

## 4 数字量 8 模拟量采集模块

### 采集模块特性:

- Ø 4 路光电隔离开关量采集.
- Ø 8 路 10 位 4-20mA 模拟量采集
- Ø RS485/RS232 标准接口.
- Ø MODBUS\_RTU 标准协议控制.
- Ø 软件设定地址等参数
- Ø 标准工业导轨安装
- Ø 提供控制示例程序.
- Ø 可按客户要求定制协议

### 工作特性:

- Ø 工作环境温度:  $-20^{\circ}\text{C} \sim 65^{\circ}\text{C}$
- Ø 相对湿度: 95% (无凝结)
- Ø 电源电压 12V(可定制其它电压)

### 典型应用:

- Ø 各种工业数据采集
- Ø PLC 接口接扩展

### 实物图

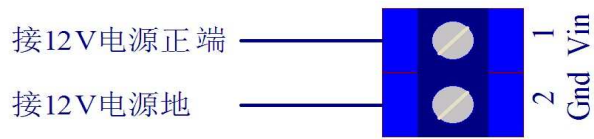


## 引脚描述

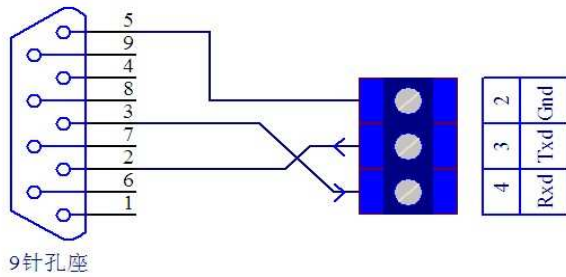
引脚	名称	描述
1	Vin	12V 电源正.
2	Gnd	电源地;Rs232 地.
3	TXD	RS232 数据发送
4	RXD	RS232 数据接收
5	D+	RS485 数据 D+(A)
6	D-	RS485 数据 D-(B)
7	AG	模拟地
8	A1	第一路模拟量输入端
9	A2	第二路模拟量输入端
10	A3	第三路模拟量输入端
11	A4	第四路模拟量输入端
12	AG	模拟地
13	AG	模拟地
14	A5	第五路模拟量输入端
15	A6	第六路模拟量输入端
16	A7	第七路模拟量输入端
17	A8	第八路模拟量输入端
18	AG	模拟地
19	X1	第一组开关量输入
20	X2	第二组开关量输入
21	X3	第三组开关量输入
22	X4	第四组开关量输入
23	Com	开关量输入公共端
24	Vio	开关量输入电源端

## 硬件连接

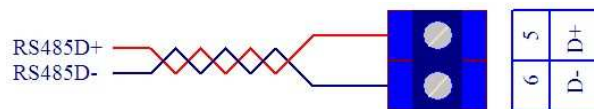
### 1. 电源的连接



### 2. RS232 串口的连接



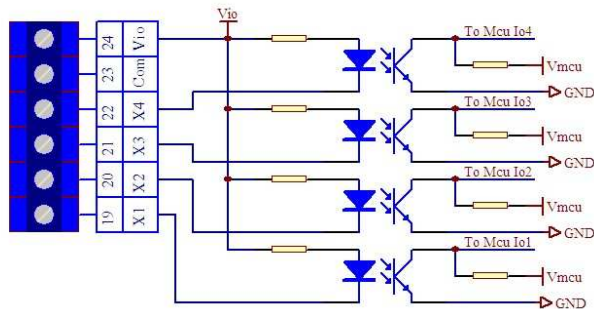
### 3. RS485 接口的连接.



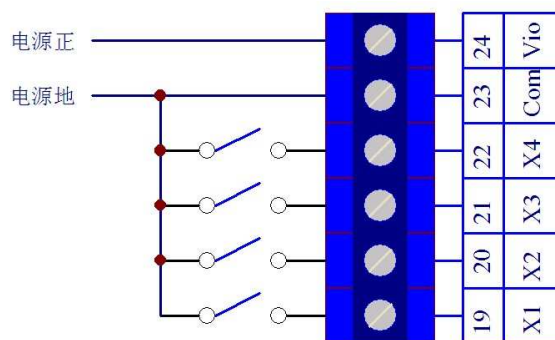
RS485 建议采用双绞线连接,当距离较长时采用带屏蔽的双绞线连接,并将屏蔽层接地.总线上挂接多个模块时采用手拉手的接线方式.具体接线方法请参考 RS485 接线规范.

### 4. 开关量输入的连接.

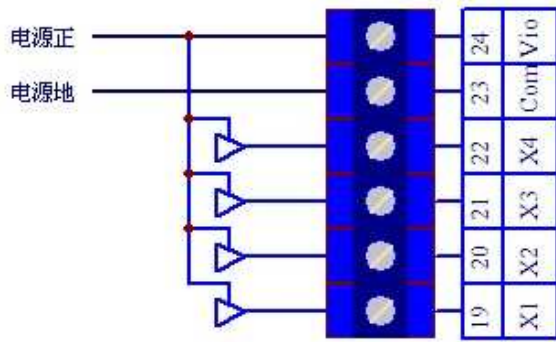
#### 4.1 开关量内部电路图



#### 4.2 采集外部干接点开关信号接线方法.

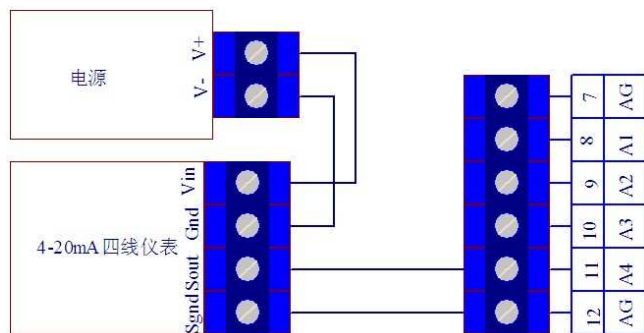


#### 4.3 采集外部 TTL/COMS 开关量信号接线方法

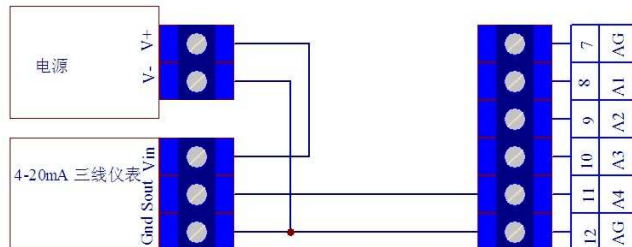


## 5. 模拟量 4-20mA 的采集电连接.

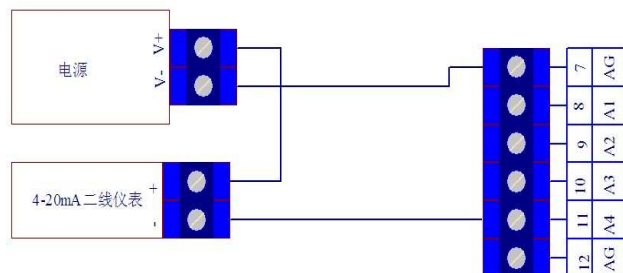
### 5.1 四线制 4-20mA 仪表的连接



### 5.1 三线制 4-20mA 仪表的连接



### 5.2 二线制 4-20mA 仪表的连接



# 软件协议

## 串口通信定义

默认波特率:9600,1 起始位,8 个数据位,1 个停止位,无校验位.

默认地址:0x01

默认通信接口:RS485 接口

## 指令简表

名称	从机地址 0x01-0xfe	命令	寄存器地址 高字节、低字节	数据域	CRC 校验 低字节、高字节
长度	1 字节	1 字节	2 字节		2 字节
读 4-20mA 模拟量	addr	0x03	0x0030	高字节=0x00, 低字节=模拟量路数 0x08	CRCL,CRCH
读温度(选配)	addr	0x03	0x003F	高字节=0x00, 低字节=温度路数 0x01	CRCL,CRCH
读取开关量输入	addr	0x02	0x0020	高字节=0x00, 低字节=开关量数 0x04	CRCL,CRCH
设定设备地址	addr	0x06	0x0040	高字节=0x00, 低字节=新设备的地址	CRCL,CRCH
设定波特率 (暂不开放)	addr	0x06	0x0041	高字节=0x00, 低字节:	CRCL,CRCH
输出断电记忆 (暂不开放)	addr	0x06	0x0042	高字节=0x00, 低字节 0x00 记忆,0x01 不记忆	CRCL,CRCH
设定校验位 (暂不开放)	addr	0x06	0x0044	高字节=0x00, 低字节 0x00=无校验,01 奇校验,02 偶校验	CRCL,CRCH

## 指令示例及详解(以地址 0x01 示例)

### 1. 读取开关量输入.

例:如是

读取地址为 0x01 模块的开关输入 发送数据为:0x01,0x02,x00,0x20,0x00,0x04,0x78,0x03

近回数据格式

名称	从机地址	命令	近回字节数	返回数据,高字节,低字节	CRC 校验
长度	1 字节	1 字节	1 字节	1 字节	2 字节
近回开关量状态	0x01	0x02	0x01	返回开关状态,低 4 位有效	CRCL,CRCH

近回结果 0x01, 0x02,x01,0x0F,0xE1,0x8c

近回 0x0F,输入的 IO 的状态 0 为低,1 为高.本模块只用低 4 位,只关心低四位.

	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
X1	H		H		H		H		H		H		H		H
X2		H	H			H	H			H	H			H	H
X3				H	H	H	H					H	H	H	H
X4								H	H	H	H	H	H	H	H

### 2. 读取模拟量输入

读取第一路: 发送数据为:0x01,0x03,x00,0x30,0x00,0x01, 0x84,0x05

返回数据格式:addr,0x03,0x02,AdcH1,AdcL1,CRCL,CRCH

计算返回采集电流简单计算方法:  $(AdcH1 * 256 + AdcL1) / 40$  单位(mA)

一次读取 8 路的指令:0x01,0x03,x00,0x30,0x00,0x08, 0x44,0x03

返回数据:

addr,0x03,0x10,AdcH1,AdcL1, ,AdcH2,AdcL2,AdcH3,AdcL3,AdcH4,AdcL4,AdcH5,AdcL5,AdcH6,AdcL6,AdcH7,AdcL7,AdcH8,AdcL8,AdcH1,AdcL1,CRCL,CRCH

### 3. 读取温度输入(选配)

发送数据为:0x01,0x03,x00,0x3F,0x00,0x01,0x78,0x03

返回数据格式:addr,0x03,0x02,Temper H, Temper L,CRCL,CRCH

计算返回温度简单计算方法:  $(Temper H * 256 + Temper L) / 10$  单位度

#### 4. 设定模块地址

将 0x01 改为 0x02    0x01,0x06,0x00,0x40,0x00,0x02,0x09,0xDF

将 0x01 改为 0x03    0x01,0x06,0x00,0x40,0x00,0x03,0xC8,0x1F

将 0x02 改为 0x01    0x02,0x06,0x00,0x40,0x00,0x01,0x49,0xED

#### 6. 设定控制端口（升级为拨码开关控制）

设定控制端口为 RS232    SW2=on

设定控制端口为 RS485    SW2=off

#### 5. 设定是否主动上传数字量输入（升级为拨码开关控制）

设定不主动上传    SW1=off

设定主动上传    SW1=on

#### 7. 设定串口校验位(暂不开放)

设定串口无校验位

设定串口偶校验位

设定串口奇校验位

#### 8. 设定串口端口波特率(暂不开放)

## 模块的外形尺寸及安装



## 附：CRC16 直接算法 C 函数

```
unsigned int crc_cal16 (unsigned char *snd, unsigned char num)
{
    unsigned char i, j;
    unsigned int c, crc=0xFFFF;
    for(i = 0; i < num; i ++)
    {
        c = snd[i] & 0x00FF;
        crc ^= c;
        for(j = 0; j < 8; j ++)
        {
            if (crc & 0x0001)
            {
                crc>>=1;
                crc^=0xA001;
            }
            else crc>>=1;
        }
    }
    return(crc);
}
```

## 附: CRC16 直接算法 Basic 函数

Public Function CRC16(Data() As Byte, Optional ByVal Offset As Integer = 0, Optional ByVal Length As Integer = 0) As Byte()

```
Dim CRC16Lo As Byte, CRC16Hi As Byte      'CRC 寄存器
Dim CL As Byte, CH As Byte                '多项式码&HA001
Dim SaveHi As Byte, SaveLo As Byte
Dim I As Integer
Dim Flag As Integer
Dim ReturnData(1) As Byte
CRC16Lo = &HFF
CRC16Hi = &HFF
CL = &H1
CH = &HA0
Length = IIf(Length < 1, UBound(Data) - Offset, Length - 1)  'Update 2007-03-15
For I = Offset To Offset + Length
    CRC16Lo = CRC16Lo Xor Data(I) '每一个数据与 CRC 寄存器进行异或
    For Flag = 0 To 7
        SaveHi = CRC16Hi
        SaveLo = CRC16Lo
        CRC16Hi = CRC16Hi \ 2          '高位右移一位
        CRC16Lo = CRC16Lo \ 2        '低位右移一位
        If ((SaveHi And &H1) = &H1) Then '如果高位字节最后一位为 1
            CRC16Lo = CRC16Lo Or &H80  '则低位字节右移后前面补 1
        End If                          '否则自动补 0
        If ((SaveLo And &H1) = &H1) Then '如果 LSB 为 1, 则与多项式码进行异或
            CRC16Hi = CRC16Hi Xor CH
            CRC16Lo = CRC16Lo Xor CL
        End If
    Next
Next
'CRC_TXT.Text = Right(("00" + (Hex(CRC16Lo))), 2)
'CRC_TXT.Text = CRC_TXT.Text + Right(("00" + (Hex(CRC16Hi))), 2)
ReturnData(0) = CRC16Lo                'CRC 低位
ReturnData(1) = CRC16Hi                'CRC 高位
CRC16 = ReturnData
End Function
```