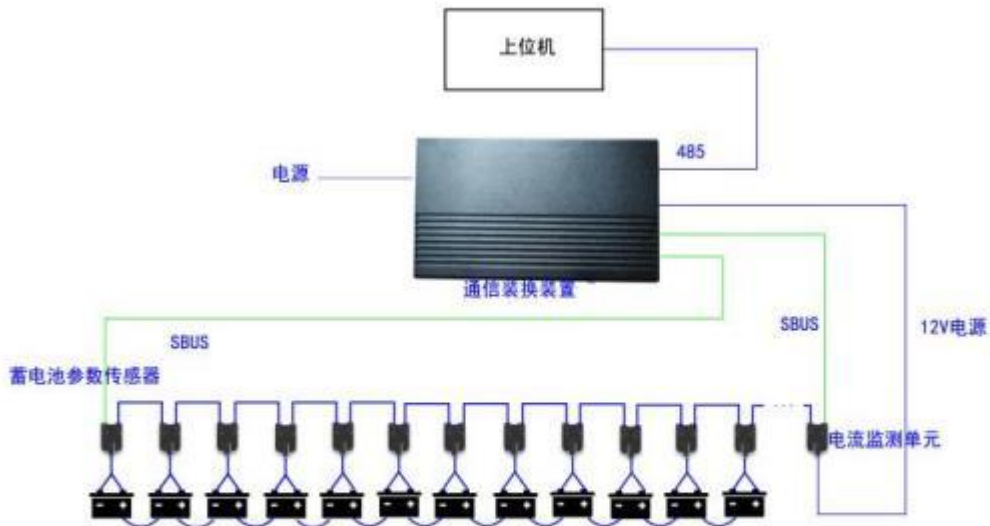


系统硬件结构与规约 V3.0.2

一、硬件结构



二、上位机与钰鑫的通讯转换装置通信规约

2.1 通信约定

- (1) 通讯转换装置的 485 作为对上位机服务的串口。
- (2) 通信为半双工 RS485，波特率默认为 9600bps 可动态设置。
- (3) 采用 Modbus 通信协议。
- (4) 对上位机提供电池串的当前数据，不提供历史数据与历史告警查询功能。

2.2 通信规约细则

(1) Modbus 读取多个寄存器的帧格式：

字节占用	1 byte	1 byte	2 bytes	2 bytes	2 bytes
字段类型	硬件地址	功能码	Modbus	寄存器数量	CRC16 校验

			起始地址		
释义		0x03:读寄存器 0x10:写寄存器	高位在前	高位在前	低位在前
示例 (十六进制)	03	03	10 00	00 0A	C0 FF

(2) Modbus 读取多个寄存器响应的帧格式:

字节占用	1 byte	1 byte	1 byte	n*2 bytes (n 代表所读取的寄存器数量)	2 bytes
字段类型	硬件地址	功能码	数据字节数		CRC16 校验
释义		0x03:读寄存器 0x10:写寄存器	数值等于寄存器数量的 2 倍	高位在前	低位在前
示例 (十六进制)	03	03	14	00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A	16 6F

send: 03 03 10 00 00 0A C0 FF (读 0x1000 开始的 10 个寄存器的内容)

recv: 03 03 14 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A CRC_L
CRC_H (所要读取的 10 个寄存器内容为红色字体)

(3) Modbus 写多个寄存器的帧格式

字节占用	1 byte	1 byte	2 bytes	2 bytes	1 bytes	n*2 bytes (n 代表所读取的寄存器数量)	2 bytes
字段类型	硬件地址	功能码	Modbus 起始地址	寄存器数量	数据字节数	数据	CRC16 校验
释义		0x03:读寄存器 0x10:写寄存器	高位在前	高位在前	数值等于寄存器数量的 2 倍	高位在前	低位在前
示例 (十六进制)	03	01	10 00	00 01	02	00 7B	EE D2

(4) Modbus 写多个寄存器响应的帧格式

字节占用	1 byte	1 byte	2 bytes	2 bytes	2 bytes
字段类型	硬件地址	功能码	Modbus 起始地址	寄存器数量	CRC16 校验
释义		0x03:读寄存器 0x10:写寄存器	高位在前	高位在前	低位在前
示例 (十六进制)	03	10	10 00	00 01	04 EB

示例:

send: 03 10 10 00 00 01 02 **00 7B** EE D2 (对 0x1000 寄存器写入 1 个寄存器, 数据为 0x007B)

recv: 03 10 10 00 00 01 04 EB (执行写寄存器成功)

2.3 CRC16 算法

```
typedef unsigned char u8;
```

```
typedef unsigned short u16;
```

```
const u8 chCRCHTable[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40
};
```

```

const u8 chCRCLTalbe[] =
{
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,
0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
0x41, 0x81, 0x80, 0x40
};

```

```

u16 CRC16_ccitt(u8 *pchMsg, u16 wDataLen)
{
    u8 chCRCHi = 0xFF;
    u8 chCRCLo = 0xFF;
    u16 wIndex;

    while (wDataLen--)
    {
        wIndex = chCRCLo ^ *pchMsg++;
        chCRCLo = chCRCHi ^ chCRCHTalbe[wIndex];
        chCRCHi = chCRCLTalbe[wIndex];
    }
    return ((chCRCHi << 8) | chCRCLo);
}

```

2.4 点表

参见文件

《【钰鑫电气】通信转换装置点表 V3.02.xlsx》

2.5 报文示例（集中器地址为 3、串数为 1）

举例：读取设备地址 **0x03** 的电压值

问询帧（16 进制）：

地址码	功能码	起始地址	数据长度	校验码低位	校验码高位
0x03	0x03	0x11 0x00	0x00 0x01	0x80	0xD4

应答帧（16 进制）：

地址码	功能码	返回有效字节数	第 1 块电池电压值	校验码低位	校验码高位
0x03	0x03	0x02	0x30 0x47	0x80	0xD4

电压计算：

电压：3047（十六进制）=12359=>电压=12.359V

举例：读取设备地址 **0x03** 的温度值

问询帧（16 进制）：

地址码	功能码	起始地址	数据长度	校验码低位	校验码高位
0x03	0x03	0x12 0x00	0x00 0x01	0x80	0x90

应答帧（16 进制）：

地址码	功能码	返回有效字节数	第 1 块电池温度值	校验码低位	校验码高位
0x03	0x03	0x02	0x0A 0x64	0xC6	0xCF

温度计算：

温度：0A64（十六进制）=2660=>温度=26.60℃

举例：读取设备地址 0x03 的内阻值

问询帧（16 进制）：

地址码	功能码	起始地址	数据长度	校验码低位	校验码高位
0x03	0x03	0x13 0x00	0x00 0x01	0x81	0x6C

应答帧（16 进制）：

地址码	功能码	返回有效字节数	第 1 块电池内阻值	校验码低位	校验码高位
0x03	0x03	0x02	0x13 0xFC	0xCC	0xF5

内阻计算：

内阻：13FC（十六进制）=5116=>内阻=5.116mΩ

举例：读取设备地址 0x03 的电流值

问询帧（16 进制）：

地址码	功能码	起始地址	数据长度	校验码低位	校验码高位
0x03	0x03	0x15 0x00	0x00 0x02	0xC1	0xE5

应答帧（16 进制）：

03 03 04 00 00 00 B4 D9 84

地址码	功能码	返回有效字节数	第 1 块电池电流值	校验码低位	校验码高位
0x03	0x03	0x04	0x00 0x00 0x00 0xB4	0xD9	0x84

电流计算：

电流: 000000B4 (十六进制) =180=>电流=1.80A