上下端通信协议

1.协议概述

该协议是二进制字节流，采用小端法，用于上下端之间的交互通讯。

上端：指主机端（比如 摄像机，pc，遥控器手操等）

下端：指辅助端（比如 变频器，传感器节点，单片机管理模块等）

注意：本协议其他章节均为通用协议，3.7章节、3.11章节、3.12章节单独针对变频器

2.接口信息

物理接口：串口TTL

串口属性

波特率：115200

数据位：8

校验位：NONE

停止位：1

流控制：NONE

物理接口：以太网

3.协议组成

字节1	字节2	字节3	字节4	字节5-6	字节7-8	字节9-10	字节11-12	字节13-13+n	字节14+n-字节15+n	字节16+n
消息包头	消息action	消息类型	预留	源设备ID	目标ID	消息子类型	消息体长度（n）	消息体	crc校验	消息包尾

3.1消息包头

单字节包头，0xBB。

3.2消息action

单字节

数据	action
0x01	状态上报
0x02	主动查询
0x03	主动设置
0x04	消息回复
0x05	组网
0xff	异常错误

3.3消息类型

单字节,代表具体不同产品或者行业

数值	数据类型
0x00	默认通用
0x01	电池管理
0x02	线切割变频器

3.4预留

单字节，预留，方便拓展后续

3.5源地址

双字节，本机设备id。

注：当为0000时表示点对点通讯

3.6目标地址

双字节，目标设备id

注：当为0000时表示点对点通讯

当为FFFF时标是广播通讯

3.7消息子类型

双字节，小端

注意：此章节只针对线切割变频器

字节1	数据类型	默认值	字节2	数据类型	默认值	单位
0x01	分段频率		0x00	0段频率	d50	HZ
			0x01	1段频率	d50	
			0x02	2段频率	d35	
			0x03	3段频率	d20	
			0x04	4段频率	d15	
			0x05	5段频率	d10	
			0x06	6段频率	d50	
			0x07	7段频率	d25	
			0x08	预设低段频率	d16	
			0x09	预设中段频率	d31	
			0x0A	预设高段频率	d51	
0x02	模块参数		0x01	变频加速时间	d08	0-10秒
			0x02	变频减速时间	d06	0-10秒
			0x03	低频力矩提升	d02	***提升调制比m，提升正弦表***
			0x04	自动省电百分比	d20	降低电压，跟上放参数相反
			0x05	过压调节	d30	单位：V
			0x06	最低开高频频率	d06	
0x03	模块参数2		0x01	开高频延时	d08	0-9.9秒（加速or减速刀指定频率之后的延时）
			0x02	允许掉电最长时间	d01	掉电容错允许时间，时间限制内继续电流驱动，不启动刹车

字节1	数据类型	默认值	字节2	数据类型	默认值	单位
			0x03	加工停机结束方式	d00	00=立即停机，01=停在右边，02=停在左边
			0x04	启动、换向、刹车起始频率	d05	单位HZ
			0x05	断丝检测时间	d00	单位秒
			0x06	丝筒启动方向	d01	00=按照之前方向，01=正向启动，02=反向启动
			0x07	加工结束信号	d00	00=常闭，01=常开
			0x08	断丝检测信号	d00	00=常闭，01=常开

3.8消息体长度

双字节，小端。

3.9校验位

双字节，小端，见附录1

3.10消息包尾

单字节消息包尾，0xCC。从消息类型开始算

3.11消息体

注意：此章节只针对线切割变频器

字节n，十进制

表示数值

3.12错误码

故障代码	含义	
E01	过压	
E02	低压	
E03	断丝	
E04	超程	
E05	左右开关坏	
E06	掉电故障	
E07		
E08		

附录1：Modbus CRC

现提供C语言版

```
#define POLY    0xA001  // The CRC-16 polynomial

/*
 * This function calculates the Modbus CRC-16 for the given data.
 *
 * Parameters:
 *      data:   The data for which to calculate the CRC-16.
 *      length: The length of the data in bytes.
 *
 * Returns:
```

```
 *      The Modbus CRC-16 value.
 */
uint16_t modbus_crc(uint8_t *data, uint16_t length) {
    uint16_t crc = 0xFFFF;
    uint16_t i, j;

    for (i = 0; i < length; i++) {
        crc ^= (uint16_t)data[i];
        for (j = 0; j < 8; j++) {
            if (crc & 0x0001) {
                crc = (crc >> 1) ^ POLY;
            } else {
                crc >>= 1;
            }
        }
    }
    return crc;
}
```