--

RS485/RS232 通信协议 Ver1.0

目录

1. 适用对象说明 1
2. 通信协议说明 1
1）主要参数 1
2）寄存器列表 1
3）应用示例 2
例 1- 主机读基本信息 2
例 2- 主机读温度、流量 4
例 3- 主机读累积流量 5
例 4- 主机对累积流量清零 6
例 5- 主机设置流量（仅适用于控制器） 7
例 6- 主机设置阀门开度（仅适用于控制器） 9
例 7- 主机读地址（仅适用于控制器） 11
例 8- 主机对零点校准 11
例 9- 恢复出厂设置 12
3. CRC检验算法（C语言） 12

1. 适用对象说明

MFM/C-C300系列 及 500系列产品。

2. 通信协议说明

1）主要参数

通信接口	RS485/RS232半双工模式
波特率	9600
数据位	8
停止位	1
校验	无
通信数据格式	MODBUS RTU（地址默认为 1）

2）寄存器列表

寄存器地址（**16**进制）	寄存器内容	数据类型	访问类型	备注
0x0003	气体类型	无符号 16位整数	只读	
0x0004	满量程	无符号 16位整数	只读	
0x0005	流量单位	无符号 16位整数	只读	
0x0014	流量小数位个数	无符号 16位整数	只读	
0x0015	温度 16位	有符号 16位整数	只读	
0x0016	流量高 16位	无符号 32位整数	只读	
0x0017	流量低 16位			
0x0018	累积流量高 32位	无符号 64位整数	读/写	

0x0019				
0x001A	累积流量低 32位			
0x001B				
0x001C	累积流量单位 0：L；1：m^3	无符号 16位整数	只读	
0x001D	累积流量的天数	无符号 16位整数	只读	
0x001E	累积流量的小时数	无符号 16位整数	只读	
0x001F	累积流量的分钟数	无符号 16位整数	只读	
0x0020	累积流量的秒数	无符号 16位整数	只读	
0x0021	阀门控制方式 0：流量控制模式 3：阀门比例模式	无符号 16位整数	读/写	仅适用于控制器
0x0022	设定流量高 16位	无符号 32位整数 有效位数 0.001	读/写	
0x0023	设定流量低 16位			
0x0024	设定阀门开度	无符号 16位整数 有效位数 0.01	读/写	
0x0025	零点校准	无符号 16位整数	只写	

3）应用示例

例 **1-** 主机读基本信息

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x03	0x00	0x03	0x00	0x03	0xF5	0xCB

从机返回：

设备地址	功能码	数据字节数	气体类型高字节	气体类型低字节	满量程高字节	满量程低字节
0x01	0x03	0x06	0x00	0x0D	0x00	0x64
流量单位高字节	流量单位低字节	CRC校验低字节	CRC校验高字节			
0x00	0x0A	0xCD	0x6C			

数据解析

①气体类型：

0x000D（十六进制）= 13（十进制）

查阅下表代码，可知 13表示氮气。

代码（十进制）	气体类型	代码（十进制）	气体类型
1	He	13	N2
2	CO	15	O2
4	Ar	25	CO2
7	H2	28	CH4
8	Air		

②流量单位：

0x000A（十六进制）= 10（十进制）

查阅下表代码，可知 10表示 SCCM。

代码（十进制）	流量单位
10	SCCM
100	SLM

③满量程：

0x0064（十六进制）= 100（十进制）

可知该设备的满量程为 100（若流量单位是 SCCM，则满流量为 100 SCCM；若流量单位是 SLM，则满流量为 100 SLM）。

例 2- 主机读温度、流量

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x03	0x00	0x14	0x00	0x04	0x04	0x0D

从机返回：

设备地址	功能码	数据字节数	小数位高字节	小数位低字节	温度数据高字节	温度数据低字节
0x01	0x03	0x08	0x00	0x01	0x00	0xFD
流量数据字节1	流量数据字节2	流量数据字节 3	流量数据字节 4	CRC校验低字节	CRC校验高字节	
0x00	0x00	0x30	0x39	0x3C	0xD1	

数据解析

①温度：

0x00FD(十六进制) = 253（十进制）

除以 10，得到实际温度 = 253/10 = 25.3（℃）。

②流量数据的小数位数：

0x0001（十六进制）= 1（十进制）

从下表可知，1表示流量数据具有三位小数。

代码（十进制）	波特率
0	两位小数
1	三位小数
2	整数

3	一位小数

③流量：

0x00003039(十六进制) = 12345（十进制）

结合流量数据的小数位数，可是当前流量为 12345*0.001 = 12.345（若流量单位是 SCCM，则流量为 100 SCCM；若流量单位是 SLM，则流量为 100 SLM）。

例 3- 主机读累积流量

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x03	0x00	0x18	0x00	0x09	0x05	0xCB

从机返回：

设备地址	功能码	数据字节数	累积流量数据字节1	累积流量数据字节2	累积流量数据字节3	累积流量数据字节4	累积流量数据字节5
0x01	0x03	0x12	0x00	0x00	0x00	0x00	0x07
累积流量数据字节6	累积流量数据字节7	累积流量数据字节8	累积流量单位高字节	累积流量单位低字节	累积流量天数高字节	累积流量天数低字节	累积流量小时数高字节
0x5B	0xCD	0x15	0x00	0x00	0x27	0x10	0x00
累积流量小时数低字节	累积流量分钟数高字节	累积流量分钟数低字节	累积流量秒数高字节	累积流量秒数低字节	CRC校验低字节	CRC校验高字节	
0x0A	0x00	0x32	0x00	0x1E	0x6E	0xF9	

数据解析

①累积流量单位：

0x0000（十六进制）= 0（十进制）

查阅寄存器列表，可知 0表示 L。

②累积流量：

0x00000000075BCD15 (十六进制) = 123456789（十进制）

除以 1000，得到实际累积流量为 123456.789。（若流量单位是 L，则满流量为 123456.789 L；若流量单位是 m^3，则满流量为 123456.789 m^3）。

③累积时间：

天数

0x2710（十六进制）= 10000（十进制）

小时数

0x000A（十六进制）= 10（十进制）

分钟数

0x0032（十六进制）= 50（十进制）

秒数

0x001E（十六进制）= 30（十进制）

可知累积时间为 10000天 10小时 50分钟 30秒。

例 4- 主机对累积流量清零

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器个数高字节	寄存器个数低字节	修改数据的字节长度	数据 1 高字节	数据 1 低字节
0x01	0x10	0x00	0x18	0x00	0x04	0x08	0x00	0x00
数据 2 高字节	数据 2 低字节	数据 3 高字节	数据 3 低字节	数据 4 高字节	数据 4 低字节	CRC校验低字节	CRC校验高字节	
0x00	0x00	0x00	0x00	0x00	0x00	0x96	0x5A	

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址高字节	寄存器个数高字节	寄存器个数低字节	CRC校验低字节	CRC校验高字节
0x01	0x10	0x00	0x18	0x00	0x04	0x41	0xCD

例 5- 主机设置流量（仅适用于控制器）

主机设置流量的方法有两种：

方法 1：先设置控制方式，再设置流量；

方法 2（推荐）：控制方式和流量同时设置。

方法 1具体如下：

首先设置控制方式，向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x21	0x00	0x00	0xD9	0xC0

数据解析

0x0000(十六进制) = 0（十进制）

从寄存器列表可知，0表示将控制方式设置为“流量控制模式”。注：主机默认采用的控制方法为“流量控制模式”，如不需要更改，可略过此步骤，直接设置流量。如需多次设置流量且控制方法相同，此步骤执行一次即可，无需重复执行。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x21	0x00	0x00	0xD9	0xC0

控制方法确定后，进行流量设置，向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	修改数据的字节长度
0x01	0x10	0x00	0x22	0x00	0x02	0x04
数据 1高字节	数据 1低字节	数据 2高字节	数据 2低字节	CRC校验低字节	CRC校验高字节	
0x00	0x00	0x30	0x34	0x65	0xB9	

数据解析

0x00003034(十六进制) = 12340（十进制）

除以 1000，得到设置的目标流量值 = 12340/1000 = 12.34（若流量单位是 SCCM，则目标值为 12.34 SCCM；若流量单位是 SLM，则目标值为 12.34 SLM）。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x10	0x00	0x22	0x00	0x02	0xE1	0xC2

方法 2具体如下：

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	修改数据的字节长度	数据 1高字节
0x01	0x10	0x00	0x21	0x00	0x03	0x06	0x00
数据 1低字节	数据 2高字节	数据 2低字节	数据 3高字节	数据 3低字节	CRC校验低字节	CRC校验高字节	
0x00	0x00	0x00	0x30	0x34	0xA3	0xF8	

数据解析

①控制方式：

0x0000(十六进制) = 0（十进制）

从寄存器列表可知，0表示将控制方式设置为“流量控制模式”。

②流量设置：

0x00003034(十六进制) = 12340（十进制）

除以 1000，得到设置的目标流量值 = 12340/1000 = 12.34（若流量单位是 SCCM，则目标值为 12.34 SCCM；若流量单位是 SLM，则目标值为 12.34 SLM）。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x10	0x00	0x21	0x00	0x03	0xD0	0x02

例 6- 主机设置阀门开度（仅适用于控制器）

主机设置阀门开度的方法有两种：

方法 1：先设置控制方式，再设置比例；

方法 2（推荐）：控制方式和比例同时设置。

方法 1具体如下：

首先设置控制方式，向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x21	0x00	0x03	0x99	0xC1

数据解析

0x0003(十六进制) = 3（十进制）

从寄存器列表可知，3表示将控制方式设置为“阀门比例模式”。

注：如需连续多次设置不同的比例，此步骤执行一次即可，无需重复执行。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x21	0x00	0x03	0x99	0xC1

控制方法确定后，进行比例设置，向机发送：

设备地址	功能码	寄存器首地址高字	寄存器首地址低字	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节

		节	节				
0x01	0x06	0x00	0x24	0x04	0xD2	0x4B	0x5C

数据解析

0x04D2(十六进制) = 1234（十进制）

除以 100，得到设置的目标比例 = 1234/100 = 12.34（%）。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数据高字节	寄存器数据低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x24	0x04	0xD2	0x4B	0x5C

方法 2具体如下：

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	修改数据的字节长度	数据 1 高字节	数据 1 低字节
0x01	0x10	0x00	0x21	0x00	0x04	0x08	0x00	0x03
数据 2 高字节	数据 2 低字节	数据 3 高字节	数据 3 低字节	数据 4 高字节	数据 4 低字节	CRC校验低字节	CRC校验高字节	
0x00	0x00	0x00	0x00	0x04	0xD2	0x7B	0x9B	

数据解析

①控制方式：

0x0003(十六进制) = 3（十进制）

从寄存器列表可知，3表示将控制方式设置为“阀门比例模式”。

②比例设置：

0x04D2(十六进制) = 1234（十进制）

除以 100，得到设置的目标比例 = 1234/100 = 12.34（%）。

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0x01	0x10	0x00	0x21	0x00	0x04	0x91	0xC0

例 7- 主机读地址（仅适用于控制器）

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器长度高字节	寄存器长度低字节	CRC校验低字节	CRC校验高字节
0xFE	0x03	0x00	0x00	0x00	0x01	0x90	0x05

从机返回：

设备地址	功能码	数据字节数	地址高字节	地址低字节	CRC校验低字节	CRC校验高字节
0xFE	0x03	0x02	0x00	0x01	0x6D	0x90

数据解析

0x0001(十六进制) = 1（十进制）

可知当前地址为 1。

例 8- 主机对零点校准

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数值高字节	寄存器数值低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x25	0x00	0x01	0x59	0xC1

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数值高字节	寄存器数值低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x25	0x00	0x01	0x59	0xC1

例 **9-** 恢复出厂设置

向机发送：

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数值高字节	寄存器数值低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x25	0x00	0x02	0x19	0xC0

若从机返回以下数据，表示操作成功。

设备地址	功能码	寄存器首地址高字节	寄存器首地址低字节	寄存器数值高字节	寄存器数值低字节	CRC校验低字节	CRC校验高字节
0x01	0x06	0x00	0x25	0x00	0x02	0x19	0xC0

3. CRC 检验算法（C 语言）

方法一**:**

```
unsigned short Crc16_Check(unsigned char *Pushdata,unsigned char length)
// 返回的 16位数据高位在前，低位在后.
{ unsigned short Reg_Crc = 0xffff;
unsigned char i,j; for(i = 0; i < length;
i++)
    { Reg_Crc ^= *Pushdata++;
    for(j = 0; j < 8;j++)
        {
            if(Reg_Crc & 0x0001)
            {
                Reg_Crc = Reg_Crc >> 1^0xA001;
            }
            else
            {
```

```
                Reg_Crc >>= 1;
            }
    } } return
    (Reg_Crc);

}
```

方法 **2:**

```
const uint16_t crctalbeabs[] = { 0x0000, 0xCC01, 0xD801, 0x1400,
0xF001, 0x3C00, 0x2800, 0xE401, 0xA001, 0x6C00, 0x7800, 0xB401,
0x5000, 0x9C01, 0x8801, 0x4400 };

uint16_t crc16tablefast(uint8_t *ptr, uint16_t len)
// 返回的 16位数据高位在前，低位在后.
{ uint16_t crc = 0xffff; uint16_t i; uint8_t ch; for (i = 0; i < len;
i++) { ch = *ptr++; crc = crctalbeabs[(ch ^ crc) & 15] ^ (crc
>> 4); crc = crctalbeabs[((ch >> 4) ^ crc) & 15] ^ (crc >> 4);




    }

    return crc;
}
```