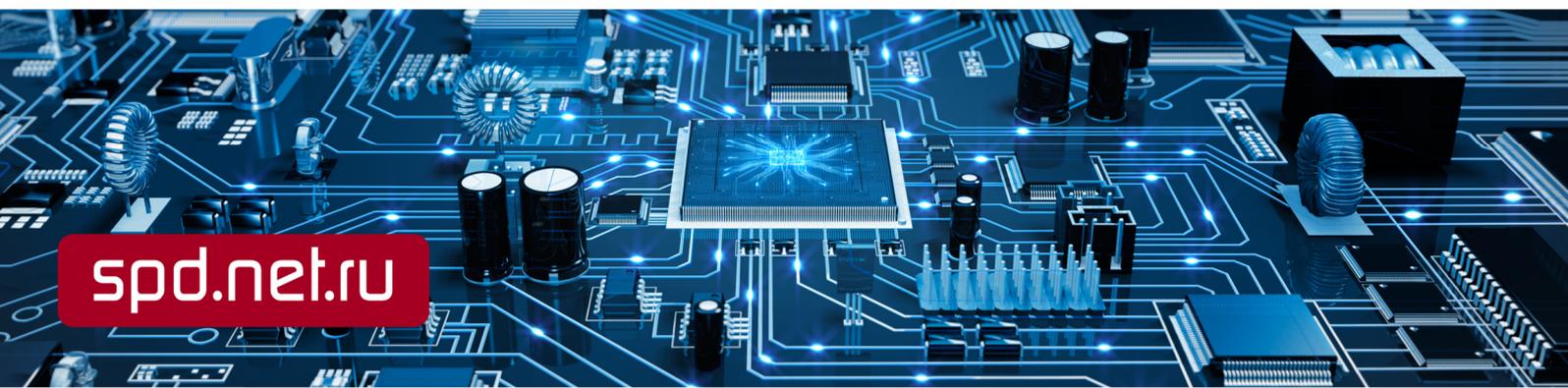


## WAKE16

### Протокол обмена



**СОДЕРЖАНИЕ**

**ОПИСАНИЕ**.....Ошибка! Закладка не определена.  
**ПРИМЕНЕНИЯ** .....Ошибка! Закладка не определена.  
**ОСОБЕННОСТИ**.....Ошибка! Закладка не определена.  
**ХАРАКТЕРИСТИКИ**.....Ошибка! Закладка не определена.  
**УСТРОЙСТВО И ПРИНЦИП РАБОТЫ** .....Ошибка! Закладка не определена.  
**КОММУТАЦИЯ ИНДУКТИВНЫХ НАГРУЗОК**.....Ошибка! Закладка не определена.  
**СТОРОЖЕВОЙ ТАЙМЕР** .....Ошибка! Закладка не определена.  
**ПРОТОКОЛ ОБМЕНА**.....Ошибка! Закладка не определена.  
**ОПИСАНИЕ КОМАНД** .....Ошибка! Закладка не определена.  
**УПРАВЛЕНИЕ USB-РЕЛЕ** .....Ошибка! Закладка не определена.

## ОПИСАНИЕ ПРОТОКОЛА

Протокол WAKE16 является логическим уровнем интерфейса управления оборудованием с помощью асинхронного последовательного канала. Физический уровень интерфейса протоколом не определяется. В качестве него может использоваться, например, RS-232 или RS-485. Протокол позволяет производить обмен пакетами данных длиной до  $2^{16}$  байт с адресуемыми устройствами, которых может быть до  $2^{15} - 1$ . Последовательный канал должен быть сконфигурирован следующим образом:

- 1) число бит в посылке – 8;
- 2) количество стоп-бит – 1;
- 3) бит чётности – нет;
- 4) скорость обмена – 1200...921600 бит/сек;
- 5) использование линий управления модемом – произвольное.

Основой протокола WAKE16 является протокол SLIP (UNIX™ Serial Link Interface Protocol). Передача данных осуществляется в двоичном виде, то есть используются все возможные значения байта (00h...FFh). Для передачи служебной информации зарезервированы два кода: FEND = C0h (Frame End) и FESC = DBh (Frame Escape). Управляющий код FEND служит для обозначения начала посылки, а код FESC служит для передачи ESC-последовательностей. Если в потоке данных встречаются байты, значения которых совпадают с управляющими кодами, производится подмена этих байт ESC-последовательностями. Этот механизм называют байт-стаффингом. Код FEND заменяется последовательностью <FESC>, <TFEND>, а код FESC – последовательностью <FESC>, <TFESC>, где TFEND = DCh (Transposed FEND), TFESC = DDh (Transposed FESC). Коды TFEND и TFESC являются управляющими только в ESC-последовательностях, поэтому при передаче данных они в подмене не нуждаются.

## СТРУКТУРА ПАКЕТА

Пакет всегда начинается управляющим кодом FEND (C0h). Затем следует необязательный байт адреса, после которого идет байт команды. За ним следует байт количества данных и собственно байты данных. Завершает пакет байт контрольной суммы CRC-16:

FEND	ADDR	CMD	N	Data1	...	DataN	CRC-16
------	------	-----	---	-------	-----	-------	--------

**FEND:** управляющий код FEND (C0h) является признаком начала пакета. Благодаря стаффингу, этот код больше нигде в потоке данных не встречается, что позволяет в любой ситуации однозначно определять начало пакета.

**ADDR:** 16-битное слово адреса используется для адресации отдельных устройств. На практике распространена ситуация, когда управление осуществляется только одним устройством. В таком случае слово адреса не требуется, и его можно не передавать. Вместо него сразу за кодом FEND передаётся байт команды CMD.

Часто встречается ситуация, когда в сети имеется всего одно устройство, которое может инициировать обмен данными (мастер-устройство), и при этом обмен ведётся по принципу «запрос – ответ». Примером могут являться сети, построенные на основе интерфейса RS-485. В этом случае пакет данных от мастера содержит адрес устройства, которому этот пакет предназначен. Однако в ответном пакете указывать адрес мастера нет необходимости, так как он единственный в сети.

Для того чтобы можно было однозначно установить, к адресу или команде относится второй байт пакета, введены некоторые ограничения. Для адресации используется 15 бит, а старший бит, передаваемый вместе с адресом, должен всегда быть установлен в единицу:

	D15	...	D1	D0
ADDR =	1	...	A1	A0

Иногда возникает необходимость передать какую-то команду или данные сразу всем устройствам. Для этого предусмотрен коллективный вызов, который осуществляется путём передачи нулевого адреса (учитывая единичный старший бит, в этом случае передаваемый адрес равен 8000h). Нужно отметить, что передача в пакете нулевого адреса полностью аналогична передаче пакета без адреса. Поэтому при реализации протокола можно автоматически исключать нулевой адрес из пакета. Учитывая разрядность адреса и один зарезервированный адрес для коллективного вызова, максимальное количество адресуемых устройств составляет  $2^{15} - 1$ .

Если возникает необходимость передать значение адреса, содержащие значения 40h или 5Bh (передаваемый байт в этом случае будет равен C0h или DBh), то производится байт-стаффинг. Поэтому следует учитывать, что устройства с такими адресами требуют большей длины пакета. Это может быть заметно в тех случаях, когда используются короткие пакеты. В

таких случаях следует избегать назначения устройствам названных адресов.

**CMD:** байт команды всегда должен иметь нулевой старший бит:

	D7	D6	D5	D4	D3	D2	D1	D0
CMD =	0	C6	C5	C4	C3	C2	C1	C0

Таким образом, код команды занимает 7 бит, что позволяет передавать до 128 различных команд. Коды команд выбираются произвольно в зависимости от нужд приложения. Рекомендуется использовать несколько стандартных кодов команд:

Код	Описание команды
01h	Передача управления основной программе центрального контроллера
02h	Передача управления загрузчику центрального контроллера
03h	Чтение блока FLASH-памяти центрального контроллера
04h	Запись блока FLASH-памяти центрального контроллера
08h	Сброс устройства
71h	Запрос информации об устройстве
7Ah	Изменение адреса устройства

Команды обычно имеют несколько параметров, которые передаются далее в виде пакета данных.

Поскольку код команды всегда имеет нулевой старший бит, этот код никогда не совпадает с управляющими кодами. Поэтому при передаче команды байт-стаффинг никогда не производится.

**N:** имеет значение, равное числу передаваемых байт данных. Под величину N отводится два байта:

	D15	...	D1	D0
N =	0	...	N1	N0

Таким образом, один пакет может содержать до  $2^{16}$  байт данных. Значение N не учитывает служебные байты пакета FEND, ADDR, CMD, N и CRC-16. В результате байт-стаффинга фактическая длина пакета может возрасти. Значение N не учитывает этот факт и отражает количество полезных байт данных (то есть значение N всегда таково, как будто байт-стаффинг не осуществляется). Если передаваемая команда не имеет параметров, то передаётся N = 0000h и байты данных опускаются.

Если возникает необходимость передать значение N, равное C0h или DBh, то производится байт-стаффинг. Однако при таких больших значениях N длина пакета столь велика, что

его удлинение ещё на один байт практически незаметно.

**Data1...DataN:** байты данных, количество которых определяется значением N. При N = 0 байты данных отсутствуют. Байты данных могут иметь любое значение, кроме FEND (C0h) и FESC (DBh). Если возникает необходимость передать одно из этих значений, то производится байт-стаффинг.

**CRC:** 16-битное слово контрольной суммы CRC-16. Контрольная сумма рассчитывается перед операцией байт-стаффинга для всего пакета, начиная с байтов адреса (или байта команды, если адреса нет) и заканчивая последним байтом данных. При вычислениях используется значение адреса вместе со старшим битом.

Для расчёта контрольной суммы используется полином  $CRC = X^{16} + X^{12} + X^5 + 1$  (0x8408). Значение CRC перед вычислением инициализируется числом FFFFh. Если CRC содержит значения байтов C0h и DBh, то они заменяются ESC-последовательностями.

Функция для вычисления CRC на языке Си показана ниже:

```
void UpdateCRC16 (char b)
{
    char i;

    CRC16 ^= b;

    for (i = 8; i > 0; i--)
        if (CRC16 & 1)
            CRC16 = (CRC16 >> 1) ^ 0x8408;
        else
            CRC16 >>= 1;
}
```

Эту функцию следует вызвать для каждого байта, в результате чего переменная CRC16 будет содержать вычисленное значение контрольной суммы.