

# Описание протокола обмена роутер <--> контроллер ШИМ

## 1 Общие положения.

- 1.1** Скорость передачи 9600 бод 8 бит данных без контроля четности с одним стоповым битом без управления потоком.
- 1.2** При обмене роутер всегда выступает ведущим, а контроллер – ведомым.
- 1.3** Пауза между символами >3сек вызывает удаление уже принятой информации (запрос, команда, запись файла) и ответ "NAK".
- 1.4** Передаче и приему подлежат последовательности символов заключенные в одинарные кавычки ' ', пробелы игнорировать, они вставлены для наглядности.
- 1.5** Обмен начинается с посылки роутером запроса/команды, после чего контроллер посылает ответ.
- 1.6** Получение от контроллера ответа на команду, содержащего 'BEL', считать фактом процесса выполнения ранее поданной команды до получения сообщения содержащего 'DONE'.
- 1.7** Получение ответа на команду, содержащего 'DONE', от контроллера, считать фактом исполнения команды.
- 1.8** Получение ответа на команду, содержащего 'FAIL', от контроллера, считать фактом несовместимости исполнения команды с текущим режимом работы контроллера.
- 1.9** Отсутствие какого-либо ответа на запрос/команду в течении более чем 6 секунд, считать отсутствием контроллера.
- 1.10** Контрольные суммы (где указано использование) при обмене — CRC16 CCITT. Текст реализации (C) табличного расчета CRC16 CCITT приведен в приложении.

## 2 Условные обозначения.

- 2.1** "HEX" - символ-заместитель однобайтового шестнадцатеричного числа.
- 2.2** "d" - символ-заместитель символа десятичной цифры 0...9 (0x30...0x39).
- 2.3** "m" - символ-заместитель символа прописной латинской буквы (0x61...0x7A).
- 2.4** "\$" ASCII-символ с кодом 0x24, служит для обозначения признака посылки-команды/запроса контроллеру.
- 2.5** "#" ASCII-символ с кодом 0x23, служит для обозначения признака посылки-ответа контроллера на посылку-запрос.
- 2.6** "SON" символ-заместитель ASCII-символа с кодом 0x01, служит для обозначения начала заголовка файла.
- 2.7** "STX" символ-заместитель ASCII-символа с кодом 0x02, служит для обозначения начала передачи тела файла.
- 2.8** "ETX" символ-заместитель ASCII-символа с кодом 0x03, служит для обозначения конца передачи тела файла и начала передачи контрольной суммы файла.
- 2.9** "EOT" символ-заместитель ASCII-символа с кодом 0x04, служит для обозначения конца передачи файла.
- 2.10** "ENQ" символ-заместитель ASCII-символа с кодом 0x05, служит для обозначения конца передачи команды/запроса.
- 2.11** "ACK" символ-заместитель ASCII-символа с кодом 0x06, служит для подтверждения корректного приема запроса или файла контроллером.
- 2.12** "BEL" символ-заместитель ASCII-символа с кодом 0x07, служит для обозначения истечения времени ожидания следующего байта при приеме.
- 2.13** "NAK" символ-заместитель ASCII-символа с кодом 0x15, служит для индикации некорректного приема команды/запроса или файла контроллером (тайм-аут передачи файла, либо ошибка контрольной суммы, либо неожиданный символ).
- 2.14** "RS" символ-заместитель ASCII-символа с кодом 0x1E, служит для разделения записей точек в теле файла.
- 2.15** "US" символ-заместитель ASCII-символа с кодом 0x1F, служит для разделения полей записи точки в теле файла.
- 2.16** "WAIT" символ заместитель ASCII-символа 'DC1' с кодом 0x11, служит для обозначения состояния обработки последней полученной команды/запроса — символы в ответ на которые получен BEL игнорируются.
- 2.17** "DONE" символ заместитель ASCII-символа 'DC2' с кодом 0x12, служит для обозначения успешного выполнения последней полученной команды/запроса.
- 2.18** "FAIL" символ заместитель ASCII-символа 'DC3' с кодом 0x13, служит для обозначения невозможности выполнения последней полученной команды/запроса.

### 3 Описание запросов.

#### 3.1 Запрос версии.

**3.1.1** Формат команды: '\$ GV ENQ HEX HEX ', где "HEX HEX" – CRC16

**3.1.2** Формат ответа: '# GV RLL h dddd s dddm ACK HEX HEX ',

где "HEX HEX" – CRC16,

"dddd" и "dddm" коды версий аппаратуры(h) и программы(s).

'NAK HEX HEX' - ошибка приема;

'BEL HEX HEX' – обработка предыдущей команды/запроса.

#### 3.2 Запрос режима работы контроллера.

**3.2.1** Запрос: '\$ GM ENQ HEX HEX ', где "HEX HEX" – CRC16

**3.2.2** Формат ответа: '# GM dd ACK HEX HEX ',

где "HEX HEX" – CRC16;

"dd" - режим работы контроллера:

"1x" - режим отработки суточного графика,

"2x" - режим управления яркостью,

"x1" - режим пониженной яркости (высокая температура прожектора),

"x2" - режим аварийного отключения ШИМ (перегрев прожектора);

'NAK HEX HEX' - ошибка приема;

'BEL HEX HEX' – обработка предыдущей команды/запроса.

#### 3.3 Запрос яркости каналов.

**3.3.1** Запрос: '\$ GC ENQ HEX HEX ', где "HEX HEX" – CRC16

**3.3.2** Формат ответа: '# GC ddddd US ddddd US ddddd US ddddd US ddddd US ddddd US ddddd US ddddd US ddddd US ddddd ACK HEX HEX ',

где "HEX HEX" – CRC16;

"dddd" - яркость канала с номером соответствующим позиции поля в сообщении (00000 – максимум, 10000 - выключен);

'NAK HEX HEX' - ошибка приема;

'BEL HEX HEX' – обработка предыдущей команды/запроса.

### **3.4    Запрос значения часов.**

**3.4.1** Формат команды: '\$ GT ENQ HEX HEX ', где "HEX HEX" – CRC16

**3.4.2** Формат ответа: '# GT dd dd dd dd dd dd dd ACK HEX HEX ',

где "HEX HEX" – CRC16,

"dd" значения часов, минут, секунд, дня месяца, дня недели, месяца, года.

'FAIL HEX HEX' – неисправность часов;

'NAK HEX HEX' - ошибка приема;

'BEL HEX HEX' – обработка предыдущей команды/запроса.

### **3.5    Запрос значения контрольной суммы актуального графика.**

**3.5.1** Формат команды: '\$ GH ENQ HEX HEX ', где "HEX HEX" – CRC16

**3.5.2** Формат ответа: '# GH dd US ddddd ACK HEX HEX ',

где "HEX HEX" – CRC16,

"dd" количество точек актуального графика 01...79;

"dddd" значение контрольной суммы актуального графика 0...65535;

'FAIL HEX HEX' – неисправность генераторов;

'NAK HEX HEX' - ошибка приема;

'BEL HEX HEX' – обработка предыдущей команды/запроса.

## 4 Описание команд.

### 4.1 Команда изменения режима работы контроллера.

**4.1.1** Формат команды: '\$ SM d ENQ HEX HEX ', где "HEX HEX" – CRC16, "d" - номер режима работы:

"1" - режим отработки суточного графика (принимается при уже загруженном суточном графике),

"2" - режим ручного управления яркостью.

**4.1.2** Ответы: 'DONE HEX HEX' - требуемый режим установлен,

'NAK HEX HEX' - ошибка приема

### 4.2 Команда установки времени.

**4.2.1** Формат команды: '\$ ST dd dd dd dd dd dd ENQ HEX HEX ', где "HEX HEX" – CRC16, "dd" значения часов, минут, дня месяца, дня недели, месяца, года.

**4.2.2** Ответы: 'DONE HEX HEX' - время установлено;

'FAIL HEX HEX' - ошибка формата;

'NAK HEX HEX' - ошибка приема.

### 4.3 Команда установки корректора хода часов.

**4.3.1** Формат команды: '\$ SL dddd ENQ HEX HEX ',

где "HEX HEX" – CRC16;

"ddd" значение корректора 0000...0511, 1000...1511,  
старшая единица означает положительное значение за собой;

**4.3.2** Ответы: 'DONE HEX HEX' - константа установлена;

'FAIL HEX HEX' - ошибка формата;

'NAK HEX HEX' - ошибка приема.

### 4.4 Команда установки яркости.

**4.4.1** Формат команды: '\$ SC dd US dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd ENQ HEX HEX',

где "HEX HEX" – CRC16,

"dd" – время действия для режима выполнения суточного графика (#GM10) (1...99 сек),

"ddd" - значение яркости в канале (0 - максимум, 10000 – выключен).

**4.4.2** Ответы: 'DONE HEX HEX' - значение установлено;

'FAIL HEX HEX' - ошибка формата;

'NAK HEX HEX' - ошибка приема.

#### **4.5 Команда установки нового делителя частоты генераторов ШИМ.**

**4.5.1** Формат команды: '\$ SP dddd ENQ HEX HEX,  
где "HEX HEX" – CRC16,  
"dddd" - значение делителя (0...65535).

**4.5.2** Ответы: 'DONE HEX HEX' - значение установлено;  
'FAIL HEX HEX' - ошибка формата;  
'NAK HEX HEX' - ошибка приема.

## 5 Передача файла суточного графика в контроллер.

### 5.1 Формат передаваемого файла.

Минимальный размер файла не может быть менее 2 записей.

Максимальный размер файла не может более 79 записей.

Минимальный интервал между соседними точками должен составлять 5 минут.

#### 5.1.1 Заголовок: 'SOH dd STX HEX HEX ',

где "HEX HEX" – CRC16 заголовка,

"dd" – количество записей в файле.

#### 5.1.2 Запись: ' dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd US dddd RS HEX HEX'

где "HEX HEX" – CRC16 записи,

"dddd" - четырехзначное значение времени часы-минуты от 00:01 до 23:59 (пример: 0159, 2300, 0001),

"dddd" - значение в канале (пример: 00000, 00001, 00022, 00033, 04444, 10000) с номером соответствующим позиции поля.

#### 5.1.3 Конец записей: 'ETX'.

#### 5.1.4 Конец файла: 'EOT'.

#### 5.1.5 Записи должны следовать в порядке увеличения поля времени.

#### 5.1.6 Контрольная сумма вычисляется над записью до символа "RS" включительно. Формат контрольной суммы: ' HEX HEX ' - двухбайтовое число контрольной суммы.

#### 5.1.7 Варианты ответа:

'ACK HEX HEX' – блок/файл принят,

'NAK HEX HEX' – блок/файл не принят по несовпадению КС.

'FAIL HEX HEX' – блок/файл не принят по ошибке в формате.

'BEL HEX HEX' – обработка предыдущего блока

## **6 Взаимодействие роутера и контроллера**

### **6.1 Действия контроллера после включения**

- 6.1.1** После включения контроллер переходит в режим ручного управления яркостью (#GM20), установив яркость всех каналов равной 10%. Через 45 секунд после включения контроллер готов к установлению связи.

### **6.2 Установление связи**

- 6.2.1** Фактом наличия связи, считается одинаковый, по содержанию (например #GVh1733s112a), ответ на три посланных запроса \$GV. Если содержание трех ответов одинаково, считать связь установленной. Максимальное время ожидания ответа – 10 секунд.
- 6.2.2** После установления связи требуется, при отсутствии других команд, каждые 40 секунд посылать запрос \$GM(T, C, A, S) “признак «жизни» роутера”. Отсутствие запросов в течении 15 мин приведет к перезагрузке роутера путем снятия с него питания.

### **6.3 Обработка команд и запросов**

- 6.3.1** Все команды/запросы исполняются в любом режиме контроллера.
- 6.3.2** В случае несовпадения в контроллере формата и/или КС команды/запроса, выдается ответ "NAK".

### **6.4 Прием файла**

- 6.4.1** Файл принимается блоками (заголовок, запись, признак конца записей/файла). Очередной блок будет принят ТОЛЬКО в случае успешного приема предыдущего блока.
- 6.4.2** В случае приема символа "SOH" в каналах ШИМ устанавливаются значения соответствующие выключенному состоянию, затем будет принят заголовок с CRC16.
- 6.4.2.1** Прием каждой записи файла заканчивается вычислением CRC16 и разбором принятой записи. В случае совпадения CRC16 и успешного разбора, выдается ответ "ACK" и ожидается прием следующего блока или "ETX". В случае несовпадения, выдается ответ "NAK" и ожидается повтор блока. После десяти неудачных попыток приема блока, либо по ошибке формата (ответ 'FAIL') контроллер удалит все принятые данные и перейдет в состояние ручного управления яркостью (#GM20).
- 6.4.2.2** После приема символа "ETX" проводится проверка на равенство количества принятых записей со значением в заголовке, в случае несовпадения выдается 'FAIL' (не соответствует число записей), удаляются все принятые данные и контроллер перейдет в состояние ручного управления яркостью (#GM20). В случае совпадения количества принятых записей со значением в заголовке выдается символ "ACK" контроллер сформирует суточный график и ожидает символ "EOT".
- 6.4.2.3** В случае корректного приема файла, после приема символа "EOT" контроллер запишет график в энергонезависимую память, запустит циклический расчет значений ШИМ, перейдет к отработке суточного графика (#GM10) и выдается символ "ACK". Время запуска графика – до 30 секунд после приема символа "EOT".
- 6.4.2.4** В случае возникновения тайм-аута в передаче более 3 сек контроллер ответив "BEL", удалит все ранее принятое и перейдет в ручное управление яркостью (#GM20).

**Конец описания**



## ПРИЛОЖЕНИЕ

### Пример реализации на языке C табличного расчета CRC16 CCITT

```
/*
Name   : CRC-16 CCITT
Poly   : 0x1021       $x^{16} + x^{12} + x^5 + 1$ 
Init   : 0xFFFF
Revert : false
XorOut : 0x0000
Check  : 0x29B1 ("123456789")
MaxLen : 4095 байт (32767 бит) - обнаружение
        одинарных, двойных, тройных и всех нечетных ошибок
*/
const unsigned short Crc16Table[256] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
    0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
    0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
    0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
    0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
    0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
    0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
    0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
    0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
    0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
    0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
    0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
    0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
    0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
    0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
    0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
    0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
    0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
    0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
    0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
    0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
    0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
    0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};

unsigned short Crc16(unsigned char * pcBlock, unsigned short len)
{
    unsigned short crc = 0xFFFF;

    while (len--)
        crc = (crc << 8) ^ Crc16Table[(crc >> 8) ^ *pcBlock++];

    return crc;
}
```