铁塔能源有限公司 BMS 与换电柜上位机 485 串口通讯协议 V1.1

2019 年 4 月 10 日

目录

1. 电池保护板通讯参数

- 通讯采用 485 口，1 位起始位 8 位数据位，无校验，1 位停止位，波特率 9600
- MODBUS 装置地址固定为 1，每个保护板单独 485 口通讯，不采用总线方式
- 协议里面只使用读取开关 (功能码 01) 和读取寄存器 (功能码 03)，其他功能码不使用
- 寄存器 1000 开始是设备 ID，采用保护板后台设备 ID，只读，不可修改
- 查询步骤：1. 查询设备 ID，2. 查询模拟量，3. 查询开关量

2. MODBUS 通讯点表

2.1. 开关量表

按照 20 个电芯定的数据表，如果电芯不足 20 个，数据用 0 补充

MODBUS 地址(开关)	内容	描述	备注
0.	保留位	默认值为 0	
1.	电芯压差过大	1 表示压差过大	
2.	充电过流	1 表示过流	
3.	放电过流	1 表示过流	
4.	短路保护	1 表示短路保护	
5.	充电高温保护	1 表示充电高温保护	
6.	放电高温保护	1 表示放电高温保护	
7.	充电低温保护	1 表示充电低温保护	
8.	放电低温保护	1 表示放电低温保护	
9.	充电 MOS 损坏	1 表示损坏	
10.	放电 MOS 损坏	1 表示损坏	
11.	内部通讯异常	1 表示异常	
12.	过充电压保护 1	1 表示过充电压保护	
13.	过充电压保护 2	1 表示过充电压保护	
14.	过充电压保护 3	1 表示过充电压保护	
15.	过充电压保护 4	1 表示过充电压保护	
16.	过充电压保护 5	1 表示过充电压保护	

17.	过充电压保护 6	1 表示过充电压保护	
18.	过充电压保护 7	1 表示过充电压保护	
19.	过充电压保护 8	1 表示过充电压保护	
20.	过充电压保护 9	1 表示过充电压保护	
21.	过充电压保护 10	1 表示过充电压保护	
22.	过充电压保护 11	1 表示过充电压保护	
23.	过充电压保护 12	1 表示过充电压保护	
24.	过充电压保护 13	1 表示过充电压保护	
25.	过充电压保护 14	1 表示过充电压保护	
26.	过充电压保护 15	1 表示过充电压保护	
27.	过充电压保护 16	1 表示过充电压保护	
28.	过充电压保护 17	1 表示过充电压保护	
29.	过充电压保护 18	1 表示过充电压保护	
30.	过充电压保护 19	1 表示过充电压保护	
31.	过充电压保护 20	1 表示过充电压保护	
32.	过放电压保护 1	1 表示过放电压保护	
33.	过放电压保护 2	1 表示过放电压保护	
34.	过放电压保护 3	1 表示过放电压保护	
35.	过放电压保护 4	1 表示过放电压保护	
36.	过放电压保护 5	1 表示过放电压保护	
37.	过放电压保护 6	1 表示过放电压保护	
38.	过放电压保护 7	1 表示过放电压保护	
39.	过放电压保护 8	1 表示过放电压保护	
40.	过放电压保护 9	1 表示过放电压保护	
41.	过放电压保护 10	1 表示过放电压保护	
42.	过放电压保护 11	1 表示过放电压保护	
43.	过放电压保护 12	1 表示过放电压保护	
44.	过放电压保护 13	1 表示过放电压保护	
45.	过放电压保护 14	1 表示过放电压保护	
46.	过放电压保护 15	1 表示过放电压保护	

47.	过放电压保护 16	1 表示过放电压保护	
48.	过放电压保护 17	1 表示过放电压保护	
49.	过放电压保护 18	1 表示过放电压保护	
50.	过放电压保护 19	1 表示过放电压保护	
51.	过放电压保护 20	1 表示过放电压保护	

2.2. 寄存器表

按照 20 个电芯定的数据表，如果电芯不足 20 个，数据用 0 补充

MODBUS 地址 (寄存器)	内容	系数	单位	备注
0.	电池包实际总电压	0.01	V	
1.	电芯数量，20、17 或其他数值	1		
2.	电量 SOC (0 ~ 100%)	1	%	
3.	剩余容量 (可能会少于设计容量)	0.01	Ah	
4.	SOH (0 ~ 100%)	1	%	
5.	充电电流	0.01	A	
6.	环境温度	1	℃	
7.	电芯温度	1	℃	
8.	板卡温度	1	℃	
9.	电芯 1 电压	0.001	V	
10.	电芯 2 电压	0.001	V	
11.	电芯 3 电压	0.001	V	
12.	电芯 4 电压	0.001	V	
13.	电芯 5 电压	0.001	V	
14.	电芯 6 电压	0.001	V	
15.	电芯 7 电压	0.001	V	
16.	电芯 8 电压	0.001	V	
17.	电芯 9 电压	0.001	V	
18.	电芯 10 电压	0.001	V	
19.	电芯 11 电压	0.001	V	

20.	电芯 12 电压	0.001	V	
21.	电芯 13 电压	0.001	V	
22.	电芯 14 电压	0.001	V	
23.	电芯 15 电压	0.001	V	
24.	电芯 16 电压	0.001	V	
25.	电芯 17 电压	0.001	V	
26.	电芯 18 电压	0.001	V	
27.	电芯 19 电压	0.001	V	
28.	电芯 20 电压	0.001	V	
1000	设备 ID (1)			
1001	设备 ID (2)			
1002	设备 ID (3)			
1003	设备 ID (4)			
1004	设备 ID (5)			
1005	设备 ID (6)			
1006	设备 ID (7)			
1007	设备 ID (8)			
1008	设备 ID (9)			
1009	设备 ID (10)			
1010	设备 ID (11)			
1011	设备 ID (12)			
1012	设备 ID (13)			

2.3. 报文示例

查询 BMS 的设备 ID (一共 13 个寄存器，返回 26 Byte, 返回的数据内容格式应符合 GBT 34014-2017)

发送 01 03 03 E8 00 0D 04 7F

接收 01 03 1A 4B 41 4D 31 32 33 34 35 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6B 2B

BMS 返回的设备 ID 为："KAM123456" (在报文里面长度不足用 00 填充)

4B	41	4D	31	32	33	34	35	36	00	00	00	00
'K'	'A'	'M'	'1'	'2'	'3'	'4'	'5'	'6'				
00	00	00	00	00	00	00	00	00	00	00	00	00

查询模拟量

发送 01 03 00 00 00 1D 85 C3

接收 01 03 3A 17 70 00 11 00 5A 06 F6 04 D2 00 00 00 16 00 17 00 18 10 1B 10 02 10 10 10 7E 0F AC 0F C1 0F CC 0F D7 0F E2 0F ED 0F F8 10 03 10 04 10 0F 10 1A 10 25 10 30 10 3B 10 46 10 51 XX XX

16 进制	10 进制	实际值	描述	地址
17 70	6000	60.0V	标称电压	0
00 11	17	17	电芯数量	1
00 5A	90	90%	电量 (0 ~ 100%)	2
06 F6	1782	17.82Ah	容量	3
04 D2	1234	12.34A	输出电流	4
00 00	0	0A	充电电流	5
00 16	22	22°C	温度 1	6
00 17	23	23°C	温度 2	7
00 18	24	24°C	温度 3	8
10 1B	4123	4.123V	电芯 1 电压	9
10 02	4098	4.098V	电芯 2 电压	10

10 10	4112	4.112V	电芯 3 电压	11
10 7E	4222	4.222V	电芯 4 电压	12
0F AC	4012	4.012V	电芯 5 电压	13
0F C1	4033	4.033V	电芯 6 电压	14
0F CC	4044	4.044V	电芯 7 电压	15
0F D7	4055	4.055V	电芯 8 电压	16
0F E2	4066	4.066V	电芯 9 电压	17
0F ED	4077	4.077V	电芯 10 电压	18
0F F8	4088	4.088V	电芯 11 电压	19
10 03	4099	4.099V	电芯 12 电压	20
10 04	4100	4.100V	电芯 13 电压	21
10 0F	4111	4.111V	电芯 14 电压	22
10 1A	4122	4.122V	电芯 15 电压	23
10 25	4133	4.133V	电芯 16 电压	24
10 30	4144	4.144V	电芯 17 电压	25
10 3B	4155	4.155V	电芯 18 电压	26
10 46	4166	4.166V	电芯 19 电压	27
10 51	4177	4.177V	电芯 20 电压	28

查询开关量：

发送 01 01 00 00 00 34 3D DD

接收 01 01 07 12 08 49 80 10 04 09 69 F0

12 二进制：00010010

二进制位	二进制值	开关量值	描述	地址
D0	0	否	正常	0
D1	1	是	故障	1
D2	0	否	充电过流	2
D3	0	否	放电过流	3
D4	1	是	短路保护	4
D5	0	否	充电高温保护	5

D6	0	否	放电高温保护	6
D7	0	否	充电低温保护	7

08 二进制：00001000

二进制位	二进制值	开关量值	描述	地址
D0	0	否	放电低温保护	8
D1	0	否	充电 MOS 损坏	9
D2	0	否	放电 MOS 损坏	10
D3	1	是	内部通讯异常	11
D4	0	否	过充电压保护 1	12
D5	0	否	过充电压保护 2	13
D6	0	否	过充电压保护 3	14
D7	0	否	过充电压保护 4	15

49 二进制：01001001

二进制位	二进制值	开关量值	描述	地址
D0	1	是	过充电压保护 5	16
D1	0	否	过充电压保护 6	17
D2	0	否	过充电压保护 7	18
D3	1	是	过充电压保护 8	19
D4	0	否	过充电压保护 9	20
D5	0	否	过充电压保护 10	21
D6	1	是	过充电压保护 11	22
D7	0	否	过充电压保护 12	23

80 二进制：10000000

二进制位	二进制值	开关量值	描述	地址
D0	0	否	过充电压保护 13	24
D1	0	否	过充电压保护 14	25

D2	0	否	过充电压保护 15	26
D3	0	否	过充电压保护 16	27
D4	0	否	过充电压保护 17	28
D5	0	否	过充电压保护 18	29
D6	0	否	过充电压保护 19	30
D7	1	是	过充电压保护 20	31

10 二进制：00010000

二进制位	二进制值	开关量值	描述	地址
D0	0	否	过放电压保护 1	32
D1	0	否	过放电压保护 2	33
D2	0	否	过放电压保护 3	34
D3	0	否	过放电压保护 4	35
D4	1	是	过放电压保护 5	36
D5	0	否	过放电压保护 6	37
D6	0	否	过放电压保护 7	38
D7	0	否	过放电压保护 8	39

04 二进制：00000100

二进制位	二进制值	开关量值	描述	地址
D0	0	否	过放电压保护 9	40
D1	0	否	过放电压保护 10	41
D2	1	是	过放电压保护 11	42
D3	0	否	过放电压保护 12	43
D4	0	否	过放电压保护 13	44
D5	0	否	过放电压保护 14	45
D6	0	否	过放电压保护 15	46
D7	0	否	过放电压保护 16	47

09 二进制：00001001

二进制位	二进制值	开关量值	描述	地址

D0	1	是	过放电压保护 17	48
D1	0	否	过放电压保护 18	49
D2	0	否	过放电压保护 19	50
D3	1	是	过放电压保护 20	51
D4	0			
D5	0			
D6	0			
D7	0			

3. MODBUS 通讯规约

3.1. 数据传送

- ✧ 主站和装置间以串行方式连接，主站端以问答方式与保护测控装置通讯。每帧报文的长度不超过 255 个字节。
- ✧ 如果装置收到的主站报文的装置地址、报文类型、数据和校验码都正确，则应在 500ms 内以正常报文响应主站。
- ✧ 如果装置收到的主站报文的装置地址或校验码不正确，则不回答。主站侧判超时后继续后续的通讯。
- ✧ 如果装置收到的报文的装置地址和校验码正确，但报文类型或数据内容不正确，则应在 500ms 内以异常报文回应主站。
- ✧ 采用 RS485，1 位起始位 8 位数据位，无校验，1 位停止位，波特率 9600 (在 1200-57600 之间的值都可以)。

3.2. 数据格式

装置地址	功能码	数据区	CRC 校验
1 个字节	1 个字节	N 个字节	2 个字节(16 位循环冗余校验码)

注：1 个字节由 8 位二进制组成，即 8 bits

3.2.1. 装置地址

装置地址是每次通讯信息帧的第一个字节，从 0 到 255。这个字节表明由用户设置为该地址的装置将接收由主站发过来的此条信息，每个装置必须有一个唯一的地址，只有符合这个地址的装置才能响应主站回送信息。当从机回送信息时，回送数据的第一个字节也是这个装置的地址。

主站发送的数据当中的装置地址表明将要发送到哪个装置，装置返回的数据当中的装置地址表明此数据来自何处。

3.2.2. 功能码

功能码是每次通讯的数据的第二个字节，MODBUS 通讯规约可以定义的功能码的范围为 1 到 127，凯信监控系统仅采用了其中一部分功能码：

功能码 (HEX)	定义	描述
01	读开关	读取一路或多路开关的状态 (遥信)
03	读寄存器	读取一个或多个寄存器 (模拟量) 数据 (遥测)
05	写单路开关	控制一路开关的分或合 (遥控)
06	写单个寄存器	写入一个寄存器/模拟量数据 (遥调)
0F	写多路开关	控制多路开关的分或合 (对多路开关同时遥控)
10	写多个寄存器	写入多个寄存器/模拟量数据 (对多个寄存器同时遥调)

主站向装置发送的命令包含以上的功能码，装置回应主站的数据时：

- ✧ 如果主站发送的命令正确，可以返回正常的数据，装置回应的数据的功能码与主站发送的功能码相同；
- ✧ 如果主站发送的命令错误，无法返回正常的数据，装置回应的数据的功能码等于主站发送的功能码与 80H 取或，即功能码字节的最高位置 1，此时装置回应的数据区只有一个字节的数据，即错误编码。

3.2.3. 装置对主站的正确和错误命令的回应

装置对主站的正确命令的回应：

装置地址	功能码	数据区	CRC 校验
1 个字节	1 个字节 和主站下发命令的功能码相同	N 个字节	2 个字节(16 位循环冗余校验码)

装置对主站的错误命令的回应：

装置地址	功能码	数据区	CRC 校验
1 个字节	1 个字节，最高位置一 即 = 功能码 | 0x80	1 个字节 错误编码	2 个字节(16 位循环冗余校验码)

错误编码：

编码	含义
1	无效的报文类型
2	无效的数据地址，包含数据长度越界
3	写入的数据值无效
6	装置忙

报文示例：

01 81 02 C1 91 收到的功能码为 01 的命令有错误(81)，错误码为 02：地址无效或长度越界

01 83 02 C0 F1 收到的功能码为 03 的命令有错误(83)，错误码为 02：地址无效或长度越界

01 85 03 02 91 收到的功能码为 05 的命令有错误(85)，错误码为 03：写入的数值无效

3.2.4. 数据区

数据区的内容以 Big Endian 形式储存，通讯时先发高位字节，后发低位字节。

数据区的内容根据不同的功能码有不同的规定，具体规定参见下面的功能码详细说明。

3.3. 功能码详细说明

3.3.1. 功能码 01：读开关 (遥信)

所有的开关都以二进制位进行编码，每个开关一位，一个字节可以容纳 8 个开关的状态，1 为合状态，0 为分状态。

开关的地址为位编码的，可以理解为地址为 0 的开关在数据区第 1 个字节的 D0 位，地址为 1 的开关在数据区的第 1 个字节的 D1 位，……地址为 7 的开关在数据区的第 1 个字节的 D7 位，地址为 8 的开关在数据区的第 2 个字节的 D0 位，地址为 X 的开关，在数据区第 X/8+1 个字节的 D[X%8]位。

主机发送的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	01: 读开关状态
起始地址	2 个字节	从哪个地址的开关开始读取开关状态 (起始 bit 位)
开关个数	2 个字节	读取几个开关的状态 (bits 数)
CRC 校验码	2 个字节	装置地址、功能码、起始地址、开关个数的 CRC 校验码

装置返回数据的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	01: 读开关状态
数据字节数 N	1 个字节	接下来有多少个字节的数据，每个字节包含 8 开关的状态 数据字节数 N= (开关个数+7)÷8
数据	N 个字节	返回的第 1 个字节的 D0 位为第一个 (起始地址) 开关的状态； 返回的第 1 个字节的 D1 位为第二个 (起始地址+1) 开关的状态； …… 返回的第 N 个字节的 D0 位为第 8N-7 个 (起始地址+8N-8) 开关的状态 返回的第 N 个字节的 D1 位为第 8N-6 个 (起始地址+8N-7) 开关的状态 …… ……
CRC 校验码	2 个字节	装置地址、功能码、数据字节数、数据的 CRC 校验码

报文示例：

假设装置的地址为 2，开关状态如下：

地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
状态	分	合	分	分	分	合	合	分	分	分	分	合	分	合	分	合

查询地址从 4 到 8 的 5 个开关的状态：

主机发送的数据：02 01 00 04 00 05 BD FB

02	1 个字节	装置地址: 02
01	1 个字节	功能码 01: 读开关状态
00 04	2 个字节	起始地址: 0004，先发高位字节 00，后发低位字节 04
00 05	2 个字节	读取 0005 个开关的状态，先发高位字节 00，后发低位字节 05
BD FB	2 个字节	02 01 00 04 00 05 的 CRC 校验码

装置返回的数据：02 01 01 06 D1 CE

<table>
<tr><td>02</td><td>1 个字节</td><td>装置地址</td></tr>
<tr><td>01</td><td>1 个字节</td><td>功能码 01: 读开关状态</td></tr>
<tr><td>01</td><td>1 个字节</td><td>接下来有 1 个字节的数据，最多可表示 8 开关的状态</td></tr>
<tr><td>06</td><td>1 个字节</td><td>由于只查询 5 个开关的状态，D0-D4: 开关状态，D5-D7 无意义
数据 06 用二进制表示等于 00000110
<table>
<tr><th>位</th><th>状态</th><th>开关地址</th><th>说明</th></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D7: 只查询 5 个开关的状态，D7 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D6: 只查询 5 个开关的状态，D6 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D5: 只查询 5 个开关的状态，D5 无意义</td></tr>
<tr><td>0</td><td>分</td><td>8</td><td>D4=0: 地址为 8 的开关状态为分</td></tr>
<tr><td>0</td><td>分</td><td>7</td><td>D3=0: 地址为 7 的开关状态为分</td></tr>
<tr><td>1</td><td>合</td><td>6</td><td>D2=1: 地址为 6 的开关状态为合</td></tr>
<tr><td>1</td><td>合</td><td>5</td><td>D1=1: 地址为 5 的开关状态为合</td></tr>
<tr><td>0</td><td>分</td><td>4</td><td>D0=0: 地址为 4 的开关状态为分</td></tr>
</table>
由于查询命令是从地址为 4 的开关开始查询的，所以返回的数据是从地址为 4 的开关状态开始的，地址为 4、5、6、7、8 的开关状态分别为分、</td></tr>
</table>

		合、合、分、分
D1 CE	2 个字节	02 01 01 06 的 CRC 校验码

3.3.2. 功能码 03：读寄存器 (遥测)

每个寄存器都是两个字节 (16 位二进制数据)，高位字节在前，低位字节在后。每个寄存器表示的数据范围为-32768 到 32767，负数用补码 (two's complement) 表示。

寄存器的地址编码，可以理解为地址为 0 的寄存器在数据区的第 1 个和第 2 个字节，地址为 1 的寄存器在数据区的第 3 个和第 4 个字节，地址为 2 的寄存器在数据区的第 5 个和第 6 个字节……

主机发送的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	03: 读寄存器
起始地址	2 个字节	从哪个地址的寄存器开始读取数据
寄存器个数	2 个字节	读取几个寄存器的数据 (字节数=寄存器个数×2)
CRC 校验码	2 个字节	装置地址、功能码、起始地址、寄存器个数的 CRC 校验码

装置返回数据的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	03: 读寄存器
数据字节数 N	1 个字节	数据字节数 N=寄存器个数×2
寄存器数据	N 个字节	寄存器个数=数据字节数÷2 返回的第一个字节和第二个字节是第一个 (起始地址) 的寄存器数据 返回的第三个字节和第四个字节是第二个 (起始地址+1) 的寄存器数据 ……
CRC 校验码	2 个字节	装置地址、功能码、数据字节数、寄存器数据的 CRC 校验码

报文示例：

假设装置的地址为 2，寄存器的数据如下：

地址	0	1	2	3	4	5	6	7	8

数据	500	1000	-900	2000	-10	800	300	-1000	600

查询地址从 2 到 5 的 4 个寄存器的数据：

主机发送的数据：02 03 00 02 00 04 E5 FA

02	1 个字节	装置地址: 02
03	1 个字节	功能码 03: 读寄存器
00 02	2 个字节	起始地址: 0002，先发高位字节 00，后发低位字节 02
00 04	2 个字节	读取 0004 个寄存器的数据，先发高位字节 00，后发低位字节 04
E5 FA	2 个字节	02 03 00 02 00 04 的 CRC 校验码

装置应答的数据：02 03 08 FC 7C 07 D0 FF F6 03 20 39 2E

02	1 个字节	装置地址: 02
03	1 个字节	功能码 03: 读寄存器
08	1 个字节	接下来有 8 个字节，即 4 个寄存器的数据
FC 7C 07 D0 FF F6 03 20	8 个字节	因为查询命令是从地址 2 的寄存器开始查询的，返回的第一个寄存器的数据就是地址为 2 的寄存器的数据： FC 7C: 地址为 2 的寄存器的数据 ＝0xFC7C，即 -900 07 D0: 地址为 3 的寄存器的数据 ＝0x07D0，即 2000 FF F6: 地址为 4 的寄存器的数据 ＝0xFFF6，即 -10 03 20: 地址为 5 的寄存器的数据 ＝0x0320，即 800
39 2E	2 个字节	02 03 08 FC 7C 07 D0 FF F6 03 20 的 CRC 校验码

3.3.3. 功能码 05：写单路开关 (遥控)

主机发送的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	05: 写单路开关
开关的地址	2 个字节	对哪个地址的开关进行遥控
控制命令	2 个字节	FF00 为合闸命令，0000 为分闸命令
CRC 校验码	2 个字节	装置地址、功能码、开关的地址、控制命令的 CRC 校验码

装置返回数据的报文格式：

装置返回的报文与主机发送的报文完全相同。装置返回这个报文，说明装置接受了遥控命令，开始执行命令，判断是否成功的执行完成了要以读开关的状态 (遥信) 等于控制的目标值为准，即读出的开关状态等于写入的开关状态，认为遥控执行成功的完成了。

报文示例：

对地址为 1 的开关进行合闸：

主机发送的数据：02 05 00 01 FF 00 DD C9

装置应答的数据：02 05 00 01 FF 00 DD C9

02	1 个字节	装置地址: 02
05	1 个字节	功能码 05: 写单路开关
00 01	2 个字节	开关的地址：0001，先发高位字节 00，后发低位字节 01
FF 00	2 个字节	开关合闸命令：0xFF00
DD C9	2 个字节	02 05 00 01 FF 00 的 CRC 校验码

对地址为 1 的开关进行分闸：

主机发送的数据：02 05 00 01 00 00 9C 39

装置应答的数据：02 05 00 01 00 00 9C 39

02	1 个字节	装置地址: 02
05	1 个字节	功能码 05: 写单路开关
00 01	2 个字节	开关的地址：0001，先发高位字节 00，后发低位字节 01

00 00	2 个字节	开关分闸命令：0x0000
9C 39	2 个字节	02 05 00 01 00 00 的 CRC 校验码

3.3.4. 功能码 06：写单个寄存器 (遥调)

主机发送的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	06: 写单个寄存器
寄存器地址	2 个字节	把数据写入哪个寄存器
写入的数据	2 个字节	写入寄存器的数据
CRC 校验码	2 个字节	装置地址、功能码、寄存器地址、写入的数据的 CRC 校验码

装置返回数据的报文格式：

装置返回的报文与主机发送的报文完全相同。装置返回这个报文，说明装置接受了写入寄存器的命令，开始执行命令，判断是否成功的执行完成了写入数据，要以读寄存器 (遥测) 的数据等于写入的值为准，即读出的寄存器数据等于写入的寄存器数据，认为写入 (遥调) 执行成功的完成了。

报文示例：

把数据-300 写入地址为 4 的寄存器：

主机发送的数据：02 06 00 04 FE D4 88 07

装置应答的数据：02 06 00 04 FE D4 88 07

02	1 个字节	装置地址: 02
06	1 个字节	功能码 06: 写单个寄存器
00 04	2 个字节	寄存器地址：0004，先发高位字节 00，后发低位字节 04
FE D4	2 个字节	写入寄存器的数据 -300 的补码为 0xFED4，先发高位字节 FE，后发低位字节 D4
88 07	2 个字节	02 06 00 04 FE D4 的 CRC 校验码

3.3.5. 功能码 0F：写多路开关 (对多路开关同时进行遥控)

主机发送的报文格式：

装置地址	1 个字节	装置地址

功能码	1 个字节	0F: 写多路开关
起始地址	2 个字节	从哪个地址的开关开始进行遥控
开关个数	2 个字节	对几个开关进行遥控
数据字节数 N	1 个字节	写入开关的数据的字节数，即接下来的遥控命令的字节数 数据字节数 N= (开关个数+7)÷8
写入的数据	N 个字节	写入的第 1 个字节的 D0 位为第一个 (起始地址) 开关的状态； 写入的第 1 个字节的 D1 位为第二个 (起始地址+1) 开关的状态； …… 写入的第 2 个字节的 D0 位为第九个 (起始地址+8) 开关的状态 ……
CRC 校验码	2 个字节	装置地址、功能码、起始地址、开关个数、字节数、数据的 CRC 校验码

装置返回数据的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	0F: 写多路开关
起始地址	2 个字节	从哪个地址的开关开始进行遥控
开关个数	2 个字节	对几个开关进行遥控
CRC 校验码	2 个字节	装置地址、功能码、起始地址、开关个数的 CRC 校验码

装置返回这个报文，说明装置接受了遥控命令，开始执行命令，判断是否成功的执行完成了要以读开关的状态 (遥信) 等于控制的目标值为准，即读出的开关状态等于写入的开关状态，认为遥控执行成功的完成了。

报文示例：

对地址为 1 的开关进行合闸、对地址为 2 的开关进行分闸、对地址为 3 的开关进行合闸：

主机发送的数据：02 0F 00 01 00 03 01 05 32 81

<table>
<tr><td>02</td><td>1 个字节</td><td>装置地址: 02</td></tr>
<tr><td>0F</td><td>1 个字节</td><td>功能码 0F: 写单路开关</td></tr>
<tr><td>00 01</td><td>2 个字节</td><td>起始地址：0001，先发高位字节 00，后发低位字节 01</td></tr>
<tr><td>00 03</td><td>2 个字节</td><td>对 3 个开关进行遥控，先发高位字节 00，后发低位字节 03</td></tr>
<tr><td>01</td><td>1 个字节</td><td>数据的字节数：1 个字节，最多能表示 8 个开关的状态</td></tr>
<tr><td>05</td><td>1 个字节</td><td>由于写入 3 个开关的状态，D0-D2: 开关状态，D3-D7 无意义
数据 05 用二进制表示等于 00000101
<table>
<tr><th>位</th><th>状态</th><th>开关地址</th><th>说明</th></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D7: 只写入 3 个开关的状态，D7 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D6: 只写入 3 个开关的状态，D6 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D5: 只写入 3 个开关的状态，D5 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D4: 只写入 3 个开关的状态，D4 无意义</td></tr>
<tr><td>0</td><td>--</td><td>--</td><td>D3: 只写入 3 个开关的状态，D3 无意义</td></tr>
<tr><td>1</td><td>合</td><td>3</td><td>D2=1: 地址为 3 的开关状态为合</td></tr>
<tr><td>0</td><td>分</td><td>2</td><td>D1=0: 地址为 2 的开关状态为分</td></tr>
<tr><td>1</td><td>合</td><td>1</td><td>D0=1: 地址为 1 的开关状态为合</td></tr>
</table>
由于查询命令是从地址为 1 的开关开始查询的，所以写入的数据是从地址为 1 的开关状态开始的，地址为 1、2、3 的开关状态分别为合、分、合</td></tr>
<tr><td>32 81</td><td>2 个字节</td><td>02 0F 00 01 00 03 01 05 的 CRC 校验码</td></tr>
</table>

装置应答的数据：02 0F 00 01 00 03 44 39

02	1 个字节	装置地址: 02
0F	1 个字节	功能码 0F: 写单路开关
00 01	2 个字节	起始地址：0001，先发高位字节 00，后发低位字节 01
00 03	2 个字节	对 3 个开关进行遥控，先发高位字节 00，后发低位字节 03
44 39	1 个字节	02 0F 00 01 00 03 的 CRC 校验码

3.3.6. 功能码 10：写多个寄存器 (对多个寄存器同时进行遥调)

主机发送的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	10: 写多个寄存器
起始地址	2 个字节	从哪个地址的寄存器开始写入
寄存器个数	2 个字节	对几个寄存器进行写入
数据字节数 N	1 个字节	写入寄存器的数据的字节数，即接下来的遥调命令的字节数 数据字节数 N=寄存器个数×2
写入的数据	N 个字节	写入的第一个字节和第二个字节是第一个 (起始地址) 的寄存器数据 写入的第三个字节和第四个字节是第二个 (起始地址+1) 的寄存器数据 ……
CRC 校验码	2 个字节	装置地址、功能码、起始地址、寄存器个数、字节数、数据 CRC 校验码

装置返回数据的报文格式：

装置地址	1 个字节	装置地址
功能码	1 个字节	10: 写多个寄存器
起始地址	2 个字节	从哪个地址的寄存器开始写入
寄存器个数	2 个字节	对几个寄存器进行写入
CRC 校验码	2 个字节	装置地址、功能码、起始地址、寄存器个数的 CRC 校验码

装置返回这个报文，说明装置接受了遥调命令，开始执行命令，判断是否成功的执行完成了要以读寄存器的数据 (遥测) 等于遥调的目标值为准，即读出的寄存器的数据等于写入寄存器的数据，认为遥调执行成功的完成了。

报文示例：

把数据 400 写入地址为 2 的寄存器，把-500 写入地址为 3 的寄存器，把 700 写入地址为 4 的寄存器：

主机发送的数据：02 10 00 02 00 03 06 01 90 FE 0C 02 BC 72 7F

02	1 个字节	装置地址: 02
10	1 个字节	功能码 10: 写多个寄存器
00 02	2 个字节	起始地址：0002，先发高位字节 00，后发低位字节 02
00 03	2 个字节	写入 3 个寄存器的数据
06	1 个字节	数据的字节数：6 个字节，包括 3 个寄存器的数据

01 90 FE 0C 02 BC	6 个字节	因为写入命令是从地址 2 的寄存器开始查询的，写入的第一个寄存器的数据就是地址为 2 的寄存器的数据： 01 90: 地址为 2 的寄存器的数据 = 0x0190，即 400 FE 0C: 地址为 3 的寄存器的数据 = 0xFE0C，即 -500 02 BC 地址为 4 的寄存器的数据 = 0x02BC，即 700
72 7F	2 个字节	02 10 00 02 00 03 06 01 90 FE 0C 02 BC 的 CRC 校验码

装置应答的数据：02 10 00 02 00 03 21 FB

02	1 个字节	装置地址: 02
10	1 个字节	功能码 10: 写多个寄存器
00 02	2 个字节	起始地址：0002，先发高位字节 00，后发低位字节 02
00 03	2 个字节	写入 3 个寄存器的数据
21 FB	2 个字节	02 10 00 02 00 03 的 CRC 校验码

3.4. CRC16 计算方法

3.4.1. 算法说明

- ✧ 预置 1 个 16 位的寄存器为十六进制的 FFFF（即全为 1）；称此寄存器为 CRC 寄存器。
- ✧ 把第一个 8 位二进制数据（即通讯信息帧的第一个字节）与 16 位的 CRC 寄存器的低 8 位相异或，把结果存放在 CRC 寄存器。
- ✧ 把 CRC 寄存器的内容右移一位（朝低位）用 0 填补最高位，并检查右移后的移出位。
- ✧ 如果移出位为 0：重复第 3 步（再次右移 1 位）；如果移出位为 1：CRC 寄存器与多项式 A001（1010 0000 0000 0001）进行异或。
- ✧ 重复步骤 3 和 4，直到右移 8 次，这样整个 8 位数据全部进行了处理。
- ✧ 重复步骤 2 到步骤 5，进行通讯信息帧下一个字节的处理。
- ✧ 将通讯信息帧的所有字节按上述步骤计算完成后，得到 16 位 CRC 寄存器的高，低字节交换。
- ✧ 最后得到的 CRC 寄存器内容即为：CRC 码。

3.4.2. MODBUS CRC 校验码的 C 语言源程序

```
unsigned short ModBusCRC16(const void *s, int n)
{
  unsigned short c = 0xffff;
  for(int k=0; k<n; k++)
   {
     unsigned short b=(((unsigned char *)s)[k]);
     for(char i=0; i<8; i++)
      {
        c = ((b^c)&1) ? (c>>1)^0xA001 : (c>>1);
        b>>=1;
      }
   }
  return (c<<8)|(c>>8);
}
```