ОАО «Завод Этон» **zavod@eton.by**

МЭП-3500

ЭЛЕКТРОННЫЙ БЛОК УПРАВЛЕНИЯ ДИСКРЕТНОГО ЭЛЕКТРОПРИВОДА

система команд

г. Новолукомль, 2010

1. Интерфейс блока МЭП-3500

Для связи блока МЭП-3500 с компьютером используется интерфейс RS-485 в полудуплексном режиме. Линия передачи данных использует 2 сигнальных провода (A и B) и общий провод (GND). Подключение к компьютеру может осуществляться через преобразователь интерфейсов RS-232 → RS-485 (например, WM-485-01) или USB→ RS-485 (например, WM-485-02).

2. Система команд блока МЭП-3500

Для передачи данных используется стандартный протокол WAKE. Блок МЭП-3500 поддерживает адресацию, поэтому возможна организация сети, состоящей из нескольких (до 127) блоков. Скорость обмена – 9600 бод. Инициатором обмена всегда выступает компьютер, который передает команды блоку МЭП-3500. Вместе с командами передаются параметры. Для разных команд число параметров может быть разным, есть команды, которые не имеют параметров вообще. В ответ на каждую команду блок передает пакет, содержащий тот же номер команды. Первый байт данных этого пакета представляет собой код ошибки (за исключением команд CMD_ECHO и CMD_INFO). Нулевой код ошибки означает успешное выполнение команды. Любой отличный от нуля код – наличие ошибки (см. описание кодов ошибок ниже). Поскольку используется полудуплексный режим, направление передачи периодически переключается. Чтобы преобразователь интерфейсов со стороны компьютера успел переключить направление, блок передает ответ с задержкой не менее 20 мс. Описание команд управления приведено ниже.

2.1. Команда CMD_NOP

Команда CMD_NOP не выполняет никакой операции. Она используется для внутренних целей и никогда не передается в блок или компьютер.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
00h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
00h	0	-							

2.2. Команда CMD_ERR

Блок передает команду CMD_ERR в качестве ответа на любую команду, если произошла ошибка приема пакета. Параметр Error Code для этой команды всегда равен ERR_TX.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
01h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
01h	1	Error Code							

2.3. Команда CMD_ECHO

Команда CMD_ECHO используется для запроса возврата пакета. Пакет может содержать до 64 байт произвольных данных. В ответ на эту команду блок передает пакет в неизменном виде обратно. Команда используется для проверки связи с блоком.

Эта команда не возвращает код ошибки Error Code.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
02h	X	Byte1							
		…							
		ByteN							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
02h	X	Byte1							
		…							
		ByteN							

2.4. Команда CMD_INFO

Команда CMD_INFO представляет собой запрос информации о типе устройства и версии встроенного программного обеспечения (firmware). В ответ передается пакет, содержащий 15 байт данных, которые представляют собой строку в коде ASCII: “MEP-3500 V1.0”, где “ MEP-3500” – тип блока, “V1.0” – версия firmware 1.0. В качестве разделителей используются пробелы (код 20h). Строка заканчивается байтом 00h.

Эта команда не возвращает код ошибки Error Code.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
03h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
03h	14	String: “MEP-3500 V1.0”, 00h							

2.5. Команда CMD_SETADDR

Команда CMD_SETADDR служит для установки сетевого адреса блока.

Параметр Signature представляет собой ключ, который необходимо передать устройству для получения прав изменения адреса. Ключ является неизменным и равен BEDAh.

Параметр Address представляет собой значение адреса подчиненного устройства (блока) в сети. Адрес может принимать значения 0…127. Значение адреса, равное 0, совпадает с адресом для коллективного вызова и может использоваться лишь в случае наличия всего одного подчиненного устройства в сети. Если подключено несколько устройств, адрес может принимать значения 1…127. Каждое устройство должно иметь уникальный адрес, иначе передаваемые по сети данные будут искажены. Для того чтобы назначить устройству адрес, устройство нужно подключить отдельно, т.е. без других устройств в сети, и выполнить команду C_SetAddr. При этом обращаться к устройству можно по ранее заданному адресу или по адресу 0, который является адресом коллективного вызова. В последнем случае новый адрес будет установлен, если даже заданный ранее адрес неизвестен.

Команда возвращает код ошибки Error Code, который может принимать значение Err_No в случае нормального выполнения команды, Err_Pa – в случае неверного значения параметров.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
04h	3	Signature (low byte)							
		Signature (high byte)							
		Address							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
04h	1	Error Code							

2.6. Команда CMD_GETADDR

Команда CMD_GETADDR служит для чтения сетевого адреса блока.

Команда возвращает значение адреса подчиненного устройства (блока) Address. Для того чтобы узнать неизвестный адрес устройства, устройство нужно подключить отдельно, т.е. без

других устройств в сети и выполнить команду C_GetAddr. Обращаться к устройству нужно по адресу 0, который является адресом коллективного вызова.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
05h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
05h	2	Error Code							
		Address							

2.7. Команда CMD_SETM

Команда CMD_SETM служит для установки минимальной скорости двигателя.

Параметр Vm представляет собой минимальную скорость двигателя. Переданное значение автоматически ограничивается диапазоном 1…4000 шаг/сек. Значение сохраняется в EEPROM. Значение по умолчанию равно 80 шаг/сек.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
06h	2	Vm (low byte)							
		Vm (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
06h	1	Error Code							

2.8. Команда CMD_GETM

Команда CMD_GETM служит для считывания текущего значения минимальной скорости двигателя.

Команда возвращает параметр Vm, который представляет собой установленное в данное время значение минимальной скорости двигателя.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
07h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
07h	3	Error Code							
		Vm (low byte)							
		Vm (high byte)							

2.9. Команда CMD_SETA

Команда CMD_SETA служит для установки величины ускорения двигателя и тока ускорения.

Параметр A представляет собой значение ускорения двигателя. Переданное значение автоматически ограничивается диапазоном 0…4000 шаг/сек^2. Значение сохраняется в EEPROM. При нулевом значении ускорения привод сразу переходит на заданную скорость без разгона. Значение по умолчанию равно 0 шаг/сек^2.

Параметр Ia представляет собой значение тока ускорения двигателя. Переданное значение автоматически ограничивается диапазоном 0…3200 мА. Значение сохраняется в EEPROM. Значение по умолчанию равно 2000 мА.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
08h	4	A (low byte)							
		A (high byte)							
		Ia (low byte)							
		Ia (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
08h	1	Error Code							

2.10. Команда CMD_GETA

Команда CMD_GETA возвращает текущее установленное ускорение двигателя и ток ускорения.

Команда возвращает параметр A, который представляет собой установленное ускорение двигателя.

Команда возвращает параметр Ia, который представляет собой установленный ток ускорения.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
09h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
09h	5	Error Code							
		A (low byte)							
		A (high byte)							
		Ia (low byte)							
		Ia (high byte)							

2.11. Команда CMD_SETP

Команда CMD_SETP служит для установки скорости, тока и хода выборки люфта.

Параметр Vp представляет собой значение скорости выборки люфта. Переданное значение автоматически ограничивается диапазоном 0…4000 шаг/сек. Значение сохраняется в EEPROM. Значение по умолчанию равно 400 шаг/сек.

Параметр Ip представляет собой значение тока выборки люфта. Переданное значение автоматически ограничивается диапазоном 0…3200 мА. Значение сохраняется в EEPROM. Значение по умолчанию равно 2000 мА.

Параметр Np представляет собой значение хода выборки люфта. Переданное значение автоматически ограничивается диапазоном 0…30000 шагов. Значение сохраняется в EEPROM. Значение по умолчанию равно 10 шагов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ah	6	Vp (low byte)							
		Vp (high byte)							
		Ip (low byte)							
		Ip (high byte)							
		Np (low byte)							
		Np (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ah	1	Error Code							

2.12. Команда CMD_GETP

Команда CMD_GETP возвращает текущее установленное значение скорости, тока и хода выборки люфта.

Команда возвращает параметр Vp, который представляет собой установленную скорость выборки люфта.

Команда возвращает параметр Ip, который представляет собой установленный ток выборки люфта.

Команда возвращает параметр Np, который представляет собой установленный ход выборки люфта.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Bh	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Bh	7	Error Code							
		Vp (low byte)							
		Vp (high byte)							
		Ip (low byte)							
		Ip (high byte)							
		Np (low byte)							
		Np (high byte)							

2.13. Команда CMD_SETL

Команда CMD_SETL служит для установки скорости, тока и хода запирания.

Параметр Vl представляет собой значение скорости запирания. Переданное значение автоматически ограничивается диапазоном 0…4000 шаг/сек. Значение сохраняется в EEPROM. Значение по умолчанию равно 100 шаг/сек.

Параметр Il представляет собой значение тока запирания. Переданное значение автоматически ограничивается диапазоном 0…3200 мА. Значение сохраняется в EEPROM. Значение по умолчанию равно 2000 мА.

Параметр No представляет собой значение хода запирания вверх. Переданное значение автоматически ограничивается диапазоном 0…30000 шагов. Значение сохраняется в EEPROM.

Значение по умолчанию равно 100 шагов.

Параметр Nc представляет собой значение хода запирания вниз. Переданное значение автоматически ограничивается диапазоном 0…30000 шагов. Значение сохраняется в EEPROM. Значение по умолчанию равно 100 шагов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ch	6	Vl (low byte)							
		Vl (high byte)							
		Il (low byte)							
		Il (high byte)							
		No (low byte)							
		No (high byte)							
		Nc (low byte)							
		Nc (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ch	1	Error Code							

2.14. Команда CMD_GETL

Команда CMD_GETL возвращает текущее установленное значение скорости, тока и хода запирания.

Команда возвращает параметр Vl, который представляет собой установленную скорость запирания.

Команда возвращает параметр Il, который представляет собой установленный ток запирания.

Команда возвращает параметр No, который представляет собой установленный ход запирания вверх.

Команда возвращает параметр Nc, который представляет собой установленный ход запирания вниз.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Dh	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Dh	7	Error Code							
		Vl (low byte)							
		Vl (high byte)							
		Il (low byte)							
		Il (high byte)							
		No (low byte)							
		No (high byte)							
		Nc (low byte)							
		Nc (high byte)							

2.15. Команда CMD_SETW

Команда CMD_SETW служит для установки величины рабочей скорости и рабочего тока двигателя. Всего имеется 4 набора рабочей скорости и тока, необходимый набор выбирается с помощью DIP-переключателя, установленного на печатной плате блока.

Параметры Vw1…Vw4 представляют собой значения рабочей скорости двигателя для каждого из 4-х наборов. Переданные значения автоматически ограничиваются диапазоном 1…4000 шаг/сек. Значения сохраняются в EEPROM. Значения по умолчанию равны 300, 400, 500 и 600 шаг/сек для каждого из наборов соответственно.

Параметры Iw1…Iw4 представляют собой значения рабочего тока двигателя для каждого из 4-х наборов. Переданные значения автоматически ограничиваются диапазоном 0…3200 мА. Значения сохраняются в EEPROM. Значения по умолчанию равны 2000 мА для каждого из наборов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Eh	16	Vw1 (low byte)							
		Vw1 (high byte)							
		Iw1 (low byte)							
		Iw1 (high byte)							
		Vw2 (low byte)							
		Vw2 (high byte)							
		Iw2 (high byte)							
		Iw2 (low byte)							
		Vw3 (high byte)							
		Vw3 (high byte)							
		Iw3 (low byte)							
		Iw3 (high byte)							
		Vw4 (high byte)							
		Vw4 (low byte)							
		Iw4 (high byte)							
		Iw4 (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Eh	1	Error Code							

2.16. Команда CMD_GETW

Команда CMD_GETW возвращает текущие установленные значения рабочей скорости и рабочего тока двигателя для каждого из 4-х наборов.

Команда возвращает параметры Vw1…Vw4, которые представляют собой значения рабочей скорости двигателя для каждого из 4-х наборов.

Команда возвращает параметры Iw1…Iw4, которые представляют собой значения

рабочего тока двигателя для каждого из 4-х наборов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Fh	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Fh	17	Error Code							
		Vw (low byte)							
		Vw (high byte)							
		Iw (low byte)							
		Iw (high byte)							
		Vw (low byte)							
		Vw (high byte)							
		Iw (low byte)							
		Iw (high byte)							
		Vw (low byte)							
		Vw (high byte)							
		Iw (low byte)							
		Iw (high byte)							
		Vw (low byte)							
		Vw (high byte)							
		Iw (low byte)							
		Iw (high byte)							

2.17. Команда CMD_SETS

Команда CMD_SETS служит для управления блоком от компьютера.

Команда имеет единственный параметр, в котором используются только 3 младших бита.

Передача бита Op = 1 равнозначна команде на открытие.

Передача бита Cl = 1 равнозначна команде на закрытие.

Передача бита En = 1 разрешает управление от компьютера.

Если En = 1, то привод выполняет команды компьютера и игнорирует сигналы местного управления. Если En = 0, то привод работает под управлением местных сигналов, а значение битов Op и Cl игнорируется.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
10h	1	-	-	-	-	-	En	Cl	Op

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
10h	1	Error Code							

2.18. Команда CMD_GETS

Команда CMD_GETS возвращает текущее состояние блока.

Команда возвращает параметр State, который представляет собой текущее состояние блока. Возможные значения параметра приведены в таблице:

Значение	Имя состояния	Состояние
0	ST_STOP	Привод остановлен
1	ST_OPEN	Выполняется открытие
2	ST_CLOSE	Выполняется закрытие
3	ST_PLAY_OPEN	Выборка люфта на открытие
4	ST_PLAY_CLOSE	Выборка люфта на закрытие
5	ST_LOCK_OPEN	Запирание на открытие
6	ST_LOCK_CLOSE	Запирание на закрытие
7	ST_LOCKED_OPEN	Блокировка открытия закончена
8	ST_LOCKED_CLOSE	Блокировка закрытия закончена
9	ST_UNLOCK_OPEN	Разблокировка на открытие
10	ST_UNLOCK_CLOSE	Разблокировка на закрытие
11	ST_CALIB_OPEN	Калибровка хода на открытие
12	ST_CALIB_CLOSE	Калибровка хода на закрытие

Команда возвращает параметр Sw, который представляет собой состояние концевых выключателей и сигналов управления. Назначение отдельных битов параметра приведено в таблице:

Бит	Имя сигнала	Сигнал
0	Sw_Opn	Сигнал открытия
1	Sw_Cls	Сигнал закрытия
2	Sw_LmO	Сигнал концевого выключателя на открытие
3	Sw_LmC	Сигнал концевого выключателя на закрытие
4	Pc_En	Блокировка сигналов местного управления
5	Sw_ERR	Сигнал ошибки управляющих сигналов

Все сигналы имеют высокий активный уровень. Сигналы открытия и закрытия являются объединением по логике «ИЛИ» соответствующих сигналов с разных источников. Сигнал ошибки управляющих сигналов формируется в том случае, если сигналы принимают запрещенные значения (например, одновременно подаются сигналы на открытие и закрытие). Если штатное управление блока запрещено, и управление осуществляется только от компьютера, то Pc_En = 1. Если штатное управление блока разрешено, Pc_En = 0.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
11h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
11h	3	Error Code							
		State							
		Sw							

2.19. Команда CMD_SETN

Команда CMD_SETN служит для установки текущей координаты оптического датчика.

Параметр StepN представляет собой желаемую координату в шагах оптического датчика. Переданное значение автоматически ограничивается диапазоном -30000…30000 шагов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
12h	2	StepN (low byte)							
		StepN (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
12h	1	Error Code							

2.20. Команда CMD_GETN

Команда CMD_GETN служит для считывания текущей координаты оптического датчика.

Команда возвращает параметр StepN, который представляет собой текущее значение координаты оптического датчика.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
13h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
13h	3	Error Code							
		StepN (low byte)							
		StepN (high byte)							

2.21. Команда CMD_SETT

Команда CMD_SETT служит для установки значения рабочего хода в шагах оптического датчика. Рабочим ходом является ход между нижней и верхней зоной запирания.

Параметр Nt представляет собой значение рабочего хода в шагах оптического датчика. Переданное значение автоматически ограничивается диапазоном 0…30000 шагов. Значение сохраняется в EEPROM. Значение по умолчанию равно 2000 шагов.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
14h	2	Nt (low byte)							
		Nt (high byte)							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
14h	1	Error Code							

2.22. Команда CMD_GETT

Команда CMD_GETT возвращает текущее установленное значение рабочего хода в шагах оптического датчика.

Команда возвращает параметр Nt, который представляет собой значение рабочего хода в шагах оптического датчика.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
15h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
15h	3	Error Code							
		Nt (low byte)							
		Nt (high byte)							

2.23. Команда CMD_GETI

Команда CMD_GETI возвращает измеренное значение по токовому входу 4 – 20 мА.

Команда возвращает параметр I, который представляет собой значение тока в микроамперах (номинальный диапазон составляет 4000…20000 мкА).

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
16h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
16h	3	Error Code							
		I (low byte)							
		I (high byte)							

2.24. Команда CMD_SETR

Команда CMD_SETR служит для установки параметров реле. Всего имеется 3 реле.

Параметры Rmode1…Rmode3 представляют собой режимы работы для каждого из 3-х реле. Значение по умолчанию равно REL_OFF. Возможные значения параметров приведены в таблице:

Значение	Имя состояния	Состояние
0	REL_OFF	Реле не используется
1	REL_IN	Реле работает по токовому входу
2	REL_OUT	Реле работает по токовому выходу

Параметры Ron1…Ron3 представляют собой значения порога включения для каждого из 3-х реле, выраженные в процентах (номинальный диапазон составляет 0…100%). Значения сохраняются в EEPROM. Значения по умолчанию равны 0%.

Параметры Roff1…Roff3 представляют собой значения порога выключения для каждого из 3-х реле, выраженные в процентах (номинальный диапазон составляет 0…100%). Значения сохраняются в EEPROM. Значения по умолчанию равны 0%.

Параметры Rhyst1…Rhyst3 представляют собой значения гистерезиса для каждого из 3-х реле, выраженные в процентах (номинальный диапазон составляет ±0…100%). Значения сохраняются в EEPROM. Значения по умолчанию равны ±0%.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
17h	12	Rmode1							
		Ron1							
		Roff1							
		Rhyst1							
		Rmode2							
		Ron2							
		Roff2							
		Rhyst2							
		Rmode3							
		Ron3							
		Roff3							
		Rhyst3							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
17h	1	Error Code							

2.25. Команда CMD_GETR

Команда CMD_GETR возвращает текущие значения параметров для каждого из 3-х реле.

Команда возвращает параметры Rmode1…Rmode3, которые представляют собой режимы работы для каждого из 3-х реле.

Команда возвращает параметры Ron1…Ron3, которые представляют собой значения порога включения для каждого из 3-х реле, выраженные в процентах.

Команда возвращает параметры Roff1…Roff3, которые представляют собой значения порога выключения для каждого из 3-х реле, выраженные в процентах.

Команда возвращает параметры Rhyst1…Rhyst3, которые представляют собой значения гистерезиса для каждого из 3-х реле, выраженные в процентах.

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
18h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
18h	13	Error Code							
		Rmode1							
		Ron1							
		Roff1							
		Rhyst1							
		Rmode2							
		Ron2							
		Roff2							
		Rhyst2							
		Rmode3							
		Ron3							
		Roff3							
		Rhyst3							

2.26. Команда CMD_GERS

Команда CMD_GETRS возвращает текущее состояние каждого из 3-х реле.

Команда возвращает параметр, в котором биты R1, R2 и R3 соответствуют текущему состоянию реле (0 – реле выключено, 1 – реле включено).

Команда возвращает код ошибки Error Code, который всегда равен Err_No.

TX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
19h	0	-							

RX

CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
19h	3	Error Code							
		-	-	-	-	-	R3	R2	R1

3. Описание протокола WAKE

Протокол WAKE является логическим уровнем интерфейса управления оборудованием с помощью асинхронного последовательного канала. Физический уровень интерфейса протоколом не определяется, может использоваться, например, RS-232, RS-485 или USB.

3.1. Характеристики протокола

Протокол позволяет производить обмен пакетами данных (data frames) длиной до 255 байт с адресуемыми устройствами, которых может быть до 127. Последовательный канал должен быть сконфигурирован следующим образом:

- число бит в посылке – 8
- количество стоп-бит – 1
- бит четности – нет
- скорость обмена – 300…115200 бод
- использование линий управления модемом – произвольное

3.2. Основа протокола

Основой протокола WAKE является протокол SLIP (UNIX™ Serial Link Interface Protocol). Передача данных осуществляется в двоичном виде, т.е. 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. Такой механизм называют байт-стаффингом (byte stuffing). 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.

Таблица 2.23.1. Управляющие коды протокола WAKE

Обозначение	Пояснение	HEX-значение
FEND	Frame End	C0h
FESC	Frame Escape	DBh
TFEND	Transposed Frame End	DCh
TFESC	Transposed Frame Escape	DDh

Таблица 2.23.2. Подмена байт данных ESC-последовательностями

Байт данных	Передаваемая последовательность
C0h	DBh, DCh
DBh	DBh, DDh

3.3. Структура пакета

Структура пакета WAKE следующая: 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 необязательный байт контрольной суммы CRC-8.

Таблица 2.23.3. Структура пакета WAKE

FEND	ADDR	CMD	N	Data1	…	DataN	CRC

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: Байт адреса используется для адресации отдельных устройств. На практике распространена ситуация, когда управление осуществляется только одним устройством. В таком случае байт адреса не требуется, и его можно не передавать. Вместо него сразу за кодом FEND передается байт команды CMD. Для того, чтобы можно было однозначно установить, адресом или командой является второй байт пакета, введены некоторые ограничения. Для адресации используется 7 бит, а старший бит, передаваемый вместе с адресом, должен всегда быть установлен:

	D7	D6	D5	D4	D3	D2	D1	D0
ADDR =	1	A6	A5	A4	A3	A2	A1	A0

Иногда возникает необходимость передать какую-то команду или данные сразу всем устройствам. Для этого предусмотрен коллективный вызов (broadcast), который осуществляется путем передачи нулевого адреса (учитывая единичный старший бит, в этом случае передаваемый байт равен 80h). 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 127.

Если возникает необходимость передать значение адреса 40h или 5Bh (передаваемый байт в этом случае будет равен C0h или DBh), то производится стаффинг, т.е. передача ESC-последовательности (см. таблицу 2.23.2). Поэтому следует учитывать, что устройства с такими адресами требуют большей на один байт 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:

Таблица 2.23.4. Стандартные коды команд протокола WAKE

Код	Название	Описание команды
00h	CMD_NOP	Нет операции
01h	CMD_ERR	Передача кода ошибки
02h	CMD_ECHO	Запрос возврата переданного пакета
03h	CMD_INFO	Запрос информации об устройстве
04h	CMD_SETADDR	Установка сетевого адреса
05h	CMD_GETADDR	Считывание сетевого адреса

Коды остальных команд выбираются в зависимости от нужд приложения. 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 стаффинг никогда не производится.

N: Байт количества данных имеет значение, равное количеству передаваемых байт данных:

	D7	D6	D5	D4	D3	D2	D1	D0
N =	N7	N6	N5	N4	N3	N2	N1	N0

Таким образом, код количества данных занимает 8 бит, в результате один пакет может содержать до 255 байт данных. Значение N не учитывает служебные байты пакета FEND, ADDR, CMD, N и CRC. В результате стаффинга фактическая длина пакета может возрасти. Значение N **не** учитывает этот факт и отражает количество полезных байт данных (т.е. значение N всегда таково, как будто стаффинг не осуществляется). Если передаваемая команда не имеет параметров, то передается N = 00h и байты данных опускаются.

Если возникает необходимость передать значение N, равное C0h или DBh, то производится стаффинг, т.е. передача ESC-последовательности (см. таблицу 2.23.2). Однако при таких больших значениях N длина пакета столь велика, что его удлинение еще на один байт практически незаметно.

Data1…DataN: Байты данных, количество которых определяется значением N. При N = 00h 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 стаффинг, т.е. передача ESC-последовательности (см. таблицу 2.23.2), состоящей из управляющего кода FESC и кода TFEND (TFESC).

CRC: Байт контрольной суммы CRC-8. Может отсутствовать в **некоторых реализациях** протокола. Контрольная сумма CRC-8 рассчитывается **перед** операцией стаффинга для всего пакета, начиная с байта FEND и заканчивая последним байтом данных. Если в пакете передается адрес, то при вычислении контрольной суммы используется его истинное значение, т.е. единичный старший бит не учитывается. Для расчета контрольной суммы используется полином CRC = $X^8 + X^5 + X^4 + 1$. Значение CRC перед вычислением инициализируется числом DEh. При передаче значения байта контрольной суммы C0h и DBh заменяются ESC-последовательностями (см. таблицу 2.23.2).

3.4. Величина избыточности протокола

Таблица 2.23.5. Величина избыточности протокола WAKE

Вид пакета	Избыточность, %
FEND, CMD, 00h, CRC	75,0
FEND, CMD, 00h	66,7
FEND, ADDR, CMD, 00h, CRC	60,0
FEND, ADDR, CMD, 00h	50,0
FEND, ADDR, CMD, 0Ah, <10 bytes of data>, CRC	20,0
FEND, ADDR, CMD, 32h, <50 bytes of data>, CRC	5,5
FEND, ADDR, CMD, 7Fh, <127 bytes of data>, CRC	2,3
FEND, CMD, 7Fh, <127 bytes of data>, CRC	2,3
FEND, ADDR, CMD, 7Fh, <127 bytes of data>	1,5
FEND, CMD, 7Fh, <127 bytes of data>	1,5

3.5. Порядок обмена и коды ошибок

Обмен всегда начинает мастер. Приняв пакет, подчиненное устройство выполняет переданную в нем команду и отправляет ответ мастеру. В ответе содержится код выполненной команды. Первый байт данных как правило является кодом ошибки. исключение составляют стандартные команды CMD_ECHO и CMD_INFO, которые не передают кода ошибки. Коды стандартных ошибок, определенных для протокола WAKE, приведены в таблице 2.23.6.

Если при выполнении команды, которая должна возвращать несколько байт данных, произошла ошибка, то возвращается всего один байт – код ошибки.

Таблица 2.23.6. Стандартные коды команд протокола WAKE

Имя ошибки	Код ошибки	Название ошибки
Err_No	00h	Нормальное завершение команды
Err_Tx	01h	Ошибка обмена с устройством
Err_Bu	02h	Устройство занято
Err_Re	03h	Устройство не готово
Err_Pa	04h	Ошибка значений параметров
Err_Nr	05h	Нет ответа
Err_Nc	06h	Нет несущей

4. Описание динамической библиотеки wsp32.dll

Со стороны компьютера протокол реализован в динамической библиотеке wsp32.dll, которая может использоваться при разработке коммуникационных программ на любом языке. Дальнейшее описание предполагает, что библиотека используется приложением, написанным на языке C++.

4.1. Заголовочный файл библиотеки wsp32.dll

```
#ifndef WSP32_H
#define WSP32_H
#ifdef WSP32_Exports
#define WSP32_API __declspec(dllexport)
#else
#define WSP32_API __declspec(dllimport)
#endif
extern "C"
 {

  WSP32_API bool WINAPI AccessCOM(char *P);
  WSP32_API bool WINAPI OpenCOM(char *P, DWORD baud);
  WSP32_API bool WINAPI CloseCOM(void);
  WSP32_API HANDLE WINAPI GetHandleCOM(void);
  WSP32_API bool WINAPI SetModeCOM(DWORD RtsMode, DWORD DtrMode);
  WSP32_API bool WINAPI SetModLns(DWORD F);
  WSP32_API bool WINAPI GetModLns(LPDWORD lpD);
  WSP32_API bool WINAPI PurgeCOM(void);
  WSP32_API bool WINAPI FlushCOM(void);
  WSP32_API bool WINAPI GetMaskCOM(LPDWORD lpEvtMask);
  WSP32_API bool WINAPI SetMaskCOM(DWORD EvtMask);
  WSP32_API bool WINAPI WaitEventCOM(LPDWORD lpEvtMask);
  WSP32_API bool WINAPI RxFrame(DWORD To, unsigned char &ADD,
             unsigned char &CMD, unsigned char &N, unsigned char *Data);
  WSP32_API bool WINAPI TxFrame(unsigned char ADDR, unsigned char CMD,
             unsigned char N, unsigned char *Data);
 }
#endif
```

4.2. Описание функций библиотеки wsp32.dll

bool AccessCOM(char *P)

Функция проверяет доступность порта. В качестве параметра передается имя порта (например, “COM1”). Возвращает true в случае доступности порта.

bool OpenCOM(char *P, DWORD baud)

Функция открывает порт. В качестве параметров передаются имя порта (например, “COM1”) и скорость в бодах, которая может принимать одно из стандартных значений (например, 115200). Возвращает true в случае успешного выполнения. Функция устанавливает на линии DTR уровень –12В, а на линии RTS уровень +12В.

bool CloseCOM(void)

Функция закрывает порт. Возвращает true в случае успешного выполнения.

HANDLE GetHandleCOM(void)

Функция возвращает значение дескриптора открытого в данный момент порта.

bool SetModeCOM(DWORD RtsMode, DWORD DtrMode)

Функция осуществляет управление линиями RTS и DTR. Параметр RtsMode может принимать значения RTS_CONTROL_DISABLE (0x00), RTS_CONTROL_ENABLE (0x01), RTS_CONTROL_HANDSHAKE (0x02) и RTS_CONTROL_TOGGLE (0x03). Режим RTS_CONTROL_TOGGLE используется для переключения направления RS-485 с помощью линии RTS и работает на платформах Win 98, ME, 2K, XP. Параметр DtrMode может принимать значения DTR_CONTROL_DISABLE (0x00), DTR_CONTROL_ENABLE (0x01) и DTR_CONTROL_HANDSHAKE (0x02). Возвращает true в случае успешного выполнения.

bool SetModLns(DWORD F)

Функция осуществляет управление линиями RTS и DTR. Возвращает true в случае успешного выполнения. Параметр может принимать такие же значения, как и у функции API EscapeCommFunction.

bool GetModLns(LPDWORD lpD)

Функция считывает линии управления модема CTS и DSR. Возвращает true в случае успешного выполнения. Параметр может принимать такие же значения, как и у функции API GetCommModemStatus.

bool PurgeCOM(void)

Функция очищает буфер COM-порта и прерывает текущие операции приема/передачи. Возвращает true в случае успешного выполнения.

bool FlushCOM(void)

Функция очищает буфер COM-порта, дождавшись завершения передачи. Возвращает true в случае успешного выполнения.

bool GetMaskCOM(LPDWORD lpEvtMask)

Функция считывает маску событий COM-порта. Возвращает true в случае успешного выполнения. Параметр может принимать такие же значения, как и у функции API GetCommMask.

bool SetMaskCOM(DWORD EvtMask)

Функция устанавливает маску Возвращает true в случае успешного выполнения. Параметр может принимать такие же значения, как и у функции API SetCommMask.

bool WaitEventCOM(LPDWORD lpEvtMask)

Функция служит для ожидания события COM-порта. Возвращает true в случае успешного выполнения.

bool RxFrame(DWORD To, unsigned char &ADD, unsigned char &CMD, unsigned char &N, unsigned char *Data)

Функция осуществляет прием WAKE-пакета. Возвращает true в случае успешного

выполнения.

bool TxFrame(unsigned char ADDR, unsigned char CMD, unsigned char N, unsigned char *Data)

Функция осуществляет передачу WAKE-пакета. Возвращает true в случае успешного выполнения.