

## 压力传感器表通讯协议

压力传感器表使用 Modbus RTU 通信协议，进行 RS485 半双工通信，读功能号 0x03，写功能号 0x10，采用 16 位 CRC 校验，仪表对校验错误不返回，数据包间隔时间为 30ms，如接收数据包间隔时间超过 30ms 则需重新发送命令。

### 数据帧格式：

起始位	数据位	停止位	校验位
1	8	1	无

### 通信异常处理：

异常应答时，将功能号的最高位置 1。例如：主机请求功能号是 0x04,则从机返回的功能号对应项为 0x84。

### 错误类型码：

0x01---功能码非法：仪表不支持接收到的功能号。

0x02---数据位置非法：主机指定的数据位置超出仪表的范围。

0x03---数据值非法：主机发送的数据值超出超出仪表对应的数据范围。

## 一、读多寄存器

例：主机读取满量程

满量程的地址编码是 0x0001，报警值采用 32 浮点数（占用 4 个字节），占用 2 个数据寄存器。十进制浮点数 200.0 的 IEEE-754 标准 16 进制内存码为 0x00004843

主机请求（读多寄存器）							
1	2	3	4	5	6	7	8
表地址	功能号	起始地址 高位	起始地址 低位	数据字长 高位	数据字长 低位	CRC 码 的低位	CRC 码 的高位
0x01	0x03	0x00	0x01	0x00	0x02	0x95	0xCB

从机正常应答（读多寄存器）								
1	2	3	4	5	6	7	8	9
表地址	功能号	数据字 节数	数据 1 高位	数据 1 低位	数据 2 高位	数据 2 低位	CRC 码 的低位	CRC 码 的高位
0x01	0x03	0x04	0x00	0x00	0x48	0x43	0x8D	0xC2

功能号异常应答：（例如主机请求功能号为 0x04）

从机异常应答（读多寄存器）				
1	2	3	8	9
表地址	功能号	错误码	CRC 码 的低位	CRC 码 的高位
0x01	0x84	0x01	0x82	0xC0

## 二、写多路寄存器

例：主机写满量程 220

满量程的地址编码是 0x0001，回差值采用 32 位浮点数（4 字节），占用 2 个数据寄存器。十进制浮点数 220 的 IEEE-754 标准 16 进制内存码为 0x00005C43

主机请求（写多寄存器）												
1	2	3	4	5	6	7	8	9	10	11	12	13
表地址	功能号	起始地址高位	起始地址低位	数据字长高位	数据字长低位	数据字节长度	数据1高位	数据1低位	数据2高位	数据2低位	CRC低位	CRC高位
0x01	0x10	0x00	0x01	0x00	0x02	0x04	0x00	0x00	0x5C	0x43	0x4A	0x92
从机正常应答（写多寄存器）												
1	2	3	4	5	6	7	8					
表地址	功能号	起始地址高8位	起始地址低8位	数据字长高位	数据字长低位	CRC码的低位	CRC码的高位					
0x01	0x10	0x00	0x01	0x00	0x02	0x10	0x08					

### 压力传感器表相关参数地址映射表

序号	地址映射	变量名称	字长	取值范围	读写允许	备注
0	0x0000	零点屏蔽值	2	-1999~9999	R/W	
1	0x0001	仪表零点 FL1	2	-1999~9999	R/W	
2	0x0002	仪表满量程 FH1	2	-1999~9999	R/W	
3	0x0003	报警值设置		-1999~9999	R/W	
4	0x0004	报警回差设置		-1999~9999	R/W	
5	0x0005	实际测量值	2	0000~9999	R	
保留						
6	0x0050	零点屏蔽权限开放	1	0~1	R/W	
7	0x0051	表地址 Add	1	0~255	R/W	
8	0x0052	波特率 bAd	1	0~1	R/W	注①
9	0x0053	小数点设置	1	0~3	R/W	
10	0x0054	采样速度设置	1	0~9	R/W	
11	0x0055	报警模式设置	1	0~1	R/W	
保留						

R:只读; R/W:可读写.

注①: 波特率

通信数值	0	1
菜单显示	4.8	9.6

#### 16 位 CRC 校验码获取程序

```
unsigned int G_CRC(unsigned char *p, uchar num)
```

```
{
    unsigned i,j;
    unsigned int Crc = 0xFFFF;
    for(i=0; i<num; i++)
    {
```

```

        Crc ^= (unsigned int)(p[i]);
        for(j=0; j<8; j++)
        {
            if(Crc & 1){Crc >>= 1; Crc ^= 0xA001; }
            else
                Crc >>= 1;
        }
    }
    return Crc;
}

```

#### 4 字节字符内码转化成十进制浮点程序

```

float  Ch_To_Float(unsigned char *ch)
{
    float result;
    unsigned char *p;
    p=(unsigned char *)&result;
    *p=*ch; *(p+1)=(ch+1); *(p+2)=(ch+2); *(p+3)=(ch+3);
    return result;
}

```

#### 十进制浮点数按转化成 4 字节字符内码程序

```

void Float_To_Char(float value, unsigned char *ch)
{
    unsigned char *p;
    p=(unsigned char *)&value;
    *ch=*p; *(ch+1)=(p+1); *(ch+2)=(p+2); *(ch+3)=(p+3);
}

```