KJ2003监控系统

MODBUS3.0馈电通讯规约(第三版)

电光防爆科技股份有限公司

MODBUS-RTU V3.0 馈电通讯规约说明

目录

一、MODBUS 通讯协议简介：

MODBUS 协议是应用于电子控制器上的一种通用语言。通过此协议，控制器相互之间、控制器经由网络（例如以太网）和其它设备之间可以通信。它已经成为一通用工业标准。有了它，不同厂商生产的控制设备可以连成工业网络，进行集中监控。

MODBUS是一个请求／应答协议，并且提供功能码规定的服务。MODBUS 是一种应用层报文传输协议，用于在通过不同类型的总线或网络连接的设备之间的客户机／服务器通信。

注:MODBUS是Modicon公司的注册商标。

数据编码：

MODBUS 使用最高有效字节在低地址存储的方式表示地址与数据项。即当发送多个字节时，首先发送最高有效字节。

例如：

寄存器大小 值

16 位 0x1234 发送的第一字节为 0x12 然后发0x34

通讯数据的类型及格式：

信息传输为异步方式，以字节为单位，每字节为10 位的格式传输：

字格式（串行数据）	10 位二进制
起始位	1 位，0
数据位	8 位，最低的有效位先被发送
奇偶校验位	偶校验
停止位	1 位，1
波特率	9600bps

通讯数据（信息帧）格式：

数据格式	地址码	功能码	数据区	CRC校验
数据长度	1字节	1字节	N字节	16位CRC校验码

数据字节： 1个字节由8 位二进制数（8Bit）组成。

CRC 校验： CRC 生成后，低字节在前，高字节在后。

MODBUS-RTU的帧结构：

在RTU 模式中，新的信息总是以至少3.5个字符的静默时间开始。紧接着传送第一个域:设备地址。整帧信息必须以一个连续的数据流进行传输。如果信息结束前存在超过1.5个字符以上的间隔时间，则出错。一帧信息的标准结构如下：

开始	地址域	功能域	数据域	CRC校验	结束
T1-T2-T3-T4	8 位	8位	n*8位	16 位	T1-T2-T3-T4

二、通讯信息传输过程：

当通讯命令由发送设备（主机）发送至接收设备（从机）时，符合相应地址码的从机接收通讯命令，并根据功能码及相关要求读取信息，如果CRC 校验无误，则执行相应的任务，然后把执行结果（数据）返送给主机。返回的信息中包括地址码、功能码、数据区及CRC 校验码。如果CRC校验出错则不返回任何信息。

地址码：

地址码是每次通讯信息帧的第一字节，从0 到255。这个字节表明由用户设置地址的从机将接收由主机发送来的信息。同一总线系统内的每个从机都必须有唯一的地址码，并且只有符合地址码的从机才能响应回送信息。当从机回送信息时，回送数据均以各自的地址码开始。主机发送的地址码表明将发送到的从机地址，而从机返回的地址码表明回送的从机地址。相应的地址码表明该信息来自于何处。

功能码：

是每次通讯信息帧传送的第二个字节。MODBUS 通讯规约可定义的功能码为1到127。作为主机请求发送，通过功能码告诉从机应执行什么动作。作为从机响应，从机返回的功能码与主机发送来的功能码一样，并表明从机已响应主机并且已进行相关的操作。

中国 电光 WZB-6GT型微机综合保护装置功能码如下表：

功能码	定义	操作
01	**读开关量输入**	**读取一路或多路开关量状态输入数（遥信）**
03	**读寄存器数据**	**读取一路或多路寄存器数据（遥测、参数、时间）**
05	**写一路开关量输出**	**控制“分/合/复位”，（遥控）**
06	**写单路寄存器**	**把1组二进制数据写入单个寄存器**
10H	**写多路寄存器**	**把多组二进制数据写入多个寄存器**

数据区：

数据区包括需要由从机返回何种信息或执行什么动作。这些信息可以是数据（如：开关量输入/输出、模拟量输入/输出、寄存器等等）、参考地址等。例如，主机通过功能码03告诉从机返回寄存器的值（包含要读取寄存器的起始地址及读取寄存器的长度），则返回的数据包括寄存器的数据长度及数据内容。对于不同的从机，地址和数据信息都不相同（可参照通讯信息表）。

CRC 校验：

MODBUS-RTU 通讯协议的 CRC（冗余循环码）包含 2 个字节，即 16 位二进制数。低字节在前，高字节在后。其详细说明见后页。

静止时间要求：

在 MODBUS-RTU 模式中，发送数据前要求数据总线静止时间即无数据发送时间至少大于 3.5 个字符的时间（如波特率 9600 时为 3.6ms）；整帧的信息必须以一个连续的数据流进行传输，如果信息结束前存在超过 1.5 个字符以上的间隔时间，则出错。

三、MODBUS 功能码简介：

3.1 功能码 01（HEX）：读 1 路或多路开关量输入状态

例：主机要读取地址为 01，开始地址为 **E200H** 的开关量 D0-D15 的输入状态

主机发送的报文格式：01 01 00 00 00 10 3D C6

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	01H	读开关量输入状态
起始地址	2	0000H	写入首地址 E200H 的偏移地址
开关量位数	2	0010H	读取开关量位数（长度可变）
CRC 码	2	3DC6H	由主机算出的 CRC 校验码

从机响应返回的报文格式：01 01 02 01 02 39 AD

从机响应	字节数	返回信息	备注
从机地址	1	01H	来自 01 从机
功能码	1	01H	读开关量输入状态
返回字节数	1	02H	开关量字节数
开关量状态	2	0102H	开关量的状态（低前高后）
CRC 码	2	39ADH	由从机计算出的 CRC 校验码

3.2 功能码 03（HEX）：读 1 路或多路寄存器（包括两个数据区：1.模拟量区 2.定值区）

例 1：主机要读取地址为 01，开始地址为 **E000H** 的 1 个从机寄存器数据**（模拟量数据）**

主机发送的报文格式：01 03 00 00 00 01 84 0A

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	03H	读取寄存器
起始地址	2	0000H	相对首地址 E000H 的起始地址
数据长度	2	0001H	读取 1 个字的内容
CRC 校验	2	840AH	主机计算出的 CRC 校验码

从机响应返回的报文格式：01 03 02 00 64 B9 AF

从机响应	字节数	返回信息	备注
从机地址	1	01H	来自 01 从机
功能码	1	03H	读取寄存器
返回字节数	1	02H	读取 2 个字节
数据	2	0064H	从机返回 1 个寄存器的数据内容
CRC 校验码	2	B9AFH	由从机计算出的 CRC 校验码

例 2：主机要读取地址为 01，开始地址为 **E100H** 的 1 个从机寄存器数据**（定值数据）**

主机发送的报文格式：01 03 01 00 00 01 85 F6

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	03H	读取寄存器
起始地址	2	0100H	相对首地址 E000H 的起始地址
数据长度	2	0001H	读取 1 个字的内容
CRC 校验	2	85F6H	主机计算出的 CRC 校验码

从机响应返回的报文格式：01 03 02 00 64 B9 AF

从机响应	字节数	返回信息	备注
从机地址	1	01H	来自 01 从机
功能码	1	03H	读取寄存器
返回字节数	1	02H	读取 2 个字节
数据	2	0064H	从机返回 **E100H** 寄存器的数据内容
CRC 校验码	2	B9AFH	由从机计算出的 CRC 校验码

3.3 功能码 05（HEX）：写一路开关量输出（遥控分合、信号复归）

偏移地址“0000H”为继电器“分”位置，“0001H”为继电器“合”位置，“0002H”为继电器“信号复归”位置。“FF00”处于 ON 状态，“0000H”处于 OFF 状态。

例：主机控制地址为 01 的从机“合”。

主机发送的报文格式：

遥 控 分：01 05 00 00 FF 00 8C 3A

遥 控 合：01 05 00 01 FF 00 DD FA

遥控复归：01 05 00 02 FF 00 2D FA

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	05H	写控制命令
起始地址	2	分：0000H 合：0001H 复归：0002H	相对首地址 E300H 的起始地址
控制命令	2	FF00H/0000H	控制“合”/“分”/“复归”
CRC 码	2	DDFAH（合）（ON）	由主机计算出的 CRC 校验码

从机响应返回的报文格式：与主机发送的报文格式及数据内容完全相同。

3.4 功能码 06（HEX）：写单路寄存器

主机利用这个功能码把一个数据保存到从机的数据寄存器中。MODBUS 规约中寄存器是 16 位，且高位在前。

例：主机把 0065 保存到 E100H 的从机寄存器中（从机地址码为 01）。

主机发送：01 06 01 00 00 65 48 1D

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	06H	写单路寄存器
起始地址	2	0100H	相对首地址 E000H 的起始地址
数据	2	0065H	将数据写入 E100H 寄存器
CRC 校验码	2	481DH	由主机计算出的 CRC 校验码

从机响应返回的报文格式：与主机发送的报文格式及数据内容完全相同。

从机返回：01 06 01 00 00 65 48 1D

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	06H	写单路寄存器
起始地址	2	0100H	相对首地址 E000H 的起始地址
数据	2	0065H	将数据写入 E100H 寄存器
CRC 校验码	2	481DH	由主机计算出的 CRC 校验码

3.5 功能码 10（HEX）：写多路寄存器

主机利用这个功能码把多个数据保存到从机的数据寄存器中。MODBUS 规约中寄存器是 16 位，且高位在前。

例：主机把 0065 保存到 E100H 的从机寄存器中（从机地址码为 01）。

主机发送：01 10 01 00 00 01 02 00 65 76 BB

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	10H	写多路寄存器
起始地址	2	0100H	相对首地址 E000H 的起始地址
寄存器个数	2	0001H	字数＝1
寄存器长度	1	02H	字节数＝2
数据	2	0065H	将数据写入 E100H 寄存器
CRC 校验码	2	76BBH	由主机计算出的 CRC 校验码

从机响应返回的报文格式：与主机发送的报文格式及数据内容完全相同。

从机返回：01 10 01 00 00 01 00 35

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	10H	写多路寄存器
起始地址	2	0100H	相对首地址 E000H 的起始地址
寄存器个数	2	0001H	字数＝1
CRC 校验码	2	0035H	由主机计算出的 CRC 校验码

3.6 功能码 10（HEX）：写多路寄存器(时间参数)

主机利用这个功能码把多个数据保存到从机的数据寄存器中。MODBUS 规约中寄存器是 16 位，且高位在前。例：主机把 00 年 01 月 01 日 00 时 00 分 00 秒保存到 E180H、E181H、E182H、E183H、E184H、E185H 的从机寄存器中（从机地址码为 01）。

主机发送： 01 10 01 80 00 06 0C 00 00 00 01 00 01 00 00 00 00 00 00 27 B7

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	10H	写多路寄存器
起始地址	2	0180H	相对首地址 E000H 的起始地址
寄存器个数	2	0006H	字数＝6
寄存器长度	1	0CH	字节数＝12
年	2	0000H	年
月	2	0001H	月
日	2	0001H	日
时	2	0000H	时
分	2	0000H	分
秒	2	0000H	秒
CRC 校验码	2	27B7H	由主机计算出的 CRC 校验码

从机响应返回的报文格式：与主机发送的报文格式及数据内容完全相同。

从机返回：01 10 01 80 00 06 40 1F

主机发送	字节数	发送信息	备注
从机地址	1	01H	发送到地址为 01 的从机
功能码	1	10H	写多路寄存器
起始地址	2	0180H	相对首地址 E000H 的起始地址
寄存器个数	2	0006H	字数＝6
CRC 校验码	2	401FH	由主机计算出的 CRC 校验码

四、错误校验码（CRC 校验）：

使用MODBUS-RTU 模式，消息包括了一基于CRC 方法的错误检测域。CRC 域检测了整个消息的内容。

主机或从机可用校验码进行判别接收信息是否正确。由于电子噪声或一些其它干扰，信息在传输过程中有时会发生错误，错误校验码（CRC）可以检验主机或从机在通讯数据传送过程中的信息是否有误，错误的数据可以放弃，这样增加了系统的可靠性及通讯效率。

CRC 域是两个字节，包含一16 位的二进制值。它由传输设备计算后加入到消息中。接收设备重新计算收到消息的CRC，并与接收到的CRC 域中的值比较，如果两值不同，则有误。

在进行CRC 计算时只用8 个数据位，起始位及停止位和奇偶校验位都不参与CRC 计算。

CRC 码的计算方法是：

1．预置1 个全“1”的16 位CRC 寄存器（0xFFFF）（即全为1）；

2．把第一个8 位二进制数据（既信息帧的第一个字节）与16 位的CRC 寄存器的低8 位相异或（XOR），把结果放于CRC 寄存器的低8 位；

3．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；

4．如果移出位为1,则CRC 寄存器与预置的值A001（1010 0000 0000 0001）异或一下；如果移出位为0，则不进行。

5．重复8 次步骤3 和4，对整个8 位数据全部进行处理；

6．重复按步骤2 到5 的方法，进行通讯信息帧的下一个字节处理；

7．将该通讯信息帧所有字节按上述步骤计算完成后，得到16 位的CRC 值；

8．CRC 添加到消息中时，低字节先加入，然后高字节。

9．如果需要获得CRC计算详解请联系我公司软件技术部

五、各指令对应的地址范围

读开关量输入（01命令）起始地址：E200H

线圈起始地址	位定义	事件代码（HEX）
00H	设备停止状态	——
01H	设备运行状态	——
02H	备用	——
03H	备用	——
04H	备用	——
05H	备用	——
06H	备用	——
07H	备用	——
08H	备用	——
09H	备用	——
0AH	备用	——
0BH	备用	——
0CH	备用	——
0DH	备用	——
0EH	备用	——
0FH	备用	——
10H	失压	01H
11H	短路	02H
12H	断相	03H
13H	不平衡	04H
14H	选漏	08H
15H	漏电检测	06H
16H	过载	07H
17H	备用	——
18H	漏电闭锁（闭锁解除）	05H(09H)
19H	备用	——
1AH	备用	——
1BH	备用	——
1CH	备用	——
1DH	备用	——
1EH	备用	——
1FH	备用	——

读多路寄存器（03命令）地址范围：**E000H—E017H （模拟量）**

寄存器地址	数据类型	系数
E000H	UAB	1
E001H	UBC	1
E002H	UCA	1
E003H	IA	1
E004H	IB	1
E005H	IC	1
E006H	RG	10
E007H	P	1
E008H	Q	1
E009H	事件代码	1
E00AH	UAB(故障时刻动作值)	1
E00BH	IA(故障时刻动作值)	1
E00CH	IB(故障时刻动作值)	1
E00DH	IC(故障时刻动作值)	1
E00EH	备用	1
E00FH	备用	1
E010H	正有功电度量(高字)	1
E011H	正有功电度量(低字)	
E012H	负有功电度量(高字)	1
E013H	负有功电度量(低字)	
E014H	正无功电度量(高字)	1
E015H	正无功电度量(低字)	
E016H	负无功电度量(高字)	1
E017H	负无功电度量(低字)	

注：读出的数据应该乘以系数：Rg数据×10＝实际值

U的单位为“ V “ ，I的单位为“ A ”，P的单位为“ kW ”，Q的单位为“ kVar ”，Rg的单位为“ Ω ”，电度量的单位为“ 度 ”。

读多路寄存器（03命令）地址范围：**E100H—E113H （定值参数）**

寄存器地址	数据类型	系数
E100H	额定电流	1
E101H	短路倍数	0.01
E102H	欠压值	0.001
E103H	欠压时间	0.01
E104H	漏电阻值	0.01
E105H	漏电时间	0.01
E106H	参数1	0.01
E107H	参数2	0.01
E108H	瓦斯电闭锁延时	0.01
E109H	额定电压	1
E10AH	额定电流	1
E10BH	密码	1
E10CH	通讯地址	1
E10DH	末端短路	0.01
E10EH	相敏保护	0.01
E10FH	过压定值	0.01
E110H	过压时间	0.01
E111H	不平衡比例	0.01
E112H	漏电调整	0.01
E113H	分级闭锁	0.01

注：读出的数据应该乘以系数：以短路倍数为例：短路倍数=短路倍数实际值*0.01

写单路寄存器（06命令）、写多路寄存器（10命令）地址范围：**E100H—E113H （定值参数）**

寄存器地址	数据类型	系数
E100H	额定电流	1
E101H	短路倍数	100
E102H	欠压值	1000
E103H	欠压时间	100
E104H	漏电阻值	100
E105H	漏电时间	100
E106H	参数1	100
E107H	参数2	100
E108H	瓦斯电闭锁延时	100
E109H	额定电压	1
E10AH	额定电流	1
E10BH	密码	1
E10CH	通讯地址	1
E10DH	末端短路	100
E10EH	相敏保护	100
E10FH	过压定值	100
E110H	过压时间	100
E111H	不平衡比例	100
E112H	漏电调整	100
E113H	分级闭锁	100

注：读出的数据应该乘以系数：以短路倍数为例：短路倍数=短路倍数实际值*100

写开关量输出（05命令）地址范围：E300H

线圈起始地址	位定义
00H	分继电器
01H	合继电器
02H	设备复归

写多路寄存器（10命令）地址范围：E180H—E185H （时间参数）

寄存器地址	数据类型
E180H	年
E181H	月
E182H	日
E183H	时
E184H	分
E185H	秒

附件：CRC 校验算法程序（直接计算）

```
function CalcCRC16(str: string): Word;
procedure CRC16(Data: Byte);
var
i: Integer;
begin
Result := Result xor Data;
begin
if ((Result and 1)=1) then
Result := (Result shr 1) XOR $A001
else
Result := Result shr 1;
end;
end;
var
i: Integer;
begin
Result := $FFFF;
for i:=1 to Length(str) do
CRC16(Byte(str[i]));
end;
```