



**СЧЕТЧИКИ ИМПУЛЬСОВ БОРЕЙ GA**

**Протокол обмена**

## Оглавление

|   |   |
|---|---|
| Система команд канала UART.....               | 2 |
| Порядок хранения и передачи байт данных ..... | 3 |
| Алгоритм генерации CRC .....                  | 4 |
| Адресация регистров .....                     | 5 |
| Коды ошибок .....                             | 6 |
| Командный регистр .....                       | 6 |
| Тип устройства .....                          | 6 |
| Поле DIB .....                                | 7 |
| Поле VIB .....                                | 7 |
| Назначение входа .....                        | 7 |
| Маски типов событий .....                     | 7 |
| Алгоритм получения архивных данных .....      | 7 |
| Протокол обмена по каналу GPRS .....          | 8 |

## Система команд канала UART

Командно-информационный обмен управляющего компьютера со счетчиком осуществляется в пакетном режиме по принципу “команда-ответ”. В качестве физической среды передачи информации используется канал RS485 со следующими параметрами:

- \* Режим передачи – 8 бит без проверки на четность, 1 стоп-бита.
- \* Скорость обмена – 9600.
- \* Максимальная длина посылки 74 байта.
- \* Порядок следования байтов в слове: старший байт вперед.
- \* Порядок следования слов: младшее слово вперед.

Протокол обмена соответствует стандарту Modbus режим RTU.

Формат кадра сообщения

| Адрес устройства | Команда | Данные | Контрольная сумма |
|------------------|---------|--------|-------------------|
| 1 Байт           | 1 Байт  | N Байт | 2 Байта           |

Для обмена данными со счётчиками используются следующие команды:

0x03 - Получение текущего значения одного или нескольких регистров,

0x10 - Установить новые значения одного или нескольких последовательных регистров.

### 03h - Получение текущего значения одного или нескольких регистров

Формат запроса:

| Адрес устройства | Команда | Начальный адрес регистра | Количество регистров | Контрольная сумма |
|------------------|---------|--------------------------|----------------------|-------------------|
| A                | 0x03    | X X (2 Байта)            | N N (2 Байта)        | C C               |

Формат ответа при выполнении без ошибок:

| Адрес устройства | Команда | Количество байт данных | Данные | Контрольная сумма |
|------------------|---------|------------------------|--------|-------------------|
| A                | 0x03    | D (1 Байт)             | D Байт | C C               |

### 10h - Установить новые значения нескольких последовательных регистров

Формат запроса:

| Адрес устройства | Команда | Начальный адрес регистра | Количество регистров | Количество байт данных | Данные | КС  |
|------------------|---------|--------------------------|----------------------|------------------------|--------|-----|
| A                | 0x10    | X X                      | N N                  | D (D = NN * 2)         | D Байт | C C |

Формат ответа при выполнении без ошибок:

| Адрес устройства | Команда | Начальный адрес регистра | Количество регистров | Контрольная сумма |
|------------------|---------|--------------------------|----------------------|-------------------|
| A                | 0x10    | X X                      | N N                  | C C               |

Формат ответа при выполнении с ошибкой:

|                  |                        |            |                   |
|------------------|------------------------|------------|-------------------|
| Адрес устройства | Команда                | Код ошибки | Контрольная сумма |
| A                | Установлен старший бит | E          | C C               |

### Порядок хранения и передачи байт данных

Для чтения и записи регистров в стандарте Modbus предусмотрены специальные функции, которые оперируют содержимым шестнадцатиразрядных регистров. Эти функции предполагают, что прибор хранит данные только типа шестнадцатиразрядное беззнаковое целое и ничего не «знают» о тех типах данных, которыми действительно представлены параметры прибора. Таким образом, получается, что в приборе данные хранятся в некоем исходном формате, а передаются по сети в виде набора шестнадцатиразрядных регистров. При передаче данных, чей размер в исходном формате превышает 16 бит (long, float, double и т.д.), используются несколько последовательных регистров. При этом младшие слова передаются в первую очередь, старшие - в последнюю. Т.о., для преобразования к порядку байт, естественному для платформы PC, требуется для каждого прочитанного/записываемого регистра изменить порядок байт.

Пример размещения данных для типа Int32

|                  |            |         |            |           |
|------------------|------------|---------|------------|-----------|
| Регистр          | Регистр A0 |         | Регистр A1 |           |
| Порядок передачи | первый     |         |            | последний |
| Байт             | B1         | B0(LSB) | B3(MSB)    | B2        |

Данные типа Char хранятся по одному в регистре.

Пример размещения массива Char[2]

|                  |            |    |            |           |
|------------------|------------|----|------------|-----------|
| Регистр          | Регистр A0 |    | Регистр A1 |           |
| Порядок передачи | первый     |    |            | последний |
| Байт             | 0          | B0 | 0          | B1        |

## Алгоритм генерации CRC

1. 16-ти битовый регистр загружается числом FFFF hex (все 1), и используется далее как регистр CRC.
2. Первый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра CRC. Результат помещается в регистр CRC.
3. Регистр CRC сдвигается вправо(в направлении младшего бита) на 1 бит, старший бит заполняется 0.
4. (Если младший бит 0): Повторяется шаг 3 (сдвиг)  
(Если младший бит 1): Делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа A001 hex.
5. Шаги 3 и 4 повторяются восемь раз.
6. Повторяются шаги со 2 по 5 для следующего сообщения. Это повторяется до тех пор пока все байты сообщения не будут обработаны.
7. Финальное содержание регистра CRC и есть контрольная сумма.  
Контрольная сумма передаётся младшим байтом вперёд.

*// Расчет контрольной суммы кадра Modbus RTU*

```
int modbus_crc( char* pdata, int len )
{
    int crc = 0xFFFF;

    while( len-- )
    {
        crc ^= *pdata++;
        for (int i=0; i<8; i++)
        {
            if (crc & 1)
                crc = (crc>>1)^0xA001;
            else
                crc >>= 1;
        }
    }

    return crc;
}
```

**Табл.1 Адресация регистров**

| Адрес  | Тип данных | Назначение                              | Диапазон        | Чтен | Зап |
|--------|------------|---|-----------------|------|-----|
| 0x0000 | Int32      | Серийный номер                          | 0-99999999h     | ✓    | ✓   |
| 0x0002 | Int16      | Версия ПО                               |                 | ✓    |     |
| 0x0003 | Int16      | Идентификатор ПО                        |                 | ✓    |     |
| 0x0004 | Int16      | Номер сборки ПО                         |                 | ✓    |     |
| 0x0007 | Int8       | День месяца сохранения данных в журнал  | 1-31            | ✓    | ✓   |
| 0x0008 | Int32      | Текущее время в формате UTC             |                 | ✓    | ✓   |
| 0x000A | Int16      | Регистр состояний                       |                 | ✓    |     |
| 0x000B | Int16      | Командный регистр                       |                 |      | ✓   |
| 0x000C | Int16      | Период передачи, ч                      | 1-65535         | ✓    | ✓   |
| 0x000D | Int16      | Период сохранения данных в журнале, мин | 1-255           | ✓    | ✓   |
|        |            |   |                 |      |     |
| 0x0100 | Struct     | Структура настроек канала 1             |                 | ✓    | ✓   |
| 0x0200 | Struct     | Структура настроек канала 2             |                 | ✓    | ✓   |
| 0x0300 | Struct     | Структура настроек канала 3             |                 | ✓    | ✓   |
| 0x0400 | Struct     | Структура настроек канала 4             |                 | ✓    | ✓   |
| 0x0500 | Struct     | Структура настроек канала 5             |                 | ✓    | ✓   |
| 0x0600 | Struct     | Структура настроек канала 6             |                 | ✓    | ✓   |
| 0x0700 | Struct     | Структура настроек канала 7             |                 | ✓    | ✓   |
| 0x0800 | Struct     | Структура настроек канала 8             |                 | ✓    | ✓   |
| 0x0900 | Struct     | Структура настроек канала 9             |                 | ✓    | ✓   |
| 0x0A00 | Struct     | Структура настроек канала 10            |                 | ✓    | ✓   |
| 0x0B00 | Struct     | Структура настроек канала 11            |                 | ✓    | ✓   |
| 0x0C00 | Struct     | Структура настроек канала 12            |                 | ✓    | ✓   |
| 0x0D00 | Struct     | Структура настроек канала 13            |                 | ✓    | ✓   |
| 0x0E00 | Struct     | Структура настроек канала 14            |                 | ✓    | ✓   |
| 0x0F00 | Struct     | Структура настроек канала 15            |                 | ✓    | ✓   |
| 0x1000 | Struct     | Структура настроек канала 16            |                 | ✓    | ✓   |
|        |            |   |                 |      |     |
| 0x2000 | Int32[N]   | Показания по каналам в импульсах        | 0-4 294 967 295 | ✓    | ✓   |
| 0x2050 | Float32[N] | Показания по каналам вычисленное        | >0              | ✓    | ✓   |
| 0x20A0 | Int32      | Состояния входов                        |                 | ✓    |     |
|        |            |   |                 |      |     |
| 0x2100 | Int16      | Индекс записи основного журнала         | 1-2047          | ✓    | ✓   |
| 0x2101 | Int16      | Индекс записи месячного журнала         | 1-341           | ✓    | ✓   |
| 0x2102 | Int16      | Индекс записи журнала событий           | 1-340           | ✓    | ✓   |
| 0x2110 | Float32[N] | Запись основного журнала                |                 | ✓    |     |
| 0x2150 | Float32[N] | Показания по месяцам                    |                 | ✓    |     |

|        |          |                            |           |   |   |
|--------|----------|----------------------------|-----------|---|---|
| 0x2200 | Struct   | Структура описания события |           | V |   |
| 0x2300 | Int16    | Номер порта сервера        | 0-65535   | V | V |
| 0x2301 | Char[4]  | Пин-код симки              | 0000-9999 | V | V |
| 0x2400 | Char[64] | Домен                      |           | V | V |
| 0x2500 | Char[64] | Имя скрипта                |           | V | V |

Табл. 2 Структура настроек канала

| Смещение | Тип данных | Назначение                             | Диапазон     |
|----------|------------|--|--------------|
| 0        | Int16      | Код производителя                      | 0-65535      |
| 1        | Int32      | Серийный номер устройства              | 0-999999999h |
| 3        | Int8       | Версия                                 | 0-255        |
| 4        | Int8       | Тип устройства                         | 0-255        |
| 5        | Int16      | DIB                                    | 0-65535      |
| 6        | Int16      | VIB                                    | 0-65535      |
| 7        | Int8       | Назначение входа                       | 0-4          |
| 8        | Float32    | Вес импульса                           | >0           |
| 10       | Int16      | Минимальная длительность импульса (мс) | 14-15999     |

Табл. 3 Структура описания события

| Смещение | Тип данных | Назначение                       |
|----------|------------|----------------------------------|
| 0        | Int32      | Время в формате UTC              |
| 2        | Int16      | Тип события                      |
| 3        | Int32      | Состояние входов                 |
| 5        | Float32[N] | Показания по каналам вычисленное |

**N** – количество каналов, для Бореев GA N=4.

#### Коды ошибок

- 0x01 – Неизвестная команда;
- 0x02 – Неизвестный адрес регистра;
- 0x03 – Неверное значение параметра;
- 0x04 – Переполнение буфера данных;
- 0x05 – Запись в журнале отсутствует.

#### Командный регистр

При записи в этот регистр, счётчик выполняет следующие действия:

- 0x0001 – очистка основного журнала,
- 0x0002 – очистка месячного журнала,
- 0x0003 – очистка журнала событий,
- 0x0005 – внеочередная передача данных.

#### Тип устройства

Тип подключаемого счётчика кодируется согласно стандарту mBus:

- 02 – электричество,
- 03 – газ,
- 04 – теплосчётчик,
- 06 – горячая вода,
- 07 – вода,
- 16 – холодная вода.

### **Поле DIB**

Поле DIB используется для указания типа данных и разбиения по тарифам:

0x0005 - число с плавающей точкой без указания тарифа

0x1085 - 1-й тариф,

0x2085 - 2-й тариф,

0x3085 - 3-й тариф,

0x4085 - энергия отпущенная,

0x5085 - энергия отпущенная 1-й тариф,

0x6085 - - - 2-й тариф,

0x7085 - - - 3-й тариф.

### **Поле VIB**

Поле VIB определяет единицы измерения параметров:

0x0003 - 1 Втч,

0x0004 - 10 Втч,

0x0013 - 1 л,

0x0014 - 10л,

0x09FB - 1 ГДж,

0x0DFB - 1 Мкал

### **Назначение входа**

0 – не подключен,

1 – импульсный счётный вход,

2 – импульсный аварийный вход,

3 – счётный вход намур,

4 – аварийный вход намур.

### **Маски типов событий**

0x0001 – перезапуск устройства,

0x0002 – отключение внешнего питания,

0x0004 – включение внешнего питания,

0x0008 – изменение состояния входов защиты.

### **Алгоритм получения архивных данных**

Записать индекс записи считываемого журнала (основного, месячного или событий), считать журнальную запись.

Максимальное количество записей в журналы:

основной – 2047,

месячный – 341,

событий – 340.



## Протокол обмена по каналу GPRS

По каналу GPRS счётчик передаёт данные POST-запросом. В запросе присутствует раздел CMD, который содержит команду DevVal. Следующий раздел DATA содержит данные от подключенных приборов в двоичном формате.

Пример запроса:

POST /chron/bin/chronos.cgi? HTTP/1.1

Host: chronometer.ru

Content-Type: multipart/form-data; boundary=BoreyGA09

Content-Length: *длина тела запроса*

--BoreyGA09

Content-Disposition: form-data; name="CMD"

Content-Type: text/plain

DevVal

--BoreyGA09

Content-Disposition: form-data; name="DATA"

Content-Type: application/octet-stream

Content-Transfer-Encoding: binary

<Пакет 1 данных от прибора><Пакет 2 данных от прибора>...

--BoreyGA09--

Структура пакета данных:

L L M M S S S S V T DIB VIB Val <DIB VIB Val >... 01 FD 17 FL 04 6D t t t t C C

L L – длина тела пакета, не включая контрольную сумму CC, 2 байта,

M M – код производителя прибора, 2 байта,

S S S S – серийный номер прибора, 4 байта,

V – версия прибора, 1 байт,

T – тип прибора согласно стандарту EN 13757-4:2005, 1 байт,

DIB – поле DIB согласно стандарту EN 13757-4:2005, берётся из соответствующего поля настроек канала, 1 или 2 байта,

VIB – поле VIB согласно стандарту EN 13757-4:2005, берётся из соответствующего поля настроек канала, 1 или 2 байта,

Val – показания от прибора в формате с плавающей точкой (4 байта) или флаг состояния канала (1 байт), если канал настроен как аварийный,

04 6D t t t t – время на момент снятия данных, формат представления Type F, согласно стандарту EN 13757-4:2005,

01 FD 17 FL – флаги состояния, 0 – нет ошибок, 1 – замкнут аварийный вход, 2 – обрыв (NAmur), 4 – короткое замыкание (NAmur).

C C – контрольная сумма стандарта mBus (2 байта).

Порядок следования байт в пакете – младшие вперёд (little endian).

Пример пакета:

18 00 92 0A 40 20 25 28 00 07 05 13 80 60 A1 48 01 FD 17 00 04 6D 00 2A 51 26 B6 18:

18 00 – длина,

92 0A – код производителя BTR,

40 20 25 28 – серийный номер прибора 28252040,

00 – версия,

07 – тип – счётчик воды,

05 – DIB – число с плавающей точкой,

13 – VIB – единицы измерения 1л,  
80 60 A1 48 – показания 330500,  
01 FD 17 00 – флаги состояния, нет ошибок,  
04 6D 00 2A 51 26 – время 2018-06-17 10:00:00,  
B6 18 – контрольная сумма.

Если 2 или более смежных канала настроены так, что у них одинаковый идентификатор прибора, состоящий из кода производителя, серийного номера, версии и типа прибора, то пакет данных состоит из длины пакета, идентификатора канала, данных канала, данных смежных каналов, времени и контрольной суммы.

В ответном сообщении от сервера счётчик ждёт строку  
<DateTime>ГГГГ-ММ-ДД ЧЧ:ММ:СС</DateTime>,  
по которой синхронизируется с сервером.

*// Расчет контрольной суммы стандарта mBus*

*int mbus\_crc( char\* pdata, int len )*

```
{  
    int crc = 0;  
  
    while( len-- )  
    {  
        crc ^= *pdata++ << 8;  
        for (int i=0; i<8; i++)  
        {  
            if (crc & 0x8000)  
                crc = (crc<<1)^0x3D65;  
            else  
                crc <<= 1;  
        }  
    }  
  
    return crc^0xFFFF;  
}
```