DBMI系列电池仪通讯协议

目　　录

1 概述

本文描述了电池巡检仪数据上报的Modbus通讯规约标准，应用于ZLBMI系列电池巡检仪向上级监控设备上报数据时的通讯规约。

2 适用范围

规约适用于各电池仪开发商开发电池仪接入到艾默生网络能源有限公司开发PSM-E20监控模块，是开发、测试电池仪通讯软件的依据。

3 参考文献

Modicon Modbus Protocol Reference Guide

4 物理接口

RS485，波特率9600BPS,字符格式采用奇校验位、1位起始位、8位数据位、1位停止位（081）的异步串行通讯格式。

5 帧结构

8Bit地址	8Bit功能码	N*8Bit数据	16BitCRC校验码

采用Modbus规约的RTU（Remote Terminal Unit）方式，每个字节以十六进制数传输，有效的数据范围为00H~FFH。

地址

指电池仪的地址，范围：112~116(70H~74H)

功能码

电池仪支持功能码03（读数据）

数据

上报的数据，按寄存器（数据地址）进行发送，每一个寄存器由两个字节组成，关于寄存器号的定义，请参阅附录A。

CRC校验码

CRC（Cyclical Redundancy Check）对地址、功能码和数据进行校验，由两字节组成，CRC由传输设备生成，附加在数据帧中，如果由接收到数据计算出来的校验和与附加在数据后的校验和不一致，则有错误发生。关于CRC生成函数，请参阅附录B内容。

6 命令解释

6.1 查询数据－功能码 03

上位机发送数据查询命令信息帧，电池仪接收到正确的查询命令后，对命令进行响应回送数据给上位机。格式如下：

查询命令帧格式

字段值	字段说明
70H	地址112
03H	功能码3
00H	起始地址高字节
00H	起始地址低字节，起始地址0
00H	数据个数高字节
18H	数据个数低字节，读24个数据
CRCHi	CRC高字节
CRCLo	CRC低字节

电池仪响应帧格式

字段值	字段说明
70H	地址112
03H	功能码3
30H	应答数据字节数

D0Hi	第1个数据高字节
D0Lo	第1个数据低字节
...	...
...	...
D24Hi	第24个数据高字节
D24Lo	第24个数据低字节
CRCHi	CRC高字节
CRCLo	CRC低字节

7 附录 A 数据地址定义

数据类型　　地址范围

模拟量 AI　　0-110

上位机如果读取设备不支持的数据地址或不存在的数据地址，设备不响应。

上位机如果下发设备不支持的功能码，则设备不响应。

读取数据分 2 段读取：

寄存器地址 0～107 一段，可从任意地址开始读取，读取数量不限。

寄存器地址 108～110 一段，可从任意地址开始读取，读取数量不限。

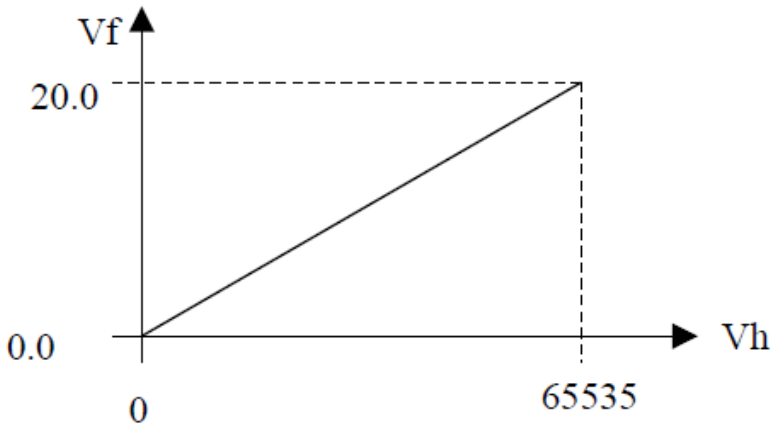
读取时寄存器的起始地址必须在上述地址段内，个数不能超过段的长度。

AI 量地址定义

地址	信号名称	范围	读写	备注
0	1#单体电压	0—20V	只读	注 1
1	2#单体电压	0—20V	只读	
…	…	…		
…	…	…		
107	108#单体电压	0—20V	只读	
108	电池组电流	-3000—3000A	只读	注 2
109	电池组总电压	0—350V	只读	
110	电池组温度	-50—100℃	只读	

注 1：

模拟量（AI）用 16Bit 表示，满量程 65535 表示 20V，0 表示 0，读回的 16Bit 的 AI 值 Vh 和实际表示的 AI 值 Vf 有如下对应关系

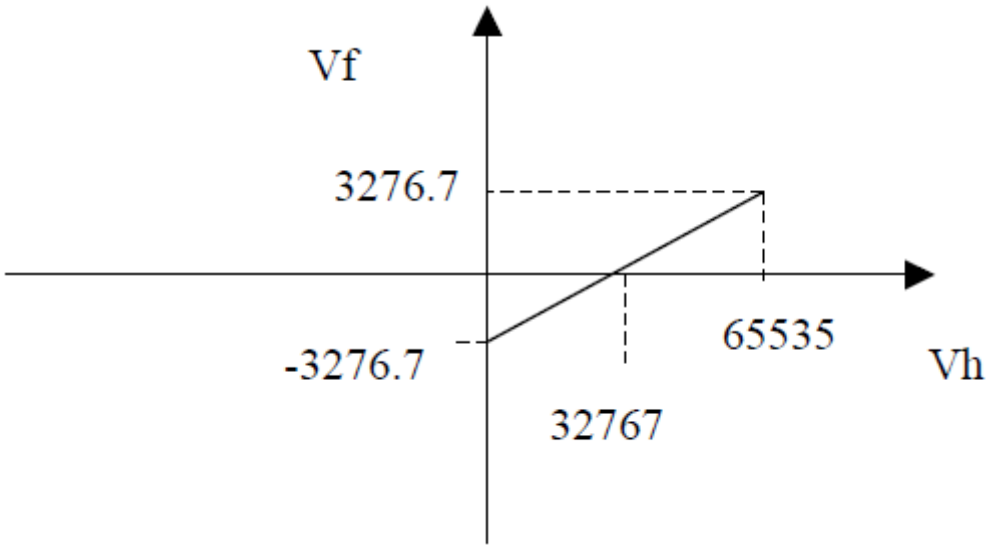


$$Vf = Vh \times \frac{20.0}{65535}$$

最小刻度为 1 个 16 进制数的 1BIT = 20.0/65535 = 0.0003V

注 2：

模拟量（AI）用 16Bit 表示，满量程 65535 表示 3276.7，0 表示-3276.7，读回的 16Bit 的 AI 值 Vh 和实际表示的 AI 值 Vf 有如下对应关系



$$Vf = (Vh - 32767) \times \frac{3276.7}{32767}$$

最小刻度为 1 个 16 进制数的 1BIT = 3276.7/32767 = 0.1

8 附录B：CRC校验的计算方法

CRC（Cyclical Redundancy Check）由两字节组成，生成函数如下：

1、CRC计算函数

```
WORD ModbusCRC(BYTE * pData, BYTE len)
{
    BYTE byCRCHi = 0xff;
    BYTE byCRCLo = 0xff;
    BYTE byIdx;
    WORD crc;

    while(len--)
    {
        byIdx = byCRCHi ^* pData++;
        byCRCHi = byCRCLo ^ gabyCRCHi[byIdx];
        byCRCLo = gabyCRCLo[byIdx];
    }

    crc = byCRCHi;
    crc <<= 8;
    crc += byCRCLo;
    return crc;
}
```

CRC码表高字节

```
BYTE gabyCRCHi[] =
{
    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,
    0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,
    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,
    0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,
    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
};

CRC码表低字节
BYTE    gabyCRCLo[] =
{
    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
};
```