

# Канальный уровень протокол обмена по RS485

Материал из wiki

В данном разделе описаны структуры и алгоритмы передачи данных по магистрали RS485. При организации взаимодействия типа "точка - точка" указанный протокол используется "как есть". При организации взаимодействия в сети с несколькими мастерами дополнительно должен использоваться сетевой контроллер, обеспечивающий управление доступом к магистрали передачи со стороны мастеров. Организация работы в этом случае описана в соответствующем разделе.

## Формат кадра данных

При передаче запросов по магистрали RS485, последние упаковываются в кадры. В одном кадре может содержаться произвольное число запросов произвольных типов с произвольными получателями. В ответном кадре будут высланы все ответы на все запросы, от всех получателей, которые помещаются в поле DATA кадра.

Формат передачи байт по магистрали RS485 следующий:

- Скорость передачи 115200/57600/19200 бит/с.
- 1 стоп бит без контроля четности.
- 8 бит данных.

Название поля	0xFF	START	DST	SRC	DATA	CRC	END
Размер, байт	1	1	1	1	0..256	2	1
Назначение	Байт синхронизации приемника и передатчика	Признак начала кадра 0xC0	Адрес получателя кадра	Адрес отправителя кадра	Данные кадра	CRC кадра, рассчитывается по полям DST, SRC, DATA, по алгоритму CRC16-MODBUS	Признак конца кадра 0xC0

Каждый кадр начинается с последовательности стартовых байт 0xFF START. Байт 0xFF служит для синхронизации приемника и передатчика, на случай, если помеха на линии была воспринята приемником как старт-бит. Таким образом этот байт необязателен для приема.

Границей кадра является уникальный флаг START/END (0xC0). Уникальность этого флага поддерживается байт-стаффингом внутри кадра с ESC-последовательностью 0xDB, причём байт START/END (0xC0) заменяется последовательностью (0xDB, 0xDC), а байт ESC (0xDB) — последовательностью (0xDB, 0xDD). Принимающее устройство должно восстанавливать правильную последовательность байт по мере их приема.

Адрес получателя задает адрес устройства, которому предназначен кадр данных. Каждое устройство имеет собственный адрес, устанавливаемый как параметр, хранящийся в энергонезависимой памяти. Также любое устройство принимает кадры, адрес получателя которых равен 0. Такие кадры могут быть использованы для первоначальной установки адреса устройства. Не допускается объединение одной магистралью устройств, имеющих одинаковые адреса.

Адрес отправителя задает адрес устройства, которому будет направлен ответ на запрос.

Поле данных содержит инкапсулированные запросы и ответы. При передаче многобайтных значений (типы int16\_t, int32\_t языка Си), передача осуществляется в порядке LITTLE\_ENDIAN (сначала младший байт, затем старший).

Поле CRC содержит значение для проверки целостности кадра. На кадры, имеющие

Устройство, принимающее кадр должно сформировать ответ на него не позднее чем через 20 мс после его получения. Если отправитель не начинает получать кадр до истечения 20 мс тайм-аута, кадр считается потерянным и отправитель может отправлять новый кадр. При повторной отправке кадра тому же получателю отправитель должен увеличить таймаут в два раза. Если после трех подряд попыток (20, 40, 80 мс) ответ не получен, считается, что связь разорвана.

**Алгоритм передачи кадра**

```

graph TD
    Start([Начало]) --> CRC[Подсчитать CRC кадра, добавить CRC в буфер кадра]
    CRC --> StartByte[Выслать байты 0xFF, 0xC0 (START)]
    StartByte --> LoopStart(( ))
    LoopStart --> IsNextByte{В = очередной байт кадра}
    IsNextByte -- ДА --> SendByte[Выслать байт 0x0B]
    IsNextByte -- НЕТ --> IsEnd{В == 0xC0 || В == 0x0B?}
    IsEnd -- ДА --> SendByte
    IsEnd -- НЕТ --> SendByte
    SendByte --> IsCRC{В == 0xC0?}
    IsCRC -- ДА --> SendByte
    IsCRC -- НЕТ --> SendByte
    SendByte --> HasMore[Есть еще байты в кадре?]
    HasMore -- ДА --> LoopStart
    HasMore -- НЕТ --> EndSend([Конец])
  
```

**Алгоритм приема кадра**

```

graph TD
    Start([Начало]) --> ClearBuf[Сбросить буфер]
    ClearBuf --> IsNextByte{В = очередной полученный байт}
    IsNextByte -- ДА --> IsStartEnd{В == START/END?}
    IsStartEnd -- ДА --> IsNextByte
    IsStartEnd -- НЕТ --> IsNextByte
    IsNextByte --> IsESC{В == ESC(0x0B)?}
    IsESC -- ДА --> IsNextByte
    IsESC -- НЕТ --> IsStartEnd
    IsStartEnd -- ДА --> IsNextByte
    IsStartEnd -- НЕТ --> CalcCRC[Подсчитать CRC кадра за исключением двух последних байт]
    CalcCRC --> StoreCRC[СХСчет в два последних байта в буфере]
    StoreCRC --> IsCRCMatch{CRCсчет == 0xC0?}
    IsCRCMatch -- ДА --> SendError[Передать кадр не валидно]
    IsCRCMatch -- НЕТ --> IsNextByte
    IsNextByte --> Is0C{В == 0C?}
    Is0C -- ДА --> Store0C[Записать 0xC0 в буфер кадра]
    Is0C -- НЕТ --> IsNextByte
    IsNextByte --> Is0B{В == 0B?}
    Is0B -- ДА --> Store0B[Записать 0x0B в буфер кадра]
    Is0B -- НЕТ --> IsNextByte
    IsNextByte --> IsError[Кадр поврежден]
    IsError --> ClearBuf
    IsError --> EndRecv([Конец])
  
```

Протокол не требует, чтобы кадр содержал что-либо кроме границ кадра, адресов получателя и отправителя и CRC. Кадр, не содержащий данных в других документах, ссылающихся на данный документ именуется PING'ом. Данный тип кадра полезен тем, что он имеет минимальный размер и хорошо подходит для проверки наличия связи с устройством или для быстрого подтверждения того, что кадр принят.

&lt;&lt;

[http://192.168.1.38/mediawiki/index.php?title=Канальный\\_уровень\\_протокол\\_обмена\\_по\\_RS485&oldid=6046»](http://192.168.1.38/mediawiki/index.php?title=Канальный_уровень_протокол_обмена_по_RS485&oldid=6046)

- 
 Powered By  
MediaWiki