

Protocol of data exchange with Dashboard via UDP

Version 2016.03.10

Client connects to UDP-server in the Dashboard via UDP-port (menu 'File/Parameters' in dashboard).

The transmitted and received data are represented in the binary format.

Multibyte values are transmitted starting from low byte (little endian).

1. Request from client to read coordinates of the beacon

Format of request

Offset	Size	Type	Description	Value
0	1	uint8_t	Address of the beacon	
1	1	uint8_t	Type of packet	0x47
2	2	uint16_t	Code of data in packet	0x0001
4	1	uint8_t	Reserved	0x04
5	2	uint16_t	Reserved	0x0000
7	1	uint8_t	Data size to read (bytes)	0x10
8	2	uint16_t	CRC-16 (see appendix)	

Format of reply from dashboard

Offset	Size	Type	Description	Value
0	1	uint8_t	Address of the beacon	
1	1	uint8_t	Type of packet	0x47
2	2	uint16_t	Code of data in packet	0x0001
4	1	uint8_t	Data size (bytes)	0x10
5	4	uint32_t	Timestamp – time from running of dashboard in 1/64 seconds on the moment of receiving coordinates	
9	2	int16_t	Coordinate X of beacon, cm	
11	2	int16_t	Coordinate Y of beacon, cm	
13	2	int16_t	Coordinate Z of beacon, cm (dashboard v4.37+)	
15	6	6 bytes	Reserved (0)	
21	2	uint16_t	CRC-16 (see appendix)	

Appendix 1. Calculating CRC-16.

For checksum the CRC-16 is used. Last two bytes of N-bytes frame are filled with CRC-16, applied to first (N-2) bytes of frame. To check data you can apply CRC-16 to all frame of N bytes, the result value should be zero.

Below is the implementation of the algorithm in the 'C'.

```
typedef ushort ModbusCrc;// ushort – 2 байта

typedef union {
    ushort w;
    struct{
        uchar lo;
        uchar hi;
    } b;
    uchar bs[2];
} Bytes;

static ModbusCrc modbusCalcCrc(const void *buf, ushort length)
{
    uchar *arr = (uchar *)buf;
    Bytes crc;
    crc.w = 0xffff;
    while(length--){
        char i;
        bool odd;

        crc.b.lo ^= *arr++;
        for(i = 0; i < 8; i++){
            odd = crc.w & 0x01;
            crc.w >>= 1;
            if(odd)
                crc.w ^= 0xa001;
        }
    }
    return (ModbusCrc)crc.w;
}
```