

开关量采集器（8 路）（MODBUS）通讯协议

VER: 1.02

1、概述

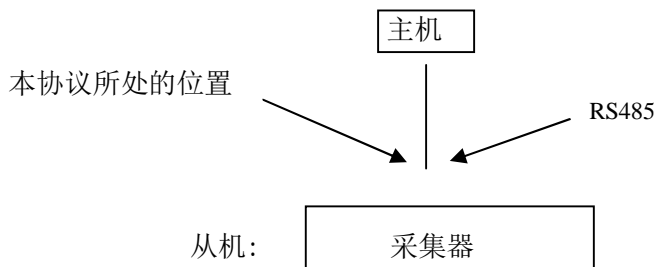
通信协议详细地描述了采集器的输入和输出命令、信息和数据，方便第三方使用和开发。

1.1 通信协议的作用

使信息和数据在上位机（主站）和采集器之间有效地传递，允许访问采集器的所有测量数据。

采集器可以实时采集现场变送器/传感器的模拟电压量, 具备一个 RS485 通讯口, 其功能和技术指标参见用户手册。

采集器采用标准的 MODBUS RTU 协议, 本协议规定了应用系统中主机与采集器之间, 在应用层的通信协议, 它在应用系统中所处的位置如下图所示:



1.2 物理接口:

连接上位机的通信口, 采用标准串行 RS485 通讯口, 使用可插拔端子。

信息传输方式为异步方式, 起始位 1 位, 数据位 8 位, 停止位 1 位, 无校验。

数据传输缺省速率为 9600b/s

2、MODBU RTU 通信协议详述

2.1 协议基本规则

以下规则确定在回路控制器和其他串行通信回路中设备的通信规则。

- 1) 所有回路通信应遵照主/从方式。在这种方式下, 信息和数据在单个主站和从站（监控设备）之间传递。
- 2) 主站将初始化和控制所有在通信回路上传递的信息。
- 3) 无论如何都不能从一个从站开始通信。
- 4) 所有环路上的通信都以“打包”方式发生。一个包裹就是一个简单的字符串（每个字符串 8 位），一个包裹中最多可含 255 个字节。组成这个包裹的字节构成标准异步串行数据, 并按 8 位数据位, 1 位停止位, 无校验位的方式传递。串行数据流由类似于 RS232C 中使用的设备产生。
- 5) 所有回路上的传送均分为两种打包方式:



A) 主/从传送

B) 从/主传送

6) 若主站或任何从站接收到含有未知命令的包裹，则该包裹将被忽略，且接收站不予响应。

2. 2 数据帧结构描述

每个数据帧组成如下：

RTU 模式

地址

功能代码

数据数量

数据 1

...

数据 n

CRC 16 位校验

CRC 校验码计算

CRC 校验码高字节是通讯信息帧的最后一个字节。

CRC 校验码由主机计算，放置于发送信息帧的尾部。从机再重新计算接收到信息的 CRC，比较计算得到的 CRC 与接收到的 CRC 是否一致，如果不一致，则表明出错。CRC 计算只用到了 8 个数据位，计算方法如下：

- ① 预置 1 个 16 位的寄存器为十六进制 FFFF（即全为 1），称此寄存器为 CRC 寄存器；
- ② 把第一个 8 位二进制数据（通讯信息帧的第 1 个字节）与 16 位 CRC 寄存器的低 8 位相异或，结果放于 CRC 寄存器；
- ③ 把 CRC 寄存器的内容右移一位（朝低位）并用 0 填补最高位，检查右移后的移出位；
- ④ 如果移出位为 0：重复第③步（再次右移一位）；
如果移出位为 1：CRC 寄存器与多项式 A001（1010 0000 0000 0001）进行异或；
- ⑤ 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；
- ⑥ 重复步骤②到步骤⑤，进行通讯信息帧下一个字节的处理；
- ⑦ 将该通讯信息帧所有字节（不包括 CRC 校验码高、低字节）按上述步骤计算完成后，CRC 寄存器内容即为 CRC 校验码。

示例：

```
unsigned short count_CRC(unsigned char *addr, int num)
{
    unsigned short CRC = 0xFFFF;
    int i;
    while (num--)
    {
        CRC ^= *addr++;
        for (i = 0; i < 8; i++)
        {
            CRC >>= 1;
            if (CRC & 1)
            {
                CRC ^= 0xA001;
            }
        }
    }
}
```



```

    }
}
return CRC;
}

```

3、传输格式

(1) 命令报文格式

读开关量输入状态：

地址	功能码	开关量状态起 始地址高位	开关量状态起 始地址低位	数据个数高 位	数据个数低 位	CRC 16 位校验
	02					低位在前

返回：

地址	功能码	字节长度	开关量状态数据	CRC16 位校验
	02			低位在前

读寄存器：

地址	功能码	数据起始地址 高位	数据起始地址 低位	数据个数高 位	数据个数低 位	CRC 16 位校验
	03					低位在前

返回：

地址	功能码	字节长度	数据 1 输入	数据 2 输入	...	CRC16 位校验
	03		高位在前			低位在前

写寄存器：

地址	功能码	寄存器地址高 位	寄存器地址低 位	待写入寄存 器的数据高 位	待写入寄存 器的数据低 位	CRC 16 位校验
	06					低位在前



返回:

地址	功能码	寄存器地址高位	寄存器地址低位	寄存器的数据高位	寄存器的数据低位	CRC 16位校验
	06					低位在前

(2)、异常应答返回

数据错误:

从站地址	功能码	异常码	CRC16 校验
	80H+原功能码	03	

操作失败:

从站地址	功能码	异常码	CRC16 校验
	80H+原功能码	04	

帧格式 (10 位)

起始位	D0	D1	D2	D3	D4	D5	D6	D7	停止位
-----	----	----	----	----	----	----	----	----	-----

4、采集器内部报文信息

读寄存器

功能码	属性	地址	字节数	项目	内容及举例	出厂设置	备注
03	系统参数						
	读写	0x0200	2	本机地址	1~127	01	0 为广播地址
	读写	0x0201	2	波特率	0: 1200 1: 2400 2: 4800 3: 9600 4: 19200	03	
	产品信息						
	只读	0x0210	2	产品年代	2014		
只读	0x0211	2	型号代码	1123: 32 路			



	只读	0x0212	2	生产年月	1307:		
	只读	0x0213	2	生产序号	0001:		
	只读	0x0214	2	硬件版本	A00:		
	只读	0x0215	2	软件版本	100(标准软件) 101(非标软件)		

写寄存器（修改地址及波特率）

功能码	属性	地址	字节数	项目	内容及举例	出厂设置	备注
系统参数							
06H	读写	0x0200	2	本机地址	1~128	01	0为广播地址
	读写	0x0201	2	波特率	0: 1200 1: 2400 2: 4800 3: 9600 4: 19200	03	

5、网络采样定时

上位机读取采集器数据每次间隔时间不小于 500ms, 推荐值 1s。

6、命令举例：

6.1 控制器的地址为 1，修改地址为 2

主机发送：

发送内容	字节数	发送数据	备注
从机地址	1	01H	从机地址
功能码	1	06H	写寄存器
寄存器地址 (高位在前)	2	0200H	0200H - 本机地址
待写入寄存器的数据 (高位在前)	2	0002H	待写入寄存器里的内容
CRC 校验 (低位在前)	2	09B3H	前面所有数据的 CRC 码

从机返回：

发送内容	字节数	发送数据	备注
从机地址	1	01H	从机地址



功能码	1	06H	写寄存器
寄存器地址 (高位在前)	2	0200H	0200H - 本机地址
寄存器的数据 (高位在前)	2	0002H	寄存器里的内容
CRC 校验 (低位在前)	2	09B3H	前面所有数据的 CRC 码

6.3 控制器的地址为 1，读开关量输入状态（使用 02 号指令）

主机发送：

发送内容	字节数	发送数据	备注
从机地址	1	01H	从机地址
功能码	1	02H	读开关量输出状态
开关量起始地址 (高位在前)	2	0000H	0000H - 开关量输入 1 的地址
开关量数量 (高位在前)	2	0008H	读取 8 个寄存器里的内容
CRC 校验 (低位在前)	2	79CCH	前面所有数据的 CRC 码

从机返回：

发送内容	字节数	发送数据	备注																																				
从机地址	1	01H	从机地址																																				
功能码	1	02H	读开关量输出状态																																				
返回字节长度	1	01H	返回 1 个字节																																				
开关量状态数据	1	01H	返回数据的每一位和开关量通道的状态依次对应，顺序为： <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>数据位</td> <td>BIT7</td> <td>BIT6</td> <td>BIT5</td> <td>BIT4</td> <td>BIT3</td> <td>BIT2</td> <td>BIT1</td> <td>BIT0</td> </tr> <tr> <td>数据</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>状态</td> <td>断开</td> <td>断开</td> <td>断开</td> <td>断开</td> <td>断开</td> <td>断开</td> <td>断开</td> <td>闭合</td> </tr> <tr> <td>通道</td> <td>8</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> </tr> </table>	数据位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	数据	0	0	0	0	0	0	0	1	状态	断开	断开	断开	断开	断开	断开	断开	闭合	通道	8	7	6	5	4	3	2	1
数据位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0																															
数据	0	0	0	0	0	0	0	1																															
状态	断开	断开	断开	断开	断开	断开	断开	闭合																															
通道	8	7	6	5	4	3	2	1																															
CRC 校验 (低位在前)	2	6048H	前面所有数据的 CRC 码																																				