

# MODBUS 协议 用户手册

Ver: E

2018-07-05

## 目录

<b>1</b>	<b>数据传输模式</b> .....	<b>4</b>
<b>2</b>	<b>寄存器和数据类型</b> .....	<b>5</b>
2.1	COIL .....	5
2.2	FLOAT .....	5
2.3	INT.....	5
2.4	LONG .....	5
<b>3</b>	<b>数据帧格式定义</b> .....	<b>6</b>
3.1	CMD=0x03(读 1 个或多个寄存器).....	6
3.2	CMD=0x05(写 COIL 变量).....	7
3.3	CMD=0x06(写单个寄存器).....	8
3.3	CMD=0x10(写多个寄存器).....	9
3.4	故障返回帧 .....	10
<b>4</b>	<b>数据帧校验算法</b> .....	<b>11</b>
4.1	LRC 校验 .....	11
4.2	CRC16 校验 .....	12
<b>5</b>	<b>常用测试软件举例</b> .....	<b>14</b>
5.1	MODSCAN32 .....	14
5.2	串口调试助手 .....	15
<b>6</b>	<b>仪表变量地址定义</b> .....	<b>16</b>
6.1	COIL 类型的变量地址.....	16
6.2	INT 类型的变量地址 .....	16
6.3	LONG 类型的变量地址 .....	17
6.4	FLOAT 类型的变量地址.....	18
6.5	可以连读的变量地址 (V10 版本以上) .....	19
6.6	可以连读的变量地址 (V12 版本以上) .....	22
6.7	可以连读的变量地址 (V13 版本以上) .....	23
<b>7</b>	<b>附录 1: 常数表: 故障码</b> .....	<b>24</b>
<b>8</b>	<b>附录 2: 常数表: 流量单位</b> .....	<b>25</b>
<b>9</b>	<b>附录 3: 报警定义</b> .....	<b>30</b>

## 重要提示:

- 1、 本手册中的通讯协议版本号只是用来表示在基本通讯功能基础上增加的功能，基本通讯功能永远有效，不随版本的升级而改变
- 2、 一定注意本手册中的红色和加粗字体的相关内容，这些内容对使用很关键

## 1 数据传输模式

MODBUS 采用 RTU 和 ASCII 两种方式进行数据传输。RTU 模式下，采用 8bit 二进制字符，ASCII 模式下采用 7bit ASCII 字符。将 RTU 模式下的一个字节的 4 位和高 4 位分开，变成 2 个字节，这样就是 ASCII 模式下传输的字节。比如 RTU 模式下的数据 0x1A，那么 ASCII 模式下就是 0x31 0x41 两个字节，所以 ASCII 模式下的帧长度为 RTU 模式下的 1 倍。

RTU 传输模式的数据帧采用 CRC 校验，ASCII 模式采用 LRC 校验。

下表总结了两种传输模式的区别：

传输模式	ASCII (7 bit)	RTU (8 bit)
编码格式	ASCII 码 ('0'-'9' 'A'-'F')	8bit 二进制 (0x00 - 0xff)
起始位	1	1
数据位	7、8	8
校验位	无、奇、偶	无、奇、偶
停止位	1、2	1、2
帧校验	LRC	CRC16

## 2 寄存器和数据类型

下表列举了几种寄存器和数据类型

寄存器类型	数据长度	寄存器数量	描述
COIL	1 bit	-	布尔变量(ON OFF)
FLOAT	32 bit	2	32 位浮点数(IEEE754 格式)
INT	16 bit	1	无符号整型(0x0 – 0xFFFF)
LONG	32 bit	2	无符号长整型(0x0 – 0xFFFFFFFF)

### 2.1 COIL

布尔变量 0xFF00 -> ON 0x0000 -> OFF

### 2.2 FLOAT

使用 2 个寄存器存储单精度 IEEE754 格式的浮点数。

每个浮点数包含 4 个字节，具体定义如下：

SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM

S:符号位 0->正 1->负

E:阶码

M:尾数的小数部分

例如：0xC1480000 = -12.5

### 2.3 INT

使用 1 个寄存器存储无符号整型数。

例如：0x0025 = 37 0x1234 = 4660

### 2.4 LONG

使用 2 个寄存器存储无符号长整型数。

例如：0x12345678 = 305419896

### 3 数据帧格式定义

#### 3.1 CMD=0x03(读 1 个或多个寄存器)

本例数据为读取瞬时流量的数据帧，仪表地址=1。

**注：瞬时流量的寄存器起始地址=0x0253，但数据帧中的寄存器起始地址必须是 0x0253-0x0001 = 0x0252**

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	03	30 33
寄存器起始地址高字节	02	30 32
寄存器起始地址低字节	52	35 32
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
帧校验	64 62	41 36
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	03	30 33
数据长度	04	30 34
寄存器 0x0253 的数据高字节	C1	43 31
寄存器 0x0253 的数据低字节	48	34 38
寄存器 0x0254 的数据高字节	00	30 30
寄存器 0x0254 的数据低字节	00	30 30
帧校验	47 D9	45 46
包尾	NONE	0D 0A

本应答帧返回的 IEEE754 格式的瞬时流量数据 C1 48 00 00 = -12.5

### 3.2 CMD=0x05(写 COIL 变量)

本例数据为清除累计流量的数据帧，仪表地址=1。

注：清累计流量的寄存器起始地址=0x0003，但数据帧中的寄存器起始地址必须是 0x0003-0x0001 = 0x0002

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	05	30 35
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	02	30 32
COIL 变量高字节	FF	46 46
COIL 变量低字节	00	30 30
帧校验	2D FA	46 39
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	05	30 35
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	02	30 32
COIL 变量高字节	FF	46 46
COIL 变量低字节	00	30 30
帧校验	2D FA	46 39
包尾	NONE	0D 0A

### 3.3 CMD=0x06(写单个寄存器)

本例数据为写流量单位=m3/h 的数据帧，仪表地址=1。

**注：流量单位的寄存器起始地址=0x0042，但数据帧中的寄存器起始地址必须是 0x0042-0x0001 = 0x0041**

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	06	30 36
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	41	34 31
变量高字节	00	30 30
变量低字节	13	31 33
帧校验	98 13	41 35
包尾	NONE	0D 0A

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	06	30 36
寄存器起始地址高字节	00	30 30
寄存器起始地址低字节	41	34 31
变量高字节	00	30 30
变量低字节	13	31 33
帧校验	98 13	41 35
包尾	NONE	0D 0A



### 3.3 CMD=0x10(写多个寄存器)

本例数据为写阻尼时间=3s 的数据帧，仪表地址=1。

**注：阻尼时间的寄存器起始地址=0x0189，但数据帧中的寄存器起始地址必须是 0x0189-0x0001 = 0x0188**

请求帧：上位机->仪表

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	10	31 30
寄存器起始地址高字节	01	30 31
寄存器起始地址低字节	88	38 38
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
数据长度	04	30 34
写入寄存器 0x0189 的高字节	40	34 30
写入寄存器 0x0189 的低字节	40	34 30
写入寄存器 0x018A 的高字节	00	30 30
写入寄存器 0x018A 的低字节	00	30 30
帧校验	E3 ED	45 38
包尾	NONE	0D 0A

本例中的 4 个数据字节为 IEEE754 格式的浮点数 40 40 00 00 = 3.0

应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	10	31 30
寄存器起始地址高字节	01	30 31
寄存器起始地址低字节	88	38 38
寄存器数量高字节	00	30 30
寄存器数量低字节	02	30 32
帧校验	C0 1E	36 43
包尾	NONE	0D 0A

### 3.4 故障返回帧

对于不能正确完成操作的请求帧，仪表将返回一个报告故障代码的返回帧，以报告不能完成操作的具体原因。

比如将流量单位设置成 Hz，由于仪表不能支持 Hz 这个流量单位，所以将返回如下的故障应答帧。

故障应答帧：仪表->上位机

数据场名称	RTU 示例数据(HEX)	ASC 示例数据(HEX)
包头	NONE	3A
仪表地址	01	30 31
功能码	86	38 36
故障码	43	34 33
帧校验	03 91	39 31
包尾	NONE	0D 0A

注：1. 故障返回帧中的功能码=请求帧的功能码+0x80

2. 具体的故障代码请参考 [附录 1：常数表：故障码](#)

## 4 数据帧校验算法

### 4.1 LRC 校验

// LRC 校验范围：从“仪表地址”到 LRC 帧校验码的前一个字节

```
void LRC(unsigned char *buf, unsigned int len)
```

```
{
```

```
    unsigned int i;
```

```
    LRC = 0;
```

```
    for (i=0; i<len; i++)
```

```
    {
```

```
        LRC += buf[i];
```

```
    }
```

```
    LRC = 0xff - LRC;
```

```
    LRC++;
```

```
}
```

## 4.2 CRC16 校验

```
const unsigned char TAB_CRC_H[] = {  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40  
};
```

```
const unsigned char TAB_CRC_L[] = {  
    0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,0x04,  
    0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,0x08,0xC8,  
    0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,0x1D,0x1C,0xDC,  
    0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,0x11,0xD1,0xD0,0x10,  
    0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,0x37,0xF5,0x35,0x34,0xF4,  
    0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,0x3B,0xFB,0x39,0xF9,0xF8,0x38,  
    0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,  
    0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,  
};
```

```

0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,
0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,
0x78,0xB8,0xB9,0x79,0xBB,0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,
0xB4,0x74,0x75,0xB5,0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,
0x50,0x90,0x91,0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,
0x9C,0x5C,0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,
0x88,0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,0x40
};

```

// **CRC** 校验范围：从“仪表地址”到 **CRC** 帧校验码的前一个字节

```

void CRC(unsigned char *buf, unsigned int len)
{
    unsigned int i;
    unsigned char CRC_H , CRC_L , index, ch;
    CRC_H = 0xff;
    CRC_L = 0xff;
    for (i=0; i<len; i++)
    {
        ch = buf[i];
        index = CRC_H ^ ch;
        CRC_H = CRC_L ^ TAB_CRC_H[index];
        CRC_L = TAB_CRC_L[index];
    }
}

```

## 5 常用测试软件举例

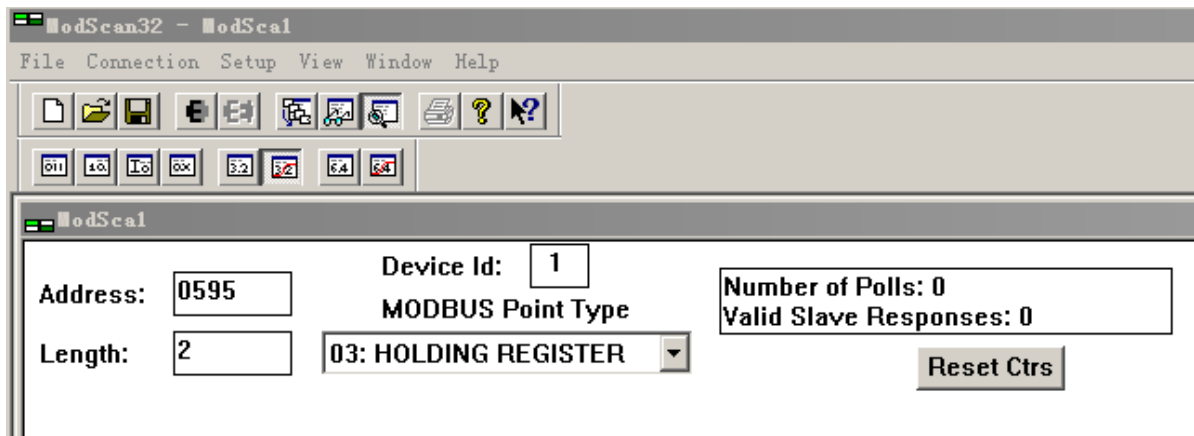
以下的举例均以读取瞬时流量(地址=0x0253)为例

通讯参数：波特率 9600 数据位=8 校验=无 停止位=1

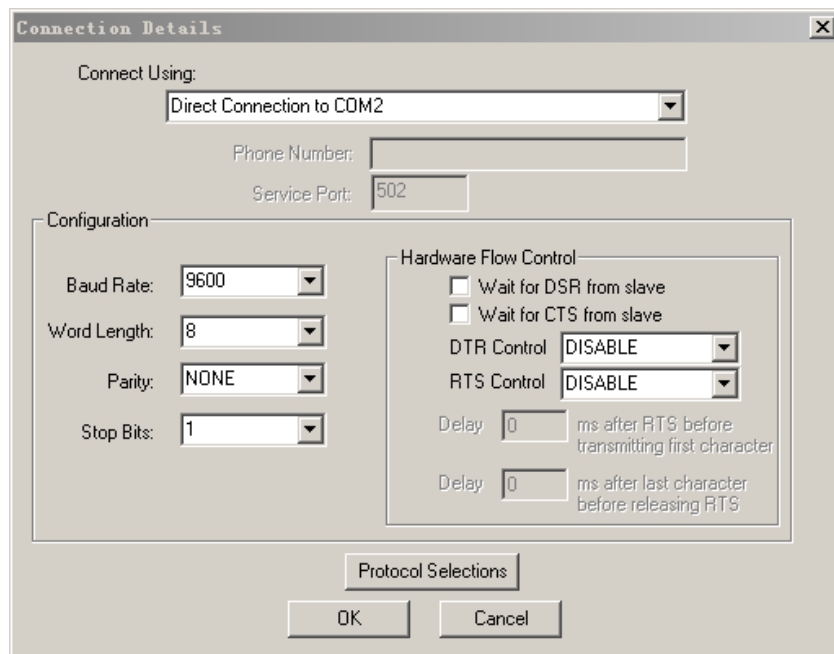
仪表地址 = 001 通讯协议=Modbus RTU

### 5.1 ModScan32

第 1 步：按下图设置

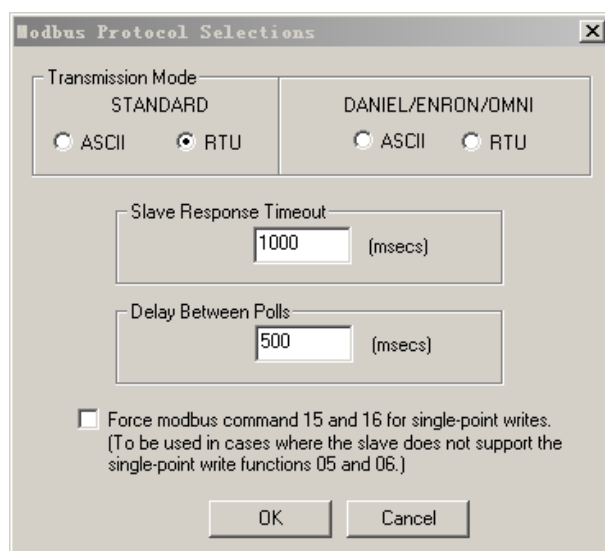


第 2 步：菜单 Connection - Connect



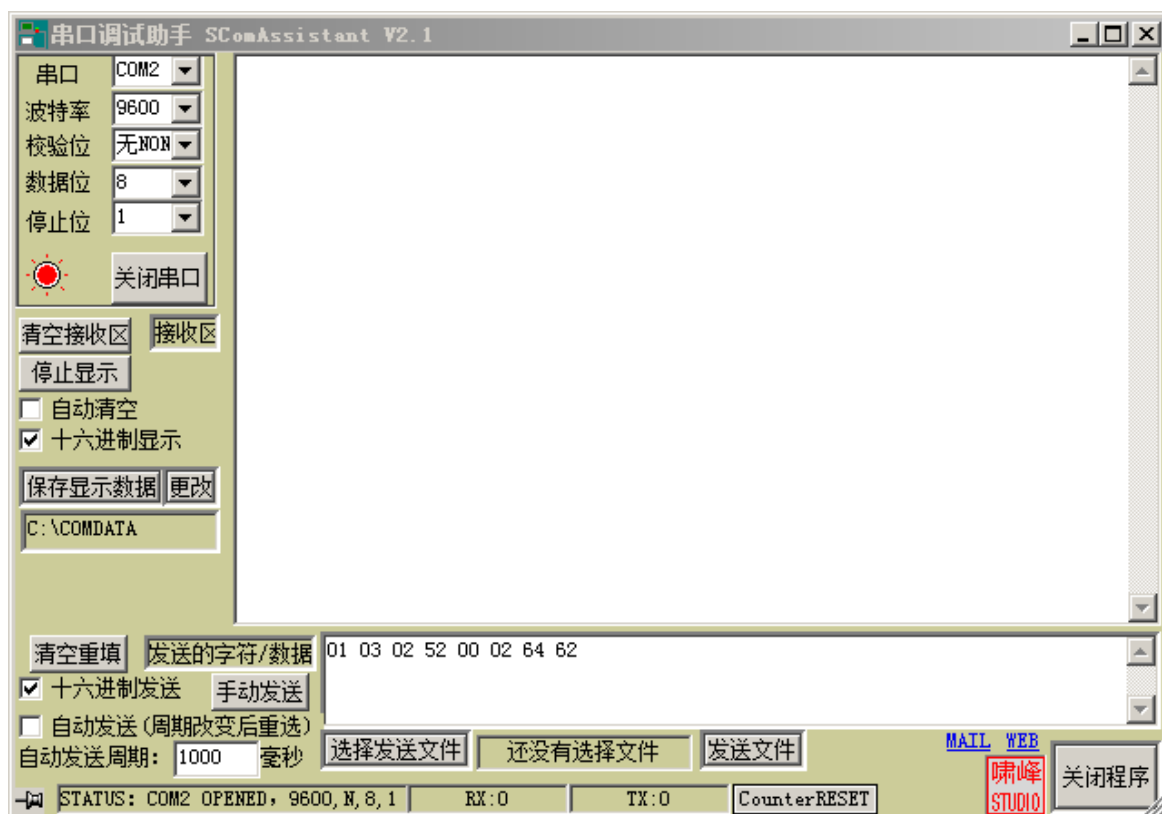
COM 口选择当前实际连接的串口

### 第 3 步: 点击 Protocol Selections



设置好后点 OK 返回

## 5.2 串口调试助手



COM 口选择当前实际连接的串口

点手动发送，观察返回数据

## 6 仪表变量地址定义

以下为仪表支持数据变量信息列表

### 6.1 COIL 类型的变量地址(所有版本均支持)

COIL 类型 (以下变量只能单个读写) !!!	寄存器地址	寄存器 长度	读 指令	写 指令
清累计	<b>0003(16 进制)</b> <b>3(10 进制)</b>	---	---	<a href="#">05</a>

### 6.2 INT 类型的变量地址(所有版本均支持)

INT 类型 (以下变量只能单个读写) !!!	寄存器地址	寄存器 长度	读 指令	写 指令
流量单位( <a href="#">附录 2:常数表流量单位</a> )	<b>0042(16 进制)</b> <b>66(10 进制)</b>	0001	<a href="#">03</a>	<a href="#">06</a>
累计单位( <a href="#">附录 2:常数表流量单位</a> )	<b>0046(16 进制)</b> <b>70(10 进制)</b>	0001	<a href="#">03</a>	<a href="#">06</a>
功率单位( <a href="#">附录 2:常数表流量单位</a> )	<b>6003(16 进制)</b> <b>24579(10 进制)</b>	0001	<a href="#">03</a>	<a href="#">06</a>
热能单位( <a href="#">附录 2:常数表流量单位</a> )	<b>6004(16 进制)</b> <b>24580(10 进制)</b>	0001	<a href="#">03</a>	<a href="#">06</a>
报警( <a href="#">附录 3::报警定义</a> )	<b>0419(16 进制)</b> <b>1049(10 进制)</b>	0001	<a href="#">03</a>	---



### 6.3 LONG 类型的变量地址(所有版本均支持)

LONG 类型 (以下变量只能单个读写) !!!	寄存器地址	寄存器 长度	读 指令	写 指令
扩展正向累计	0309(16 进制)    777(10 进制)	0002	<a href="#">03</a>	---
正向累计	0311(16 进制)    785(10 进制)	0002	<a href="#">03</a>	---
扩展反向累计	0313(16 进制)    787(10 进制)	0002	<a href="#">03</a>	---
反向累计	0315(16 进制)    789(10 进制)	0002	<a href="#">03</a>	---
扩展正向累计热量	0317 (16 进制)    791(10 进制)	0002	<a href="#">03</a>	---
正向累计热量	0319 (16 进制)    793(10 进制)	0002	<a href="#">03</a>	---
扩展反向累计热量	0321 (16 进制)    801(10 进制)	0002	<a href="#">03</a>	---
反向累计热量	0323 (16 进制)    803(10 进制)	0002	<a href="#">03</a>	---

注：累计量的计算方法如下

假设读出的“扩展正向累计” = 2，“正向累计” = 1234

则总的正向累计流量 =  $2 * 10000000 + 1234 = 20001234$

## 6.4 FLOAT 类型的变量地址(所有版本均支持)

<b>FLOAT 类型</b> (以下变量只能单个读写) !!!	寄存器地址	寄存器 长度	读 指令	写 指令
主变量(瞬时流量)	0253( <b>16 进制</b> ) <b>595(10 进制)</b>	0002	<a href="#">03</a>	----
瞬时功率	2000 ( <b>16 进制</b> ) <b>8192(10 进制)</b>	0002	<a href="#">03</a>	----
入口温度	2002 ( <b>16 进制</b> ) <b>8194(10 进制)</b>	0002	<a href="#">03</a>	----
出口温度	2004 ( <b>16 进制</b> ) <b>8196(10 进制)</b>	0002	<a href="#">03</a>	----
阻尼时间 s	0189( <b>16 进制</b> ) <b>393(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
小信号切除%	0197( <b>16 进制</b> ) <b>407(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
刻度流量	0209( <b>16 进制</b> ) <b>521(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
4-20mA 电流测试(mA)	0143( <b>16 进制</b> ) <b>323(10 进制)</b>	0002	---	<a href="#">10</a>
输出电流值 mA	0203( <b>16 进制</b> ) <b>515(10 进制)</b>	0002	<a href="#">03</a>	---
输出频率值 Hz	0229( <b>16 进制</b> ) <b>553(10 进制)</b>	0002	<a href="#">03</a>	---
频率上限 Hz	0223( <b>16 进制</b> ) <b>547(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
脉冲当量 L/p	1103( <b>16 进制</b> ) <b>4355(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
脉冲宽度 ms	0227( <b>16 进制</b> ) <b>551(10 进制)</b>	0002	<a href="#">03</a>	<a href="#">10</a>
压力	153B( <b>16 进制</b> ) <b>5435(10 进制)</b>	0002	<a href="#">03</a>	----

## 6.5 可以连读的变量地址（V10 版本及以上）

以下变量可以连续读 (通讯协议 V10 版本以上支持!!!)	寄存器地址	寄存器 长度	读 指令	写 指令
主变量(瞬时流量) float 与地址 <b>0253(16 进制)</b> 的数据相同	<b>0500(16 进制)</b> <b>1280(10 进制)</b>	0002	<a href="#">03</a>	---
扩展正向累计 unsigned long 与地址 <b>0309(16 进制)</b> 的数据相同	<b>0502(16 进制)</b> <b>1282(10 进制)</b>	0002	<a href="#">03</a>	---
正向累计 unsigned long 与地址 <b>0311(16 进制)</b> 的数据相同	<b>0504(16 进制)</b> <b>1284(10 进制)</b>	0002	<a href="#">03</a>	---
扩展反向累计 unsigned long 与地址 <b>0313(16 进制)</b> 的数据相同	<b>0506(16 进制)</b> <b>1286(10 进制)</b>	0002	<a href="#">03</a>	---
反向累计 unsigned long 与地址 <b>0315(16 进制)</b> 的数据相同	<b>0508(16 进制)</b> <b>1288(10 进制)</b>	0002	<a href="#">03</a>	---

注：累计量的计算方法如下

假设读出的“扩展正向累计” = 2，“正向累计” = 1234

则总的正向累计流量 =  $2 * 10000000 + 1234 = 20001234$

以下变量可以连续读 (通讯协议 V10 版本以上支持!!!)		寄存器地址	寄存器 长度	读 指令	写 指令
主变量(瞬时流量)	float	0510(16 进制)    1296(10 进制)	0002	<a href="#">03</a>	---
与地址 0253(16 进制)的数据相同					
扩展正向累计	unsigned long	0512(16 进制)    1298(10 进制)	0002	<a href="#">03</a>	---
正向累计	unsigned long	0514(16 进制)    1300(10 进制)	0002	<a href="#">03</a>	---
扩展反向累计	unsigned long	0516(16 进制)    1302(10 进制)	0002	<a href="#">03</a>	---
反向累计	unsigned long	0518(16 进制)    1304(10 进制)	0002	<a href="#">03</a>	---

此处的累计量读出的数据是表头显示值\*1000 的结果。

假设表头显示的正向累计量是：**20001.234 m3**

那么读出的数据是：“扩展正向累计” = 2，“正向累计” = 1234

则读出总的正向累计流量 =  $2 * 10000000 + 1234 = 20001234$

因为读出的数据是表头实际显示值的 1000 倍，

所以实际的正向累计流量 =  $20001234 / 1000.0 = 20001.234 \text{ m3}$

以下变量可以连续读 (通讯协议 V10 版本以上支持!!!)		寄存器地址	寄存器 长度	读 指令	写 指令
主变量(瞬时流量)	float	0520(16 进制)    1312(10 进制)	0002	<a href="#">03</a>	---
与地址 0253(16 进制)的数据相同					
瞬时功率	float	0522(16 进制)    1314(10 进制)	0002	<a href="#">03</a>	---
与地址 2000(16 进制)的数据相同					
入口温度 0C	float	0524(16 进制)    1316(10 进制)	0002	<a href="#">03</a>	---
与地址 2002(16 进制)的数据相同					
出口温度 0C	float	0526(16 进制)    1318(10 进制)	0002	<a href="#">03</a>	---
与地址 2004(16 进制)的数据相同					
扩展正向累计	unsigned long	0528(16 进制)    1320(10 进制)	0002	<a href="#">03</a>	---
与地址 0309(16 进制)的数据相同					
正向累计	unsigned long	052A(16 进制)    1322(10 进制)	0002	<a href="#">03</a>	---
与地址 0311(16 进制)的数据相同					
扩展反向累计	unsigned long	052C(16 进制)    1324(10 进制)	0002	<a href="#">03</a>	---
与地址 0313(16 进制)的数据相同					
反向累计	unsigned long	052E(16 进制)    1326(10 进制)	0002	<a href="#">03</a>	---
与地址 0315(16 进制)的数据相同					
扩展正向热能累计	unsigned long	0530(16 进制)    1328(10 进制)	0002	<a href="#">03</a>	---
与地址 0317(16 进制)的数据相同					
正向热能累计	unsigned long	0532(16 进制)    1330(10 进制)	0002	<a href="#">03</a>	---
与地址 0319(16 进制)的数据相同					
扩展反向热能累计	unsigned long	0534(16 进制)    1332(10 进制)	0002	<a href="#">03</a>	---
与地址 0321(16 进制)的数据相同					
反向热能累计	unsigned long	0536(16 进制)    1334(10 进制)	0002	<a href="#">03</a>	---
与地址 0323(16 进制)的数据相同					

注：累计量的计算方法如下

假设读出的“扩展正向累计” = 2，“正向累计” = 1234

则总的正向累计流量 = 2 \* 10000000 + 1234 = 20001234

## 6.6 可以连读的变量地址（V12 版本及以上）

以下变量可以连续读 (通讯协议 V12 版本以上支持!!!)		寄存器地址		寄存器 长度	读 指令	写 指令
正向累计	float	0550(16 进制)	1360(10 进制)	0002	03	---
反向累计	float	0552(16 进制)	1362(10 进制)	0002	03	---
正向热能累计	float	0554(16 进制)	1364(10 进制)	0002	03	---
反向热能累计	float	0556(16 进制)	1366(10 进制)	0002	03	---

以下变量可以连续读 (通讯协议 V12 版本以上支持!!!)		寄存器地址		寄存器 长度	读 指令	写 指令
正向累计	double	0560(16 进制)	1376(10 进制)	0004	03	---
反向累计	double	0564(16 进制)	1380(10 进制)	0004	03	---
正向热能累计	double	0568(16 进制)	1384(10 进制)	0004	03	---
反向热能累计	double	056C(16 进制)	1388(10 进制)	0004	03	---

## 6.7 可以连读的变量地址（V13 版本及以上）

以下变量可以连续读 (通讯协议 V13 版本以上支持!!!)		寄存器地址		寄存器 长度	读 指令	写 指令
主变量(瞬时流量)	float	054E(16 进制)	1358(10 进制)	0002	03	---
正向累计	float	0550(16 进制)	1360(10 进制)	0002	03	---
反向累计	float	0552(16 进制)	1362(10 进制)	0002	03	---
正向热能累计	float	0554(16 进制)	1364(10 进制)	0002	03	---
反向热能累计	float	0556(16 进制)	1366(10 进制)	0002	03	---
瞬时功率	float	0558(16 进制)	1368(10 进制)	0002	03	---
入口温度 0C	float	055A(16 进制)	1370(10 进制)	0002	03	---
出口温度 0C	float	055C(16 进制)	1372(10 进制)	0002	03	---

## 7 附录 1: 常数表: 故障码

0x01:	无效指令码
0x02:	无效的寄存器地址
0x30:	参数超上限
0x31:	参数超下限
0x32:	参数选择项错误
0x40:	无效的寄存器长度
0x41:	寄存器不支持当前的指令码
0x42:	寄存器未指定功能
0x43:	流量单位不存在
0x44:	总量单位不存在
0x45:	最高频率输出超上限
0x46:	最低频率输出超下限
0x47:	最高流速超上限
0x48:	占空比超上限



## 8 附录 2: 常数表: 流量单位

1	inH <sub>2</sub> O	压力	43	Cum	容积
2	inHg	压力	44	Ft	长度
3	ftH <sub>2</sub> O	压力	45	m	长度
4	mmH <sub>2</sub> O	压力	46	bbl	容积
5	mmHg	压力	47	in	长度
6	psi	压力	48	cm	长度
7	bar	压力	49	mm	长度
8	mbar	压力	50	min	时间
9	g/Sqcm	压力	51	s	时间
10	Kg/Sqcm	压力	52	h	时间
11	Pa	压力	53	d	时间
12	kPa	压力	54	cSt	粘度
13	torr	压力	55	cP	粘度
14	atm	压力	56	uMho	电导
15	Cuft/min	体积流量	57	%	分析
16	gal/min	体积流量	58	V	电动势
17	L/min	体积流量	59	pH	分析
18	Impgal/min	体积流量	60	g	质量
19	Cum/h	体积流量	61	kg	质量
20	ft/s	速度	62	MetTon	质量
21	m/s	速度	63	lb	质量
22	gal/s	体积流量	64	STon	质量
23	MMgal/d	体积流量	65	LTon	质量
24	L/s	体积流量	66	mMho/cm	电导
25	ML/d	体积流量	67	uMho/cm	电导
26	Cuft/s	体积流量	68		力
27	Cuft/d	体积流量	69		能量

28	Cum/s	体积流量	70	g/s	质量流量
29	Cum/d	体积流量	71	g/min	质量流量
30	Impgal/h	体积流量	72	g/h	质量流量
31	Impgal/d	体积流量	73	Kg/s	质量流量
32	degC	温度	74	Kg/min	质量流量
33	degF	温度	75	Kg/h	质量流量
34	degR	温度	76	Kg/d	质量流量
35	Kelvin	温度	77	MetTon/min	质量流量
36	mV	电动势	78	MetTon/h	质量流量
37	ohm	电阻	79	MetTon/d	质量流量
38	Hz	频率	80	lb/s	质量流量
39	mA	电流	81	lb/min	质量流量
40	gal	容积	82	lb/h	质量流量
41	L	容积	83	lb/d	质量流量
42	Impgal	容积	84	STon/min	质量流量
85	STon/h	质量流量	128	kWh	能量
86	STon/d	质量流量	129	Hp	功率
87	LTon/h	质量流量	130	Cuft/h	体积流量
88	LTon/d	质量流量	131	Cum/min	体积流量
89		能量	132	bbl/s	体积流量
90	SGU	密度	133	bbl/min	体积流量
91	g/Cucm	密度	134	bbl/h	体积流量
92	Kg/Cum	密度	135	bbl/d	体积流量
93	lb/gal	密度	136	gal/h	体积流量
94	lb/Cuft	密度	137	Impgal/s	体积流量
95	g/mL	密度	138	L/h	体积流量
96	kg/L	密度	139		浓度
97	g/L	密度	140	Mcal/h	功率
98	lb/Cuin	密度	141	MJ/h	功率

99	STon/Cuyd	密度	142	BTU/h	功率
100	degTwad	密度	143		角度
101	degBrix	浓度	144		角度
102	degBaum hv	密度	145		压力
103	degBaum lt	密度	146	ug/L	密度
104	degAPI	密度	147	ug/Cum	密度
105	% sol-wt	浓度	148		密度
106	% sol-vol	浓度	149		容积比率
107	degBall	体积质量	150	% Stm Qual	分析
108	proof/vol	浓度	151	ft.in16	长度
109	proof/mass	浓度	152	Cuft/lb	体积质量
110	bush	容积	153	pF	电容
111	Cuyd	容积	154	mL/L	容积比率
112	Cuft	容积	155	uL/L	容积比率
113	Cuin	容积	156		
114	in/s	速度	157		
115	in/min	速度	158		
116	ft/min	速度	159		
117		角速度	160	% plato	分析
118		角速度	161		分析
119		角速度	162	Mcal	能量
120	m/h	速度	163		电阻
121		体积流量	164	MJ	能量
122		体积流量	165	BTU	能量
123		体积流量	166		容积
124		容积	167		容积
125	Oz	质量	168		容积
126		能量	169		浓度
127	kW	功率	170		压力

171		压力	214		
172		压力	215		
173		压力	216		
174		压力	217		
175		压力	218	lton/s	质量流量
176		压力	219	lton/m	质量流量
177		压力	220	ston/s	质量流量
178		压力	221	g/d	质量流量
179		压力	222	a-ft/s	体积流量
180		体积流量	223	a-ft/m	体积流量
181		体积流量	224	a-ft/h	体积流量
182		体积流量	225	a-ft/d	体积流量
183		体积流量	226	Mgal/s	体积流量
184		体积流量	227	Mgal/m	体积流量
185		体积流量	228	Mgal/h	体积流量
186		体积流量	229	L/d	体积流量
187		体积流量	230	mL/d	体积流量
188		体积流量	231	mL	容积
189		体积流量	232	Mgal	容积
190		体积流量	233	a-ft	容积
191			234	cal/h	功率
192			235	gal/d	体积流量
193			236	hL	容积
194			237	Mpa	压力
195			238	inH2O @4DegC	压力
196			239	mmH2O @4DegC	压力
197			240	MetTon/s	质量流量

198			241	mL/s	体积流量
199			242	mL/min	体积流量
200			243	mL/h	体积流量
201			244	KJ/h	功率
202			245	GJ/h	功率
203			246	MW	能量
204			247	kBTU/h	功率
205			248	kcal/h	功率
206			249	Ps	功率
207			250	KJ	能量
208			251	GJ	能量
209			252	MWh	能量
210			253	kBTU	能量
211			254	kcal	能量
212					
213					

### 9 附录 3: 报警定义

15	14	13	12	11	10	9	8
Unused	Unused	Unused	Unused	Unused	Unused	Unused	Unused

7	6	5	4	3	2	1	0
Unused	Unused	ADC	ZERO	COIL	EMPTY	Unused	MEM

Unused: Bits 15 –6 Unused

ADC: Bit 5 超 ADC 范围

ZERO: Bit 4 零点过大(<-99.9mV 或 > +99.9mV)

COIL: Bit 3 励磁报警

EMPTY: Bit 2 空管报警

Unused: Bit 1 Unused

MEM: Bit 0 EEPROM 未安装