I/O 采集模块

M0802

8 路输入 2 路(继电器)输出

智能数字量采集器

使用说明

目 录

第1章　产品概述

1.1 概述

M0802 为智能数字量采集器，产品具有 8 路干接点数字量输入和 2 路 C 型继电器输出数字量；电源及 RS485 接口均加入防雷保护电路，产品稳定可靠；丰富的指示灯方便调试，运行状态一目了然；采用标准 Modbus RTU 协议，方便系统集成商、工程商使用；通过 RS-485 即可实现对远程数字量设备的数据采集和控制。DI 输入通常有接入接近开关、机械开关、按钮、继电器、光电开关、烟感、水浸、红外探测器、气体泄漏报警器等数字量开关设备；DO 通常可控制继电器、接触器、SSR 及电灯等负载设备。

针对工业应用，RS485 通讯接口采用光电隔离设计，避免工业现场信号对通讯接口的影响，具有良好的兼容性及稳定性，同一总线上支持最多 255 个设备在线通讯；标准 Modbus RTU 通讯协议及常用功能码，使得用户可以更加轻松实现与监控软件、Modbus RTU 协议的 PLC 等设备和系统的整合应用；提供协议和示例代码，使您的二次开发更加灵活、简便、高效。

本产品广泛应用于：医疗、工矿产品开发 ；工控教学应用远程通讯 ；机房动力环境监控；移动数据采集站 ；智能楼宇控制数据、安防工程等应用系统 ；机械、消防、石化、建筑、电力、交通等各行业 RS-232/485 总线工业自动化控制系统等领域。

1.2 性能特点

- 8 路干接点数字量输入（DI）
- 2 路 C 型继电器输出（DO）
- DI 输入范围：干接点
- DO 输出范围：C 型继电器输出
- 驱动能力：3A，250VAC
- 双硬件看门狗，绝不死机
- 采用 32 位 ARM 嵌入式 CPU，高性能低功耗
- 采用 Modbus RTU 通信协议
- 丰富的的指示灯，方便调试
- RS485 通信接口提供光电隔离及防雷保护
- 电源具有过流、过压、防反接及防雷保护
- 工业级温度范围，应对严苛现场环境

- 标准导轨安装或螺钉固定

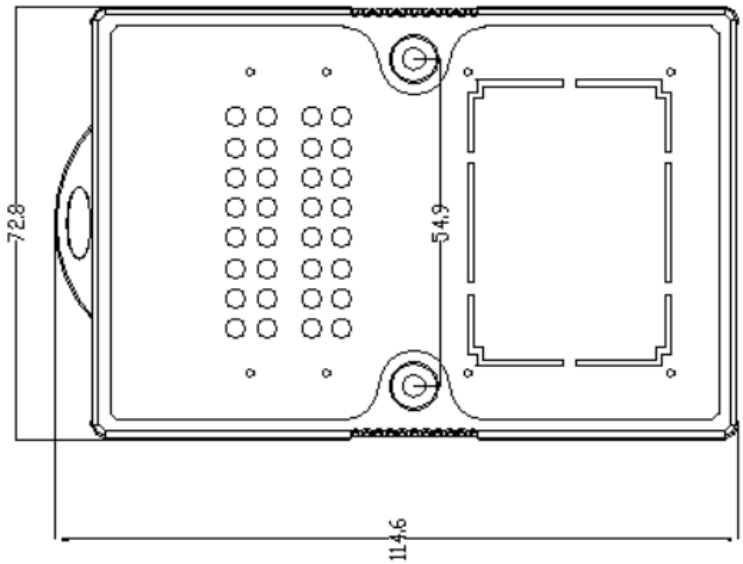
1.3 技术参数

DI 接口	DI	8 路干接点
	接点电压	无
	触发电流	小于 1mA
	防雷防护	共模，600W
	过压过流保护	30V，500mA
DO 接口	DO	2 路 C 型继电器
	接点电压	0~250VAC
	触点容量	3A
串口通信参数	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	N.8.1
	通讯协议	Modbus RTU
	防雷防护	共模，600W
	过压过流保护	30V，100mA
电源参数	电源规格	6-24VDC (推荐 12VDC)
	功耗	60mA@12VDC
	防雷防护	1300W
	过压过流保护	30V，500mA
工作环境	工作温度、湿度	-40~85℃，5~90%RH，不凝露
	储存温度、湿度	-60~125℃，5~90%RH，不凝露
其他	尺寸	110mm*75mm*30mm
	保修	2 年保修

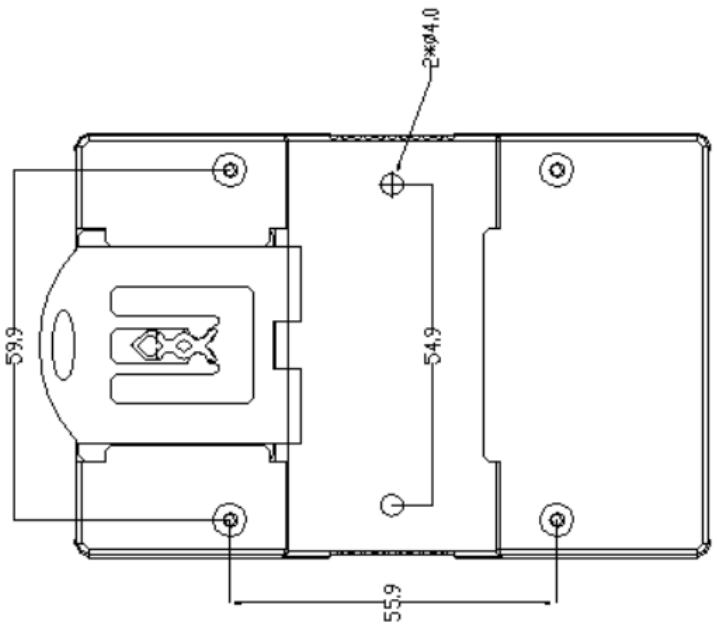
第2章　外观尺寸

2.1 产品外观

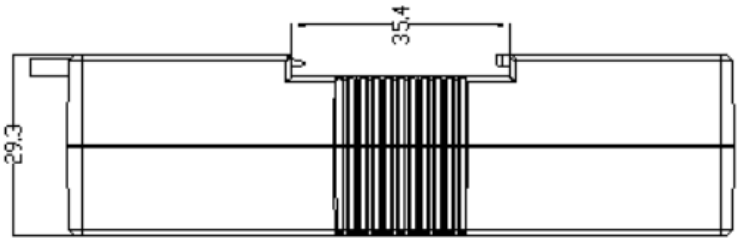
2.2.1 前视图



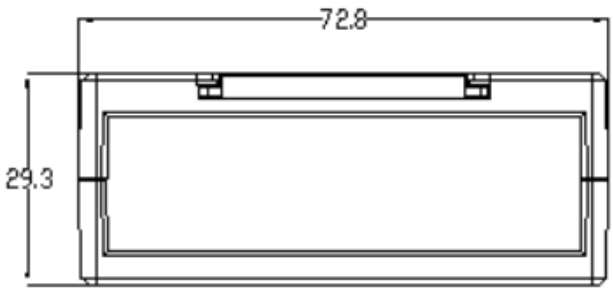
2.2.2 后视图



2.2.3 侧视图

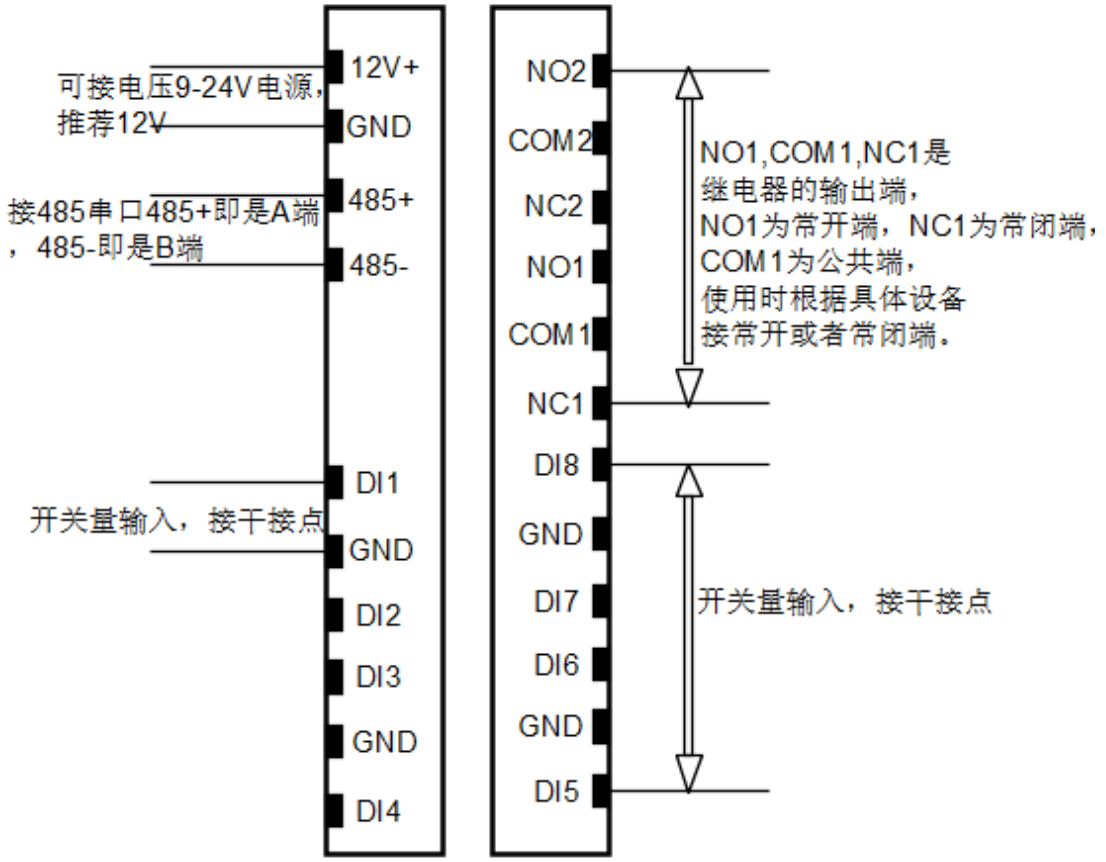


2.2.4 顶视图



第3章　产品接线图

产品接线图



第4章　引脚说明及指示灯

4.1 引脚定义

引脚定义	说明
VS+	电源正
GND	电源负
485+	RS485+
485-	RS485-
DI(GND)	数字量信号输入公共端
DI1~8	数字量信号输入端
DO1~2(COM)	数字量信号输出端
DO1~2(NC)	数字量信号常闭输出端
DO1~2(NO)	数字量信号常开输出端空

4.2 LED 指示灯

M0802 外设 14 个状态 LED 指示灯，能够准确及时报告设备的工作状态，为工程的施工和调试带来极大的方便。其说明如下表所示:

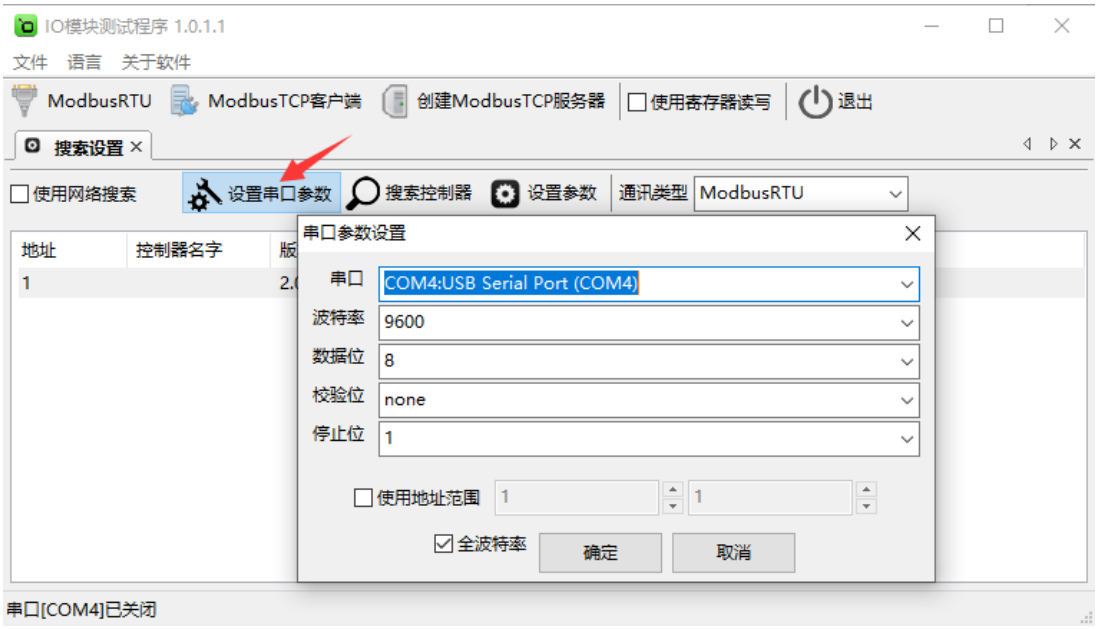
指示灯	指示灯说明
PWR	电源指示灯（亮：有电源连接；灭：无电源连接）
RUN	闪烁：正常运行；常亮或者不亮：工作不正常
TX	RS485 接口发送数据
RX	RS485 接口接收数据
DI1~DI8	亮：对应 DI 有输入
DO1~DO2	亮：对应 DO 有输出

第5章　软件操作

本软件为无安装的绿色测试软件，拷贝过来即可使用，软件只对设备产品进行配置和测试，不做其他用途，在使用软件对IO模块进行操作时，请保证模块正常加点并连接好通讯线缆。

5.1 搜索 IO 模块

打开 IO 模块测试程序，点“使用串口搜索”图标，再点“设置串口参数”图标，在弹出对话框中设置串口相关参数，包括波特率（IO 模块默认出厂波特率为 9600）,数据位设置为 8，停止位设置为 1，校验位设置为 none，一切就绪后，点击“确定”按钮：



设置好串口参数后，点“确定”按钮。

备注：如果已知 IO 模块的波特率（比如出厂的 9600），并且在上一步中设置了和 IO 模块匹配的波特率，则不用勾选“全波特率”复选框；如果未知 IO 模块的波特率，则需要勾选“全波特率”复选框，软件会从低波特率开始尝试搜索设备（1200bps~115200bps）。

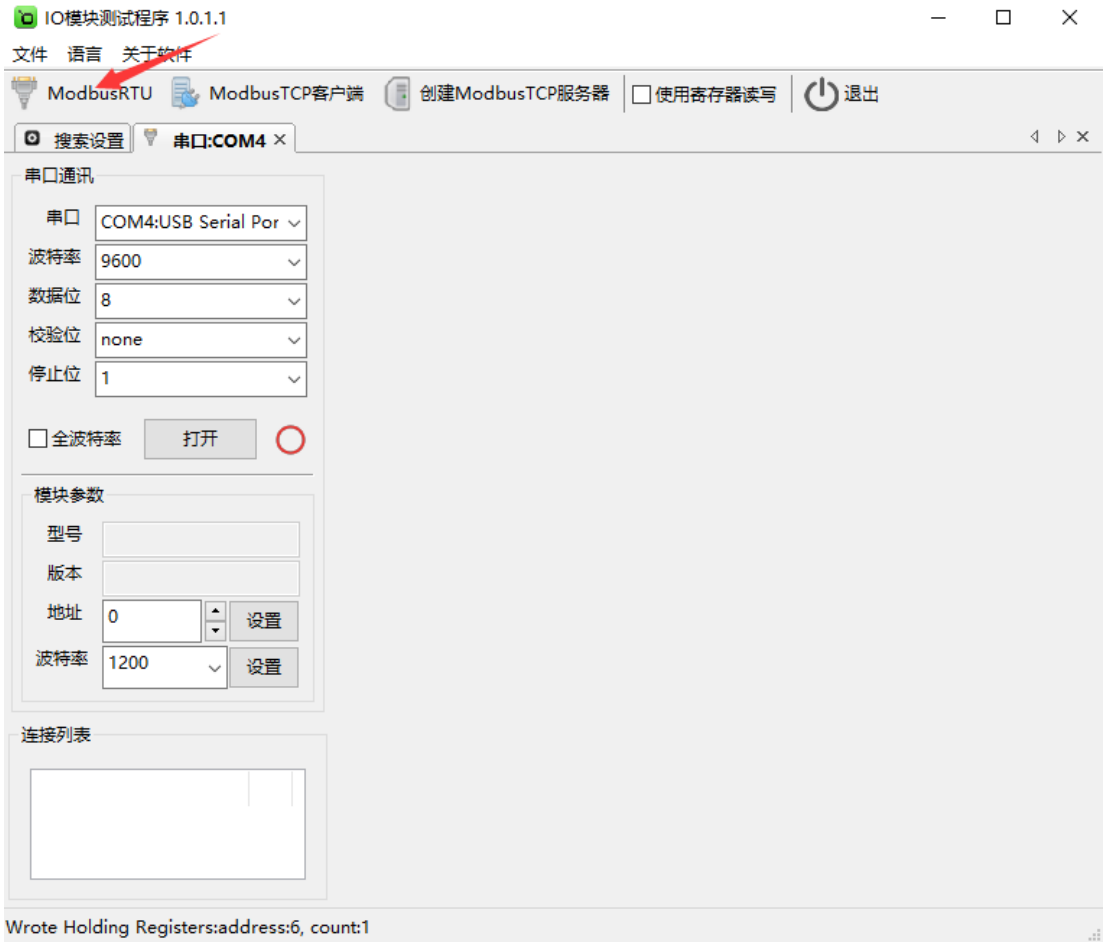
点“搜索控制器”按钮，页面会显示设备的地址、控制器名字、版本号和波特率等，如下图：

5.2 设置 IO 模块

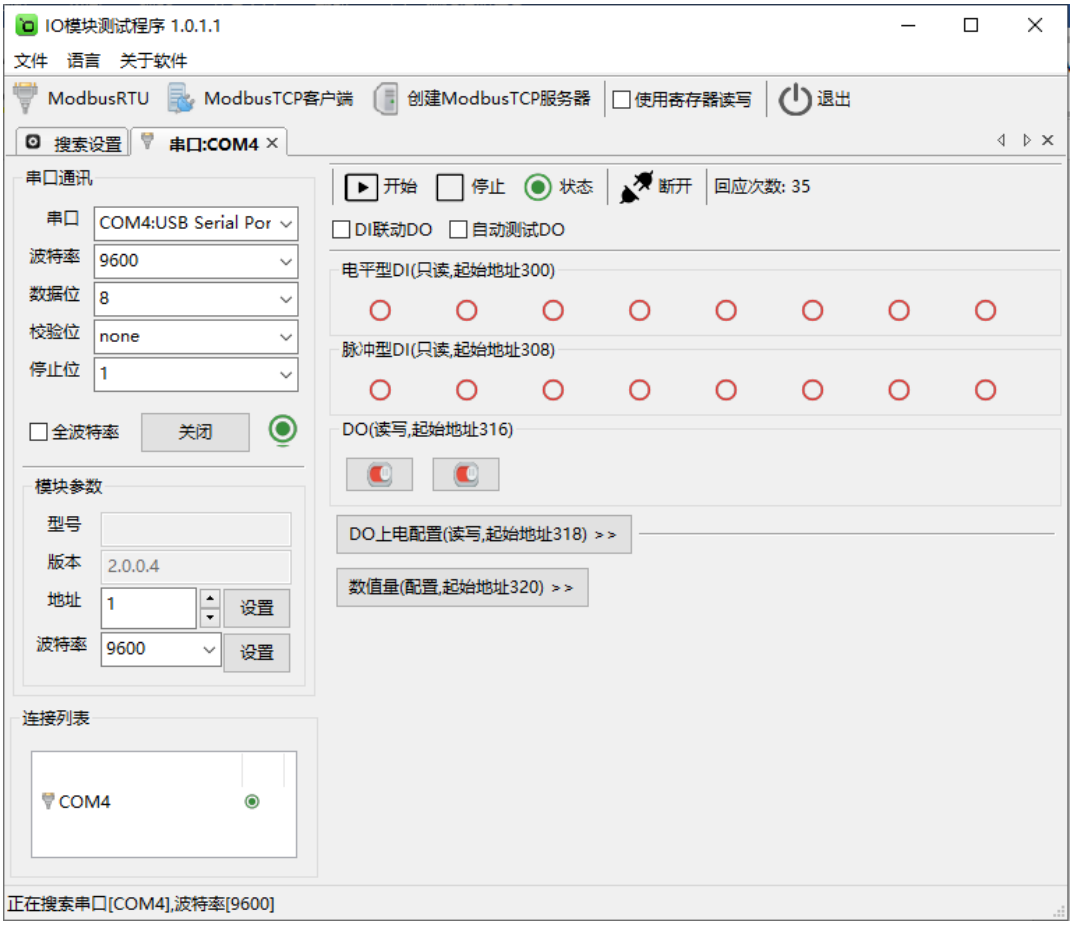
选中模块，点“设置参数”按钮，在弹出的对话框中设置 IO 模块的地址（范围是 1~255），在“波特率”下拉菜单中选择波特率，点“确定”按钮，参数设置成功。如下图：

和其他模块联动：勾选后，该模块会与其他模块联动工作。否则，该模块独立工作。

点击“ModbusRTU”图标或者选中模块点右键弹出“ModbusRTU”，然后左键选择，弹出以下界面，此时测试界面的左上方显示“串口通讯”参数设置界面。如下图：

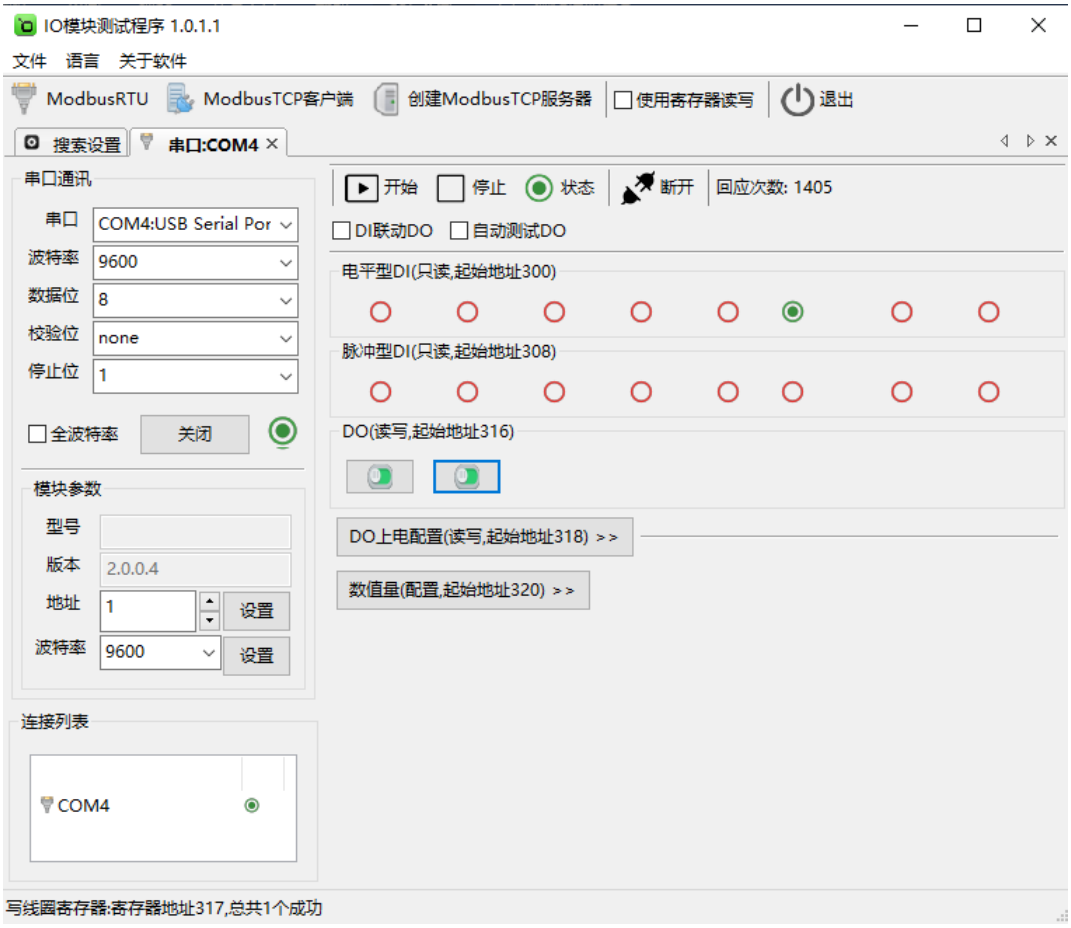


在弹出的对话框中再点击“打开”按钮，、“模块参数”和“连接列表”等设置参数，且可以更改。如下图:



5.3 测试 IO 模块

点击“打开”按钮，测试软件界面左下方显示“模块参数”和“连接列表”设置界面。下面状态栏会有写寄存器的提示。右半边测试软件会根据产品型号自动显示所对应的测试界面，显示为采集 DI 和 DO 的数值，DI 状态为只读值，红色色表示断开，绿色表示接通。DO 的各路状态均为读写值，可以很方便地改变其状态。写入值 0 表示常闭点闭合而常开点断开，写入值 1 表示常闭点断开而常开点闭合；上电状态 0 表示加电时常闭点闭合而常开点断开，上电状态 1 表示加电时常闭点断开而常开点闭合。可以很直观地看到它各路的状态。如下图：


IO模块测试程序 1.0.1.1
文件 语言 关于软件
ModbusRTU
ModbusTCP客户端
创建ModbusTCP服务器
使用寄存器读写
退出
搜索设置
串口:COM4
串口通讯
串口
COM4:USB Serial Por
波特率
9600
数据位
8
校验位
none
停止位
1
全波特率
关闭
模块参数
型号
版本
2.0.0.4
地址
1
设置
波特率
9600
设置
连接列表
COM4
开始
停止
状态
断开
回应次数: 1405
DI联动DO
自动测试DO
电平型DI(只读,起始地址300)
脉冲型DI(只读,起始地址308)
DO(读写,起始地址316)
DO上电配置(读写,起始地址318) >>
数值量(配置,起始地址320) >>
写线圈寄存器:寄存器地址317,总共1个成功

第6章　通讯协议及寄存器定义

6.1 通讯协议

遵循标准 MODBUS RTU 协议，协议格式如下:

从设备地址	功能码	数据	校验
1 字节	1 字节	N 字节	2 字节

从设备地址：即 IO 模块的地址，地址可设置;

功能码：读写 IO 模块 DIO 状态的功能码;

数据：根据功能码和寄存器个数确定数据的大小;

校验：CRC16 校验，校验低位在前，高位在后。

6.1.1 读线圈状态

功能码：0x01

上位机报文:

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x01
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文:

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x01
字节数	1 字节，从读寄存器个数计算得出： 如果寄存器个数被 8 整除： 字节数 = 寄存器个数/8 如果寄存器个数不能被 8 整除： 字节数 = 寄存器个数/8+1
数据	每一位表示一路 DIO 的状态，第一个字节的第一位表示起始寄存器的状态，依次类推
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文:

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x01
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.2 写单个线圈状态

功能码：0x05

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x05
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前，写 0x0000 表示输出 0，写 0xff00 表示输出 1
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x05
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前，回应 0x0000 表示 0，回应 0xff00 表示 1
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x05
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.3 写多个线圈状态

功能码：0x0f

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff

功能码	1 字节，内容为 0x0f
起始寄存器	2 字节，高位在前
寄存器个数	2 字节，高位在前
字节数	1 字节，字节数从寄存器个数计算得出： 如果寄存器个数被 8 整除： 字节数 = 寄存器个数/8 如果寄存器个数不能被 8 整除： 字节数 = 寄存器个数/8+1
数据	每一位表示一路线圈状态（即是 DO 或其配置），第一个字节的第一位表示起始寄存器的状态
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文:

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x0f
起始寄存器	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文:

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x0f
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.4 读保持寄存器

功能码：0x03

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x03
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x03

字节数	1 字节，即是寄存器个数 x2，因为每个保持寄存器两个字节
数据	各个保持寄存器的值，每个保持寄存器占用 2 字节，并且高位在前
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x03
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.5 写单个保持寄存器

功能码：0x06

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x06
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x06
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x06
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.6 写多个保持寄存器

功能码：0x10

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x10
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
字节数	1 字节，即是寄存器个数 x2，因为每个保持寄存器占用 2 个字节
数据	各个保持寄存器的值，每个保持寄存器占用 2 字节，并且高位在前
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x10
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x10
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

6.1.7 错误码表

错误码	意义
0x01	无效功能码
0x02	无效寄存器地址
0x03	寄存器值无效
0x04	从机设置错误
0x05	ACK，一般用于长时间执行某项任务
0x06	从机忙状态
0x07	NEGATIVE ACK
0x08	MEMORY PARITY ERROR

6.2 寄存器定义

6.2.1 M0802 寄存器

寄存器地址	功能	种类	读写状态	取值范围

300	DI 电平输入 1	线圈状态	只读	0 表示无输入，1 表示有输入
301	DI 电平输入 2	线圈状态	只读	0 表示无输入，1 表示有输入
302	DI 电平输入 3	线圈状态	只读	0 表示无输入，1 表示有输入
303	DI 电平输入 4	线圈状态	只读	0 表示无输入，1 表示有输入
304	DI 电平输入 5	线圈状态	只读	0 表示无输入，1 表示有输入
305	DI 电平输入 6	线圈状态	只读	0 表示无输入，1 表示有输入
306	DI 电平输入 7	线圈状态	只读	0 表示无输入，1 表示有输入
307	DI 电平输入 8	线圈状态	只读	0 表示无输入，1 表示有输入
308	DI 脉冲输入 1	线圈状态	只读	0 表示无输入，1 表示有输入
309	DI 脉冲输入 2	线圈状态	只读	0 表示无输入，1 表示有输入
310	DI 脉冲输入 3	线圈状态	只读	0 表示无输入，1 表示有输入
311	DI 脉冲输入 4	线圈状态	只读	0 表示无输入，1 表示有输入
312	DI 脉冲输入 5	线圈状态	只读	0 表示无输入，1 表示有输入
313	DI 脉冲输入 6	线圈状态	只读	0 表示无输入，1 表示有输入
314	DI 脉冲输入 7	线圈状态	只读	0 表示无输入，1 表示有输入
315	DI 脉冲输入 8	线圈状态	只读	0 表示无输入，1 表示有输入
316	DO 输出 1	线圈状态	读写	0 表示无输出，1 表示有输出
317	DO 输出 2	线圈状态	读写	0 表示无输出，1 表示有输出
318	上电 DO 配置 1	线圈状态	读写	0 表示无输出，1 表示有输出
319	上电 DO 配置 2	线圈状态	读写	0 表示无输出，1 表示有输出
300	DI 电平输入 1	保持寄存器	只读	0 表示无输入，1 表示有输入
301	DI 电平输入 2	保持寄存器	只读	0 表示无输入，1 表示有输入
302	DI 电平输入 3	保持寄存器	只读	0 表示无输入，1 表示有输入
303	DI 电平输入 4	保持寄存器	只读	0 表示无输入，1 表示有输

				入
304	DI 电平输入 5	保持寄存器	只读	0 表示无输入，1 表示有输入
305	DI 电平输入 6	保持寄存器	只读	0 表示无输入，1 表示有输入
306	DI 电平输入 7	保持寄存器	只读	0 表示无输入，1 表示有输入
307	DI 电平输入 8	保持寄存器	只读	0 表示无输入，1 表示有输入
308	DI 脉冲输入 1	保持寄存器	只读	0 表示无输入，1 表示有输入
309	DI 脉冲输入 2	保持寄存器	只读	0 表示无输入，1 表示有输入
310	DI 脉冲输入 3	保持寄存器	只读	0 表示无输入，1 表示有输入
311	DI 脉冲输入 4	保持寄存器	只读	0 表示无输入，1 表示有输入
312	DI 脉冲输入 5	保持寄存器	只读	0 表示无输入，1 表示有输入
313	DI 脉冲输入 6	保持寄存器	只读	0 表示无输入，1 表示有输入
314	DI 脉冲输入 7	保持寄存器	只读	0 表示无输入，1 表示有输入
315	DI 脉冲输入 8	保持寄存器	只读	0 表示无输入，1 表示有输入
316	DO 输出 1	保持寄存器	读写	0 表示无输出，1 表示有输出
317	DO 输出 2	保持寄存器	读写	0 表示无输出，1 表示有输出
318	上电 DO 配置 1	保持寄存器	读写	0 表示无输出，1 表示有输出
319	上电 DO 配置 2	保持寄存器	读写	0 表示无输出，1 表示有输出
320	DO1 输出脉冲时间	保持寄存器	读写	0~65535ms
321	DO2 输出脉冲时间	保持寄存器	读写	0~65535ms

6.3 协议应用范例

6.3.1 读寄存器命令举例

以下为读取 IO 模块 8 路电平型 DI 的命令举例，假定 IO 模块的地址为 1，寄存器起始地址为 300（十六进制为 0x12c），个数为 8，上位机发送的数据如下（十六进制表示）：

01 01 01 2c 00 08 fd f9

各项分别表示：

01 IO 模块的地址，1 字节；

01 功能码：读取线圈状态的功能码；

01 2c 起始寄存器，即是寄存器 300；

00 08 需要读取的寄存器个数，这里举例为 8 路，8 路电平型 DI；

fd f9 CRC16 校验，从地址到数据域的校验，计算结果为 0xf9fd，因为要低在前，所以是 fd f9。

从机应答举例，假定 8 路电平 DI 状态状态分别：1 0 1 0 0 0 0 0，则回应的数据如下（十六进制表示）：

01 01 01 05 91 8b

各项分别表示：

01 IO 模块的地址，1 字节；

01 功能码：读取线圈状态的功能码；

01 字节数，因为是 8 个寄存器，所以字节数=寄存器个数/8=1；

05 各个寄存器的值，从低位开始对应的电平 DI 的第一路；

91 8b CRC16 校验，从地址到数据域的校验，计算结果为 0x8b91，因为要低位在前，所以是 91 8b。

6.3.2 写单个寄存器命令举例

以下为写 DO1 输出的应用举例，假定 IO 模块的地址为 1，寄存器地址为 316（十六进制为 0x13c），写 DO1 状态 1 的数据如下（十六进制表示）：

01 05 01 3c ff 00 4d ca

01 IO 模块的地址，1 字节；

05 功能码：写线圈状态的功能码；

01 3c 寄存器地址，高位在前，DO1 的寄存器；

ff 00 向 DO1 写 1 的操作，如果写 0，则填 00 00；

4d ca CRC16 校验，从地址到数据域的校验，计算结果为 0xca4d，因为要低位在前，所以是 4d

ca。

如果执行正常，从机应答数据如下（十六进制表示）：

01 05 01 3c ff 00 4d ca

01 IO 模块的地址，1 字节；

05 功能码：写线圈状态的功能码；

01 3c 寄存器地址，高位在前，DO1 的寄存器；

ff 00 DO1 的状态返回值，ff 00 表示 DO1 状态为 1，00 00 表示 DO1 状态为 0；

4d ca CRC16 校验，从地址到数据域的校验，计算结果为 0xca4d，因为要低位在前，所以是 4d ca。

6.3.3 写多个寄存器命令举例

以下为写从 DO1 开始的 2 路 DO，假定 IO 模块的地址为 1，寄存器地址为 316（十六进制为 0x13c），写从 DO1 到 DO2 的数据（全部输出）如下（十六进制表示）：

01 0f 01 3c 00 02 01 03 cf 42

01IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

01 3c 寄存器地址，高位在前，从 DO1 寄存器开始写；

00 02 寄存器个数，写 2 路，即是 DO1~DO2；

01 字节数，寄存器个数不能被 8 整除，所以字节数=寄存器个数/8+1=1；

03 各路 DO 寄存器的值，该字节的第一位表示第一路 DO，以此类推；

cf 42 CRC16 校验，从地址到数据域的校验，计算结果为 0x42cf，因为要低位在前，所以是 cf 42。

如果执行正常，从机应答数据如下（十六进制表示）：

01 0f 01 3c 00 02 1a f5

01IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

01 3c 寄存器地址，高位在前，从 DO1 寄存器开始写；

00 02 寄存器个数，总共写 2 路，即是 DO1~DO2；

1a f5 CRC16 校验，从地址到数据域的校验，计算结果为 0xfa15，因为要低位在前，所以是 1a f5。

第7章　装箱清单

序号	名称	数量	单位	备注
1	主设备 M0802	1	台	
2	产品简易说明书	1	张	
3	合格证	1	张	