



NKWOD[®] 凌柯沃

LW T7004

4 路输出

网络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.....	3
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.....	9
5.3 测试 IO 模块.....	11
5.3.1 模块作为服务器模式.....	11

5.3.2 模块作为客户端模式.....	13
第 6 章 通讯协议及寄存器定义.....	17
6.1 通讯协议.....	17
6.1.1 读线圈状态.....	17
6.1.2 写单个线圈状态.....	18
6.1.3 写多个线圈状态.....	18
6.1.4 读保持寄存器.....	19
6.1.5 写单个保持寄存器.....	20
6.1.6 写多个保持寄存器.....	20
6.1.7 错误码表.....	21
6.2 寄存器定义.....	21
6.2.1 公共寄存器.....	21
6.2.2 T7004 寄存器.....	22
6.3 协议应用范例.....	23
6.3.1 读寄存器命令举例.....	23
6.3.2 写单个寄存器命令举例.....	24
6.3.3 写多个寄存器命令举例.....	24
第 7 章 装箱清单.....	26

第 1 章 产品概述

1.1 概述

LW T7004 为网络型智能数字量采集器，具有 4 路 C 型继电器输出数字量(DO)；电源及 RJ-45 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 TCP 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 TCP/IP 网络即可实现对远程数字量设备的数据采集和控制。

本产品采用标准 Modbus TCP 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 TCP 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 C 型继电器输出 (DO)
- DO 输出：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 TCP 通信协议，支持客户端和服务器模式
- 丰富的的指示灯，方便调试
- RJ-45 通信接口提供防雷保护
- 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

DO 接口	DO	4 路 C 型继电器
	触点容量	3A / 250VAC; 3A / 30VDC
网络通信参数	通讯接口	RJ-45



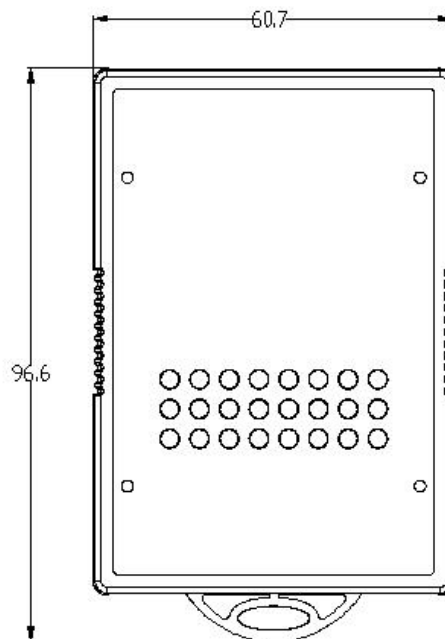
	速率	10/100M 自适应
	通讯协议	Modbus TCP
	嵌入协议	ARP, ICMP, IP, TCP, UDP, DHCP, DNS
	设置方式	设置程序
	防雷防护	250W
电源参数	电源规格	9-28VDC (推荐 12VDC)
	功耗	60mA@12VDC
	防雷防护	3000W
	端口压保护	30V (可自恢复)
工作环境	工作温度、湿度	-40~85°C, 5~90%RH, 不凝露
	储存温度、湿度	-60~125°C, 5~90%RH, 不凝露
其他	尺寸	87mm*58mm*26mm
	质保	6 年质保

第 2 章 外观尺寸

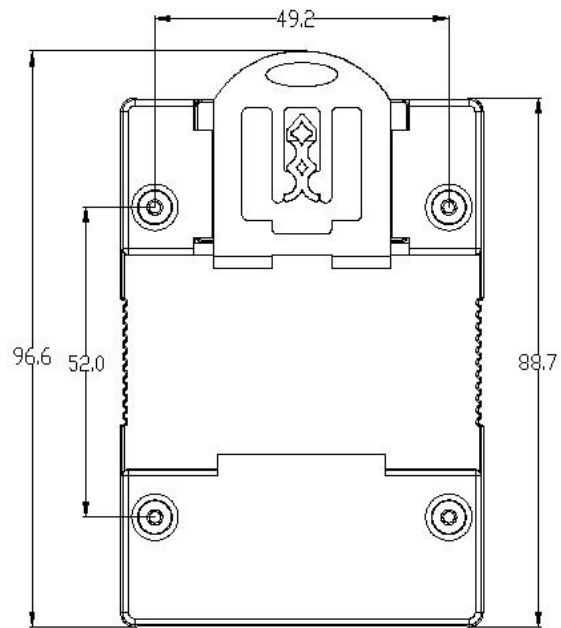
2.1 产品外观



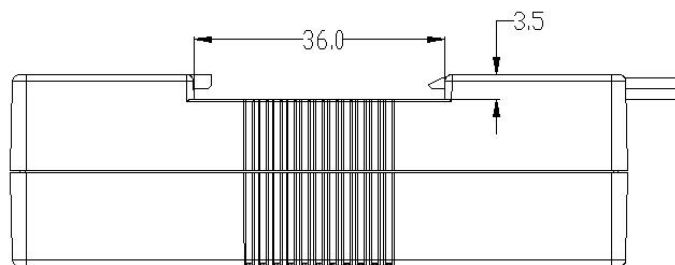
2.2.1 前视图



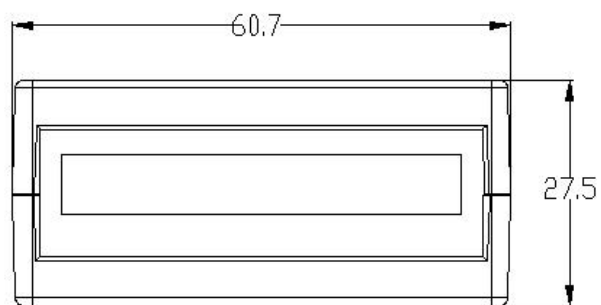
2.2.2 后视图



2.2.3 侧视图

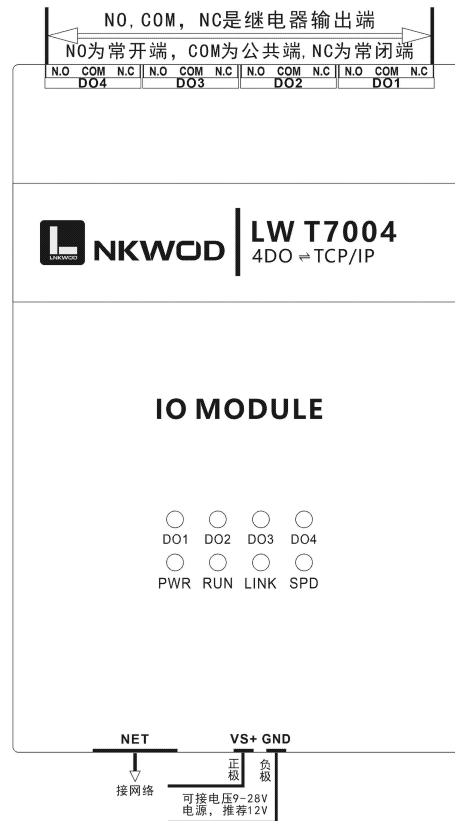


2.2.4 顶视图



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
NET	RJ-45 接口
DO1~4(COM)	数字量信号输出端
DO1~4(NC)	数字量信号常闭输出端
DO1~4(NO)	数字量信号常开输出端空

4.2 LED 指示灯

LW T7004 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：

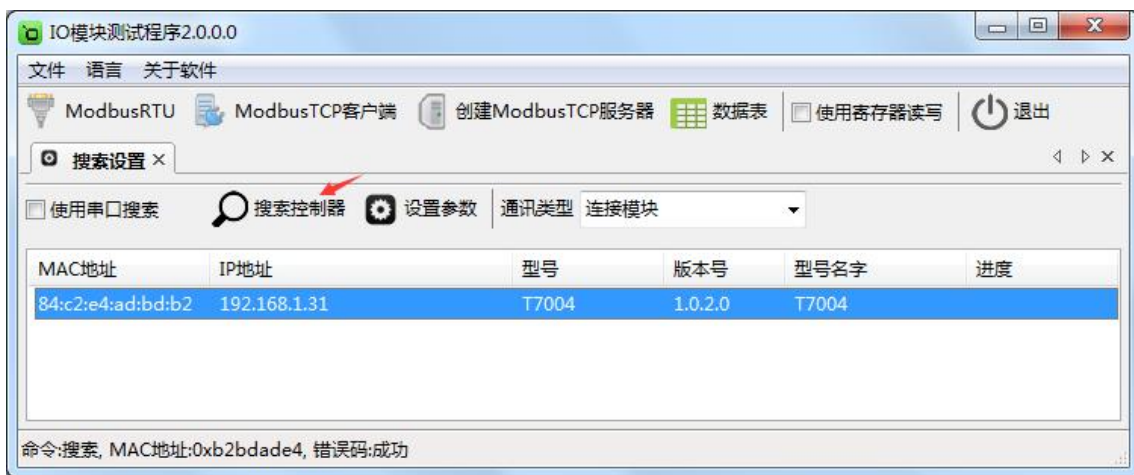
指示灯	指示灯说明
PWR	电源指示灯（亮：有电源连接；灭：无电源连接）
RUN	闪烁：正常运行；常亮或者不亮：工作不正常
LINK	亮：表示有网络连接，闪烁：表示有网络数据收发
SPD	亮：表示 100M 网速，不亮：表示 10M 网速
DO1~DO4	亮：对应 DO 有输出

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。

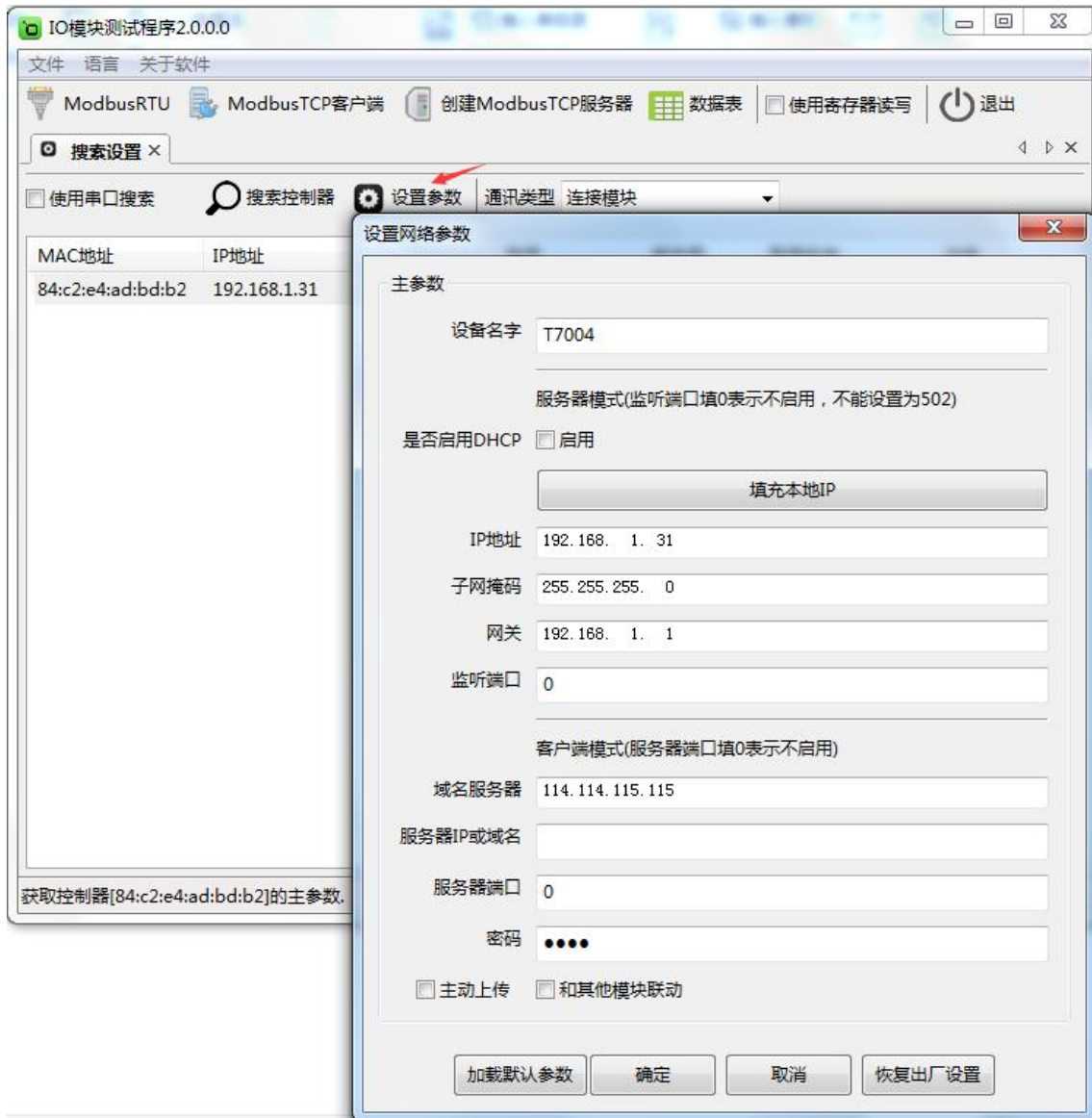
5.1 搜索 IO 模块

打开 IO 模块测试程序，该程序默认“使用网络搜索”，点击“搜索控制器”图标，局域网内的所有模块会展示在设备列表框中，页面会显示设备的参数包括 MAC 地址，IP 地址（IO 模块默认出厂 IP 地址为 192.168.1.31），型号，版本号，控制器名字。如下图：

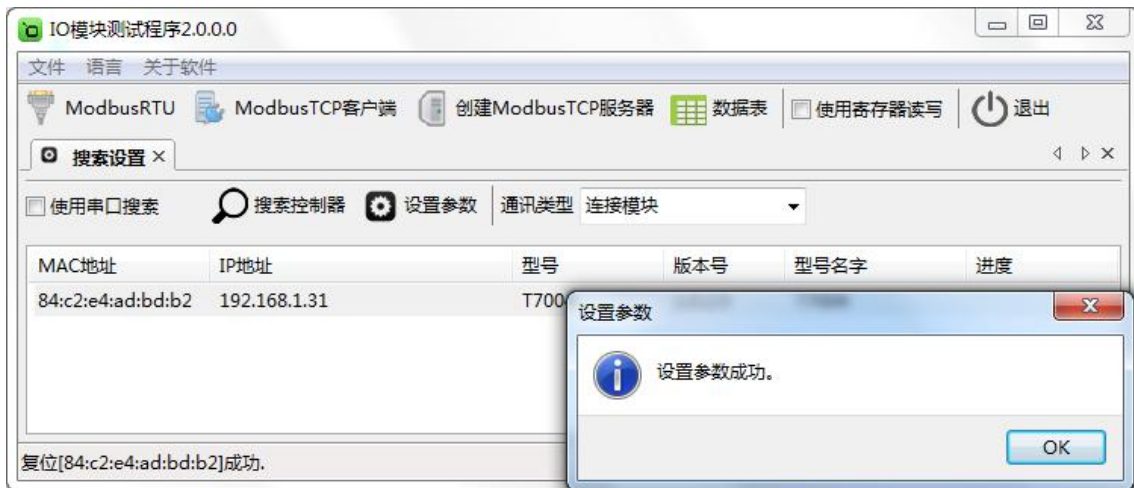


5.2 设置 IO 模块

选中模块，双击或者点击“设置参数”图标（双击设备列表中的模块会把 IP 地址自动导入“设置网络参数”界面，使用“设置网络参数”界面之前请确保要测试的模块 IP 地址与电脑在同一网段），该模块的默认参数会显示于“设置网络参数”界面中，按需要修改其参数，如下图：



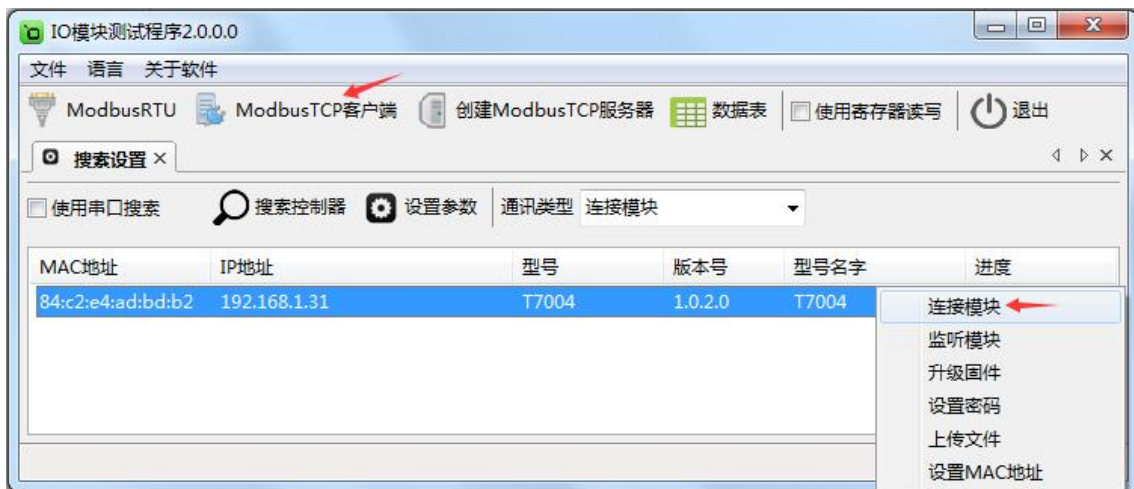
如果模块作为客户端连接到一个服务器，则“服务器IP或域名”和“服务器端口”也要设置，再点击“确定”图标，模块会保存新的参数并重启。如下图：



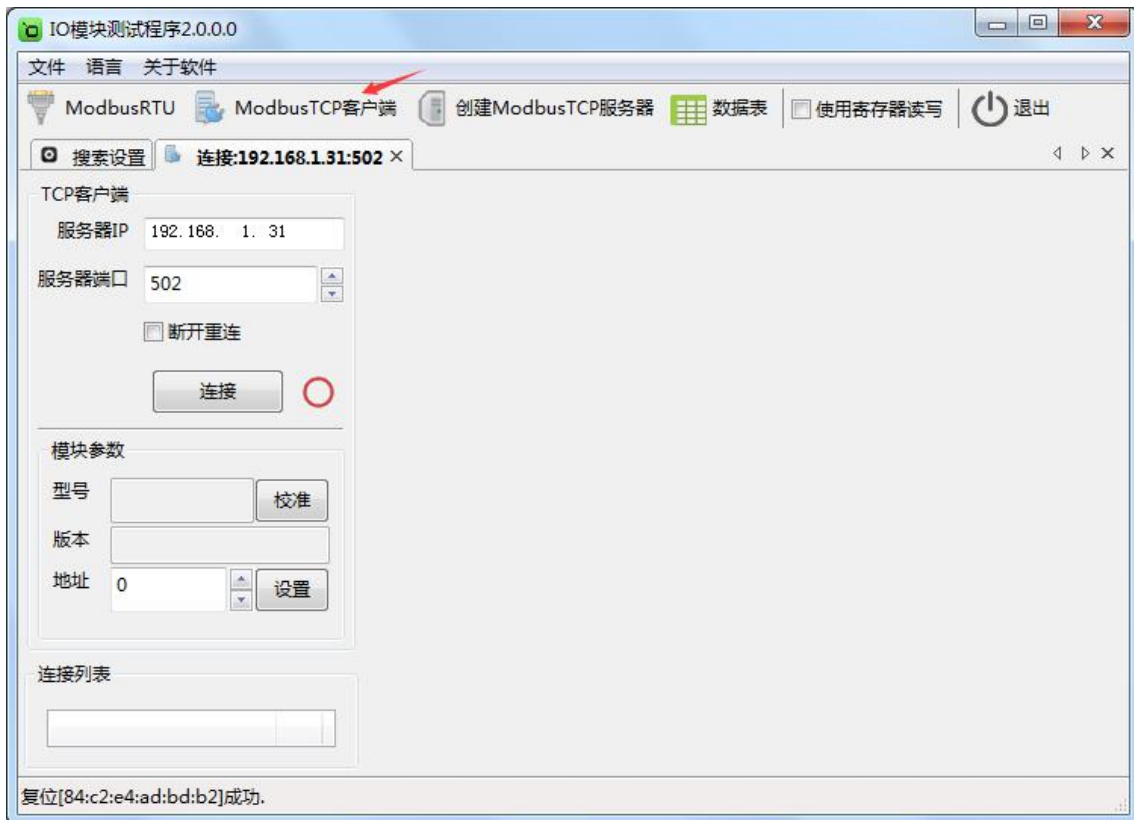
5.3 测试 IO 模块

5.3.1 模块作为服务器模式

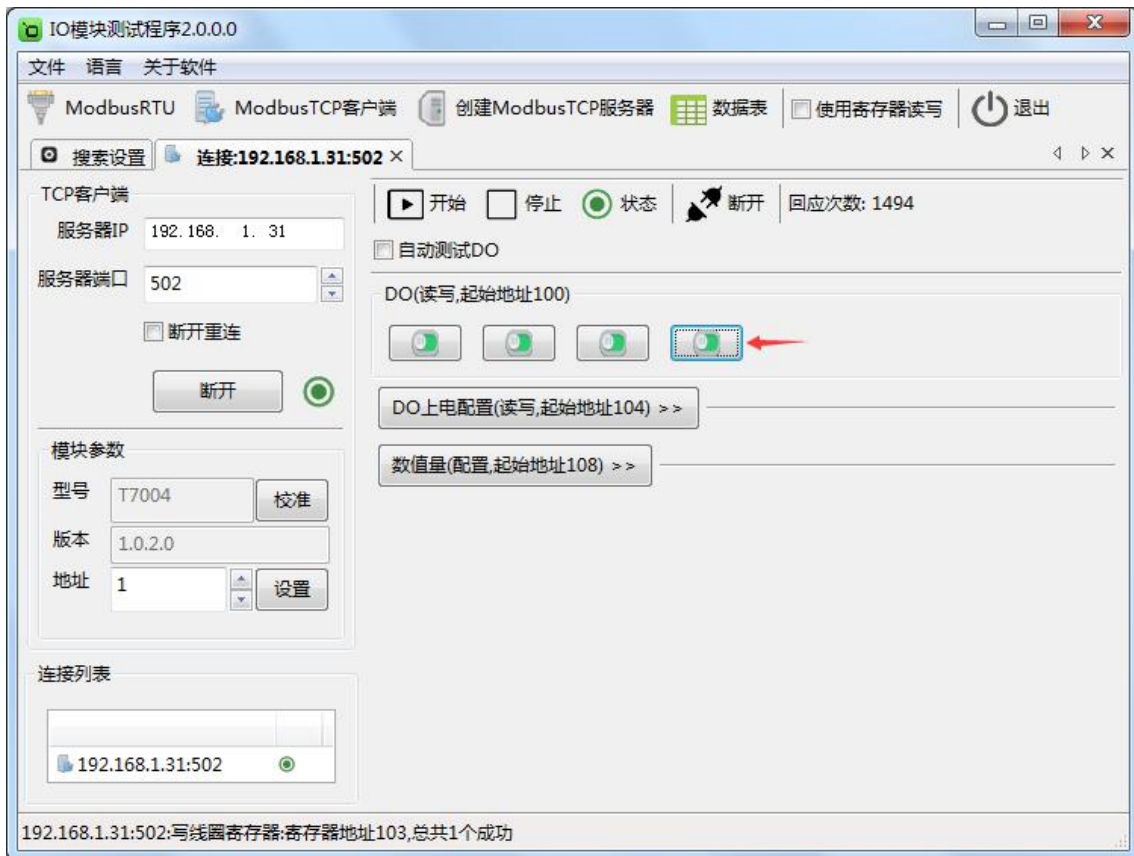
选中要设置的模块，点击“Modbus TCP 客户端”图标或者鼠标右键选择“连接模块”，如下图：



然后鼠标左键选择“连接模块”，如下图：



点击“连接”图标，模块网络连接建立之后，“连接”按钮会变为“断开”按钮，同时测试界面左半部分显示为设备的 TCP 客户端参数（IP 地址（默认 192.168.1.31），服务器端口（默认 502）），模块参数（型号，版本，地址），连接列表会显示所有该局域网的 IO 网络模块。测试界面右半部分为采集 DO 的数值，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：



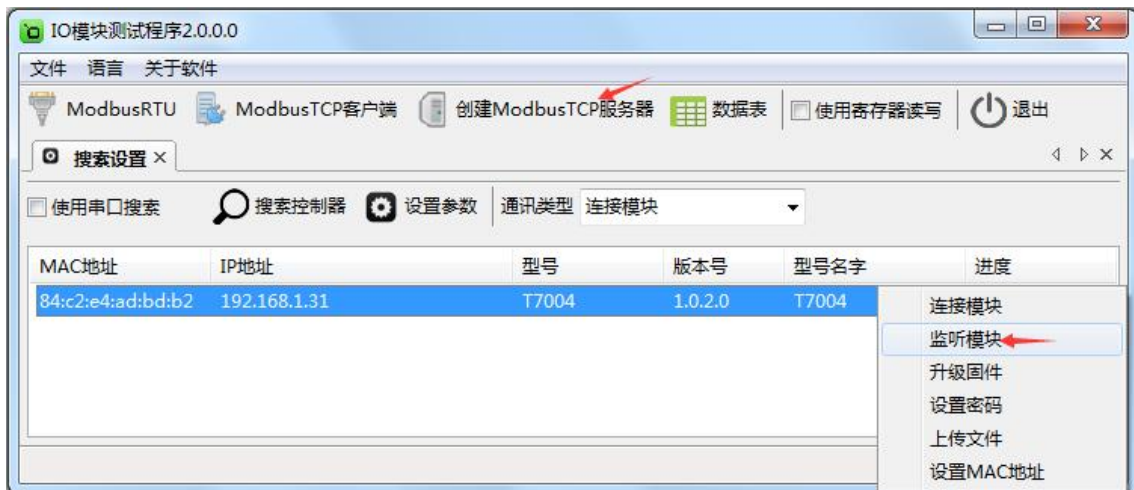
5.3.2 模块作为客户端模式

模块作为客户端连接到一个服务器，需设置“服务器IP或域名”和“服务器端口”参数，例：服务器IP或域名设为“192.168.1.122”，服务器端口设为“8008”，再点击“确定”图标，模块会保存新的参数并重启。如下图：

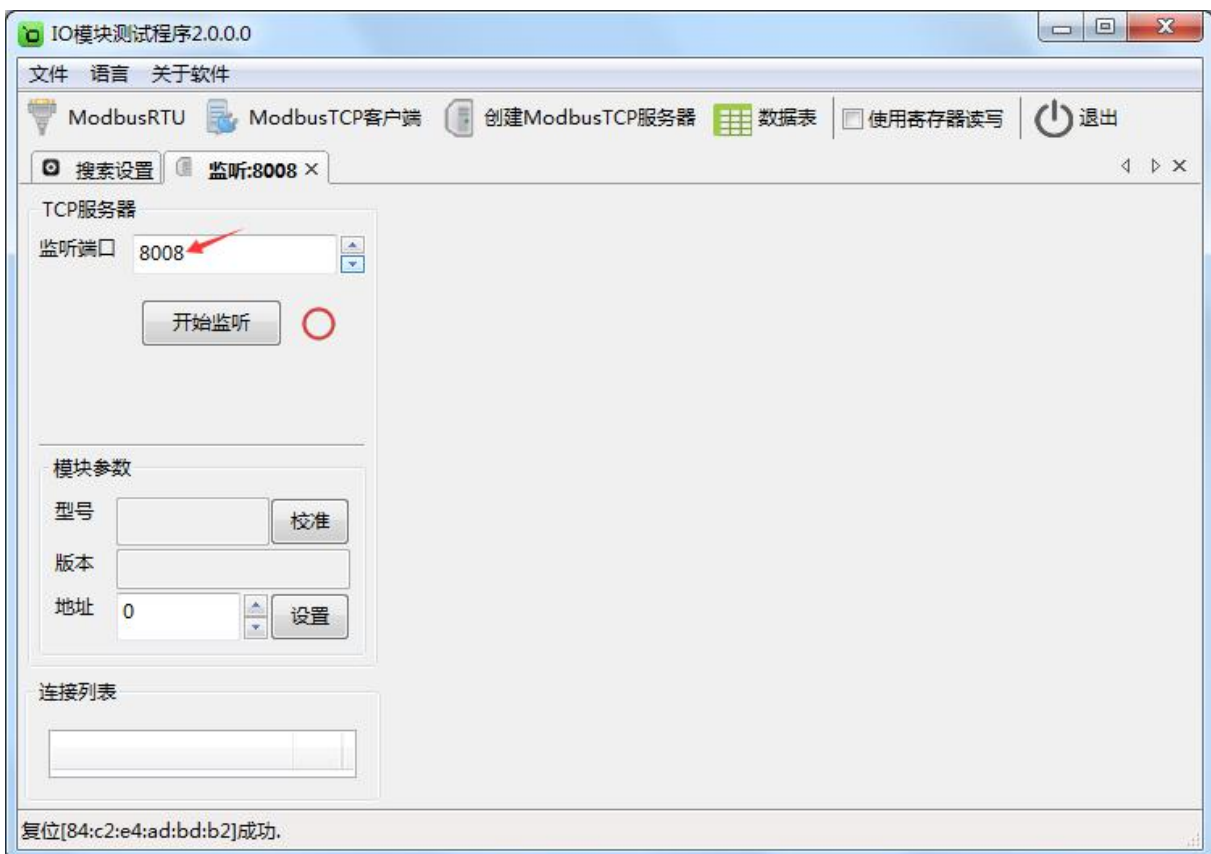
备注：如果在广域网中使用IO模块，需设置“域名服务器”参数。



选中要设置的模块,点击“创建 Modbus TCP 服务器”图标或者鼠标右键选择“监听模块”,如下图:

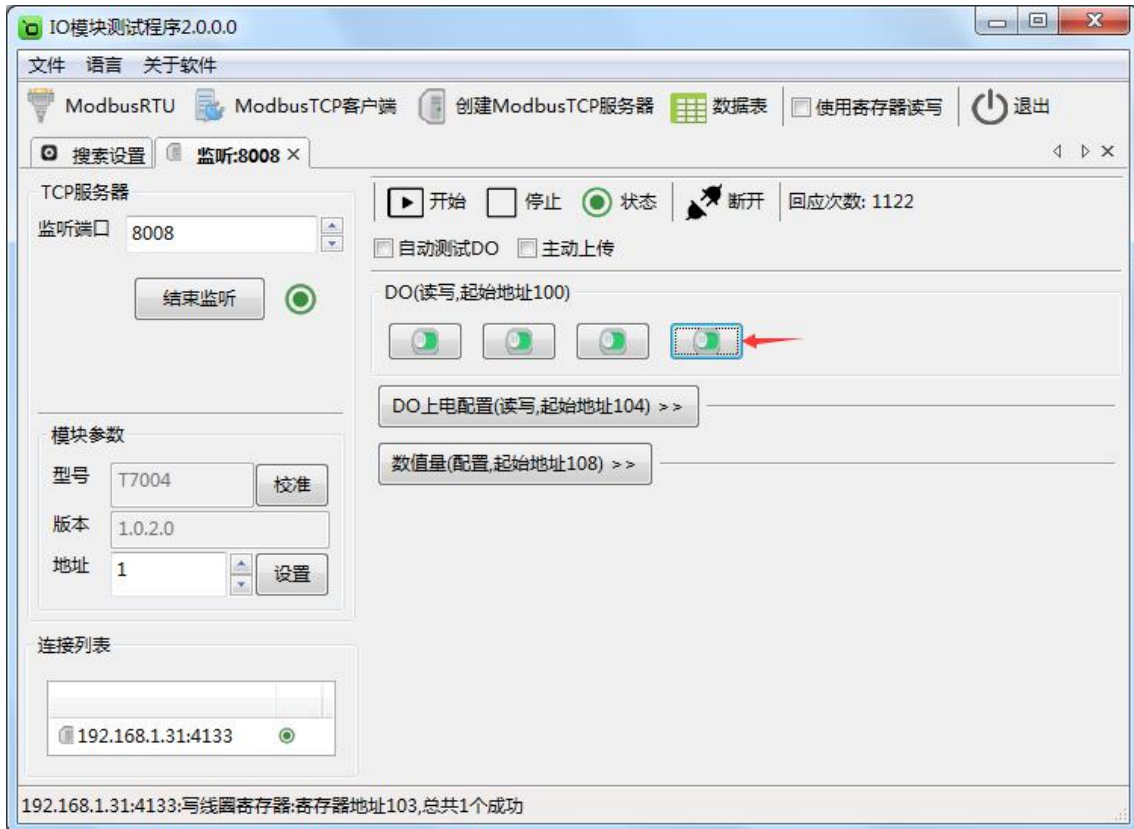


然后鼠标左键选择“监听模块”，如下图：



监听端口输入“8008”（注意：此端口号一定和服务器端口号一致，否则不能通讯），点击“开始监听”图标，模块网络连接建立之后，“开始监听”按钮会变为“结束监听”按钮，同时测试界面左半部分显示为设备的 TCP 服务器参数（监听端口“8008”），模块参数（型号，版本，地址），连接列表会

显示所有该局域网的 IO 网络模块。测试界面右半部分为采集 DO 的数值，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：



第 6 章 通讯协议及寄存器定义

6.1 通讯协议

遵循标准 MODBUS TCP 协议，协议格式如下：

传输标志	协议标志	长度	单元标志	功能码	数据
2 字节	2 字节	2 字节	1 字节	1 字节	N 字节

传输标志：MODBUS 请求和响应传输过程中序列号，客户端生成，应答时复制该值，高位在前；

协议标志：Modbus 协议默认为 0，高位在前；

长度：后续字节的长度，高位在前；

单元标志：从机标志（从机地址）；

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；

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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	保留	保持寄存器	读写	

9	保留	保持寄存器	读写	
10	地址	保持寄存器	读写	0~255, 单元标志
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: T7004
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 T7004 寄存器

寄存器地址	功能	种类	读写状态	取值范围
100	DO 输出 1	线圈状态	读写	0 表示无输出, 1 表示有输出
101	DO 输出 2	线圈状态	读写	0 表示无输出, 1 表示有输出
102	DO 输出 3	线圈状态	读写	0 表示无输出, 1 表示有输出
103	DO 输出 4	线圈状态	读写	0 表示无输出, 1 表示有输出
104	上电 DO 配置 1	线圈状态	读写	0 表示无输出, 1 表示有输出
105	上电 DO 配置 2	线圈状态	读写	0 表示无输出, 1 表示有输出
106	上电 DO 配置 3	线圈状态	读写	0 表示无输出, 1 表示有输出
107	上电 DO 配置 4	线圈状态	读写	0 表示无输出, 1 表示有输出
108	DO1 输出脉冲时间	保持寄存器	读写	0~65535ms
109	DO2 输出脉冲时间	保持寄存器	读写	0~65535ms
110	DO3 输出脉冲时间	保持寄存器	读写	0~65535ms
111	DO4 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 2 路电平型 DI 和 2 路脉冲型 DI 的命令举例，假定 IO 模块的地址为 1，寄存器起始地址为 100（十六进制为 0x64），个数为 4，上位机发送的数据如下（十六进制表示）：

00 01 00 00 00 06 01 01 00 64 00 04

各项分别表示：

00 01 传输标志，序列号，表示 0x0001；

00 00 协议标志，默认为 0x0000，表示 Modbus 协议；

00 06 后续字节长度，6 个字节；

01 IO 模块的地址，1 字节；

01 功能码：读取线圈状态的功能码；

00 64 起始寄存器，即是寄存器 100；

00 04 需要读取的寄存器个数，这里举例为 4 路，2 路电平型 DI 和 2 路脉冲型 DI；

从机应答举例，假定 2 路电平 DI 状态状态分别：1 0，脉冲型 DI 状态是电平型 DI 状态的脉冲表示，瞬间值为：1 0，则回应的数据如下（十六进制表示）：

00 01 00 00 00 04 01 01 01 05

各项分别表示：

00 01 传输标志，序列号，表示 0x0001；

00 00 协议标志，默认为 0x0000，表示 Modbus 协议；

00 04 后续字节长度，4 个字节；

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，所以字节数=寄存器个数/8+1=1；

05 各个寄存器的值，从低位开始对应的电平 DI 的第一路；

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 104(十六进制为 0x68),写 DO1 状态 1 的数据如下(十六进制表示):

00 01 00 00 00 06 01 05 00 68 ff 00

00 01 传输标志, 序列号, 表示 0x0001;

00 00 协议标志, 默认为 0x0000, 表示 Modbus 协议;

00 06 后续字节长度, 6 个字节;

01 IO 模块的地址, 1 字节;

05 功能码: 写线圈状态的功能码;

00 68 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;

如果执行正常, 从机应答数据如下(十六进制表示):

00 01 00 00 00 06 01 05 00 68 ff 00

00 01 传输标志, 序列号, 表示 0x0001;

00 00 协议标志, 默认为 0x0000, 表示 Modbus 协议;

00 06 后续字节长度, 6 个字节;

01 IO 模块的地址, 1 字节;

05 功能码: 写线圈状态的功能码;

00 68 寄存器地址, 高位在前, DO1 的寄存器;

ff 00 DO1 的状态返回值, ff00 表示 DO1 状态为 1, 00 00 表示 DO1 状态为 0;

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 104(十六进制为 0x68), 写从 DO1 到 DO2 的数据(全部输出)如下(十六进制表示):

00 01 00 00 00 08 01 0f 00 68 00 02 01 03

00 01 传输标志, 序列号, 表示 0x0001;

00 00 协议标志，默认为 0x0000，表示 Modbus 协议；

00 08 后续字节长度，8 个字节；

01 IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

00 68 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；

如果执行正常，从机应答数据如下（十六进制表示）：

00 01 00 00 00 06 01 0f 00 68 00 02

00 01 传输标志，序列号，表示 0x0001；

00 00 协议标志，默认为 0x0000，表示 Modbus 协议；

00 06 后续字节长度，6 个字节；

01 IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

00 68 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；



第 7 章 装箱清单

序号	名称	数量	单位	备注
1	主设备 LW T7004	1	台	
2	产品简易说明书	1	张	
3	合格证	1	张	