приоритет: низкий/высокий</li> <li>• скорость передачи: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Бод</li> <li>• кадр символа: 10, 11 или 12 бит</li> <li>• время задержки символа (CDT): от 20 мс до 65530 мс шагами по 10 мс</li> <li>• время задержки квитирования: от 20 мс до 65530 мс шагами по 10 мс</li> <li>• число попыток установления соединения: от 1 до 255</li> <li>• число попыток передачи: от 1 до 255</li> <li>• настройка линии приема: нет, R(A)5V/R(B)0V, R(A)0V/R(B)5V</li> </ul>

### 6.10.4 Технические данные компьютерного интерфейса RK 512

В следующей таблице вы найдете технические данные компьютерного интерфейса RK 512:

Компьютерная процедура RK 512	
Макс. длина кадра сообщения	1024 байта
Параметры	параметрируется: <ul style="list-style-type: none"> <li>• скорость передачи: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Бод</li> <li>• кадр символа: 10, 11 или 12 бит</li> <li>• время задержки символа (CDT): от 20 мс до 65530 мс шагами по 10 мс</li> <li>• время задержки квитирования: от 20 мс до 65530 мс шагами по 10 мс</li> <li>• число попыток установления соединения: от 1 до 255</li> <li>• число попыток передачи: от 1 до 255</li> <li>• настройка линии приема: нет, R(A)5V/R(B)0V, R(A)0V/R(B)5V</li> </ul>

### 6.10.5 Минимальное число циклов CPU

Следующая таблица описывает минимальное число циклов CPU (вызовов SFB), необходимое для выполнения задания:

Блок	Название	Число циклов CPU при обработке ...		
		конца без ошибки	конца с ошибкой	Сброса/запуска (RESET/ANLAUF)
SFB 60	SEND_PTP	≥ 2	≥ 2	≥ 3
SFB 61	RCV_PTP	≥ 2	≥ 2	≥ 3
SFB 62	RES_RCVB	≥ 2	≥ 2	≥ 3
SFB 63	SEND_RK	≥ 2	≥ 2	≥ 3
SFB 64	FETCH_RK	≥ 2	≥ 2	≥ 3
SFB 65	SERVE_RK	≥ 2	≥ 2	≥ 3

### 6.10.6 Времена передачи

Следующие таблицы содержат измеренные времена передачи в зависимости от выбранного протокола передачи.

Для измерения были соединены друг с другом два CPU 314C-2PtP. Измерялось время, прошедшее от появления на линии связи первого символа одного кадра сообщения до появления первого символа непосредственно следующего за ним кадра.

У драйвера ASCII в основе измерения лежит самый быстрый вариант протокола (распознавание конца кадра сообщения с помощью конечного символа без программного управления потоком данных).

У процедуры 3964(R) и у компьютерного интерфейса RK 512 измерения проводились в каждом случае с настройками по умолчанию, т.е. стандартные значения с BCC.

#### Драйвер ASCII (времена передачи в мс)

Скорость передачи (Бод)/ Полезные данные	38400	19200	9600	4800	2400	1200	600	300
1 байт	5	6	7	9	13	23	41	78
10 байтов	7	11	17	28	51	97	190	376
20 байтов	11	17	28	51	97	190	374	744
50 байтов	19	34	62	120	236	465	927	1847
100 байтов	35	64	121	236	466	926	1846	3685
200 байтов	64	120	237	467	927	1845	3686	7363
500 байтов	154	298	586	1160	2309	4607	9204	13398
1000 байтов	305	591	1168	2316	4613	9210	18402	36788

#### Процедура 3964(R) (времена передачи в мс)

Скорость передачи (Бод)/ Полезные данные	38400	19200	9600	4800	2400	1200	600	300
1 байтов	8	11	14	22	38	71	137	267
10 байтов	11	16	25	43	80	154	302	601
20 байтов	14	22	36	66	126	246	487	966
50 байтов	23	38	71	136	264	522	1037	2071
100 байтов	38	68	130	250	494	982	1958	3907
200 байтов	67	126	246	482	956	1902	3798	7586
500 байтов	158	303	595	1175	2838	4664	9316	18620
1000 байтов	308	597	1177	2330	4642	9266	18515	37011

## Компьютерный интерфейс RK 512 (времена передачи в мс)

Скорость передачи (Бод)/ Полезные данные	38400	19200	9600	4800	2400	1200	600	300
1 байт	21	29	44	75	134	253	501	1002
10 байтов	33	42	63	101	180	337	667	1334
20 байтов	37	48	74	124	228	430	851	1701
50 байтов	48	71	112	199	368	709	1402	2804
100 байтов	70	105	178	321	605	1176	2323	4642
200 байтов	126	196	336	618	1173	2293	4543	9064
500 байтов	278	445	778	1450	2784	5450	10836	21608
1000 байтов	545	878	1554	2876	5534	10860	21571	43027

### 6.10.7 Соединительные кабели

Если вы сами готовите соединительные кабели, обратите, пожалуйста, внимание, что вы должны применять только экранированные корпуса. Экран кабеля должен быть с обеих сторон на большой поверхности соединен с корпусом штекера и экранирующим проводом.



#### Осторожно

Никогда не соединяйте экран кабеля с контактом GND, так как это может привести к повреждению интерфейса.

GND (контакт 8) в любом случае должен быть подключен на обеих сторонах, так как иначе возможно повреждение интерфейса.

На следующих страницах вы найдете некоторые примеры соединительных кабелей для двухточечного соединения между CPU и модулями S7 или SIMATIC S5.

### Соединительный кабель X 27/RS422 (CPU 31xC - CPU 31xC/CP 340/CP 341/CP 440/CP 441)

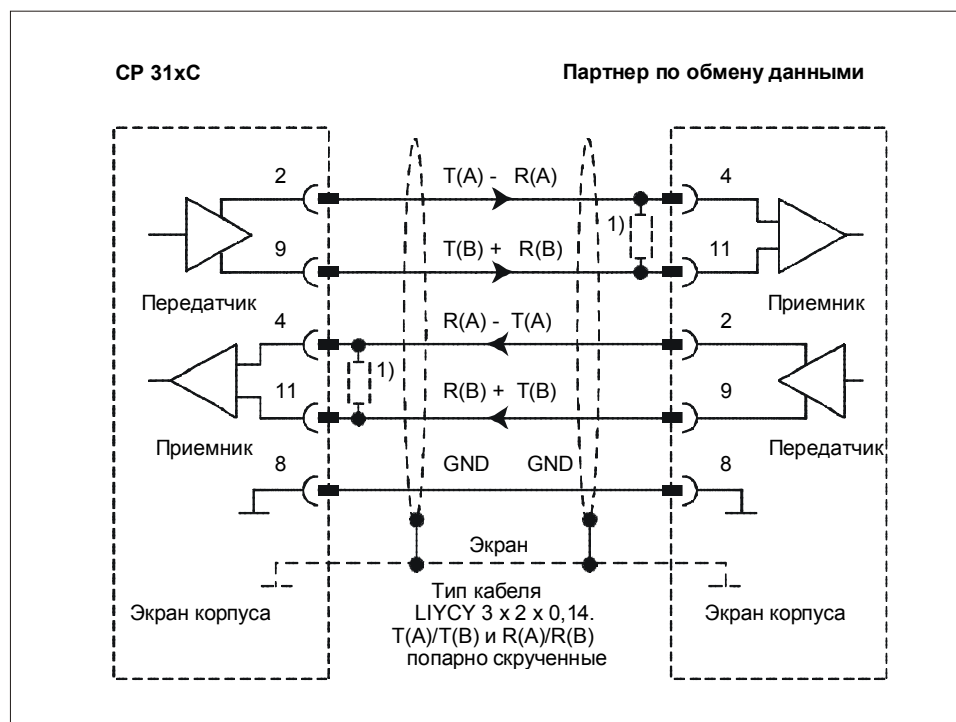
В распоряжении имеются соединительные кабели с предпочтительными длинами: 5 м, 10 м и 50 м.

Исполнение	Номер для заказа
X27 (RS 422), 5 м	6ES7 902-3AB00-0AA0
X27 (RS 422), 10 м	6ES7 902-3AC00-0AA0
X27 (RS 422), 50 м	6ES7 902-3AG00-0AA0

На следующем рисунке представлен соединительный кабель для режима RS 422 между CPU 31xC и CPU 31xC/CP 340/CP341/CP 440/CP 441.

Для соединительных кабелей вам нужны следующие штифтовые штекеры

- на стороне CPU 31xC: 15-контактный миниатюрный D-образный штифтовой штекер с винтовым креплением
- у партнера по обмену данными: 15-контактный миниатюрный D-образный штифтовой штекер с винтовым креплением



- 1) При длинах кабеля > 50 м вы должны для бесперебойного обмена данными впасть на стороне приемника конечное сопротивление около 330 Ом.

#### Замечание

При использованном типе кабеля возможны следующие длины:

- макс. 1200 м при 19200 Бод
- макс. 500 м при 38400 Бод

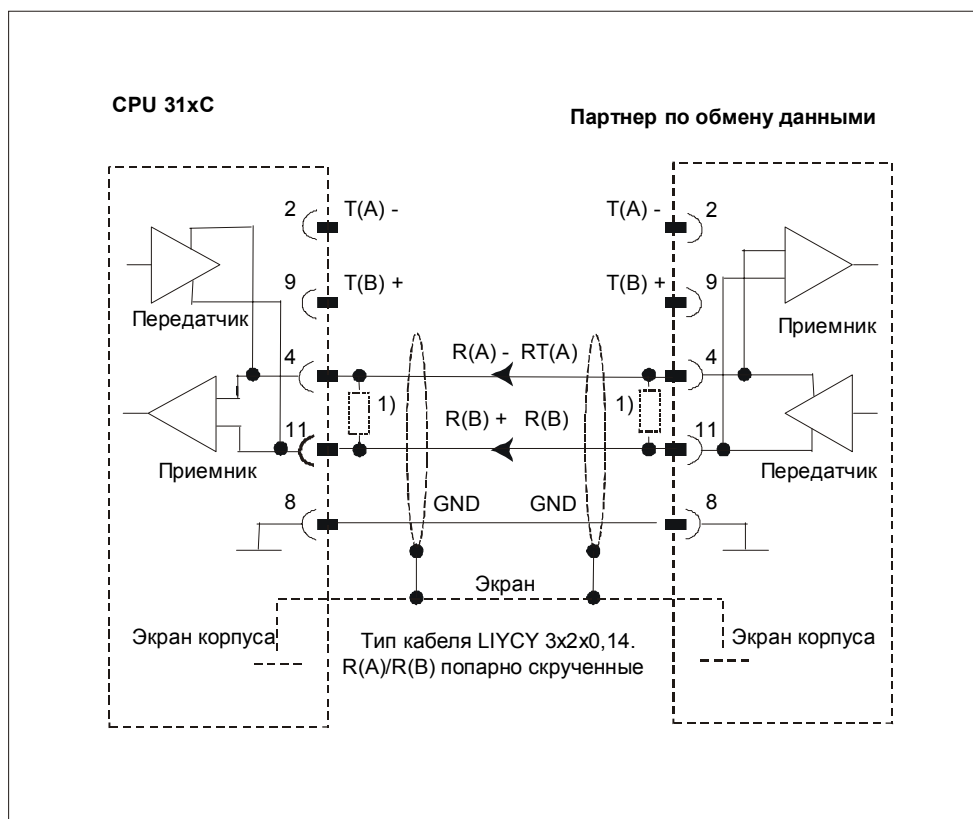
### Соединительный кабель X 27/RS485 (CPU 31xC – CPU 31xC/CP 340/CP 341/CP 440/CP 441)

На следующем рисунке представлен соединительный кабель для режима RS485 между CPU 31xC и CPU 31xC/CP 340/CP 341/CP 440/CP 441.

Фирма Siemens не предлагает готовых кабелей.

Для соединительных кабелей вам нужны следующие штифтовые штекеры

- на стороне CPU 31xC: 15-контактный миниатюрный D-образный штифтовой штекер с винтовым креплением
- у партнера по обмену данными: 15-контактный миниатюрный D-образный штифтовой штекер с винтовым креплением



- 1) При длинах кабеля > 50 м вы должны для бесперебойного обмена данными впаять на стороне приемника оконечное сопротивление около 330 Ом.



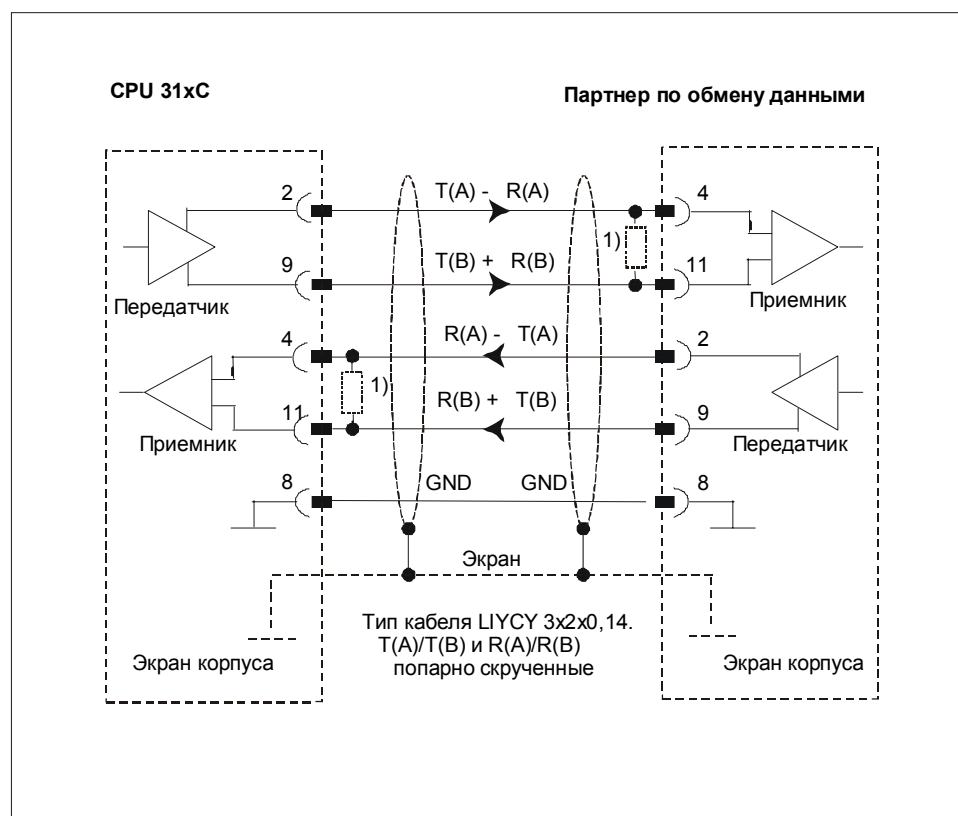
### Соединительный кабель X 27/RS422 (CPU 31xC - CP 544, CP 524, CPU 928B, CPU 945, CPU 948)

На следующем рисунке представлен соединительный кабель для режима RS 422 между CPU 31xC и CP 544, CP 524, CPU 928B, CPU 945, CPU 948.

Фирма Siemens не предлагает готовых кабелей.

Для соединительных кабелей вам нужны следующие штифтовые штекеры

- на стороне CPU 31xC: 15-контактный миниатюрный D-образный штифтовой штекер с винтовым креплением
- у партнера по обмену данными: 15-контактный миниатюрный D-образный штифтовой штекер со **сдвигаемым креплением**



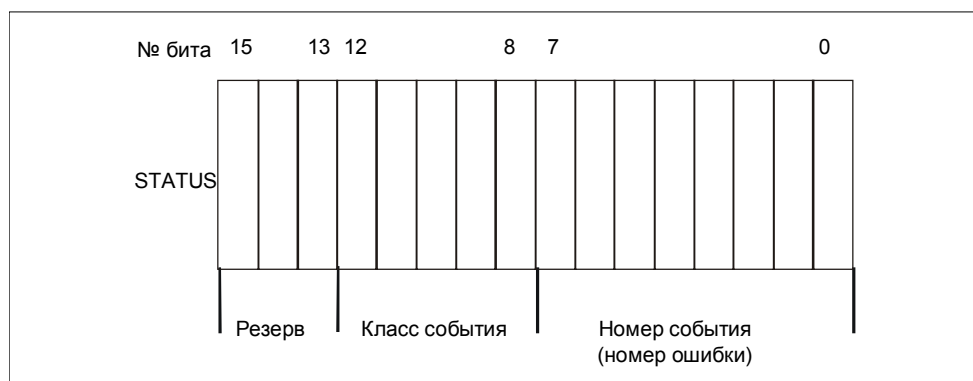
- 1) При длинах кабеля > 50 м вы должны для бесперебойного обмена данными впаять на стороне приемника оконечное сопротивление около 330 Ом.

## 6.10.8 Сообщения об ошибках

Для диагностики ошибок каждый функциональный блок обладает параметром STATUS. Каждый номер сообщения STATUS независимо от используемого системного функционального блока имеет одно и то же значение.

### Схема нумерации – класс события/номер события

На следующем рисунке представлена структура параметра STATUS:



### Пример

На следующем рисунке показано содержимое параметра STATUS для события "Прекращение задания из-за нового пуска или сброса" (Класс события: 05h, номер события 01h):



## Классы событий

В следующих таблицах вы найдете описание различных классов и номеров событий:

Класс события 3 (03h): "Ошибка при параметризации SFB"		
№ события	Событие	Устранение
(03)01h	<ul style="list-style-type: none"> <li>Тип данных источника или цели недопустим или отсутствует.</li> <li>Область (начальный адрес, длина) недопустима.</li> <li>DB отсутствует или недопустим (напр., DB 0) или</li> <li>Другой тип данных отсутствует или недопустим.</li> <li>Номер байта или бита меркера связи недействителен.</li> </ul>	<ul style="list-style-type: none"> <li>Проверить и, возможно, исправить параметризацию.</li> <li>Партнер передает недопустимые параметры в заголовке кадра сообщения.</li> <li>Проверить параметризацию, возможно, установить блок.</li> <li>Возьмите допустимые типы данных из таблиц задания.</li> <li>Партнер передает неверные параметры в заголовке кадра сообщения.</li> </ul>
(03)03h	Невозможен доступ к области.	Проверить параметризацию. Возьмите из таблиц задания допустимые начальные адреса и длины, или партнер передает неверные параметры в заголовке кадра сообщения.

Класс события 5 (05h): "Ошибка при обработке задания"		
№ события	Событие	Устранение
(05)01h	Выполнение текущего задания было прекращено из-за нового пуска или сброса.	Повторите прекращенное задание. При перепараметризации из PG вам следует перед описанием интерфейса обратить внимание на то, чтобы не исполнялось больше ни одно задание.
(05)02h	Задание в этом режиме не разрешено (напр., не параметризован интерфейс устройства).	Параметризируйте интерфейс устройства.
(05)0Eh	<ul style="list-style-type: none"> <li>Недопустимая длина кадра сообщения</li> <li>или</li> <li>Указанные при параметризации конечные символы не появились внутри максимально допустимой длины.</li> </ul>	<ul style="list-style-type: none"> <li>Длина кадра сообщения &gt; 1024 байтов. Выберите меньшую длину кадра сообщения или</li> <li>Введите дополнительно конечные символы в буфере передачи на желаемом месте.</li> </ul>
(05)13h	Ошибка у типа данных (DB ...): <ul style="list-style-type: none"> <li>Неизвестный тип данных или тип данных не разрешен (напр., DE)</li> <li>Указанные на SFB типы данных источника и цели не соответствуют друг другу.</li> </ul>	Допустимые типы данных и их комбинации возьмите из таблиц задания.
(05)15h	У координирующего меркера указан неверный номер бита.	Допустимые номера битов: от 0 до 7

<b>Класс события 5 (05h): "Ошибка при обработке задания"</b>		
<b>№ события</b>	<b>Событие</b>	<b>Устранение</b>
(05)16h	Указан слишком большой номер CPU.	Разрешенные номера CPU: 0, 1, 2, 3 или 4
(05)17h	Длина передачи > 1024 байтов слишком велика	Разбейте задание на несколько заданий меньшей длины.
(05)1Dh	Выполнение задание на прием или передачу было прекращено <ul style="list-style-type: none"> <li>Сброс коммуникационного блока</li> <li>Перепараметризация</li> </ul>	Повторите задание коммуникационного блока.
(05)22h	Было запущено новое задание SEND, хотя старое задание еще не завершено.	Запускайте новое задание SEND только тогда, когда старое задание завершено с параметром DONE или ERROR.

<b>Класс события 6 (06h): "Ошибка при обработке задания партнера" только у RK512</b>		
<b>№ события</b>	<b>Событие</b>	<b>Устранение</b>
(06)01h	Ошибка в 1-м командном байте (не 00 или FFh)	Принципиальная ошибка в структуре заголовка у партнера. Доказать, при необходимости, неверное поведение партнера помощью устройства для тестирования интерфейса, которое включается в передающую линию.
(06)02h	Ошибка в 3-м командном байте (не A, 0 или E)	Принципиальная ошибка в структуре заголовка у партнера. Доказать, при необходимости, неверное поведение партнера помощью устройства для тестирования интерфейса, которое включается в передающую линию.
(06)03h	Ошибка в 3-м командном байте у дополнительных кадров сообщений (команда не такая, как в 1-м кадре сообщения)	Принципиальная ошибка в структуре заголовка у партнера. Доказать, при необходимости, неверное поведение партнера помощью устройства для тестирования интерфейса, которое включается в передающую линию.
(06)04h	Ошибка в 4-м командном байте (неверная буква команды)	Принципиальная ошибка в структуре заголовка у партнера или была затребована недопустимая комбинация команд. Проверьте допустимость команд. Доказать, при необходимости, неверное поведение партнера помощью устройства для тестирования интерфейса, которое включается в передающую линию.
(06)06h	Ошибка в 5-м командном байте (номер DB недопустим)	Возьмите допустимые номера DB, начальные адреса и длины из таблиц задания.
(06)07h	Ошибка в 5-м или 6-м командном байте (начальный адрес слишком велик)	Возьмите допустимые номера DB, начальные адреса и длины из таблиц задания.
(06)09h	Ошибка в 9. и 10-м командном байте (координирующий меркер у этого типа данных недопустим, или номер бита слишком велик).	Принципиальная ошибка в структуре заголовка у партнера. Выясните из таблиц задания, когда разрешен координационный меркер.
(06)0Ah	Ошибка в 10-м командном байте (номер CPU не разрешен)	Принципиальная ошибка в структуре заголовка у партнера.

Класс события 7 (07h): "Ошибка передачи"		
№ события	Событие	Устранение
(07)01h	Только у 3964(R): Передача первого повторения: <ul style="list-style-type: none"> <li>При передаче кадра сообщения была распознана ошибка или</li> <li>Партнер передачей символа отрицательного квитирования (NAK) затребовал повторения.</li> </ul>	Повторение не является ошибкой, однако, оно может быть указанием на то, что в передающей линии возникают помехи или имеет место неправильное поведение устройства партнера. Если после максимального числа повторений кадр сообщения не удалось все же передать, то сообщается номер ошибки, которая встретилась первой.
(07)02h	Только у 3964(R): Ошибка при установлении соединения: После того как был передан STX, был принят NAK или любой символ (кроме LE или STX).	Проверьте, нет ли ошибок в поведении устройства партнера, при необходимости, с помощью устройства для тестирования интерфейса, включаемого в передающую линию.
(07)03h	Только у 3964(R): <ul style="list-style-type: none"> <li>время задержки квитирования (ADT) превышено</li> <li>после передачи STX от партнера не было получено ответа в течение времени задержки квитирования.</li> </ul>	Устройство партнера работает слишком медленно или не готово к приему, или имеет место, например, обрыв в передающей линии. При необходимости, докажите ошибочное поведение устройства партнера с помощью устройства для тестирования интерфейса, которое включается в линию передачи.
(07)04h	Только у 3964(R): Прекращение передачи партнером: Во время текущего режима передачи партнером был принят один или несколько символов.	Проверьте, произошло ли отображение ошибки и у партнера, так как, возможно, не все переданные данные прибыли (напр., обрыв в передающей линии), или имеют место серьезные помехи, или имеет место неправильное поведение устройства партнера. Если необходимо, докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(07)05h	Только у 3964(R): Отрицательное квитирование при передаче	Проверьте, произошло ли отображение ошибки и у партнера, так как, возможно, не все переданные данные прибыли (напр., обрыв в передающей линии) или имеют место серьезные помехи или имеет место неправильное поведение устройства партнера. Если необходимо, докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(07)06h	Только у 3964(R): Ошибка при завершении соединения: <ul style="list-style-type: none"> <li>Кадр сообщения был отклонен партнером в конце символом NAK или любым символом (кроме DLE) или</li> <li>Символ квитирования (DLE) был принят слишком рано.</li> </ul>	Проверьте, произошло ли отображение ошибки и у партнера, так как, возможно, не все переданные данные прибыли (напр., обрыв в передающей линии) или имеют место серьезные помехи, или имеет место неправильное поведение устройства партнера. Если необходимо, докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.

Класс события 7 (07h): "Ошибка передачи"		
№ события	Событие	Устранение
(07)07h	Только у 3964(R): Превышено время задержки квитирования в конце соединения или время контроля ответа после переданного кадра сообщения: После установления соединения с помощью DLE ETX в течение времени задержки квитирования не пришло ответа от партнера.	Устройство партнера работает слишком медленно или повреждено. Если необходимо, докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(07)08h	Только у драйвера ASCII: Время ожидания XON истекло.	Партнер по обмену данными поврежден, работает слишком медленно или включен в режиме offline. Проверьте партнера по обмену данными или, если необходимо, измените параметризацию.
(07)09h	Только у 3964(R): Установление соединения невозможно, было превышено разрешенное число попыток установления соединения.	Проверьте кабель интерфейса или параметры передачи. Проверьте также у партнера, правильно ли установлены параметры функции приема между CPU и CP.
(07)0Ah	Только у 3964(R): Данные не удалось передать, было превышено разрешенное число попыток передачи.	Проверьте кабель интерфейса или параметры передачи.
(07)0Bh	Только у 3964(R): конфликт инициализации неразрешим, так как у обоих партнеров установлен высокий приоритет.	Измените параметризацию.
(07)0Ch	Только у 3964(R): конфликт инициализации неразрешим, так как у обоих партнеров установлен низкий приоритет.	Измените параметризацию.

Класс события 8 (08h): "Ошибка приема"		
№ события	Событие	Устранение
(08)01h	Только у 3964(R): Ожидание первого повторения: При приеме кадра сообщения была распознана ошибка, и CPU посредством отрицательного квитирования (NAK) затребовал у партнера повторения.	Повторение не является ошибкой, однако, оно может быть указанием на то, что в передающей линии возникают помехи или имеет место неправильное поведение устройства партнера. Если после максимального числа повторений кадр сообщения не удалось все же передать, то сообщается номер ошибки, которая встретилась первой.

Класс события 8 (08h): "Ошибка приема"		
№ события	Событие	Устранение
(08)02h	Только у 3964(R): Ошибка при установлении соединения: <ul style="list-style-type: none"> <li>В исходном состоянии был принят один или несколько любых символов (кроме NAK или STX) или</li> <li>После принятого STX партнером были переданы дальнейшие символы, не ожидая ответа DLE.</li> </ul> После включения питания у партнера: <ul style="list-style-type: none"> <li>При включении партнера, CPU принимает неопределенный символ.</li> </ul>	При необходимости докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включаемого в линию передачи.
(08)05h	Только у 3964(R): Логическая ошибка при приеме: После приема DLE был принят еще один любой символ (кроме DLE, ETX).	Проверьте, не удваивает ли партнер всегда символ DLE в заголовке кадра сообщения и в строке данных, или установление соединения предпринято с помощью DLE ETX. При необходимости докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включаемого в линию передачи.
(08)06h	Превышено время задержки символа (CDT): <ul style="list-style-type: none"> <li>Два следующих друг за другом символа не были приняты в течение времени задержки символа (CDT) или</li> </ul> Только у 3964(R): <ul style="list-style-type: none"> <li>Первый символ после передачи DLE при установлении соединения не был принят в течение времени задержки символа.</li> </ul>	Устройство партнера работает слишком медленно или повреждено. Если необходимо, докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(08)07h	Недопустимая длина кадра сообщения: Был принят кадр сообщения длиной 0.	Прием кадра сообщения длиной 0 не является ошибкой. Проверьте, почему партнер по обмену данными ведет передачу без полезных данных.
(08)08h	Только у 3964(R): Ошибка в символе контроля блока BCC: Внутренне сформированное значение BCC не совпадает с BCC, принятым партнером в конце соединения.	Проверьте, нет ли сильных помех в линии связи, в этом случае следует также время от времени наблюдать за кодами ошибок. При необходимости докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включаемого в линию передачи.
(08)09h	Только у 3964(R): Истекло время ожидания повторения блока	Установите в параметрах у партнера по обмену данными такое же время ожидания блока, как у вашего модуля. При необходимости докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включаемого в линию передачи.

Класс события 8 (08h): "Ошибка приема"		
№ события	Событие	Устранение
(08)0Ah	Отсутствует свободный приемный буфер: При приеме не было свободного приемного буфера.	Должен чаще вызываться SFB RCV.
(08)0Ch	Ошибка передачи: <ul style="list-style-type: none"> <li>Была распознана ошибка передачи (ошибка четности, ошибка стопового бита, ошибка переполнения).</li> </ul> Только у 3964(R): <ul style="list-style-type: none"> <li>Если в исходном состоянии принимается искаженный символ, немедленно передается сообщение об ошибке, чтобы своевременно можно было распознать влияние помех на линию передачи.</li> </ul> Только у 3964(R): <ul style="list-style-type: none"> <li>Если это происходит в режиме передачи или приема, то запускаются повторения.</li> </ul>	Помехи в линии передачи вызывают повторения кадров сообщений и снижают тем самым пропускную способность для полезных данных. Возрастает опасность нераспознанной ошибки. Измените структуру своей системы или прокладку кабелей.  Проверьте соединительный кабель партнера по обмену данными или проверьте, одинаково ли установлены у обоих устройств скорость передачи, контроль четности и число стоповых битов.
(08)0Dh	BREAK: Обрыв принимающего кабеля у партнера.	Снова установите связь или включите партнера.
(08)0Eh	Переполнение приемного буфера при не разблокированном управлении потоком данных.	В программе пользователя должен чаще вызываться SFB для приема или при параметризации должна быть установлена связь с управлением потоком данных.
(08)10h	Ошибка четности	Проверьте соединительный кабель партнера по обмену данными или проверьте, одинаково ли установлены у обоих устройств скорость передачи, контроль четности и число стоповых битов.
(08)11h	Ошибка рамки символа	Проверьте соединительный кабель партнера по обмену данными или проверьте, одинаково ли установлены у обоих устройств скорость передачи, контроль четности и число стоповых битов.  Измените структуру своей системы или прокладку кабелей.
(08)12h	Только у драйвера ASCII: После того как CPU передал XOFF, были еще приняты символы.	Снова выполните параметризацию партнера или быстрее удаляйте ненужные данные.
(08)14h	Только у драйвера ASCII: Был потерян один или несколько кадров сообщений, так как работа велась без управления потоком данных.	Насколько это возможно, работайте с управлением потоком данных. Используйте весь приемный буфер. При настройке основных параметров установите параметр "Reaction to CPU STOP [Реакция на переход CPU в STOP]" на "Continue [Продолжить]".



Класс события 8 (08h): "Ошибка приема"		
№ события	Событие	Устранение
(08)16h	Длина принятого кадра сообщения была больше, чем согласованная длина.	Требуется коррекция у партнера.

Класс события 9 (09h): "Принят ответный кадр сообщения с ошибкой или ошибочный кадр сообщения от партнера по соединению"		
№ события	Событие	Устранение
(09)02h	Только у RK 512: Ошибка доступа к памяти у партнера (отсутствует память) у SIMATIC S5 как партнера: <ul style="list-style-type: none"> <li>• Неверная область у индикаторного слова или</li> <li>• Отсутствует область данных (кроме DB/DX) или</li> <li>• Область данных слишком коротка (кроме DB/DX)</li> </ul>	Проверьте, есть ли у партнера желаемая область данных и достаточно ли она велика или проверьте параметры вызванного системного функционального блока.  Проверьте длину, указанную в системном функциональном блоке.
(09)03h	Только у RK 512: Ошибка доступа к DB/DX у партнера (DB/DX отсутствует или слишком мал) у SIMATIC S5 как партнера: <ul style="list-style-type: none"> <li>• DB/DX отсутствует или</li> <li>• DB/DX слишком мал или</li> <li>• номер DB/DX недопустим.</li> </ul> У задания FETCH превышена допустимая область источника.	<ul style="list-style-type: none"> <li>• Проверьте наличие и достаточность желаемой области данных у партнера.</li> <li>• Проверьте параметры вызванного системного функционального блока.</li> <li>• Проверьте длину, указанную в системном функциональном блоке.</li> </ul>
(09)04h	Только у RK 512: Партнер сообщает "Вид задания не разрешен".	Ошибочное поведение партнера, так как CPU не выдавал системных команд.
(09)05h	Только у RK 512: Ошибка у партнера или у SIMATIC S5 как партнера: <ul style="list-style-type: none"> <li>• Недопустимый тип источника или цели</li> <li>• Ошибка памяти в устройстве автоматизации (AG) партнера</li> <li>• Ошибка связи CP/CPU у партнера</li> <li>• AG партнера находится в состоянии STOP</li> </ul>	<ul style="list-style-type: none"> <li>• Проверьте, может ли партнер передавать желаемый тип данных.</li> <li>• Проверьте структуру аппаратного обеспечения у партнера.</li> <li>• Переключите AG партнера в состояние RUN.</li> </ul>

Класс события 9 (09h): "Принят ответный кадр сообщения с ошибкой или ошибочный кадр сообщения от партнера по соединению"		
№ события	Событие	Устранение
(09)08h	Только у RK 512: Партнер распознает ошибку синхронизации: Нарушена последовательность кадров сообщений.	Эта ошибка возникает при новом пуске собственного AG или AG партнера. При этом речь идет о нормальном поведении установки при запуске. Вам не нужно ничего устранять. Во время работы эта ошибка возможна также вследствие ранее произошедших ошибок. В ином случае вы можете исходить из неправильного поведения партнера.
(09)09h	Только у RK 512: DB/DX у партнера заблокирован координирующим меркером.	<ul style="list-style-type: none"> <li>В программе партнера: После обработки последних переданных данных снова сбрасывайте координирующий меркер!</li> <li>В программе: Повторите задание!</li> </ul>
(09)0Ah	Только у RK 512: Ошибка в заголовке кадра сообщения, которая была распознана партнером: 3-й командный байт в заголовке неверен	Проверьте, вызвана ли ошибка помехами или неправильным поведением партнера. Докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(09)0Ch	Только у RK 512: Партнер распознает неверную (общую) длину кадра сообщения.	Проверьте, вызвана ли ошибка помехами или неправильным поведением партнера. Докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(09)0Dh	Только у RK 512: До сих пор у партнера не было нового пуска.	Произвести новый пуск у AG партнера или перевести переключатель выбора режима работы в положение RUN.
(09)0Eh	Только у RK 512: Принят неизвестный номер ошибки в ответном кадре сообщения.	Проверьте, вызвана ли ошибка помехами или неправильным поведением партнера. Докажите это с помощью устройства для тестирования интерфейса, включенного в линию передачи.

Класс события 10 (0Ah): "Ошибки в ответном кадре сообщения партнера, которые были распознаны CPU "		
№ события	Событие	Устранение
(0A)02h	Только у RK 512: Ошибка в структуре принятого ответного кадра сообщения (1-й байт не 00 или FF)	При необходимости, докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включенного в линию передачи.
(0A)03h	Только у RK 512: Принятый кадр сообщения содержит слишком много или слишком мало данных.	При необходимости, докажите неправильное поведение устройства партнера с помощью устройства для тестирования интерфейса, включенного в линию передачи.

Класс события 10 (0Ah): "Ошибки в ответном кадре сообщения партнера, которые были распознаны CPU "		
№ события	Событие	Устранение
(0A)05h	Только у RK 512: В течение времени контроля от партнера не пришло ответного кадра сообщения.	не является ли партнер слишком медленным устройством? Эта ошибка часто отображается вследствие ранее произошедших ошибок. Например, могут отображаться ошибки приема (класс события 8), после чего был послан кадр сообщения FETCH. Причина: Ответный кадр сообщения не мог быть принят из-за помех, прошло время контроля. Эта ошибка может возникать также, когда у партнера был произведен новый пуск, прежде чем он смог ответить на последний полученный кадр сообщения FETCH.

Класс события 11 (0Bh): "Предупреждения"		
№ события	Событие	Устранение
(0B)01h	Приемный буфер заполнен больше, чем на 2/3	Чаще вызывайте блок приема, чтобы избежать переполнения приемного буфера.

## 6.10.9 Параметры SFB

### Параметры SFB 60 "SEND\_PTP"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>REQ</b>	IN	BOOL	Инициализация задания при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Задание прерывается. Передача блокируется.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>DONE</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0
<b>SD_1</b>	IN_OUT	ANY	Параметр передачи: Здесь вы указываете: <ul style="list-style-type: none"> <li>Номер DB, из которого передаются данные.</li> <li>Номер байта данных, начиная с которого должны передаваться данные.</li> </ul> Напр.: DB10 от байта 2 -> DB10.DBB2	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Здесь вы указываете длину подлежащих передаче данных в байтах	от 1 до 1024	1

### Параметры SFB 61 "RCV\_PTP"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>EN_R</b>	IN	BOOL	Деблокировка приема	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Выполнение задания прекращено	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>NDR</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>RD_1</b>	IN_OUT	ANY	Параметр приема: Здесь вы указываете: <ul style="list-style-type: none"> <li>Номер DB, в котором сохраняются принимаемые данные.</li> <li>Номер байта данных, начиная с которого сохраняются принимаемые данные.</li> </ul> Напр.: DB20, начиная с байта 5 -> DB20.DBB5	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Вывод длины данных (число байтов)	от 0 до 1024	0

### Параметры SFB 62 "RES\_RCVB"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>REQ</b>	IN	BOOL	Инициализация задания при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Выполнение задания прекращено	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>DONE</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0

## Параметры SFB 63 "SEND\_RK"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации SFB RK (Минимальная длина 240 байт).	Зависит от CPU, ноль не разрешен.	0
<b>REQ</b>	IN	BOOL	Инициализация задания при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Задание прерывается. Передача блокируется.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>R_CPU</b>	IN	INT	Номер CPU партнера (только в многопроцессорном режиме)	от 0 до 4	1
<b>R_TYPE</b>	IN	CHAR	Тип адреса на CPU партнера <ul style="list-style-type: none"> <li>'D' = блок данных</li> <li>'X' = расширенный блок данных</li> </ul>	'D', 'X'	'D'
<b>R_DBNO</b>	IN	INT	Номер блока данных на CPU партнера	от 0 до 255	0
<b>R_OFFSET</b>	IN	INT	Номер байта данных на CPU партнера	от 0 до 510 (только четные значения)	0
<b>R_CF_BYT</b>	IN	INT	Байт меркеров связи на CPU партнера (255: означает: без меркера связи)	от 0 до 255	255
<b>R_CF_BIT</b>	IN	INT	Бит меркера связи на CPU партнера	от 0 до 7	0
<b>DONE</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0
<b>SD_1</b>	IN_OUT	ANY	Параметр передачи: Здесь вы указываете: <ul style="list-style-type: none"> <li>Номер DB, из которого передаются данные.</li> <li>Номер байта данных, начиная с которого должны передаваться данные.</li> </ul> Напр.: DB10 от байта 2 -> DB10.DBB2	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Здесь вы указываете длину подлежащих передаче данных в байтах	от 1 до 1024	1

## Параметры SFB 64 "FETCH\_RK"

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации SFB RK (Минимальная длина 240 байт).	Зависит от CPU, ноль не разрешен.	0
<b>REQ</b>	IN	BOOL	Инициализация задания при положительном фронте	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Выполнение задания прекращено.	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>R_CPU</b>	IN	INT	Номер CPU партнера (только в многопроцессорном режиме)	от 0 до 4	1
<b>R_TYPE</b>	IN	CHAR	Тип адреса на CPU партнера <ul style="list-style-type: none"> <li>'D' = блок данных</li> <li>'X' = Расширенный блок данных</li> <li>'M' = меркеры</li> <li>'E' = входы</li> <li>'A' = выходы</li> <li>'Z' = счетчики</li> <li>'T' = таймеры</li> </ul>	'D', 'X', 'M', 'E', 'A', 'Z', 'T'	'D'
<b>R_DBNO</b>	IN	INT	Номер блока данных на CPU партнера	от 0 до 255	0
<b>R_OFFSET</b>	IN	INT	Номер байта данных на CPU партнера	См. таблицу: „Параметры на FB для источника данных (CPU партнера)“	0
<b>R_CF_BYT</b>	IN	INT	Байт меркеров связи на CPU партнера (255: означает: без меркера связи)	от 0 до 255	255
<b>R_CF_BIT</b>	IN	INT	Бит меркера связи на CPU партнера	от 0 до 7	0
<b>DONE</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0
<b>RD_1</b>	IN_OUT	ANY	Параметр приема: Здесь вы указываете: <ul style="list-style-type: none"> <li>номер DB, в котором сохраняются извлеченные данные.</li> <li>номер байта данных, с которого сохраняются извлеченные данные.</li> </ul> Напр.: DB10 от байта 2 -> DB10.DBB2	Зависит от CPU	0
<b>LEN</b>	IN_OUT	INT	Здесь указывается длина подлежащего извлечению кадра сообщения в байтах. Для каждого таймера и счетчика в качестве длины нужно указывать два байта.	от 1 до 1024	1

### Параметры SFB 65 "SERVE\_RK" для приема и подготовки данных

Параметр	Объявление	Тип данных	Описание	Диапазон значений	По умолчанию
<b>SYNC_DB</b>	IN	INT	Номер DB, в котором сохраняются общие данные для синхронизации SFB RK (Минимальная длина 240 байт).	Зависит от CPU, но не разрешен.	0
<b>EN_R</b>	IN	BOOL	Деблокировка задания	TRUE/FALSE	FALSE
<b>R</b>	IN	BOOL	Выполнение задания прекращено	TRUE/FALSE	FALSE
<b>LADDR</b>	IN	WORD	Адрес входов/выходов вашего субмодуля, который вы установили в «HW Konfig».	Зависит от CPU	3FFh
<b>L_TYPE</b>	OUT	CHAR	Данные приняты: Тип целевой области на локальном CPU (разрешены только большие буквы): <ul style="list-style-type: none"> <li>'D' = блок данных</li> </ul> Подготовка данных: Тип области-источника на локальном CPU (разрешены только большие буквы): <ul style="list-style-type: none"> <li>'D' = блок данных</li> <li>'M' = меркеры</li> <li>'E' = входы</li> <li>'A' = выходы</li> <li>'Z' = счетчики</li> <li>'T' = таймеры</li> </ul>	'D'  'D', 'M', 'E', 'A', 'Z', 'T'	' '
<b>L_DBNO</b>	OUT	INT	Номер блока данных на локальном CPU (цель)	Зависит от CPU, но не разрешен.	0
<b>L_OFFSET</b>	OUT	INT	Номер байта данных на локальном CPU (цель)	0-510	0
<b>L_CF_BYT</b>	OUT	INT	Байт меркеров связи на локальном CPU (255: означает: без меркера связи)	от 0 до 255	0
<b>L_CF_BIT</b>	OUT	INT	Бит меркера связи на локальном CPU	от 0 до 7	0
<b>NDR</b>	OUT	BOOL	Задание завершено без ошибок	TRUE/FALSE	FALSE
<b>ERROR</b>	OUT	BOOL	Задание завершено с ошибкой	TRUE/FALSE	FALSE
<b>STATUS</b>	OUT	WORD	Номер ошибки (см. раздел 6.10.8)	от 0 до FFFFh	0
<b>LEN</b>	IN_OUT	INT	Длина кадра сообщения, количество в байтах	от 0 до 1024	0



## 6.11 Предметный указатель, двухточечное соединение

### А

Адресация операндов данных..... 6-49

### Б

Биты данных..... 6-11, 6-21  
Буферизованный принятый кадр  
сообщения ..... 6-16, **6-24**

### В

Ввод в действие интерфейса на  
физическом уровне..... 6-52  
Возможности использования..... 6-1  
Времена передачи..... 6-93  
Время задержки квитирования (ADT)..... 6-23  
Время задержки символа (CDT).....  
**6-5, 6-13, 6-23, 6-56, 6-60**  
Время контроля при отсутствующем  
символе конца ..... 6-13  
Время переключения..... 6-56  
Встроенная помощь ..... 6-9

### Д

Данные параметризации  
драйвер ASCII..... 6-11  
процедура 3964(R)..... 6-21  
RK 512 ..... 6-26  
Данные приняты  
драйвер ASCII..... 6-58, 6-59  
процедура 3964(R)..... 6-69  
Двунаправленный обмен данными ..... 6-3  
Двухпроводный режим ..... 6-4, 6-11, 6-17  
Двухточечное соединение ..... 6-18  
Длина кадра сообщения..... 6-13, 6-15  
Дополнительный кадр сообщения..... 6-78  
Дополнительный кадр сообщения GET.... 6-85  
Дополнительный кадр сообщения SEND. 6-82  
Драйвер ASCII ..... 6-56  
данные приняты..... 6-58, 6-59  
параметры ..... 6-11  
передача данных ..... 6-56  
приемный буфер..... 6-65  
технические данные ..... 6-90  
управление потоком данных..... 6-65

### З

Заголовок кадра сообщения  
структура командного кадра сообщения  
RK 512..... 6-78  
Зажим для экрана..... 6-6  
Запрет переписывания..... 6-16, 6-24

### И

Извлечение данных  
RK 512 ..... 6-83  
Интерфейс X27  
определение..... 6-2  
свойства ..... 6-2  
Интерфейс X27 (RS 422/485) ..... **6-2, 6-94**

### К

Кадр символа ..... 6-4  
Кадр сообщения FETCH..... 6-78  
Кадр сообщения SEND ..... 6-78  
Класс события..... 6-98  
Кодовая независимость ..... 6-59  
Командный кадр ..... 6-77  
Компьютерный интерфейс RK 512..... 6-77  
извлечение данных ..... 6-83  
командный кадр..... 6-77, 6-78  
ответный кадр сообщения..... 6-79  
параметры..... 6-26  
передача данных..... 6-80  
Контрольная сумма блока..... 6-67  
Контроль четности ..... 6-11, 6-21  
Конфликт инициализации ..... 6-72  
Косвенная параметризация ..... 6-51  
Критерий окончания ..... 6-56, 6-60  
истечение времени задержки символа 6-60  
конечный символ..... 6-63  
фиксированная длина кадра сообщения  
6-61

### М

Меркер связи..... 6-48, 6-78  
Минимальное число циклов CPU..... 6-92  
Многоточечное соединение ..... 6-18

### Н

Настройка приемной линии по умолчанию  
6-17, 6-24  
Настройки по умолчанию 6-17, 6-19, 6-24, 6-52  
Непосредственная параметризация ..... 6-51  
пример..... 6-51  
Номер события..... 6-98

### О

Операнды данных  
адресация..... 6-49  
Основные параметры..... 6-10  
Ответный кадр сообщения..... 6-77, 6-79  
структура и содержимое..... 6-79

Ошибка процедуры.....	6-73
-----------------------	------

## П

Параметризация.....	6-8
Параметризация	
косвенная.....	6-51
непосредственная .....	6-51
Параметрируемая процедура 3964 .....	6-22
Параметры	
SFB 60 SEND_PTP.....	6-108
SFB 61 RCV_PTP.....	6-108
SFB 62 RES_RCVB .....	6-109
SFB 63 SEND_RK.....	6-110
SFB 64 FETCH_RK.....	6-111
SFB 65 SERVE_RK .....	6-112
Параметры модуля.....	6-8
Параметры SFB.....	6-8
Пауза в передаче .....	6-13
Передача данных	
драйвер ASCII.....	6-56
процедура 3964(R).....	6-68
RK 512 .....	6-80
Подключение	
фронтштекера.....	6-6
Полнодуплексный режим .....	6-3, 6-17
Полудуплексный режим .....	6-3, 6-17
Полярность.....	6-52
Попытки передачи.....	6-23
Попытки установления соединения .....	6-23
Приемный буфер.....	6-16, <b>6-65, 6-77</b>
Примеры	
ссылка на.....	6-55
Приоритет.....	6-21, <b>6-66</b>
Процедура 3964	
приемный буфер.....	6-77
Процедура 3964 со стандартными	
значениями.....	6-22
Процедура 3964(R).....	6-66
данные приняты.....	6-69
конфликт инициализации .....	6-72
обработка данных, содержащих ошибки	
6-71	
ошибка процедуры .....	6-73
параметры .....	6-21
передача данных .....	6-68
приоритет.....	6-66
символ контроля блока.....	6-67
технические данные .....	6-91
управляющие символы.....	6-66
Процедура 3964(R), запуск.....	6-73
Процедура 3964(R), передача.....	6-74
Процедура 3964(R), прием .....	6-75
Процедуры квитирования.....	6-65

## Р

Распознавание конца принимаемого	
кадра сообщения.....	6-13
Режим RS422.....	6-56
Режим RS485.....	6-56

## С

Сертификаты.....	6-89
Символическая адресация фактического	
операнда .....	6-51
Символ конца текста.....	6-14, <b>6-56</b>
Символ XOFF.....	6-12
Символ XON.....	6-12
Согласованность данных.....	6-30, 6-33, 6-39, 6-43
Соединительный кабель .....	6-94
Стандартная библиотека .....	6-26
Стартовый бит.....	6-11, 6-21
Стоповые биты.....	6-11, 6-21
Структура программы .....	6-27

## Т

Таблица	
SFB 60 SEND_PTP .....	6-108
SFB 61 RCV_PTP.....	6-108
SFB 62 RES_RCVB.....	6-109
SFB 63 SEND_RK.....	6-110
SFB 64 FETCH_RK .....	6-111
SFB 65 SERVE_RK.....	6-112
Технические данные .....	6-89
Технические данные интерфейса	
X27 (RS 422/485) .....	6-89

## У

Управление потоком данных.....	6-65
Управляющие символы.....	6-66

## Ф

Фактический операнд	
символическая адресация .....	6-51
Физический уровень интерфейса.....	6-52
Фиксированная длина кадра сообщения	
6-56, <b>6-61</b>	
Фронтштекер	
подключение.....	6-6

## Ч

Четырехпроводный режим.....	6-4, 6-11, 6-17
-----------------------------	-----------------

## Э

Экземплярный DB .....	6-26
Экран кабеля.....	6-7, 6-94
Экранные формы для параметризации .....	6-8

## В

BCC (символ контроля блока).....	6-67
----------------------------------	------

## Ф

FETCH_RK .....	6-35, 6-40
----------------	------------

## M

Master ..... 6-18

## R

RCV\_PTP ..... 6-28, 6-31  
RES\_RCVB ..... 6-28, 6-33  
RS 422 ..... 6-4, 6-11, **6-17**  
RS 422/485 ..... 6-2  
RS 485 ..... 6-4, 6-11, **6-17**

## S

SEND\_PTP ..... 6-28  
SEND\_RK ..... 6-35, 6-36  
SERVE\_RK ..... 6-35, 6-45

SFB 60 ..... 6-28  
SFB 61 ..... 6-28  
SFB 62 ..... 6-28  
SFB 63 ..... 6-35  
SFB 64 ..... 6-35  
SFB 65 ..... 6-35  
SFB FETCH\_RK ..... 6-40  
SFB RCV\_PTP ..... 6-31  
SFB RES\_RCVB ..... 6-33  
SFB SEND\_PTP ..... 6-28  
SFB SEND\_RK ..... 6-36  
SFB\_SERVE\_RK ..... 6-45  
Slave ..... 6-18

## X

XON/XOFF ..... 6-11

