



NKWOD[®] 凌柯沃

LW MA7348V

隔离型 4 路数字量 (DI) 输入

8 路电压模拟量 (AI) 输入

混合型智能数字量采集器

使用说明

目 录

第 1 章 产品概述	3
1.1 概述	3
1.2 性能特点	3
1.3 技术参数	4
第 2 章 外观尺寸	5
2.1 产品外观	5
2.2.1 前视图	5
2.2.2 后视图	6
2.2.3 侧视图	6
2.2.4 顶视图	6
第 3 章 产品接线图	7
产品接线图	7
第 4 章 引脚说明及指示灯	8
4.1 引脚定义	8
4.2 LED 指示灯	8
第 5 章 软件操作	9
5.1 搜索 IO 模块	9
5.2 设置 IO 模块	10
5.3 测试 IO 模块	12
5.3.1 测试量程为 0~20mA	12

5.3.2 测试量程为 4~20mA.....	13
5.3.3 开关量测试.....	14
第 6 章 通讯协议及寄存器定义	16
6.1 通讯协议.....	16
6.1.1 读线圈状态.....	16
6.1.2 写单个线圈状态.....	17
6.1.3 写多个线圈状态.....	17
6.1.4 读保持寄存器.....	18
6.1.5 写单个保持寄存器.....	19
6.1.6 写多个保持寄存器.....	20
6.1.7 错误码表.....	20
6.2 寄存器定义.....	21
6.2.1 公共寄存器.....	21
6.2.2 MA7348V 寄存器.....	21
6.2.3 浮点数说明.....	23
6.3 协议应用范例.....	23
6.3.1 读 DI 输入寄存器命令举例.....	23
6.3.2 读模拟量寄存器命令举例.....	24
第 7 章 装箱清单	26

第1章 产品概述

1.1 概述

LW MA7348V 为混合型隔离混合智能采集器，具有 4 路干接点数字量输入（DI），8 路电压型模拟量输入（量程为 0~5V），采用单端输入设计。RS485 接口光电隔离和电源隔离技术，有效抑制干扰；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，符合工业标准，方便系统集成商、工程商使用；方便与上位机通讯，可实现快速组网，构建监测系统；适用于各种工业场合及自动化系统，通过 RS-485 即可实现对远程模拟量设备的数据采集和控制。

本产品采用标准 Modbus RTU 协议及常用功能码，使用户可以更加轻松实现与 SCADA 软件、HMI 设备及支持 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。

1.2 性能特点

- 4 路干接点数字量输入（DI）
- DI 输入范围：0~5V
- 8 路模拟量电流输入
- AI 输入测量范围：0~5V
- 16 位分辨率
- 精度 1‰
- AI 输入通道采取单端输入
- AI 输入与系统采用光电隔离
- AI 通道具有防雷保护功能
- 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

- 宽电源电压设计
- 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

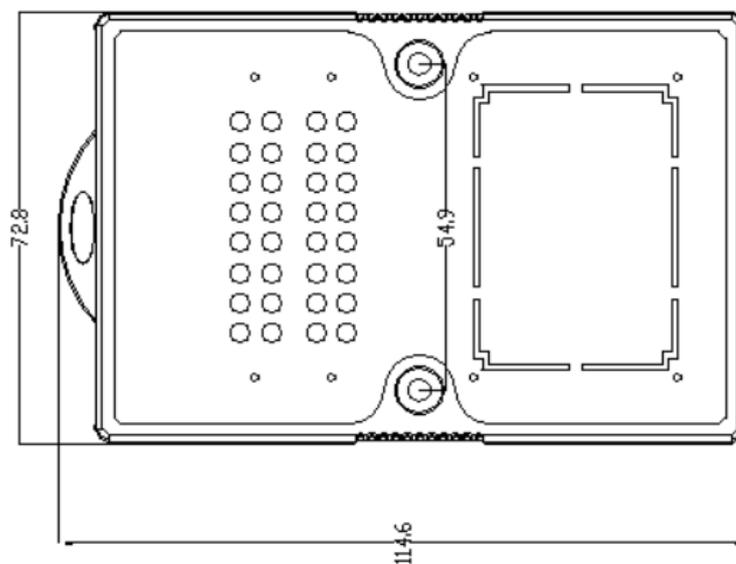
DI 接口	DI	4 路干接点
	触发电压	小于 1V 为逻辑 1，大于 2.5V 为逻辑 0
	触发电流	小于 1mA
	防雷防护	600W
	端口过压保护	30V
模拟量接口	AI	8 路单端输入
	AI 分辨率	16bit
	AI 量程	0~5V
	精度	1‰
	采集速度	100Hz
	AI 输入阻抗	200Ω
	电源隔离度	1500V
串口通信参数	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	N.8.1
	通讯协议	Modbus RTU
	防雷防护	共模，600W
	过压过流保护	30V，100mA
电源参数	电源规格	9-28VDC (推荐 12VDC)
	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°C，5~90%RH，不凝露
	储存温度、湿度	-60~125°C，5~90%RH，不凝露
其他	尺寸	110mm*75mm*30mm
	保修	6 年质保

第2章 外观尺寸

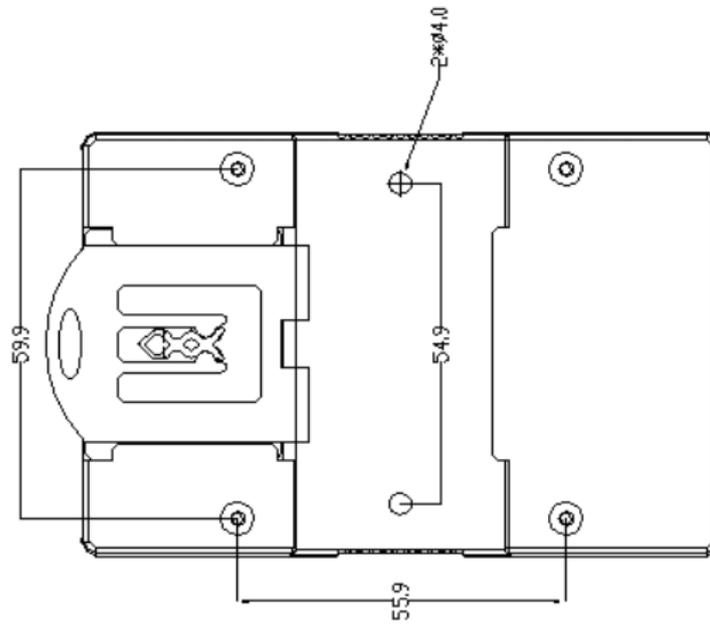
2.1 产品外观



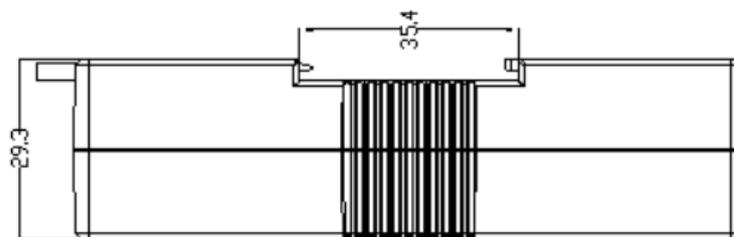
2.2.1 前视图



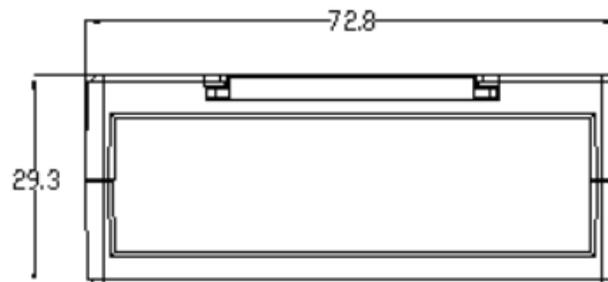
2.2.2 后视图



2.2.3 侧视图

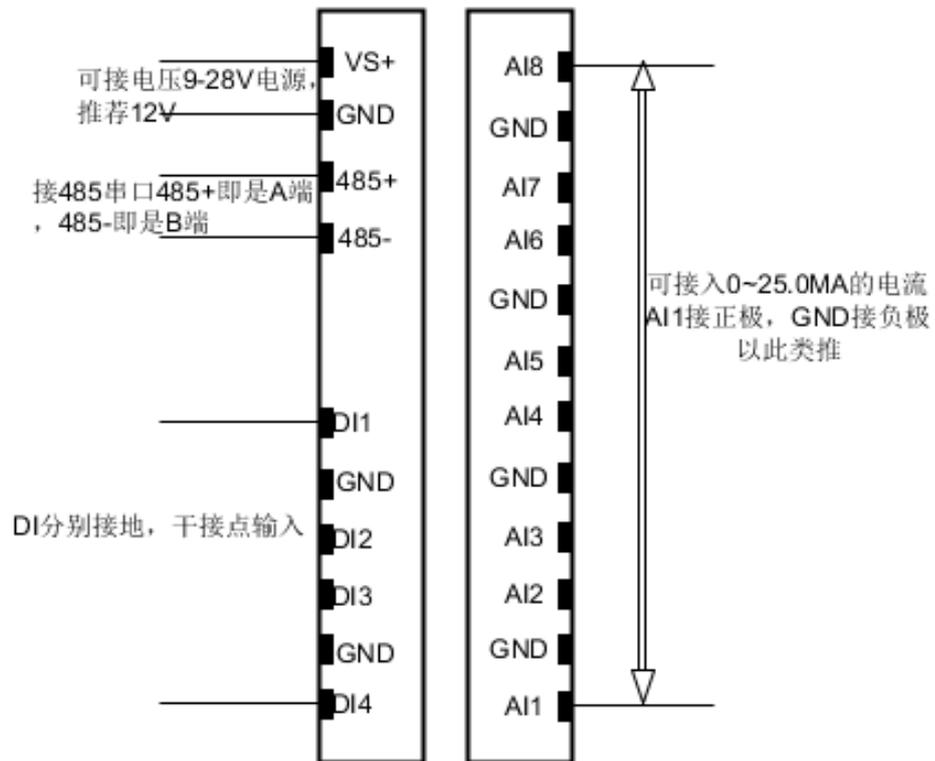


2.2.4 顶视图



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
DI1~4	数字量信号输入端
AI(GND)	模拟量信号输入公共端
AI1~8	模拟量信号输入端

4.2 LED 指示灯

LW MA7348V 外设 8 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~DI4	亮：对应 DI 有输入

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，请保证模块正常加电并连接好通讯线缆。

5.1 搜索 IO 模块

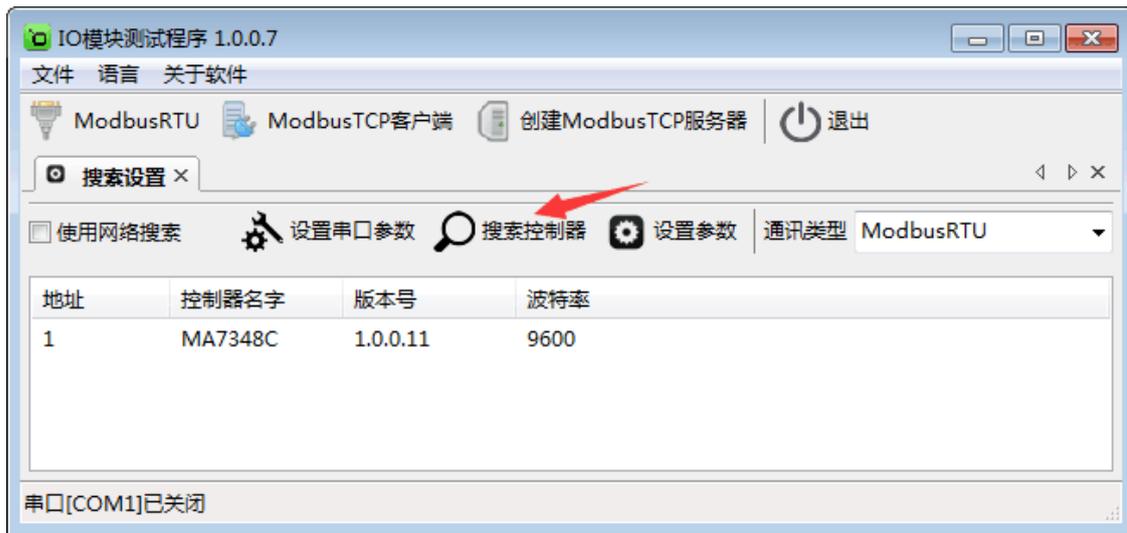
打开 IO 模块测试程序，点“使用串口搜索”图标，再点“设置串口参数”图标，在弹出对话框中设置串口相关参数，包括波特率（IO 模块默认出厂波特率为 9600），数据位设置为 8，停止位设置为 1，校验位设置为 none，一切就绪后，点击“确定”按钮：



设置好串口参数后，点“确定”按钮。

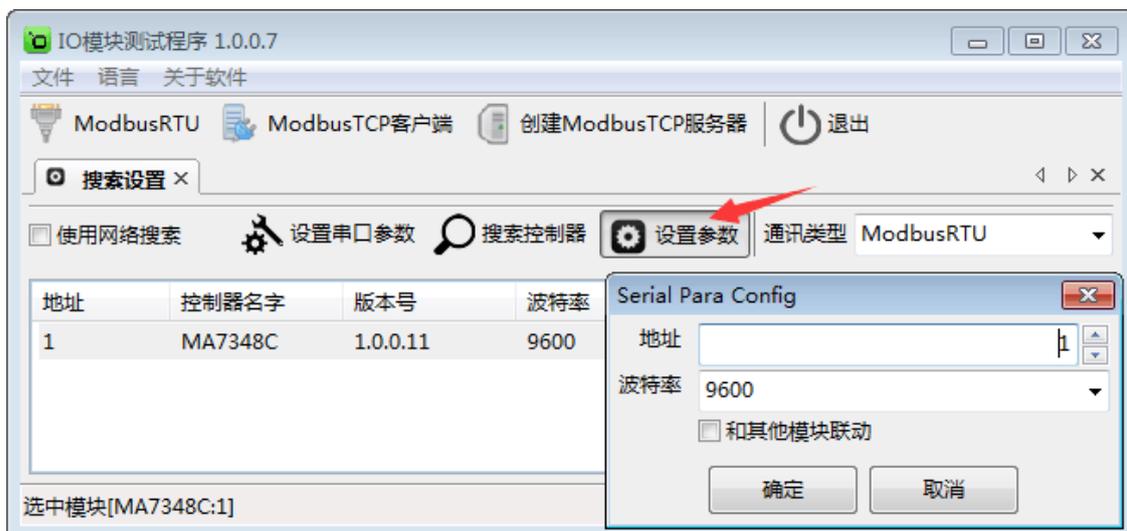
备注：如果已知 IO 模块的波特率（比如出厂的 9600），并且在上一步中设置了和 IO 模块匹配的波特率，则不用勾选“全波特率”复选框；如果未知 IO 模块的波特率，则需要勾选“全波特率”复选框，软件会从低波特率开始尝试搜索设备（1200bps~115200bps）。

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：

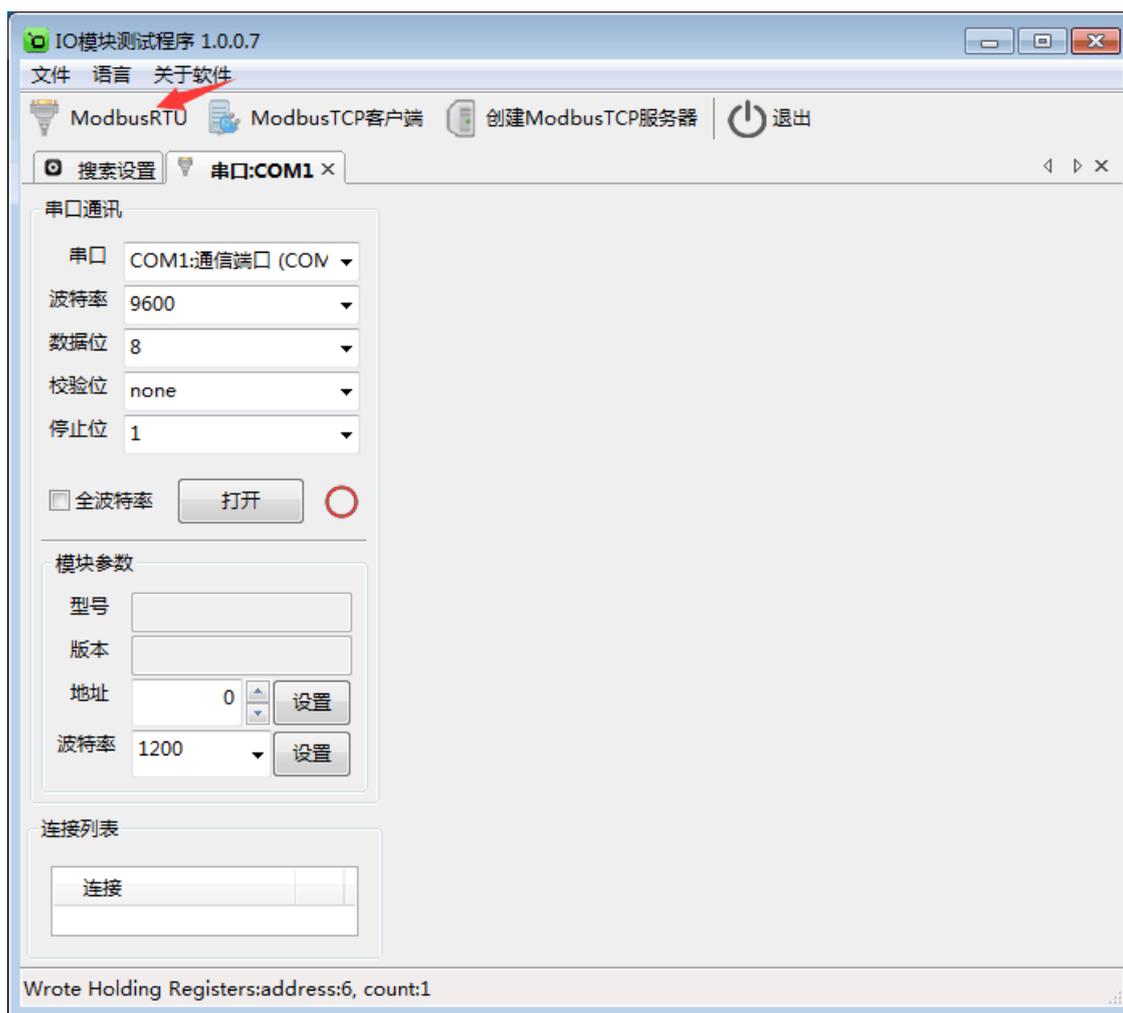


5.2 设置 IO 模块

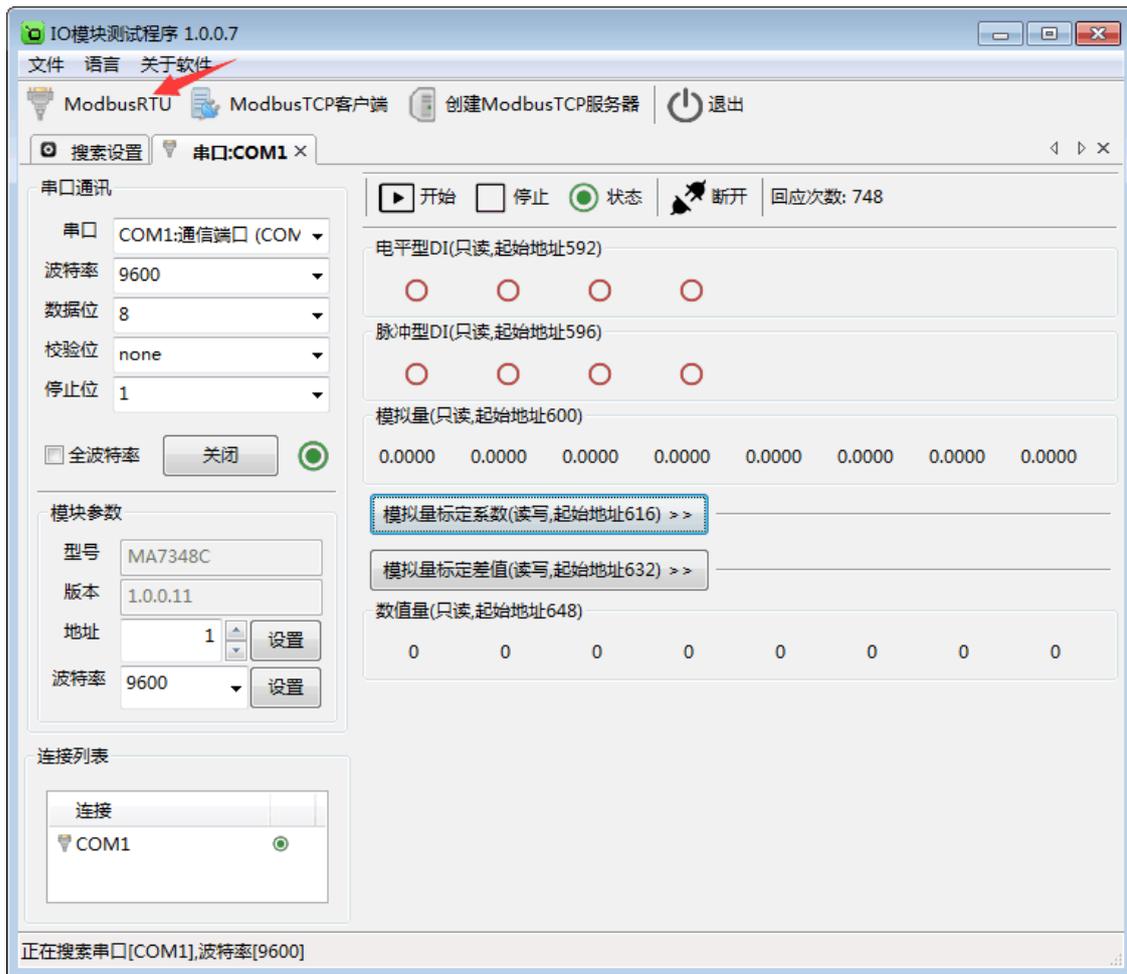
选中模块，点“设置参数”按钮，在弹出的对话框中设置 IO 模块的地址（范围是 1~255），在“波特率”下拉菜单中选择波特率，点“确定”按钮，参数设置成功。如下图：



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：



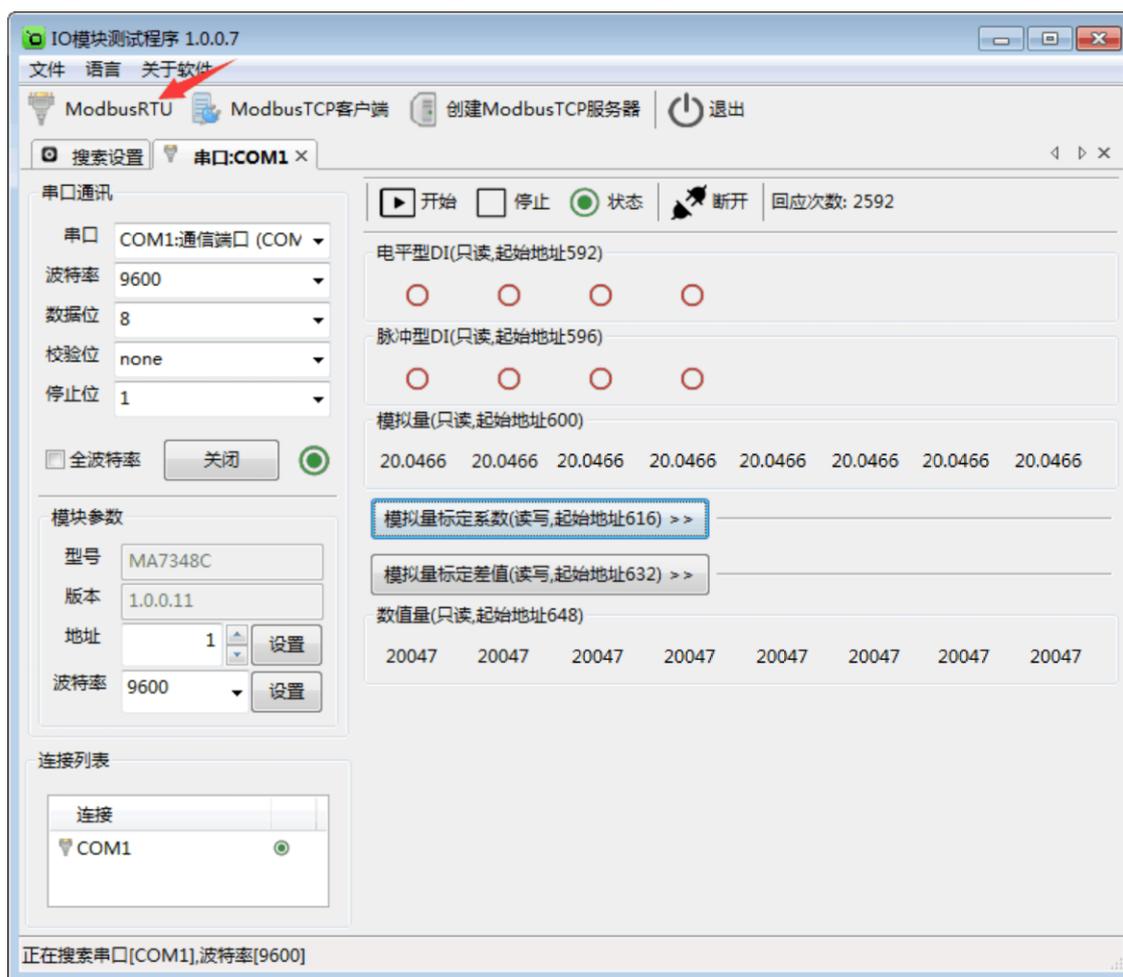
在弹出的对话框中再点击“打开”按钮、“模块参数”和“连接列表”等设置参数，且可以更改。如下图所示：



5.3 测试 IO 模块

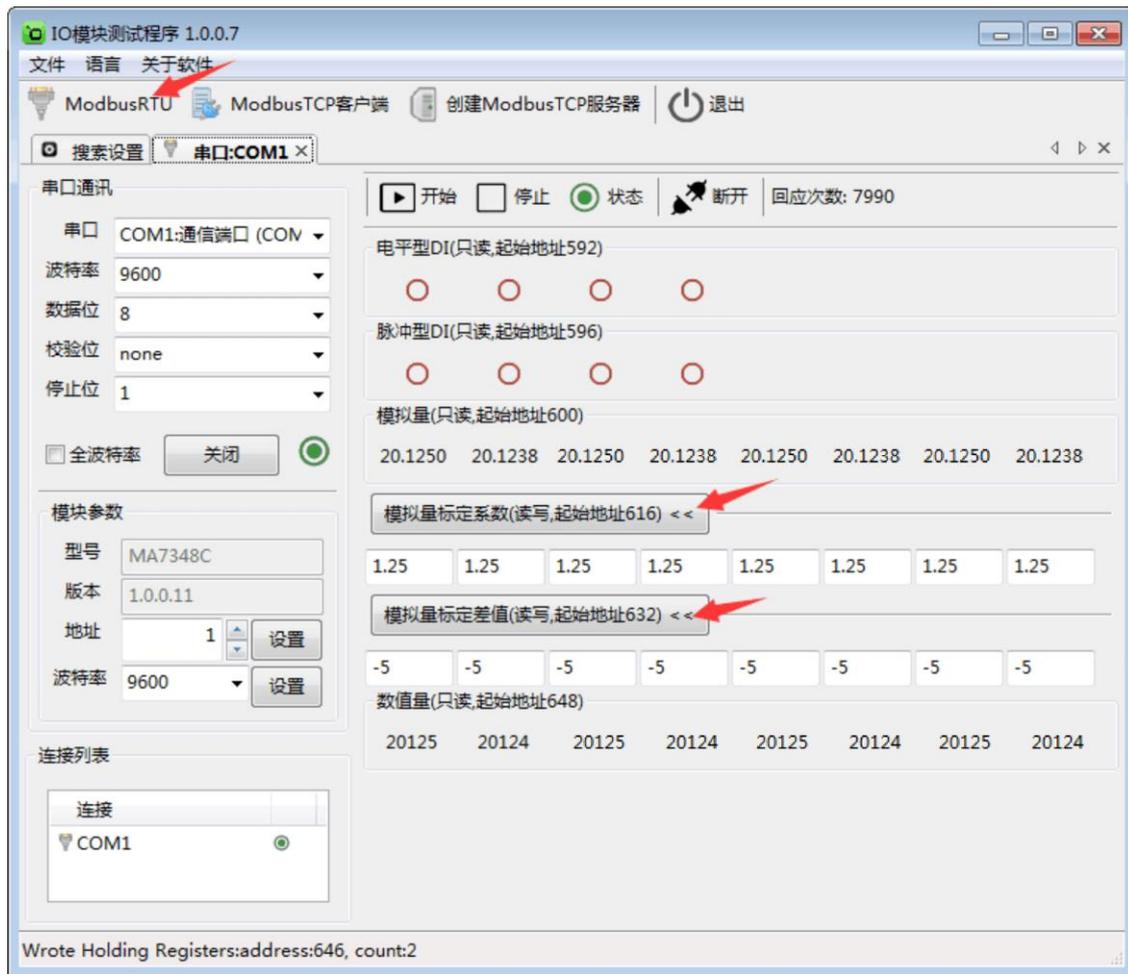
5.3.1 测试量程为 0~20mA

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。下面状态栏会有串口号和波特率的提示。采集 0~20mA 模拟量值时，模拟量标定系数默认为“1”，模拟量标定差值默认为“0”。右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 AI 的数值，AI 状态为只读值，显示模拟值（0~20mA）和数值量，可以很直观地看到它各路的状态。如下图：



5.3.2 测试量程为 4~20mA

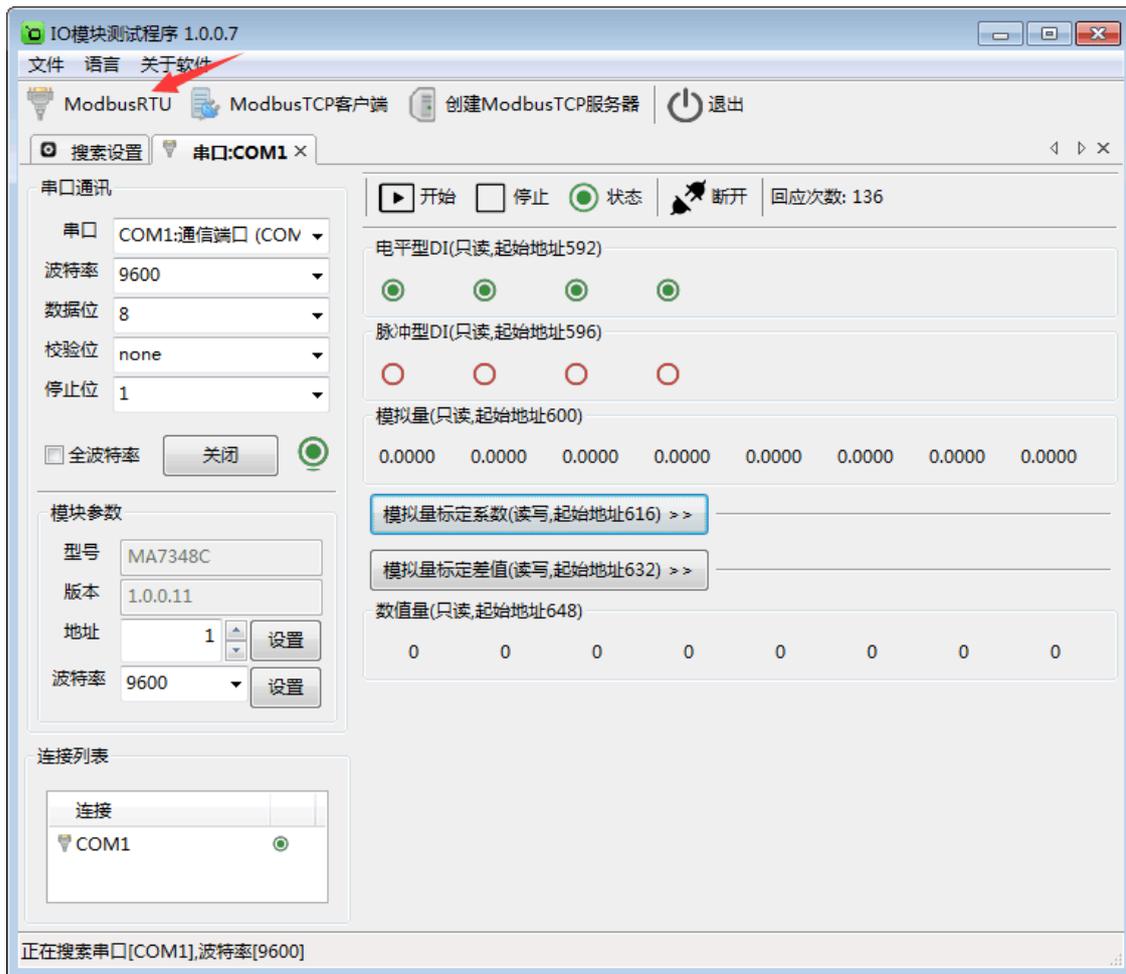
点击“打开”按钮，测试软件界面左下方显示“模块参数”和“连接列表”设置界面。下面状态栏会有串口号和波特率的提示。采集 4~20mA 模拟量值时，点击“模拟量标定系数”按钮，标定为“1.25”，点击“模拟量标定差值”按钮，标定为“-5”。右半边测试软件会根据产品型号自动显示所对应的测试界面，显示为采集 AI 的数值，AI 状态为只读值，显示模拟值（4~20mA）和数值量，可以很直观地看到它各路的状态。如下图：



备注：标定系数和标定差值的方法，分别在相应的输入框内填入需要标定的数值，直接敲击键盘“回车”键，即可保存。

5.3.3 开关量测试

点击“打开”按钮，测试软件界面左边显示“模块参数”和“连接列表”设置界面。下面状态栏会有串口号和波特率的提示。右边测试软件会根据产品型号自动显示所对应的测试界面，采集 DI 的数值，DI 的各路状态均为只读值，红色表示断开，绿色表示接通。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：



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 1 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
寄存器地址	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
寄存器地址	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 公共寄存器

寄存器地址	功能	种类	读写状态	取值范围
0	型号高位	保持寄存器	只读	
1	型号低位	保持寄存器	只读	
2	序列号高位	保持寄存器	只读	
3	序列号低位	保持寄存器	只读	
4	版本高位	保持寄存器	只读	
5	版本低位	保持寄存器	只读	
6	BOOT 版本高位	保持寄存器	只读	
7	BOOT 版本低位	保持寄存器	只读	
8	波特率高位	保持寄存器	读写	300~115200
9	波特率低位	保持寄存器	读写	
10	地址	保持寄存器	读写	0~255
11	型号名字 1	保持寄存器	只读	字符串格式的型号名字，例如：MA7348C
12	型号名字 2	保持寄存器	只读	
13	型号名字 3	保持寄存器	只读	
14	型号名字 4	保持寄存器	只读	
15	型号名字 5	保持寄存器	只读	
16	IO 模块寄存器基址	保持寄存器	只读	
17	IO 模块寄存器个数	保持寄存器	只读	
18	IO 模块配置总个数	保持寄存器	只读	
19	寄存器个数 1	保持寄存器	只读	某种类型的寄存器个数
20	寄存器类型 1	保持寄存器	只读	类型取值为：0 保持寄存器，1 线圈寄存器
21	寄存器属性 1	保持寄存器	只读	属性按位表示：1 可读，2 可写，3 可配置，4 电平型 DI，5 脉冲型 DI
22	是否浮点数 1	保持寄存器	只读	
...		保持寄存器	只读	
22+4N	寄存器个数 N	保持寄存器	只读	某种类型的寄存器个数
23+4N	寄存器类型 N	保持寄存器	只读	类型取值为：0 保持寄存器，1 线圈寄存器
24+4N	寄存器属性 N	保持寄存器	只读	属性按位表示：1 可读，2 可写，3 可配置，4 电平型 DI，5 脉冲型 DI
25+4N	是否浮点数 N	保持寄存器	只读	

6.2.2 MA7348V 寄存器

寄存器地址	功能	种类	读写状态	取值范围
-------	----	----	------	------



592	DI 电平输入 1	线圈状态	只读	0 表示无输入, 1 表示有输入
593	DI 电平输入 2	线圈状态	只读	0 表示无输入, 1 表示有输入
594	DI 电平输入 3	线圈状态	只读	0 表示无输入, 1 表示有输入
595	DI 电平输入 4	线圈状态	只读	0 表示无输入, 1 表示有输入
596	DI 脉冲输入 1	线圈状态	只读	0 表示无输入, 1 表示有输入
597	DI 脉冲输入 2	线圈状态	只读	0 表示无输入, 1 表示有输入
598	DI 脉冲输入 3	线圈状态	只读	0 表示无输入, 1 表示有输入
599	DI 脉冲输入 4	线圈状态	只读	0 表示无输入, 1 表示有输入
592	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
593	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
594	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
595	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
596	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
597	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
598	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
599	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
600	电流测量值 1	保持寄存器	读写	0~25.0MA, 浮点数表示
602	电流测量值 2	保持寄存器	读写	0~25.0MA, 浮点数表示
604	电流测量值 3	保持寄存器	读写	0~25.0MA, 浮点数表示
606	电流测量值 4	保持寄存器	读写	0~25.0MA, 浮点数表示
608	电流测量值 5	保持寄存器	读写	0~25.0MA, 浮点数表示
610	电流测量值 6	保持寄存器	读写	0~25.0MA, 浮点数表示
612	电流测量值 7	保持寄存器	读写	0~25.0MA, 浮点数表示
614	电流测量值 8	保持寄存器	读写	0~25.0MA, 浮点数表示
616	系数标定 1	保持寄存器	读写	0~65535
618	系数标定 2	保持寄存器	读写	0~65535
620	系数标定 3	保持寄存器	读写	0~65535
622	系数标定 4	保持寄存器	读写	0~65535
624	系数标定 5	保持寄存器	读写	0~65535
626	系数标定 6	保持寄存器	读写	0~65535
628	系数标定 7	保持寄存器	读写	0~65535
630	系数标定 8	保持寄存器	读写	0~65535
632	差值标定 1	保持寄存器	读写	0~25.0MA
634	差值标定 2	保持寄存器	读写	0~25.0MA
636	差值标定 3	保持寄存器	读写	0~25.0MA
638	差值标定 4	保持寄存器	读写	0~25.0MA
640	差值标定 5	保持寄存器	读写	0~25.0MA
642	差值标定 6	保持寄存器	读写	0~25.0MA
644	差值标定 7	保持寄存器	读写	0~25.0MA
646	差值标定 8	保持寄存器	读写	0~25.0MA
648	整数电流测量值 1	保持寄存器	只读	0~25000, 正整数表示
649	整数电流测量值 2	保持寄存器	只读	0~25000, 正整数表示
650	整数电流测量值 3	保持寄存器	只读	0~25000, 正整数表示
651	整数电流测量值 4	保持寄存器	只读	0~25000, 正整数表示
652	整数电流测量值 5	保持寄存器	只读	0~25000, 正整数表示
653	整数电流测量值 6	保持寄存器	只读	0~25000, 正整数表示

654	整数电流测量值 7	保持寄存器	只读	0~25000, 正整数表示
655	整数电流测量值 8	保持寄存器	只读	0~25000, 正整数表示

6.2.3 浮点数说明

浮点数数据格式与 IEEE-754 标准 (32) 有关, 长度 32 位。四个字节的浮点数传送顺序为先低字节后高字节。浮点数格式见表 A8:

表 A8 浮点数格式

D31	D30~D23	D22~D0
浮点数符号位	阶码	尾数

$$\text{浮点数的数值} = ((-1)^{\text{符号位}}) \times 1. \text{尾数} \times 2^{(\text{阶码}-127)}$$

浮点数需要使用两个寄存器表示, 假设电流测量 1 的值为 10.56mA, 则在内存表示的字节如下:

0xC3	0xF5	0x28	0x41
低内存地址		高内存地址	

在使用两个寄存器表示则是:

寄存器 1: 0x4128

寄存器 2: 0xF5C3

因为在传输过程中保持寄存器需要高位在前, 则传输的字节顺序为:

0x41	0x28	0xF5	0xC3
低内存地址		高内存地址	

6.3 协议应用范例

6.3.1 读 DI 输入寄存器命令举例

以下为读取 IO 模块 4 路 DI 的命令举例, 假定 IO 模块的地址为 1, 寄存器起始地址为 592 (十六进制为 0x250), 个数为 4, 上位机发送的数据如下 (十六进制表示):

01 01 02 50 00 04 3c 60

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;

02 50 起始寄存器，即是寄存器 592；

00 04 需要读取的寄存器个数，这里举例为 4 路 DI；

3c 60 CRC16 校验，从地址到数据域的校验，计算结果为 0x603c，因为要低在前，所以是 3c 60。

从机应答举例，假定 4 路 DI 状态状态分别：1100，则回应的数据如下（十六进制表示）：

01 01 01 03 11 89

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 4 个寄存器，不够 8 整除，所以字节数=寄存器个数/8+1=1；

03 各个寄存器的值，从低位开始对应的 DI 的第一路；

11 89 CRC16 校验，从地址到数据域的校验，计算结果为 0x8911，因为要低位在前，所以是 11 89。

6.3.2 读模拟量寄存器命令举例

以下为读取 IO 模块的 4 路电流浮点数举例，假定 4 路电流各个电流值分别为：1.20, 2.45, 5.10, 12.12，IO 模块地址为 1，则上位机发送的数据如下（十六进制表示）：

01 03 02 58 00 08 c4 67

01 IO 模块的地址，1 字节；

03 功能码：读取线圈状态的功能码；

02 58 起始寄存器，即是寄存器 600；

00 08 寄存器个数，8 个，每个浮点数占用了两个寄存器；

c4 67 CRC16 校验，从地址到数据域的校验，计算结果为 0x67c4，因为要低在前，所以是 c4 67。

如果一切正常，则从机应答的数据如下（十六进制表示）：

01 03 10 3f 99 99 9a 40 1c cc cd 40 a3 33 33 41 41 eb 85 8a d8

01 IO 模块的地址，1 字节；

03 功能码：读取线圈状态的功能码；

10 所有寄存器总共占用的字节数，十进制的 16；

3f 99 99 9a 40 1c cc cd 40 a3 33 33 41 41 eb 85 分别表示各路电流值，每 4 个字节表示一路，例如 3f 99 99 9a 表示第一路的电流为 1.20，以此类推。

8a d8 CRC16 校验，从地址到数据域的校验，计算结果为 0xd88a，因为要低在前，所以是 8a d8。



第7章 装箱清单

序号	名称	数量	单位	备注
1	主设备 LW MA7348V	1	台	
2	产品简易说明书	1	张	
3	合格证	1	张	