

## Протокол обмена «OPC сервер – управляющий контроллер» (Tag Based Protocol)

Общий формат пакета в сети:

**PP NN DD ... DD XX FF,**

где

**PP** – префикс пакета (биты 0...6 – идентификатор системы (адрес блока), бит 7 – направление пакета: запрос (0) / ответ (1))

**NN** – длина поля данных

**DD** – поле данных (команда или ответ на команду – см. список команд и ответов)

**XX** – контрольная сумма без учета стопового байта (CRC8)

**FF** – стоповый байт

### Список команд и ответов

Ниже приводится только поле данных. Выделенные значения указаны в HEX.

#### Проверка связи

Запрос: **00**

Ответ: **10 20**

Полный пакет запроса: pp 01 00 CC FF

Полный пакет ответа: px 02 10 20 CC FF

pp – идентификатор системы (адрес блока),

px – идентификатор системы (адрес блока) | 0x80

02 – длина поля данных

CC – контрольная сумма

#### Чтение тега:

Запрос: **10 TN**

Ответ: **10 TN F1 F2 F3 F4 ... ,**

где

**TN** – внешний номер тега в системе OPC сервер – управляющий контроллер (см. mbu\_tagstable.pdf)

**F1, F2, F3, F4 ...** – значение запрашиваемого тега (число заданного типа)

Полный пакет запроса: pp 02 10 TN CC FF

Полный пакет ответа: px 0X 10 TN F1 F2 F3 ... CC FF

### Запись тега:

Запрос: **20 TN F1 F2 F3 F4 ...**

Ответ: **20 TN,**

где

**TN** – внешний номер тега в системе OPC сервер – управляющий контроллер (определяется в конфигурационном INI файле)

**F1, F2, F3, F4 ...** – значение, присваиваемое соответствующему тегу и записываемое в устройство (число заданного типа).

Полный пакет запроса:       pp 0X 20 TN F1 F2 F3 ... CC FF

Полный пакет ответа:       rx 03 20 TN CC FF

### Пример чтения тега типа float32

Запрос: 0x11 0x02 0x10 0x01 CS 0xFF

0x11 - идентификатор системы (адрес блока)

0x02 - длина поля данных запроса (0x10 0x01)

0x10 - код команды (чтение)

0x01 - номер тега

CS - контрольная сумма CRC8 (считается для последовательности байт с префикса системы (0x11) по номер тега (0x01)).

0xFF - стоповый байт

Ответ: 0x91 0x06 0x10 0x01 0x3E 0x34 0x12 0x32 CS 0xFF

0x91 - номер системы (0x11 | 0x80)

0x06 - длина поля данных ответа (4 байта float + 2 байта)

0x10 - код команды (чтение)

0x01 - номер тега

0x3E 0x34 0x12 0x32 - данные (float32)

CS - контрольная сумма CRC8 (считается для последовательности байт с префикса системы (0x91) по последний байт данных (0x32)).

0xFF - стоповый байт

## Расчет CRC8

```
//-----  
BYTE CRC8_Table[256] =  
{  
    0x00,0xf7,0xb9,0x4e,0x25,0xd2,0x9c,0x6b,0x4a,0xbd,0xf3,0x04,0x6f,0x98,0xd6,0x21,  
    0x94,0x63,0x2d,0xda,0xb1,0x46,0x08,0xff,0xde,0x29,0x67,0x90,0xfb,0x0c,0x42,0xb5,  
    0x7f,0x88,0xc6,0x31,0x5a,0xad,0xe3,0x14,0x35,0xc2,0x8c,0x7b,0x10,0xe7,0xa9,0x5e,  
    0xeb,0x1c,0x52,0xa5,0xce,0x39,0x77,0x80,0xa1,0x56,0x18,0xef,0x84,0x73,0x3d,0xca,  
    0xfe,0x09,0x47,0xb0,0xdb,0x2c,0x62,0x95,0xb4,0x43,0x0d,0xfa,0x91,0x66,0x28,0xdf,  
    0x6a,0x9d,0xd3,0x24,0x4f,0xb8,0xf6,0x01,0x20,0xd7,0x99,0x6e,0x05,0xf2,0xbc,0x4b,  
    0x81,0x76,0x38,0xcf,0xa4,0x53,0x1d,0xea,0xcb,0x3c,0x72,0x85,0xee,0x19,0x57,0xa0,  
    0x15,0xe2,0xac,0x5b,0x30,0xc7,0x89,0x7e,0x5f,0xa8,0xe6,0x11,0x7a,0x8d,0xc3,0x34,  
    0xab,0x5c,0x12,0xe5,0x8e,0x79,0x37,0xc0,0xe1,0x16,0x58,0xaf,0xc4,0x33,0x7d,0x8a,  
    0x3f,0xc8,0x86,0x71,0x1a,0xed,0xa3,0x54,0x75,0x82,0xcc,0x3b,0x50,0xa7,0xe9,0x1e,  
    0xd4,0x23,0x6d,0x9a,0xf1,0x06,0x48,0xbf,0x9e,0x69,0x27,0xd0,0xbb,0x4c,0x02,0xf5,  
    0x40,0xb7,0xf9,0x0e,0x65,0x92,0xdc,0x2b,0x0a,0xfd,0xb3,0x44,0x2f,0xd8,0x96,0x61,  
    0x55,0xa2,0xec,0x1b,0x70,0x87,0xc9,0x3e,0x1f,0xe8,0xa6,0x51,0x3a,0xcd,0x83,0x74,  
    0xc1,0x36,0x78,0x8f,0xe4,0x13,0x5d,0xaa,0x8b,0x7c,0x32,0xc5,0xae,0x59,0x17,0xe0,  
    0x2a,0xdd,0x93,0x64,0x0f,0xf8,0xb6,0x41,0x60,0x97,0xd9,0x2e,0x45,0xb2,0xfc,0x0b,  
    0xbe,0x49,0x07,0xf0,0x9b,0x6c,0x22,0xd5,0xf4,0x03,0x4d,0xba,0xd1,0x26,0x68,0x9f  
};  
//-----  
BYTE CRC8(BYTE* buff, int len)  
{  
    BYTE crc = 0xFF;  
    for (int i=0; i<len; i++) crc = CRC8_Table[(crc ^ *buff++)];  
    return crc;  
}  
//-----
```