

智嵌物联 Smart IO 系列（网络版）产品使用手册



深圳总部

地址：广东省深圳市宝安区新桥街道新桥社区

新和大道 6-18 号 1203

网址：www.zhiqw.com

电话：0755-23203231

北京办事处

地址：北京市房山区德润街6号院8号楼3层

电话：18210365439



天猫店铺



淘宝店铺



京东店铺



微信公众号



公司官网

修订历史

版本	日期	原因
V1.0	2021.09.11	发布
V1.1	2022.08.18	1、替换封面图片 2、增加 NET-AA80 硬件接口及尺寸

目 录

1	关于手册.....	1
2	产品快速入门.....	2
1.1	功能简介	2
1.2	功能特点	2
1.3	快速使用说明.....	3
1.3.1	使用前硬件准备.....	3
1.3.2	使用配置软件控制.....	3
1.3.3	使用网络调试助手控制.....	6
1.3.4	串口服务器功能测试.....	7
1.4	设备参数配置.....	9
1.4.1	设备的基本参数配置.....	9
1.4.2	设备控制.....	9
1.4.3	DI 输入消抖配置	10
1.4.4	自动上传设置.....	10
1.4.5	联动策略设置.....	11
1.4.6	定时控制.....	12
1.4.7	通信超时.....	13
1.4.8	掉电检测.....	13
1.4.9	电机控制.....	14
1.4.10	输出反馈.....	15
1.4.11	频闪控制.....	16
1.4.12	顺序控制.....	16
1.4.13	用户代码.....	17
2.	产品规格	18
2.1	产品参数	18
2.1.1	电气参数.....	18
2.1.2	工作环境参数.....	18
2.1.3	EMC 防护等级	18
2.1.4	继电器规格.....	19
2.1.5	数字量输入参数.....	19
2.1.6	ADC 模拟量输入参数.....	20
2.1.7	通信参数.....	20
3.	设备接口接线说明	21
3.1	产品硬件接口布局.....	21
3.1.1	NET-8800M 硬件接口及尺寸	21
3.1.2	NET-8880D 硬件接口及尺寸.....	22
3.1.3	NET-AA80M 硬件接口及尺寸	23
3.1.4	NET-BC84D 硬件接口及尺寸	24
3.2	电源接口	25
3.3	RS485 接口	25
3.4	RJ45 网口	25
3.5	拨码开关	26

3.6	恢复出厂设置.....	26
3.7	产品指示灯	26
3.8	模拟量输入接口.....	27
3.9	数字量输入接口.....	28
3.9.1	干节点接线方式.....	28
3.9.2	湿节点接线方式.....	28
3.9.3	NPN 型传感器接线方式	29
3.9.4	PNP 型传感器接线方式.....	29
3.10	继电器输出接口.....	30
4.	产品功能	31
4.1	工作模式	31
4.1.1	TCP/UDP 连接	31
4.1.2	连接 MQTT 服务器.....	33
4.1.3	HTTP Post 控制	34
4.1.4	Web Socket 控制.....	34
4.1.5	串口数据透传.....	35
4.2	注册包、心跳包.....	35
4.2.1	心跳包	35
4.2.2	注册包	36
4.3	DI 输入信号检测	36
4.4	DI 脉冲计数	37
4.5	继电器输出（DO）	37
4.6	模拟量测量	38
4.7	联动控制	39
4.8	自动上报	41
4.9	RTC 定时	41
4.10	ModBus TCP 转 RTU	42
4.11	二次开发	42
5.	控制协议	43
5.1	ASCII 格式	43
5.1.1	设置 DO 继电器输出状态.....	43
5.1.2	设置单路 DO 继电器状态.....	43
5.1.3	设置 DO 继电器延时断开时间.....	44
5.1.4	读取设备所有接口状态.....	45
5.1.5	只读取 DI 状态	45
5.1.6	只读 DO 继电器状态.....	46
5.1.7	只读模拟量 AI 值	46
5.1.8	只读 DI 的脉冲计数值	46
5.1.9	ASCII 格式主动上传	47
5.2	JSON 格式	47
5.2.1	设置设备的所有输出状态.....	47
5.2.2	读取设备所有接口状态.....	48
5.2.3	JSON 格式主动上报	49
5.3	ModBus RTU 格式.....	50
5.3.1	地址域（Additional address）	50
5.3.2	功能码（Function code）	50

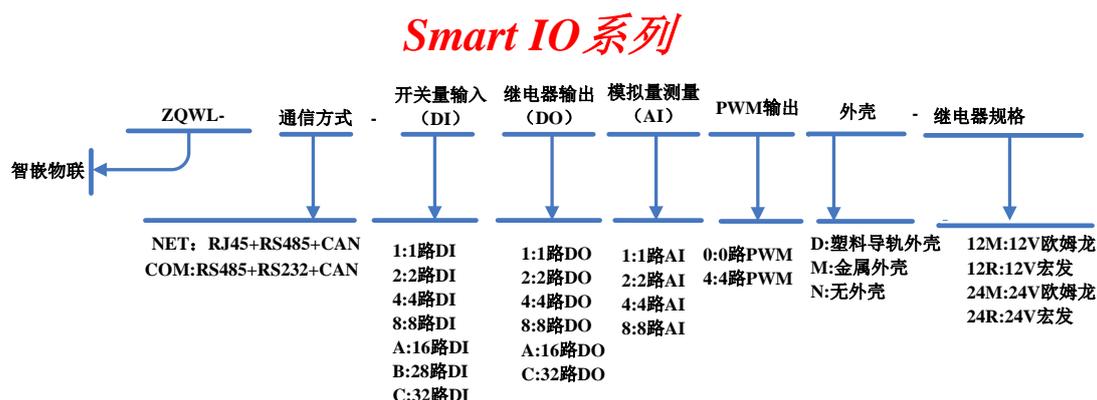
5.3.3	ModBus 格式主动上报	56
5.4	自定义指令格式.....	56
5.4.1	自定义格式主动上报.....	60
6.	固件升级	62
7.	恢复出厂设置	63
8.	应用案例	64
8.1	设备接入智嵌云平台实现远程控制案例.....	64
8.2	设备 DI、DO、AI、脉冲计数主动上报.....	70
8.3	设备与组态王等组态软件通信.....	75
8.4	设备 MQTT 协议测试案例.....	83
	销售网络.....	89

1 关于手册

本手册适用型号：ZQWL-NET-8880D-12M、ZQWL-NET-8880D-12R、ZQWL-NET-8880D-24M、ZQWL-NET-8880D-24R；ZQWL-NET-BC84D-12M、ZQWL-NET-BC84D-12R、ZQWL-NET-BC84D-24M、ZQWL-NET-BC84D-24R。文档中若无特别说明，所介绍功能均适用于以上型号。

1. 命名规则：

智嵌物联 Smart IO 系列产品命名规则如图 1.1 所示。



如：ZQWL-NET-BC84D-12M

图 1.1 命名规则

2. 相关资料下载

二次开发资料下载：[点击下载](#)

配置工具下载：[点击下载](#)

接线说明下载地址：[点击下载](#)

2 产品快速入门

1.1 功能简介

智嵌物联深耕工业控制领域，为满足客户各种工业控制需求，特推出了 Smart IO 系列产品。

Smart IO 系列产品均支持二次开发，提供丰富的 SDK，用户只需关心项目的业务逻辑，而无需关心设备内部繁杂的联网机制，可极大缩短用户项目周期。

Smart IO 系列产品均内置 RTC 时钟，可实现脱机定时控制，再也不用担心因网络不稳定而造成的误控。

Smart IO 系列均支持 JSON、ASCII、标准 ModBus、自定义格式等多种控制指令格式，用户可选用适合自己的控制协议，让控制指令不再单一。

Smart IO 系列支持设备的级联扩展，用户可根据不同的项目需求任意组合不同功能型号的产品。

Smart IO 系列按通信方式不同，分为网络版本（net）和串口版本（com）两个子系列型号，用户可根据需要选择合适的型号，本文档主要介绍网络版本的 Smart IO 系列的主要使用方法。

Smart IO 系列网络版本（net），外围接口丰富：32 路继电器输出（DO）、28 路数字量输入（DI）、4 路 PWM 输出、8 路模拟量输入测量（AI）、1 路 RS485、1 路 CAN、1 路以太网等。外围接口、业务逻辑可根据用户需求进行定制。

1.2 功能特点

- ◆ 电源具有防反接功能。
- ◆ ARM9、Linux 系统。
- ◆ 最大支持 32 路继电器输出，继电器品牌、继电器线圈电压可选。
- ◆ 最大支持 28 路数字量输入检测，支持干节点、湿节点、NPN、PNP 接线方式。
- ◆ 最大支持 4 路 PWM 输出，频率范围 10Hz~500KHz，占空比 0%~100%可调。
- ◆ 最大支持 8 路模拟量输入测量（兼容 0~5V、0~10V，0~20mA）。
- ◆ 支持网络控制、RS485 总线控制、CAN 总线控制。
- ◆ 内置 RTC，支持脱机定时控制。
- ◆ 支持掉电检测，可保存异常掉电前的继电器状态、脉冲计数值。
- ◆ 支持频闪控制功能。
- ◆ 支持顺序控制，用户可按照业务需求编写自己的控制逻辑。
- ◆ 支持控制直流电机、220V 交流电机。
- ◆ 支持串口服务器功能、ModBus TCP 转 RTU 功能、ModBus 主动轮询，定时上报。
- ◆ 支持主动上报、定时控制、条件控制、一对多联动控制、延时断开、脉冲计数等。
- ◆ 支持 JSON、ASCII、ModBus、自定义格式等多种控制指令格式。
- ◆ 支持同时连接 2 个 TCP/UDP 服务器、1 个 MQTT 服务器。
- ◆ 支持 websocket、HTTP Post，提供相应控制 Demo。

- ◆ 支持心跳包、注册包。
- ◆ 支持接入智嵌云、阿里云、阿里云物模型。
- ◆ 支持 APP 控制。
- ◆ 支持二次开发，提供二次开发 SDK。
- ◆ 丰富的 LED 状态指示灯，快速定位问题。
- ◆ 支持本地升级。

1.3 快速使用说明

通过该设备，用户可以实现远程控制、远程检测、本地控制、本地检测及本地数据上云的功能。本节以远程控制功能为例，简要说明本设备的使用方法。

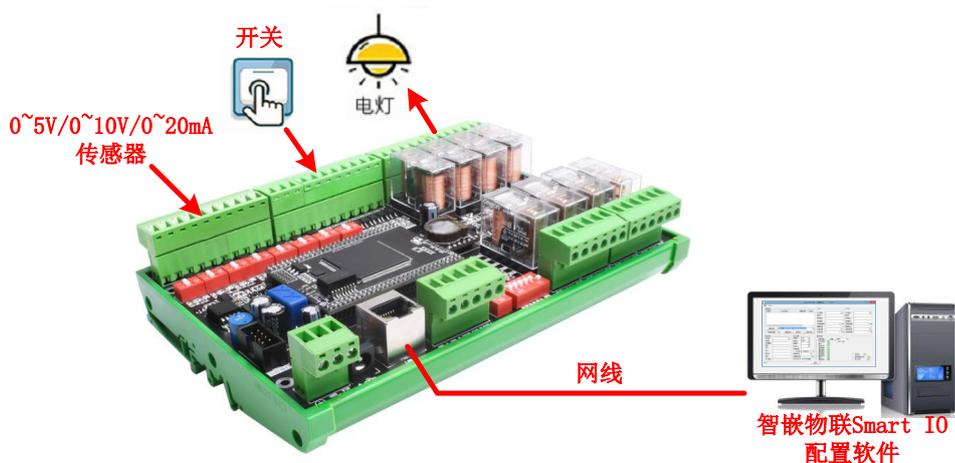


图 2.1 远程控制应用示例

注意，测试前请务必检查电源适配器的电压是否与控制板供电电压相符合。本文档以 12VDC 供电为例说明。

所需要的测试软件可以到官网下载：<http://www.zhiqwl.com/>

1.3.1 使用前硬件准备

需要以下硬件：

- 待测设备一台；
- DC12V 2A 电源适配器一个；
- 网线一根；



图 2.2 硬件准备

1.3.2 使用配置软件控制

本公司提供有测试用的控制软件，第一次使用时建议使用该智嵌物联的配置软件来控

制。

控制板的出厂默认参数如表 2.1 所示。

表 2.1 设备默认参数

项目	参数	说明
用户名	admin	此两项用于网页登录
密码	admin	
IP 地址	192.168.1.253	
子网掩码	255.255.255.0	
网关	192.168.1.1	
工作模式	TCP_SERVER	
本地端口	502	
RS485 波特率	9600	
RS485 参数	None/8/1	

3. 电脑 IP 设置

测试之前必须保证电脑的 IP 是和设备 IP 处于一个网段内，如果不在一个网段内，需要重新设置电脑的 IP 地址（静态 IP），如图 2.3 所示。

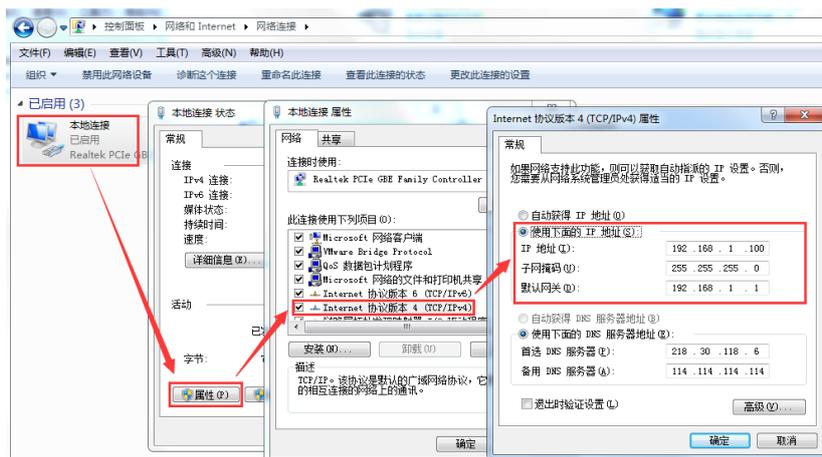


图 2.3 电脑 IP 地址设置

4. 硬件连接

用网线将电脑的网口和设备的网口连起来，并接上电源适配器(注意，“VCC”接电源正极（红线），“GND”接电源负极（黑线）。

上电后观察指示灯状态，如表 2.2 所示。待 RUN 灯闪烁、LINK 灯常亮、ACT 灯闪烁之后，再进行下一步操作。

表 2.2 指示灯基本含义

面板丝印	含义	亮	灭
PWR	电源指示灯	电源正常	电源异常
RUN	运行指示灯	正常运行时，亮灭频率约 1Hz	
485	RS485 数据指示灯	当 485 有数据时，该灯闪烁	

CAN	CAN 数据指示灯	当 CAN 有数据时，该灯闪烁	
LINK	网络连接指示灯	连接到网络时，常亮	
ACT	网络数据指示灯	网络有数据时，闪烁	
X1~X28	28 路开关量输入指示	外部开关量输入触发	外部开关量输入无触发
Y1~Y32	32 路继电器输出指示	继电器常开触点闭合	继电器常开触点断开
PWM1~PWM4	4 路 PWM 输出指示	有波形输出时，灯常亮或 闪烁	无波形输出时，灯不亮

5. 搜索设备

打开 Smart IO-NET 配置软件，下载地址：[点击下载](#)

点配置软件中的“搜索设备”按钮，如果硬件连接正常并且 IP 设置正常，则会搜到设备，如图 2.4 所示。



图 2.4 搜索设备

6. 对设备进行控制

选中搜到的设备，点击软件界面中的【控制】按钮，在弹出来的界面，可对每路继电器输出进行控制，观察每一路 DI 状态、脉冲计数值、模拟量值，如图 2.5 所示。

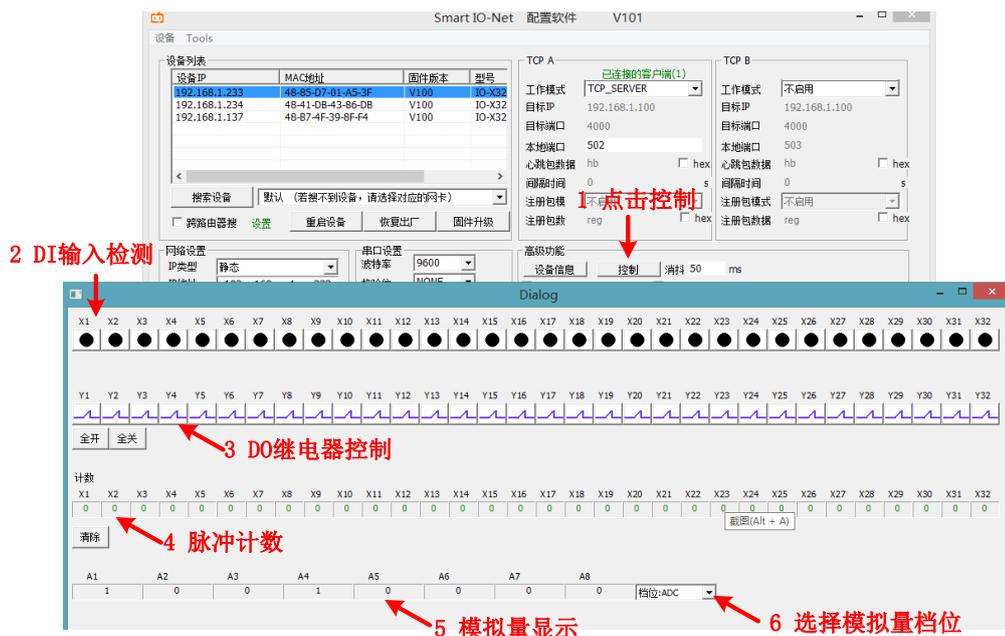


图 2.5 控制设备

说明：

开关量输入：X1~X32 即为控制板的 DI 输入状态，红色表示无信号，绿色表示有信号；

开关量输出：Y1~Y32 即为控制板的 DO 继电器输出状态，红色表示继电器的常开触点与公共端断开，绿色表示继电器的常开触点与公共端闭合，可以通过单击来改变状态。

脉冲计数：X1~X32 下面的数字即为每路 DI 的脉冲计数值。

模拟量：A1~A8，可以通过下拉菜单，选择模拟量的档位，来显示不同档位的模拟量值。此处选择的模拟量档位，要和设备上拨码开关选择的模拟量档位一致。

1.3.3 使用网络调试助手控制

通过网络调试助手向设备发送控制指令，即可实现对设备的控制，指令格式详见第 5 章介绍。

打开网络调试助手，在网络助手的“协议类型”下拉列表中，选择“TCP Client”（控制板的工作模式是 TCP SERVER）；将“服务器 IP 地址”一栏中输入设备的 IP 地址：192.168.1.253。在“服务器”端口一栏中输入控制板的本地端口：502。以上都设置好后，点击“连接”，连接成功后，连接按钮的状态将变成红色灯，如图 2.6 所示。

如图 2.8 所示。



图 2.8 串口调试助手

3. 打开网络调试助手

选择“TCP Client”（控制板的工作模式是 TCP SERVER）；将“服务器 IP 地址”一栏中输入设备的 IP 地址：192.168.1.253。在“服务器”端口一栏中输入控制板的本地端口：502。以上都设置好后，点击“连接”，连接成功后，连接按钮的状态将变成红色灯，如图 2.9 所示。



图 2.9 网络调试助手

4. 数据透传成功

分别点击网络调试助手和串口调试助手的【发送】按钮，可看到数据会成功发送和接收，

如图 2.10 所示。



图 2.10 数据透传成功

1.4 设备参数配置

1.4.1 设备的基本参数配置

使用上位机软件，用户可以查看设备的版本信息，并可设置设备的所有参数，比如工作模式、服务器 IP 端口号、心跳包、注册包、串口参数、以及一些高级功能。如图 2.11 所示。保存参数后，新参数即可生效。

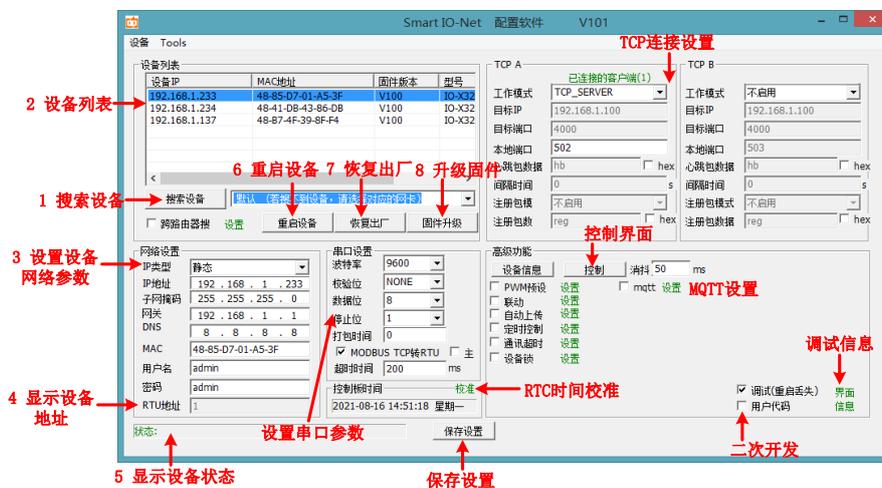


图 2.11 配置设备的基本参数

1.4.2 设备控制

用户可使用该上位机软件对设备的 DI、DO、AI、脉冲计数等接口进行控制及检测。如图 2.12 所示，在此界面下，可以观察 DI 状态、脉冲计数值、AI 模拟量值，也可对每一路继电器进行控制。

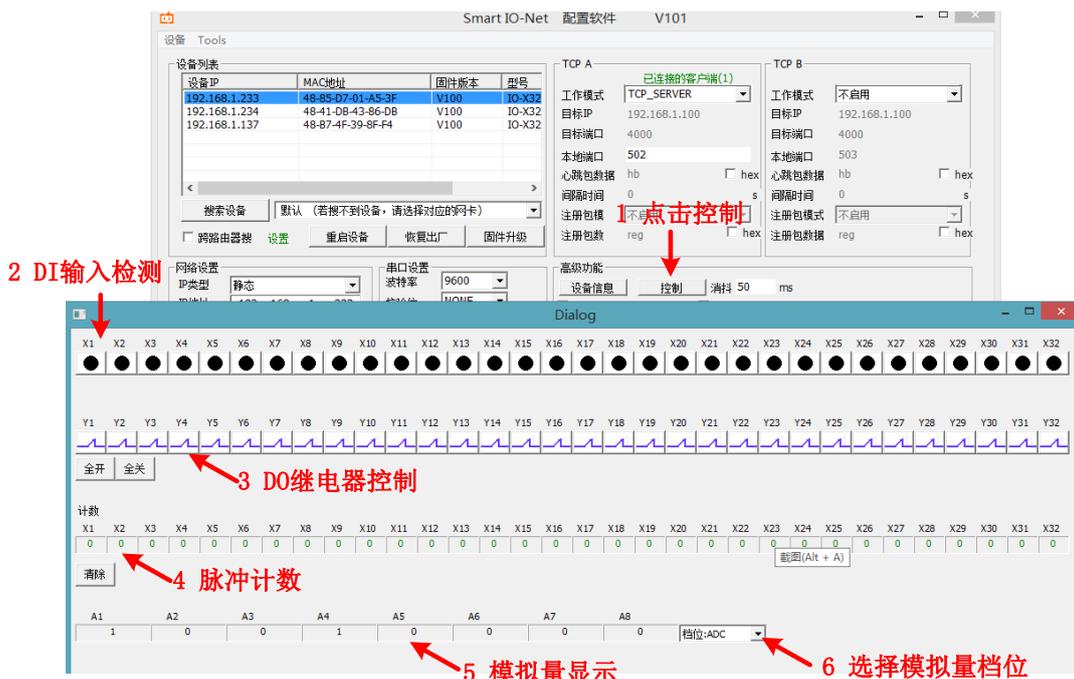


图 2.12 设备控制

1.4.3 DI 输入消抖配置

当设备的 DI 输入接机械按键时，机械按键的触点闭合和断开时，都会产生抖动，为了避免这种情况造成的系统误触发，用户可设置设备的输入消抖时间，一般 50ms~100ms。配置方法如图 2.13 所示。



图 2.13 DI 输入消抖配置

1.4.4 自动上传设置

通过上位机软件配置自动上报策略，可以配置条件触发上报和定时上报，具体设置依用户需求。设置完成后，保存参数，新参数即可生效。配置方法如图 2.14 所示。

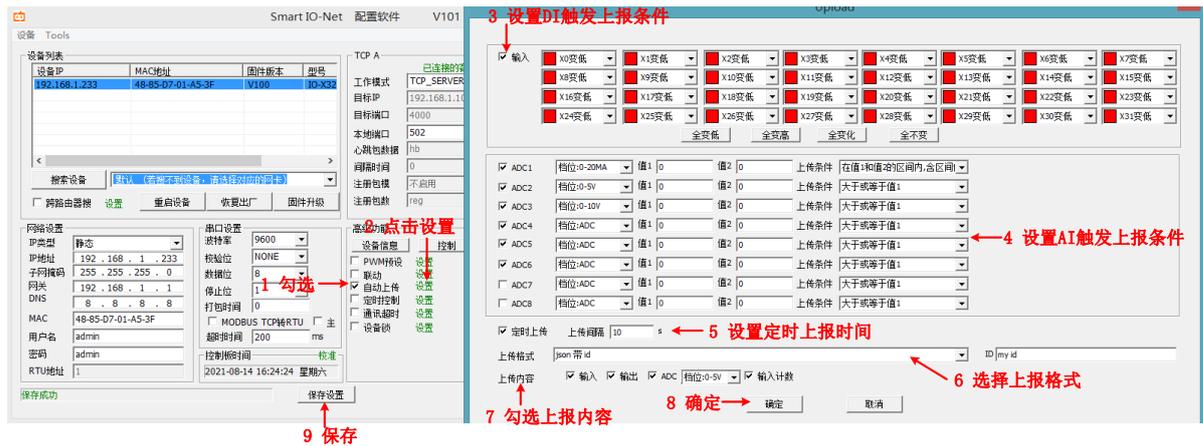


图 2.14 自动上报策略配置

- ① DI 输入触发主动上报：每一路 DI 输入均可设置四种不同的触发条件：DI 变化（上升沿/下降沿）、DI 变低（下降沿）、DI 变高（上升沿）、DI 无视（不起作用）。
- ② ADC 阈值触发主动上报：ADC 档位选择要与设备背面拨码开关选择的档位一致，触发上报条件可设置为“大于或等于值 1”、“在值 1 和值 2 的区间内，含区间”。
- ③ 定时上传：设置定时上传时间间隔，勾选使能。
- ④ 上传格式：选择数据上传的格式，目前支持：“自定义 48 3a 01 41...45 2e 只上报输入”、“自定义 48 3a 01 21...45 2e”、“modbus”、“ascii”、“JSON 带 MAC”、“JSON 带 ID”等格式上传。具体使用方法详见第 5 章。
- ⑤ 上传内容：勾选使能要上传的内容，若此处不勾选，即使达到设置的触发条件，也不会上传。

1.4.5 联动策略设置

联动控制功能支持用户自主设置触发 DO 输出变化的条件，使设备使用起来更加灵活，能应用于更多场景。

触发条件：

X1~X28，28 路 DI 输入：DI 输入存在有效信号

8 路 AI 模拟量阈值：大于、小于、区间内、区间外

DO 继电器输出动作：

断开：继电器的常开触点与公共端触点断开，常闭触点与公共端闭合。

闭合：继电器的常开触点与公共端触点闭合，常闭触点与公共端断开。

翻转：继电器状态和之前相反。

不变：继电器状态不动作。

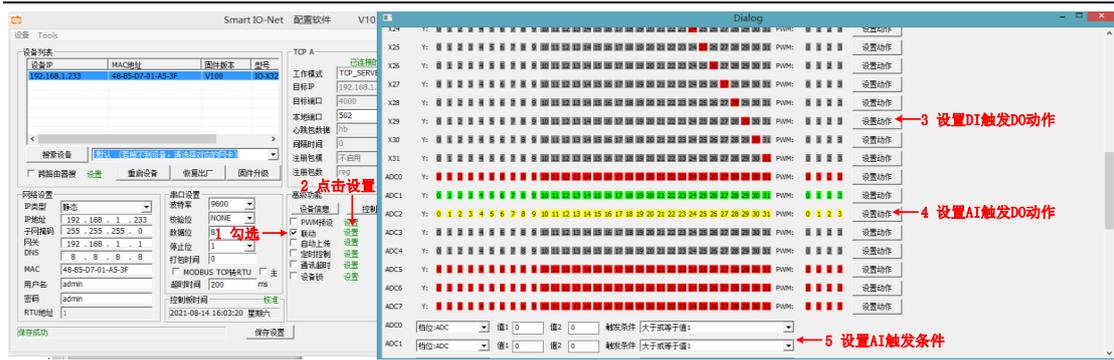


图 2.15 联动触发条件设置

- ① 需要联动的 DO 继电器的状态可以是断开、闭合、翻转、不动作，用户可根据需求进行设置。
- ② 用户可每种状态都试一试，可对设备的联动逻辑有更深的体会。

1.4.6 定时控制

设备内置 RTC 时钟，可实现设备的脱机定时控制功能。设备支持 32 个定时器，用户可设置具体的时间以及循环周期，当到达用户设置的时间后，设备的 DO 继电器输出就会按照用户设置好的动作执行。具体设置方法如图 2.16 所示，保存设置后，参数即可生效。



图 2.16 定时控制配置

1.4.7 通信超时

当网络/串口长时间无数据通信时，用户可设置每一路继电器做出不同的动作：断开、闭合、不动作。设置方法如图 2.17 所示。保存参数后，新参数即可生效。

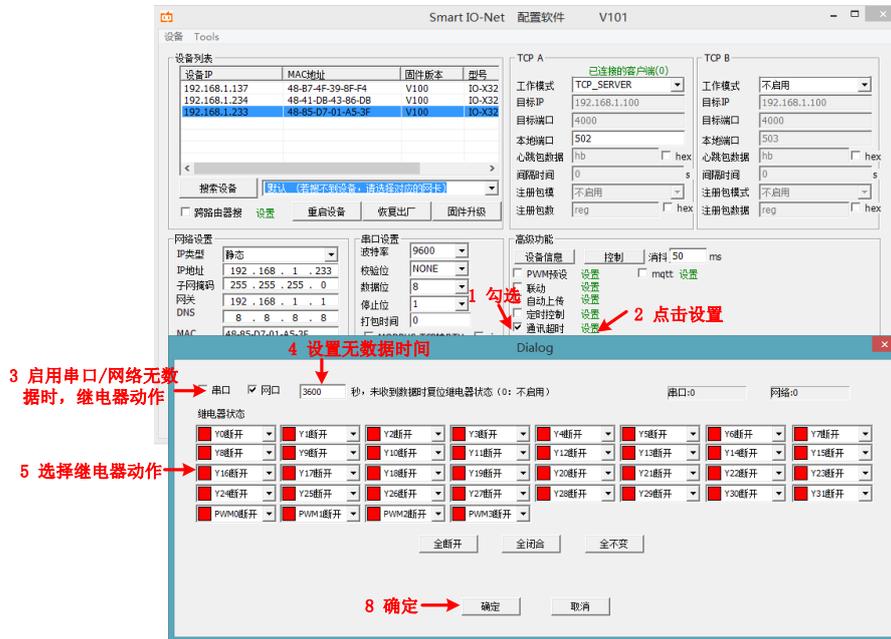


图 2.17 通信超时后继电器动作配置

1.4.8 掉电检测

设备内置掉电检测电路，设备会检测到设备是否断电，并将设备的继电器状态及开光量的脉冲计数值保存，当设备再次上电后，设备会自动恢复到断电前的状态。设备默认不开启掉电检测功能，用户可通过以下配置方法实现设备的掉电检测功能。

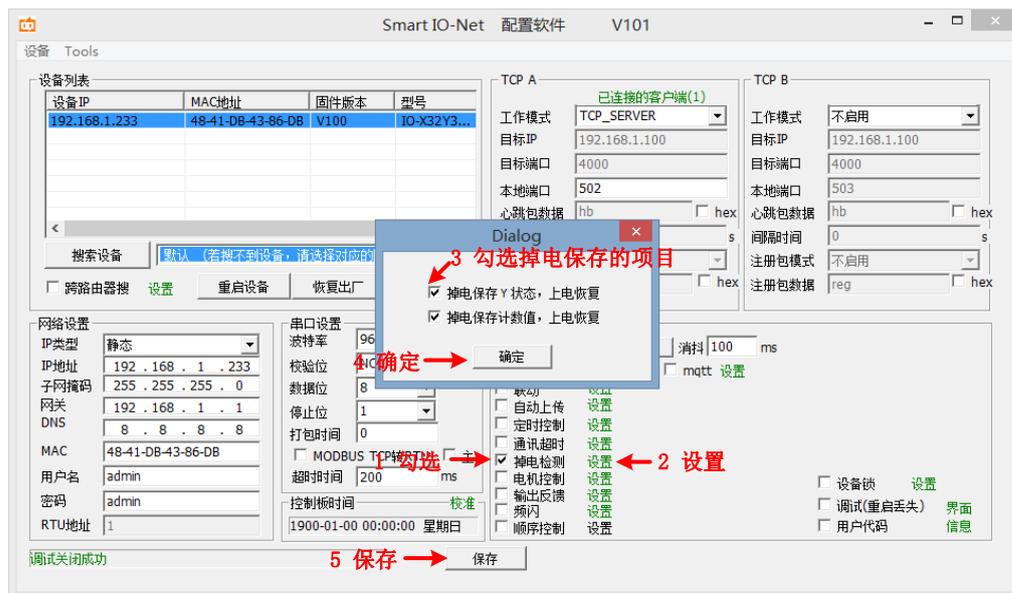


图 2.18 掉电检测配置

1.4.9 电机控制

设备支持直流电机、交流电机控制：正转、反转、停转。每两路继电器输出可实现一个电机的控制，比如型号 NET-8880D，有 8 路继电器输出，可以实现 4 个电机的控制。直流电机的接线图如图 2.19 所示。

电机正转：第一路继电器公共端与常开触点闭合，第二路继电器公共端与常开触点断开。

电机反转：第一路继电器公共端与常开触点断开，第二路继电器公共端与常开触点闭合。

电机停转：第一、二路继电器公共端与常开触点都断开。

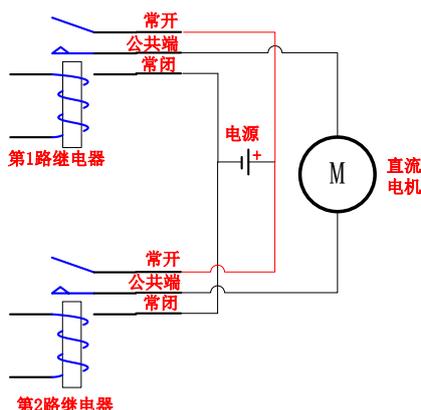


图 2.19 两路继电器控制 1 个直流电机接线图

交流电机的接线图如图 2.20 所示。

电机正转：第一路继电器公共端与常开触点闭合，第二路继电器公共端与常开触点断开。

电机反转：第一、二路继电器公共端与常开触点都闭合。

电机停转：第一、二路继电器公共端与常开触点都断开。

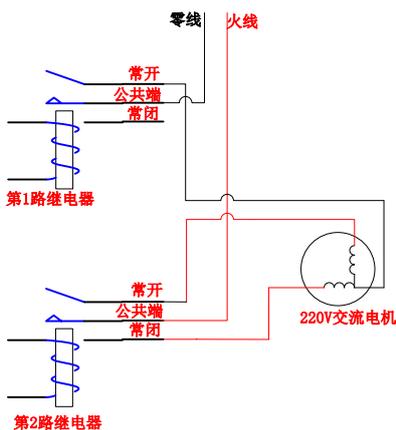


图 2.20 两路继电器控制 1 个 220V 电机接线图

设备默认不开启电机控制功能，可通过以下配置方法，开启该功能。

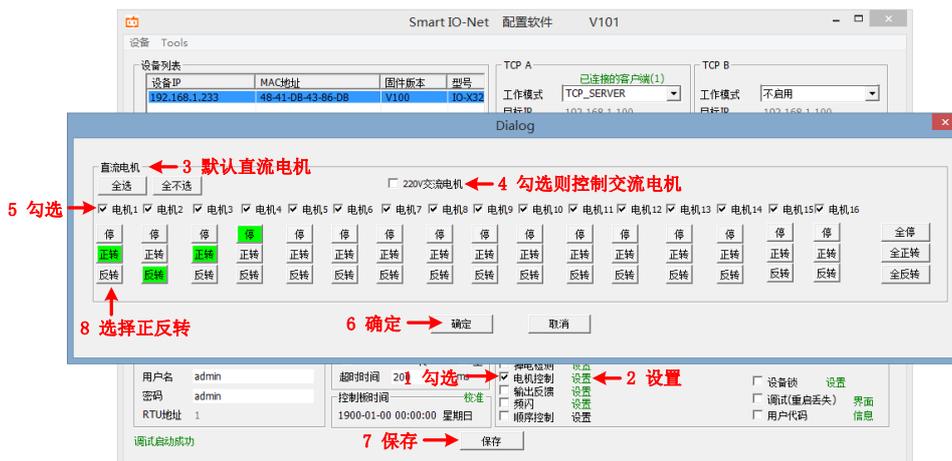


图 2.21 电机控制配置方法

- ① 每两路继电器可控制一个电机，比如电机 1 是由第 1、2 路继电器输出控制，电机 2 是由第 3、4 路继电器控制，依次类推。
- ② 电机的控制指令协议详见第 5 章。

1.4.10 输出反馈

设备支持检测继电器输出负载的电流大小及是否通断。用户可外接霍尔传感器或电流互感器，将传感器的电流输出接口接到设备的模拟量检测接口上，设备通过模拟量检测接口，可知道传感器的电流值，从而知道，继电器输出负载上的电流大小。

设备默认不使能输出反馈功能，可通过以下配置方法开启。



图 2.22 输出反馈配置

1.4.11 频闪控制

设备支持频闪控制每一路继电器输出。可以实现将继电器输出按照一定频率（周期）闭合和断开。闭合时间和断开时间可以通过配置软件设置。

设备默认不使能频闪控制功能，可通过以下配置方法开启。

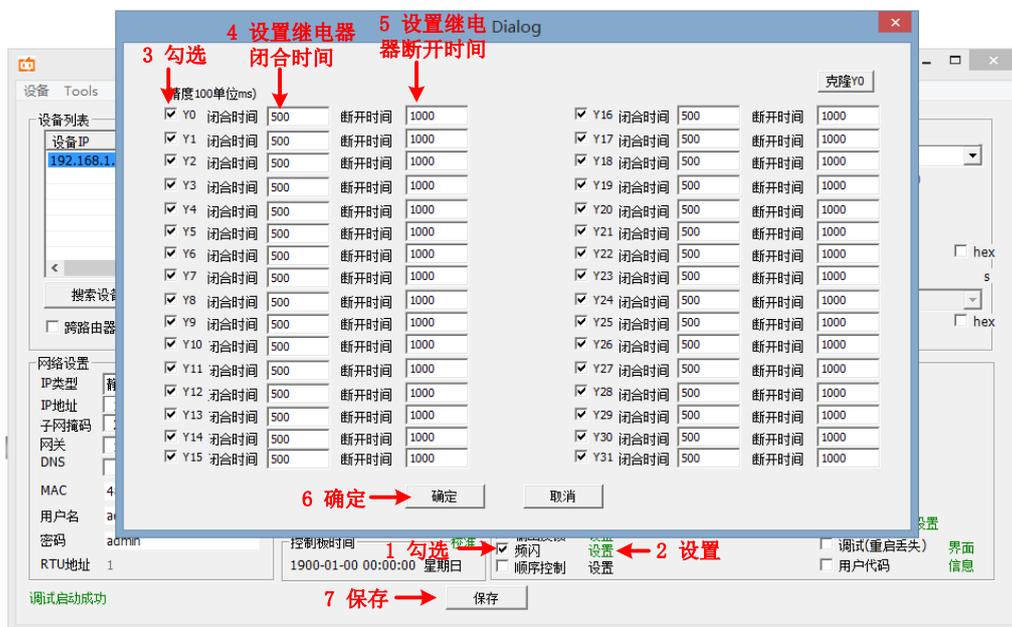


图 2.23 频闪控制功能配置

1.4.12 顺序控制

设备支持多路继电器之间的顺序控制，用户可通过简单的编程即可实现多路继电器的顺序控制及延时间隔；设备还支持 DI 触发启动顺序控制。指令格式如下：

y0=1 代表第一路继电器常开触点与公共端触点闭合；

y0=0 代表第一路继电器常开触点与公共端触点断开；

s100 代表延时时间，100ms；

各指令之间以英文格式的“;” 隔开。

例如：y0=1;s200;y1=1;s200;y2=1;s200;y3=1;s200; 第 1~4 路继电器间隔 200ms 依次打开。

设备默认不使能顺序控制功能，可通过以下配置方法开启。

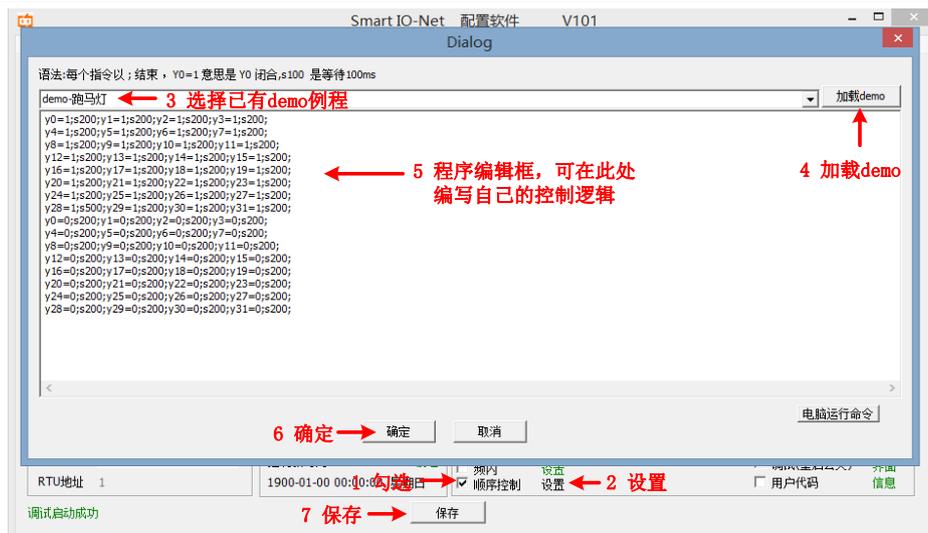


图 2.24 顺序控制配置方法

1.4.13 用户代码

设备支持二次开发，提供丰富的函数库，方便满足用户的各种需求。具体介绍详见产品相关的二次开发文档。



图 2.25 用户代码使能

2. 产品规格

设备通信接口有 RS485、CAN 和 RJ45 网络，支持 2~32 路继电器输出、2~28 路数字量输入、0~8 路 AI 模拟量采集、0~4 路 PWM 输出，满足用户的各种需求。设备采用工业级设计，具有很强的抗干扰能力。

2.1 产品参数

2.1.1 电气参数

除非特别说明，所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 2.1 电气参数

适用型号	参数名称	额定值			单位
		最小值	典型值	最大值	
ZQWL-NET-8880D-12M	电源电压	11V	12	13V	V
	工作电流 (8 路继电器全打开)	-	500	-	mA
ZQWL-NET-8880D-24M	电源电压	20V	24	25V	V
	工作电流 (8 路继电器全打开)	-	260	-	mA
ZQWL-NET-BC84D-12M	电源电压	11V	12	13V	V
	工作电流 (32 路继电器全打开)	-	1450	-	mA
ZQWL-NET- BC84D -24M	电源电压	20V	24	25V	V
	工作电流 (32 路继电器全打开)	-	940	-	mA

2.1.2 工作环境参数

表 2.2 工作环境参数

参数名称	额定值			单位
	最小值	典型值	最大值	
工作环境温度	-40	-	85	°C
存贮温度	-40	-	85	°C
工作环境湿度	5~95%RH			-

2.1.3 EMC 防护等级

除非特别说明，所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 2.3 防护等级参数

接口	浪涌等级	ESD 等级
电源接口	8/20 μ S 波形: \pm 2KV	空气放电: \pm 15KV 接触放电: \pm 8KV
网口、RS485、CAN 接口	10/70 μ S 波形: \pm 4KV	
其他接口	-	

2.1.4 继电器规格

除非特别说明，所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 2.4 两种继电器具体参数如表：

项目	参数	
	欧姆龙继电器	宏发继电器
额定负载	10A@AC250V 10A@30V	10A@AC277V 10A@28V
触点接触电阻	50m Ω 以下	100m Ω 以下
动作时间	15ms 以下	10ms 以下
复位时间	5ms 以下	5ms 以下
最大开关频率	机械: 18,000 次/小时	-
	额定负载: 1,800 次/小时	-
寿命	机械: AC 1,000 万次以上、DC 2,000 万次以上（开关频率 18,000 次/小时）	机械: 10 万次
	额定负载: 10 万次以上@额定负载（开关频率 18,00 次/小时）	额定负载: 5 万次以上@额定负载（开关频率 360 次/小时）

2.1.5 数字量输入参数

设备数字量输入电平有两种规格：2.7V~7V 规格和 6V~30V 规格，默认 6V~30V 规格，如有特殊需求，请联系公司销售。

除非特别说明，所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 2.5 数字量输入参数

设备输入规格	参数名称	额定值			单位
		最小值	典型值	最大值	
6V~30V 规格	高电平输入电压	6.0	-	30	V
	低电平输入电压	-	-	4V	V
2.7V~7V 规格	高电平输入电压	2.7	-	7	V
	低电平输入电压	-	-	1.5	V

2.1.6 ADC 模拟量输入参数

AI 接口支持测量“0~5V”、“0~10V”、“0~20mA”三种档位，用户可通过设备背面的拨码开关选择模拟量档位。

表 2.6 ADC 模拟量输入参数

采集信号类型	模拟量档位	最小值	典型值	最大值	测量精度
电压	0~5V	0V	--	5V	0.5%FS
电压	0~10V	0V	--	10V	0.5%FS
电流	0~20mA	0V	--	20mA	0.5%FS

2.1.7 通信参数

除非特别说明，所列参数是指 $T_{amb}=25^{\circ}\text{C}$ 时的值。

表 2.7 产品通信参数

项目	参数	指标
RS485	波特率	600bps~460800bps（出厂默认参数：9600bps，8，N，1）
	通信距离	大于 1200 米
CAN		符合 CAN 2.0B 规范，兼容 CAN 2.0A；符合 ISO 11898-1/2/3。 波特率支持 10kbps~1000kbps。
网口	数据速率	10/100M 自适应，MDI/MDIX 交叉直连自动切换
	支持协议	ETHERNET、TCP、UDP、IP、ARP、DHCP、DNS、ICMP、HTTP、MQTT

3. 设备接口接线说明

本章主要介绍产品的硬件接口信息及使用方法。

3.1 产品硬件接口布局

3.1.1 NET-8800M 硬件接口及尺寸

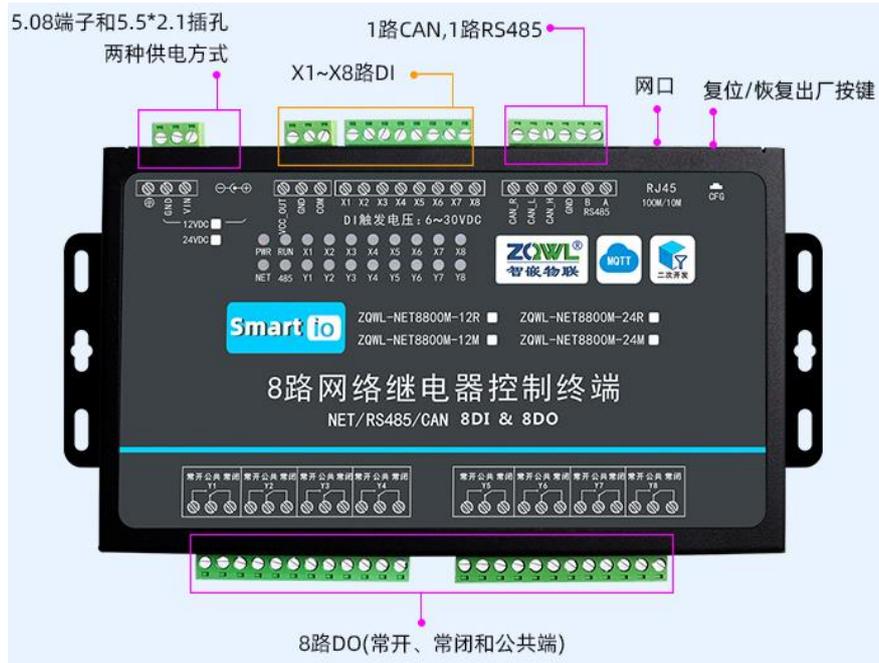


图 3.1 ZQWL-NET-8800M 硬件接口

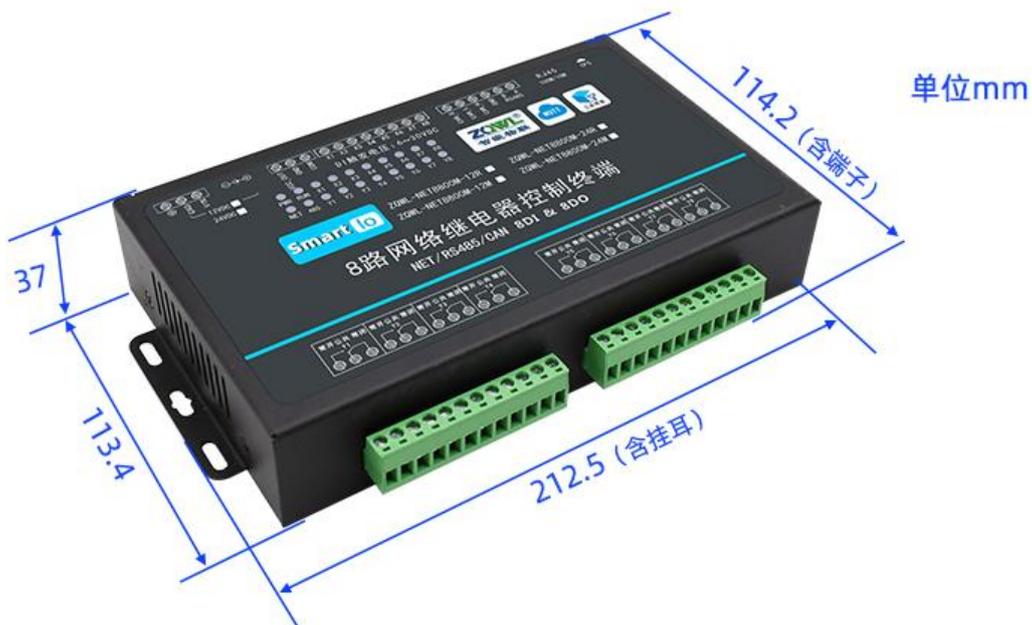


图 3.2 ZQWL-NET-8800M 结构尺寸

3.1.2 NET-8880D 硬件接口及尺寸

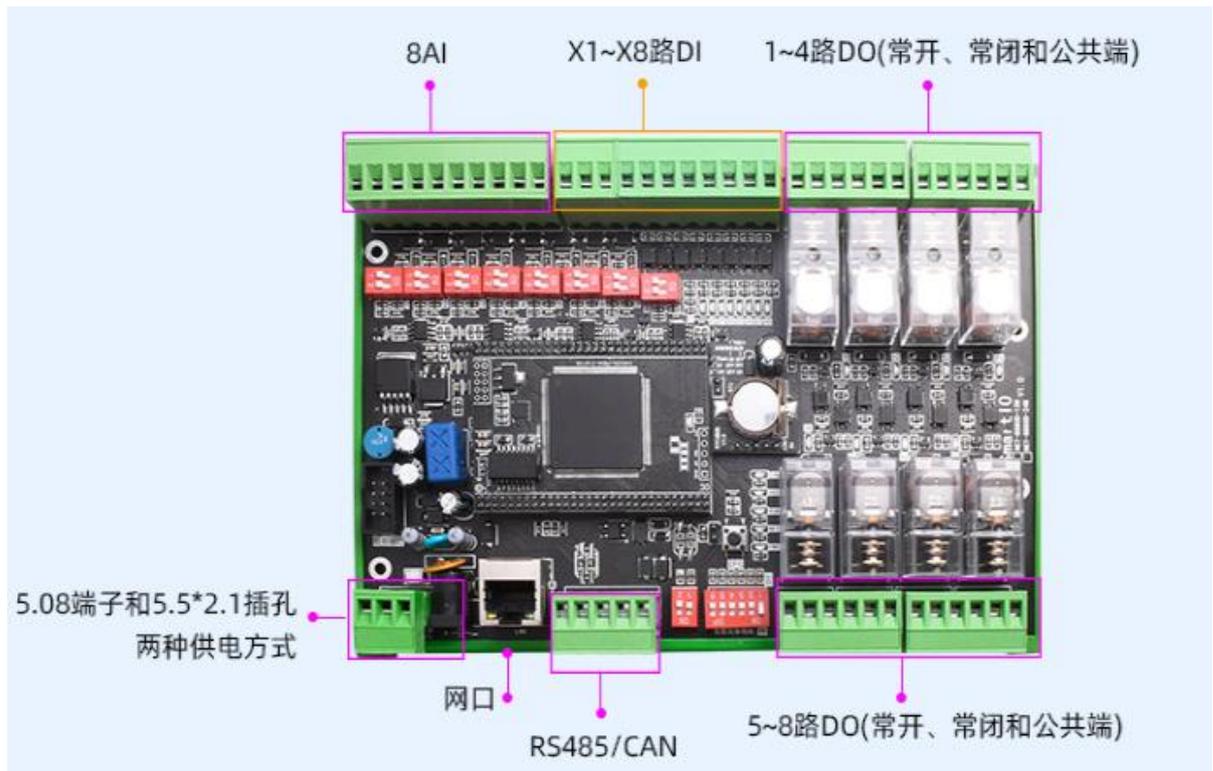


图 3.3 ZQWL-NET-8880D 硬件接口

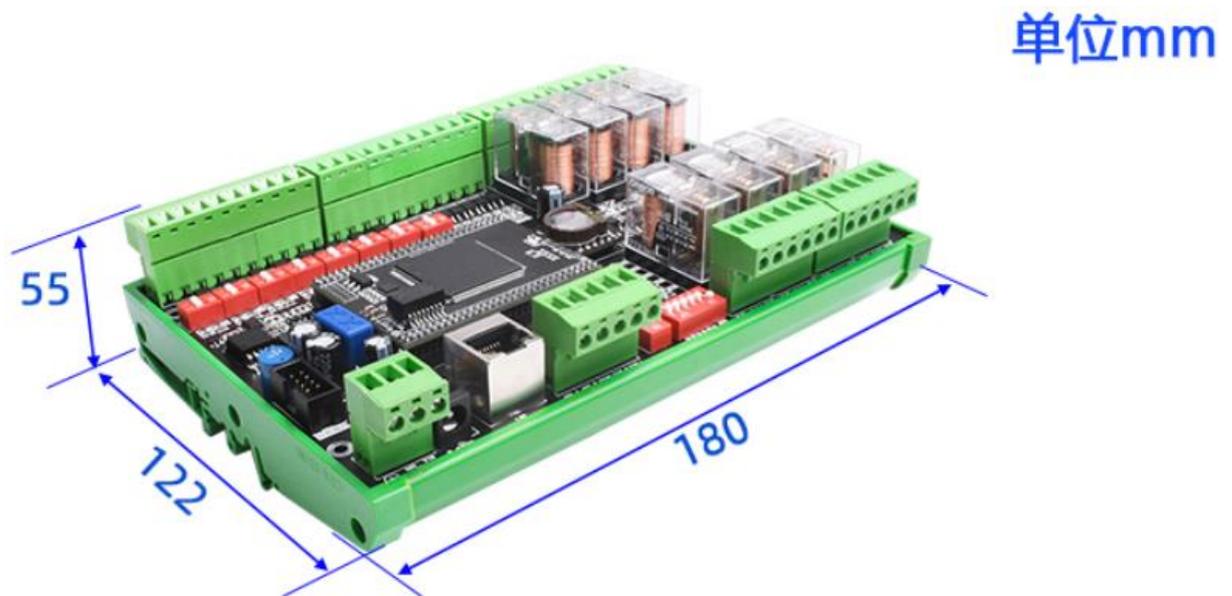


图 3.4 ZQWL-NET-8880D 结构尺寸

3.1.3 NET-AA80M 硬件接口及尺寸

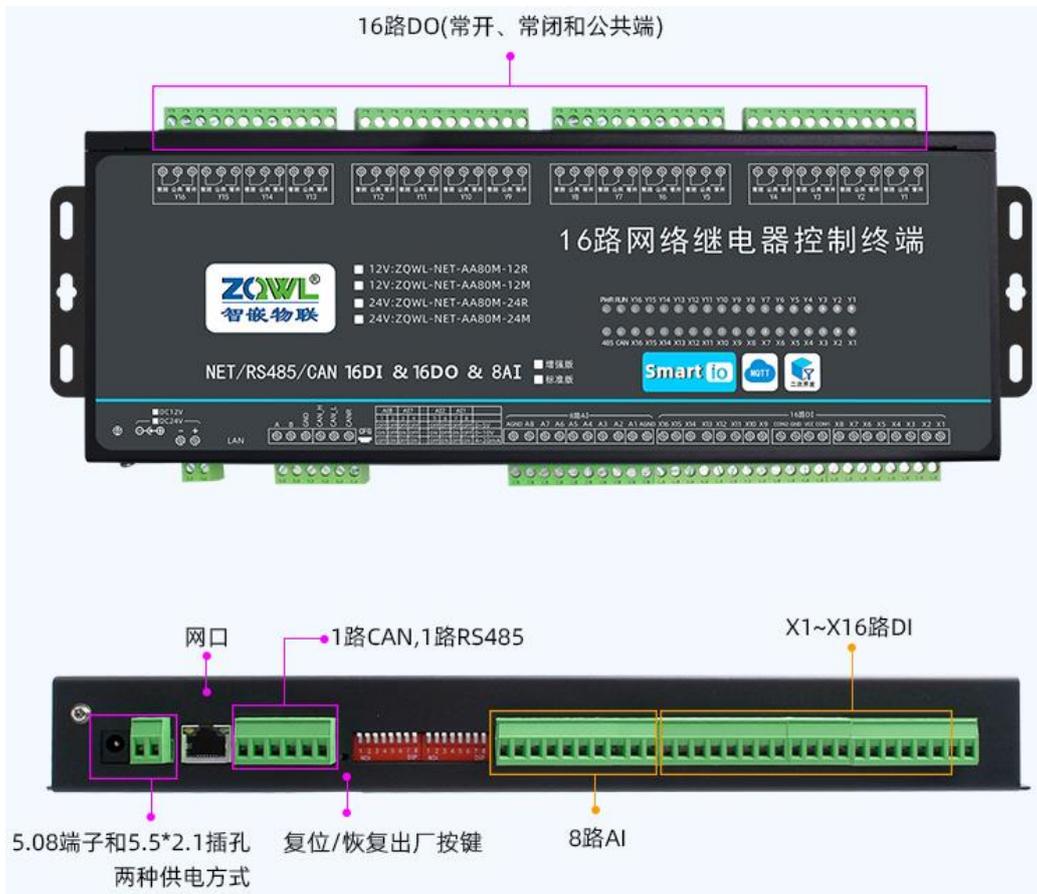


图 3.5 NET-AA80M 硬件接口



图 3.6 NET-AA80M 结构尺寸

3.1.4 NET-BC84D 硬件接口及尺寸

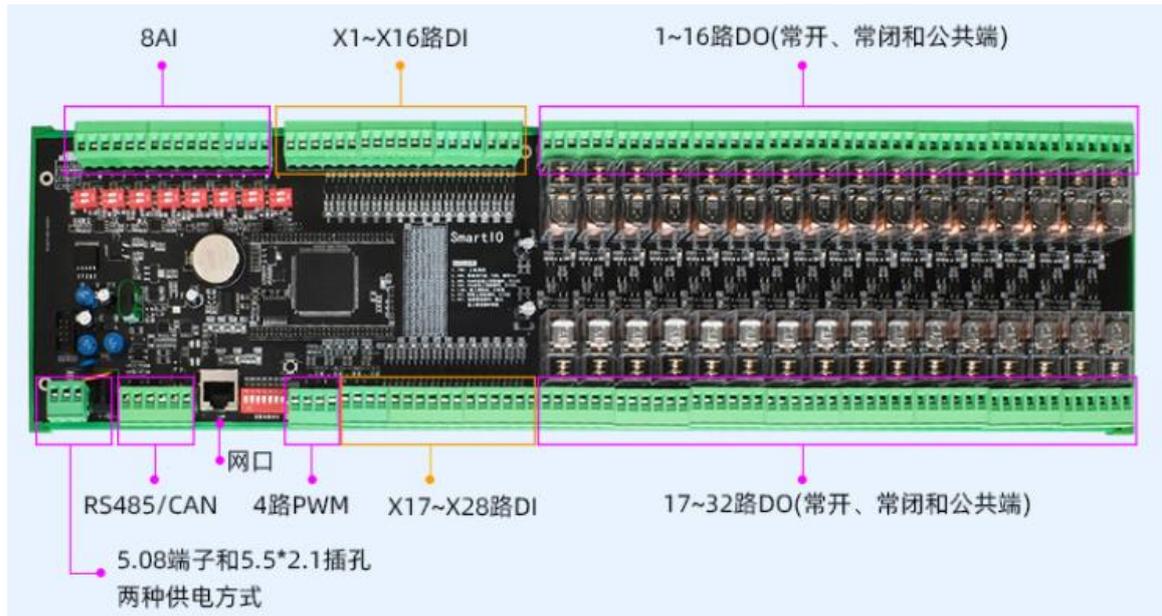


图 3.7 NET-BC84D 硬件接口

单位mm

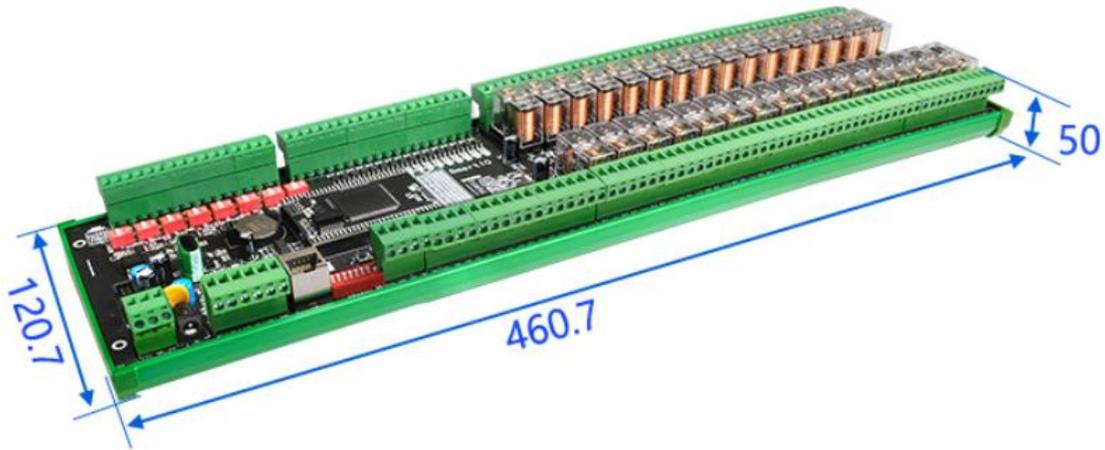


图 3.8 ZQWL-NET-BC84D 结构尺寸

3.2 电源接口

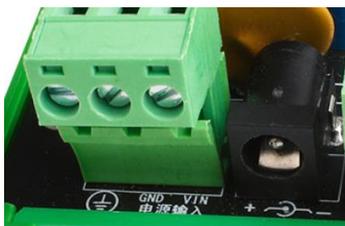


图 3.9 电源及保护地接口

电源接口支持两种形式：DC 插座（5.5*2.1 规格）和 5.08 绿色端子座。用户可以根据自己的实际情况灵活选用。

设备电源接口处会有接地端子，丝印为 \oplus 。

保证设备与大地的良好接触，为外部干扰提供泄放路径，提高产品的稳定性及可靠性。

① 为提高产品的稳定性及可靠性，强烈建议用户将保护地接口良好接地。

3.3 RS485 接口

RS485 接口有三种功能：继电器本地控制、RS485 总线数据到网络的透明传输。

RS485 接口采用 5.08 绿色端子座的形式，具体功能如表 3.1 所示。内置 ESD、浪涌防护器件，具体防护等级详见表 2.3 所示。

表 3.1 RS485 接口功能

面板丝印	功能
A	控制板 RS485 正极，接对端的 A(正)信号
B	控制板 RS485 负极，接对端的 B(负)信号

① RS485 长距离通信且周围环境干扰较大时，建议选择使用屏蔽双绞线。

② 本设备的通信距离不小于 1200 米@115200bps。

③ 在设备外部加 120 Ω 终端电阻的条件下，本设备的通信距离不小于 1200 米@256000bps。

3.4 RJ45 网口

以太网接口：1 个 RJ45

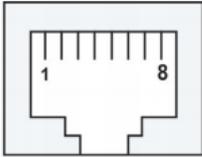
电气隔离：1.5KV 电磁隔离

数据速率：10/100M 自适应，MDI/MDIX 交叉直连自动切换

支持协议：ETHERNET、TCP、UDP、IP、ARP、DHCP、DNS、ICMP、HTTP、MQTT

接口定义如表 3.2 所示。

表 3.2 RJ45 引脚定义

网口		1、Tx+ 2、Tx- 3、Rx+ 6、Rx-
----	---	----------------------------------

3.5 拨码开关

设备的地址可通过拨码开关设置，设备有一个 8 位拨码开关，其中，1~6 为设备地址设置，7、8pin 为设备预留。如图 3.10 所示。

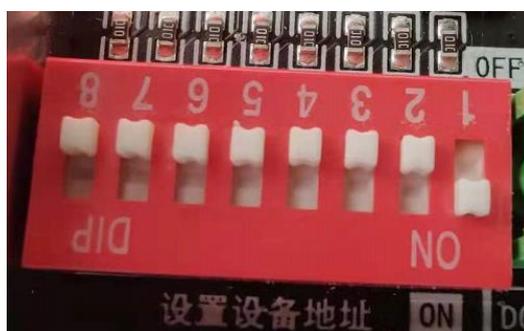


图 3.10 拨码开关

拨码开关的第 1~6 位为地址选择位，二进制表示。拨到“ON”时该位取“1”，反之该位取“0”。

如图 3.10 对应拨码开关值为：

拨码开关上的丝印	8	7	6	5	4	3	2	1
拨码位置	任意	任意	OFF	OFF	OFF	OFF	OFF	ON
取值	无意义	无意义	0	0	0	0	0	1

换算成 16 进制的值为 0x01，即地址为 1。

注：SW7：最高位 SW1：最低位。

3.6 恢复出厂设置

短按“CFG”按键，设备复位；按住“CFG”按键 6 秒以上，产品则恢复出厂设置。

3.7 产品指示灯

本设备具有丰富的指示灯：PWR、RUN、RS485、CAN、NET、LINK、ACT、X1~X28、Y1~Y32、PWM1~PWM4，其基本含义，如表 3.3 所示。

表 3.3 指示灯基本含义

面板丝印	含义	亮	灭
PWR	电源指示灯	电源正常	电源异常
RUN	运行指示灯	正常运行时，亮灭频率约 1Hz	
485	RS485 数据指示灯	当 485 有数据时，该灯闪烁	
CAN	CAN 数据指示灯	当 CAN 有数据时，该灯闪烁	
LINK	网络连接指示灯	连接到网络时，常亮	
ACT	网络数据指示灯	网络有数据时，闪烁	
X1~X28	28 路开关量输入指示	外部开关量输入触发	外部开关量输入无触发
Y1~Y32	32 路继电器输出指示	继电器常开触点闭合	继电器常开触点断开
PWM1~PWM4	4 路 PWM 输出指示	有波形输出时，灯常亮或闪烁	无波形输出时，灯不亮

3.8 模拟量输入接口

设备支持 8 路模拟量接口，可测量模拟量档位有：支持测量“0~5V”、“0~10V”、“0~20mA”。可通过设备的拨码开关选择对应的档位，设备默认“0~5V”档位。如表 3.4 所示。

表 3.4 模拟量档位选择

档位	AI1~AI8		拨码位置
	1	2	
0~5V	OFF	OFF	
0~10V	OFF	ON	
0~20mA	ON	OFF	

AI 模拟量接线如图 3.11 所示。



图 3.11 AI 接线图

3.9 数字量输入接口

本产品共有 28 路数字量输入检测，每路数字量输入接口均支持干节点、湿节点、NPN、PNP 接法。

3.9.1 干节点接线方式

若用户需要检测的是无源开关信号，可以使用干节点的接线方式。具体接线方式如图 3.12 所示。

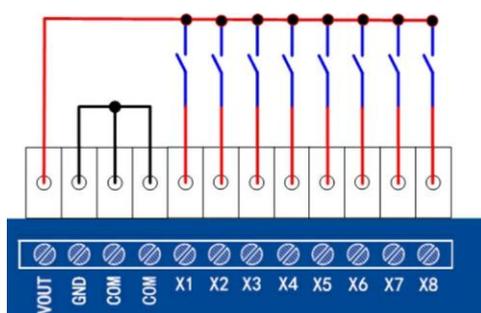


图 3.12 干节点接线图

3.9.2 湿节点接线方式

若用户需要检测有源的开关信号，可以采用湿节点的接线方式。具体接线方式如图 3.13 所示。

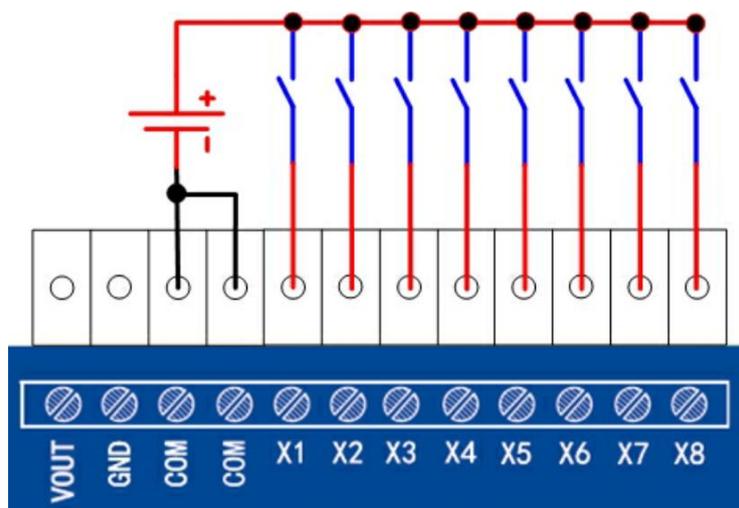


图 3.13 湿节点接线图

- ① COM 与 X1~X28 之间无需分正负，只要构成电流回路即可。
- ① 外接电源电压范围：6.0V~30V / 2.7V~7V，具体参数，详见 2.1.5。

3.9.3 NPN 型传感器接线方式

若用户需检测 NPN 型传感器的开关量信号，可以采用如图 3.14 所示的接线方式。

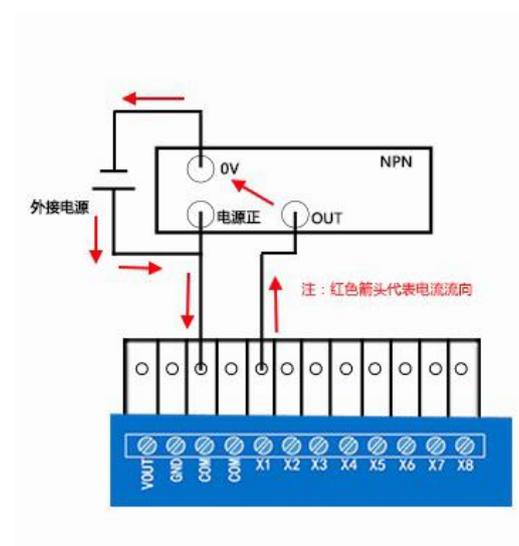


图 3.14 NPN 型传感器接线图

- ① 采用此种接法时，NPN 型传感器的供电电源即为本产品的供电电源。

3.9.4 PNP 型传感器接线方式

若用户需检测 PNP 型传感器的开关量信号，可以采用如图 3.15 所示的接线方式。

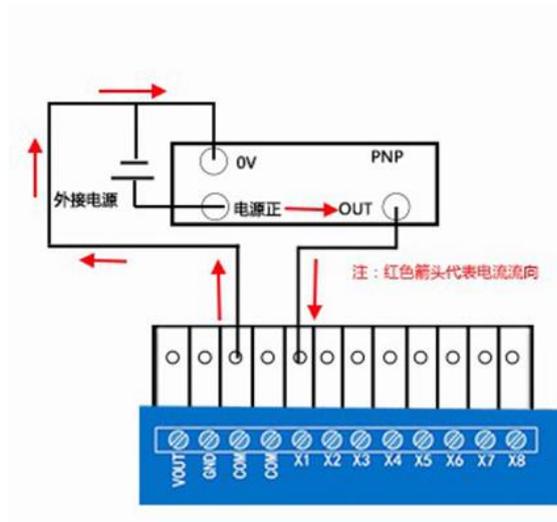


图 3.15 PNP 型传感器接线方式

3.10 继电器输出接口

本产品共有 32 路继电器输出（Y1~Y32），每路继电器都有三个触点：常开、常闭和公共端。继电器的电气参数详见表 2.4。

产品面板上每路都有对应的指示灯，具体含义，详见表 3.3。

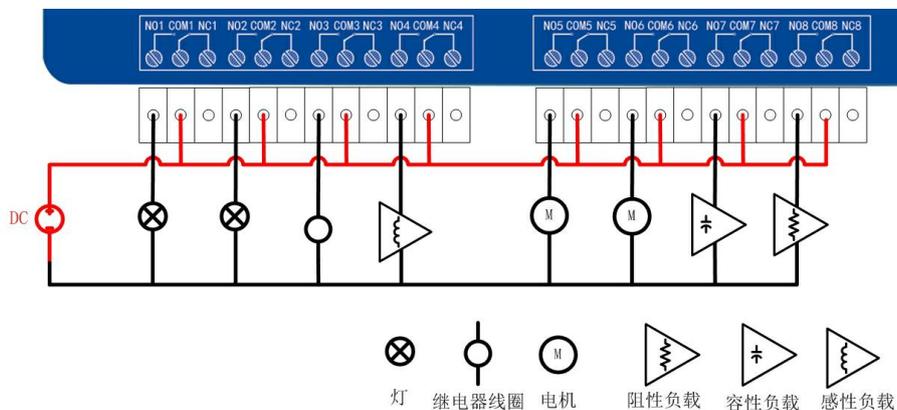


图 3.16 继电器输出接线图

- ① 使用继电器输出控制大功率负载或感性负载时，请确保不要超过本产品的额定负载参数，具体参数详见表 2.4。

4. 产品功能

4.1 工作模式

设备可同时支持 2 个 TCP/UDP 连接、1 个 MQTT 连接、websocket 控制、HTTP Post 控制。

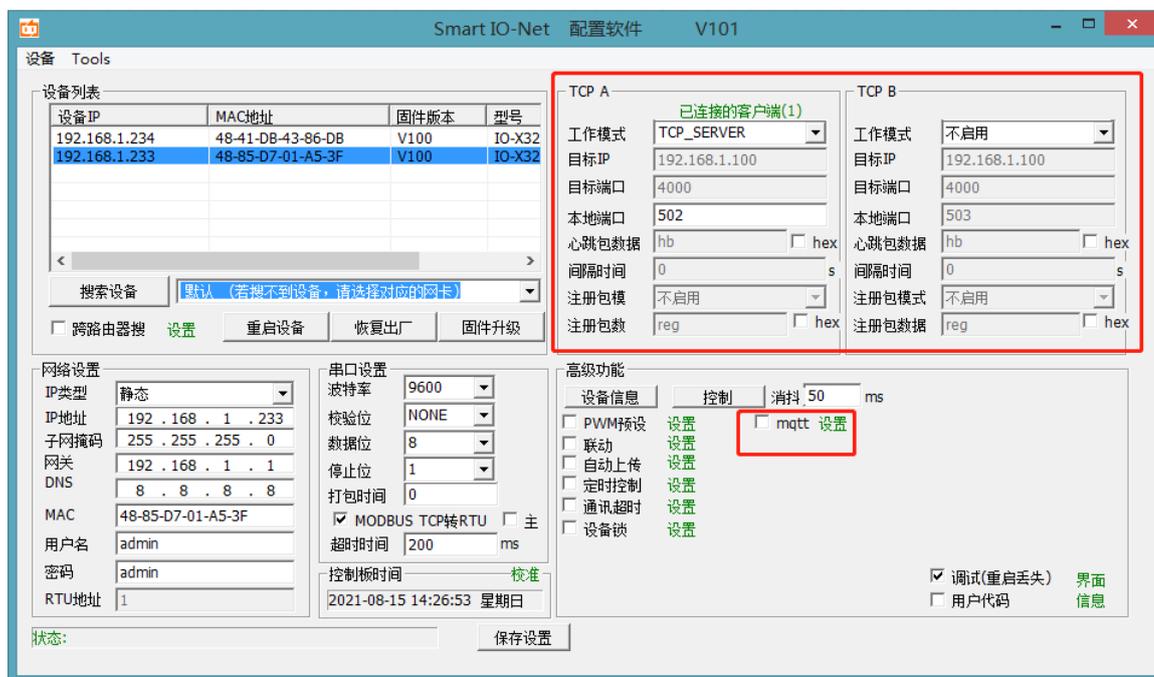


图 4.1 同时连接多个服务器

4.1.1 TCP/UDP 连接

设备同时支持 2 个 TCP/UDP 连接，可支持的工作模式有：TCP_SERVER、TCP_CLIENT、UDP_SERVER、UDP_CLIENT、虚拟串口模式等。用户在选择设备的工作模式时，可参考图 4.2 所示的流程图。

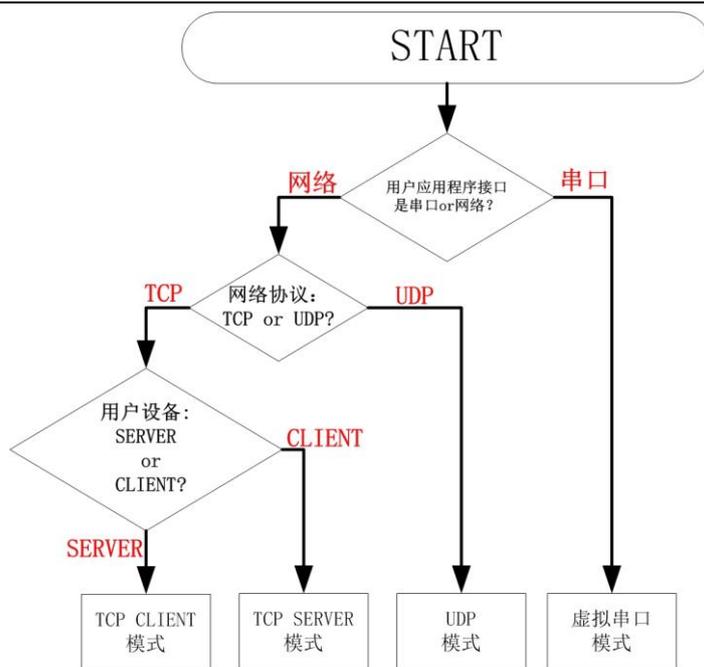


图 4.2 工作模式选择流程图

4.1.1.1 TCP_SERVER 模式工作原理

在 TCP 服务器（TCP Server）模式下，设备始终等待用户的客户端（TCP Client）设备的连接，在与客户端建立 TCP 连接后即可进行双向数据通信。如图 4.3 所示，在 TCP_SERVER 工作模式下，数据传输步骤如下：

1. 建立连接。用户 TCP 客户端向设备发送建立连接请求。
2. 连接一旦建立，TCP 客户端与设备之间即可进行数据的双向传输。

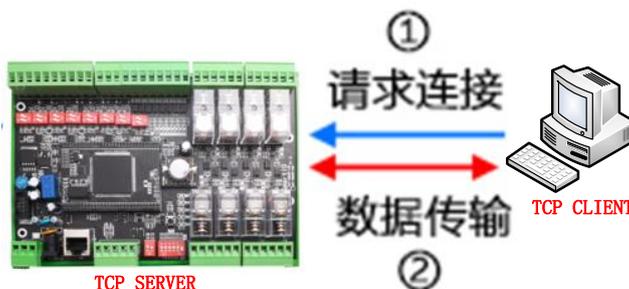


图 4.3 TCP SERVER 模式工作原理

4.1.1.2 TCP_CLIENT 模式工作原理

在 TCP 客户端（TCP Client）模式下，设备作为 TCP 客户端，将主动向“目标 IP/域名”和“目标端口”所指定的 TCP 服务器发起连接请求。如果连接不成功，TCP 客户端将会根据设置的连接条件不断尝试与 TCP 服务器建立连接。在与 TCP 服务器端建立 TCP 连接后即可进行双向数据通信。

1. 被配置为 TCP Client 模式的设备向 TCP 服务器发送建立连接请求。
2. 连接一旦建立，设备与 TCP 服务器之间即可进行数据传输。

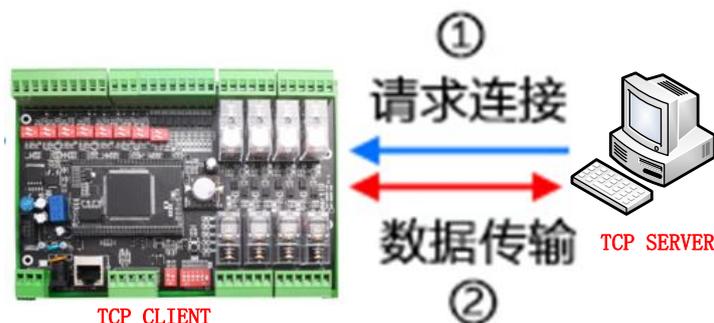


图 4.4 TCP CLIENT 模式工作原理

4.1.1.3 UDP SERVER 模式工作原理

在 UDP_SERVER 模式下，设备的数据总是发往最后一个与设备通讯的 UDP 对象（IP 和端口），并且必须先有对方向设备发数据后，设备才能记录这个 IP 和端口号。

该模式的特点是，设备的数据可以和不同的 UDP 对象（IP 和端口）通讯。

1. 在该模式下，UDP 客户端首先向设备发送一包数据。
2. 设备收到 UDP 客户端的第一包数据后，两者即可进行双向数据传输。



图 4.5 UDP SERVER 模式工作原理

4.1.1.4 UDP CLIENT 模式工作原理

该模式下设备的数据总是发往预先设置的“目标 IP/域名”和“目标端口”，并且多个 UDP 对象（IP 和端口）都可以将数据发到设备 IP 和该 PORT 的“本地端口”上，从而转发到该 PORT 口上。

该模式下，支持“目标 IP/域名”为“255.255.255.255”的广播发送。

4.1.2 连接 MQTT 服务器

设备支持连接 1 个 MQTT 服务器。设备会将串口数据或设备本身接口状态数据发送到用户配置的发布 Topic 里；设备也会接收用户配置的订阅 Topic 里的数据，数据解析后，若符合控制协议，设备做出相应的动作，若不符合控制协议，则转发到串口。



图 4.6 MQTT 配置

MQTT 通信案例详见第 8.4 节。

4.1.3 HTTP Post 控制

设备支持 HTTP Post 协议对设备进行控制及信号采集，提供 HTTP Post 控制 demo，方便用户快速使用产品。

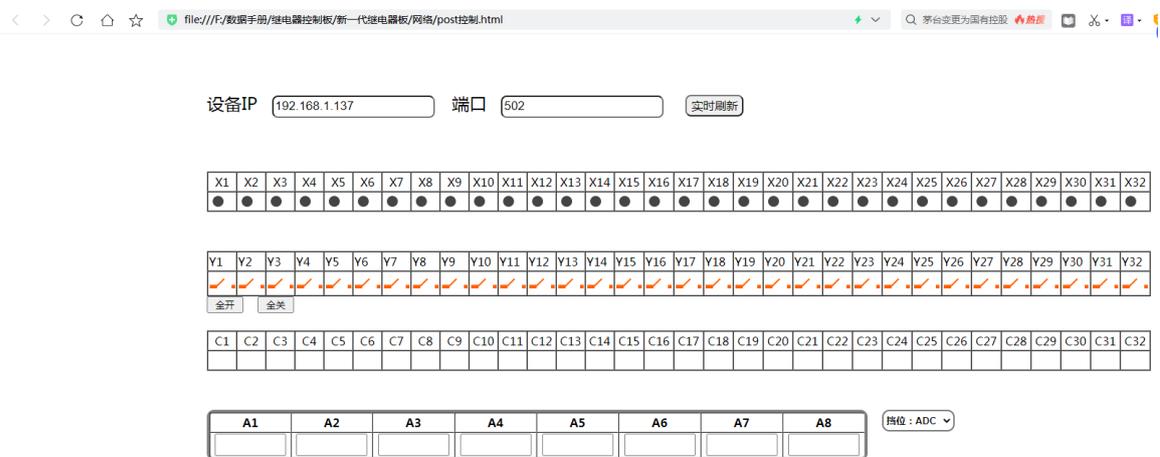


图 4.7 HTTP POST 控制 demo 界面

4.1.4 Web Socket 控制

设备支持 Web Socket 协议对设备进行控制及信号采集，提供 Web Socket 控制 demo，方便用户快速使用产品。

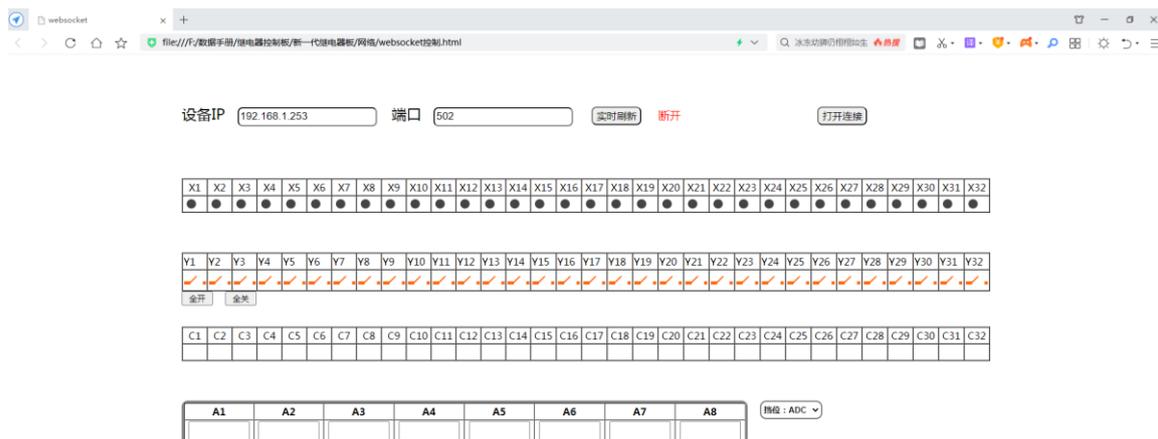


图 4.8 Web Socket 控制 demo 界面

4.1.5 串口数据透传

设备支持 RS485 接口数据与网络之间的数据透传：设备接收到数据之后，首先解析是否符合设备的控制协议，若符合控制协议，则按照指令控制设备做出相应的动作；若不符合控制协议，则数据透传，如图 4.9 所示。控制协议具体详见第 5 章。

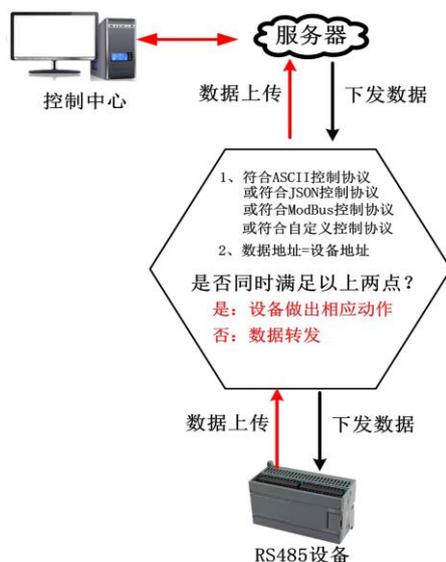


图 4.9 串口数据透传策略

4.2 注册包、心跳包

4.2.1 心跳包

心跳包只能用在网络透传模式下，用户可以根据需求设置心跳包数据和心跳包时间。当心跳包间隔设置为 0 或不勾选“启用心跳包”，心跳包功能不启用。

向服务器发送心跳包主要目的是为了保持连接稳定可靠，保证连接正常的同时还可以让服务器通过心跳包知道设备在线情况。用户可以选择让设备发送心跳包以实现特定的需求。

④ 网络心跳包是在透传模式下，一个心跳时间内没有数据向网络发送的时候才会发送，如果数据交互小于心跳时间，则不会发送心跳包。

4.2.2 注册包

注册包只能用在网络透传模式下，用户可以根据需求来设置注册包的发送方式和注册包数据，也可以不使用注册包功能。

注册包可以作为设备获取服务器功能的识别码，也可以作为数据包头，方便服务器识别数据来源。

设备支持三种注册包发送方式，如表 4.1 所示。

表 4.1 注册包发送方式

注册包发送方式	说明
与服务器建立连接时，向服务器发送一次	连接服务器成功后，发送注册包到服务器，并且只发送一次
向服务器发送的每个数据包前都加上	向服务器发送数据时，在数据前增加注册包后发送到服务器
同时支持以上两种	连接服务器成功后，发送注册包到服务器，同时在向服务器发送数据时，在数据前增加注册包后再发送到服务器端

4.3 DI 输入信号检测

1. 接线方法

设备每路数字量输入接口均支持干节点、湿节点、NPN、PNP 接法。具体接线方式请参考 3.9 节。

2. 指令格式

获取设备开关量状态的指令格式支持 ASCII 格式、JSON 格式、ModBus RTU 格式、自定义指令格式，用户可根据需要选择适合自己的指令格式。具体详见第 5 章。

3. 输入消抖滤波

DI 输入检测具有滤波功能，用户可根据实际需要，通过上位机软件进行配置。详见 1.4 节：设备参数配置。

当设备的 DI 输入接机械按键时，机械按键的触点闭合和断开时，都会产生抖动，为了避免这种情况造成的系统误触发，用户可设置设备的输入消抖时间，一般 50ms~100ms。

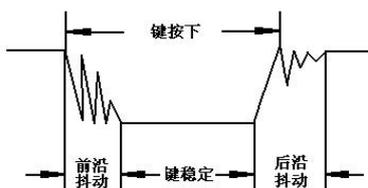


图 4.10 按键抖动波形

4. 自动上报

设备支持条件触发上报、定时上报。可通过上位机软件进行配置，配置方法详见 1.4。自动上报案例详见 8.2 小节。

设备上报数据格式支持：JSON 格式、ASCII 格式、ModBus RTU 格式、自定义格式，用户可根据需要选择合适的数据格式。具体详见第 5 章。

4.4 DI 脉冲计数

检测设备每路 DI 输入信号的上升沿，每检测到一个上升沿信号，计数加 1。设备断电/复位重启后，计数是否清零可配置。脉冲计数最大值为 0xFF FF FF FF FF FF FF，在计数值大于最大值后将会从 0 开始重新计数。

1. DI 脉冲计数断电保存

设备内置掉电检测电路，当设备断电时，设备可将脉冲计数值保存起来，当设备再次上电后，恢复到掉电前的计数值。通过配置软件勾选使能该功能，默认不使能该功能，配置方法详见 1.4 节：设备参数配置。

2. 指令格式

设置或获取 DI 脉冲计数值的指令格式支持 ASCII 格式、JSON 格式、ModBus RTU 格式、自定义指令格式，用户可根据需要选择适合自己的指令格式。具体详见第 5 章。

3. 主动上报

设备支持条件触发上报、定时上报。可通过上位机软件进行配置，配置方法详见 1.4。自动上报案例详见 8.2 小节。

设备上报数据格式支持：JSON 格式、ASCII 格式、ModBus RTU 格式、自定义格式，用户可根据需要选择合适的数据格式。具体详见第 5 章。

4.5 继电器输出（DO）

1. 接线方法

具体接线方法详见 0 节。

2. DO 继电器状态掉电保持功能

设备内置掉电检测电路，当设备断电时，设备会自动保存设备在掉电前的所有 DO 继电器输出的状态，设备重新上电后，会自动恢复掉电前的 DO 继电器输出状态。通过配置软件勾选使能该功能，默认不使能该功能，配置方法详见 1.4 节：设备参数配置。

3. DO 继电器输出延时断开功能

设备的每路 DO 继电器支出延时断开功能：从 DO 继电器常开触点闭合开始计时，设置的延时时间到后，DO 继电器常开触点断开。延时时间范围 0ms~2147483647ms。延时时间可通过手机 APP 设置，详见 8.1 小节介绍，也可通过控制指令设置，具体详见第 5 章。

4. DO 继电器定时控制功能

设备内置 RTC 时钟，可实现设备的脱机定时控制。

设备支持 32 个定时器，用户可设置具体的时间以及循环周期，当到达用户设置的时间后，设备的 DO 继电器输出就会按照用户设置好的动作执行。配置方法详见 1.4。

5. 电机控制

设备支持直接控制直流电机和 220V 交流电机（支持正转、反转、停转），每两路继电器输出可控制 1 个电机。电机与继电器的接线方法详见 1.4 小节，电机的控制指令详见第 5 章。

6. 频闪控制功能

设备支持频闪控制每一路继电器输出。可以实现将继电器输出按照一定频率（周期）闭合和断开。闭合时间和断开时间可以通过配置软件设置，配置方法详见 1.4 小节。

7. 顺序控制功能

设备支持多路继电器之间的顺序控制，用户可通过简单的编程即可实现多路继电器的顺序控制及延时间隔；设备还支持 DI 触发启动顺序控制。配置方法详见 1.4 小节。

8. 继电器输出负载反馈

设备支持检测继电器输出负载的电流大小及是否通断。用户可外接霍尔传感器或电流互感器，将传感器的电流输出接口接到设备的模拟量检测接口上，设备通过模拟量检测接口，可知道传感器的电流值，从而知道，继电器输出负载上的电流大小。配置方法及接线详见 1.4 小节。

9. 通信超时，输出继电器状态配置

当设备的网络或串口上长时间没有数据时（例如网线或串口线断了），用户可配置每一路继电器做出相应的动作。配置方法详见 1.4 小节。

10. 指令格式

设置或获取 DO 继电器输出的指令格式支持 ASCII 格式、JSON 格式、ModBus RTU 格式、自定义指令格式，用户可根据需要选择适合自己的指令格式。具体详见第 5 章。

11. 主动上报

设备支持条件触发上报、定时上报。可通过上位机软件进行配置，配置方法详见 1.4。自动上报案例详见 8.2 小节。

设备上报数据格式支持：JSON 格式、ASCII 格式、ModBus RTU 格式、自定义格式，用户可根据需要选择合适的数据格式。具体详见第 5 章。

4.6 模拟量测量

设备支持 8 路模拟量输入，支持测量“0~5V”、“0~10V”、“0~20mA”档位的模拟量。

1. 接线方法

具体接线方法详见 3.8 小节。

2. 档位选择

两路模拟量档位可通过设备的拨码开关来选择。如表 4.2 所示。

表 4.2 模拟量档位选择

档位	AI1~AI8		拨码位置
	1	2	

0~5V	OFF	OFF	
0~10V	OFF	ON	
0~20mA	ON	OFF	

3. 指令格式

读取 AI 模拟量测量值的指令格式支持 ASCII 格式、JSON 格式、ModBus RTU 格式、自定义指令格式，用户可根据需要选择适合自己的指令格式。具体详见第 5 章。

4. 主动上报

设备支持条件触发上报、定时上报。可通过上位机软件进行配置，配置方法详见 1.4。自动上报案例详见 8.2 小节。

设备上报数据格式支持：JSON 格式、ASCII 格式、ModBus RTU 格式、自定义格式，用户可根据需要选择合适的数据格式。具体详见第 5 章。

4.7 联动控制

联动控制功能支持用户自主设置触发 DO 输出变化的条件，使设备使用起来更加灵活，能应用于更多场景。

触发条件：

X1~X28，28 路 DI 输入：DI 输入存在有效信号

8 路 AI 模拟量阈值：大于、小于、区间内、区间外

DO 继电器输出动作：

断开：继电器的常开触点与公共端触点断开，常闭触点与公共端闭合。

闭合：继电器的常开触点与公共端触点闭合，常闭触点与公共端断开。

翻转：继电器状态和之前相反。

不变：继电器状态不动作。

④ 对于同一路 DO 输出，可以有多种触发条件，触发条件之间的关系是“或”的关系。

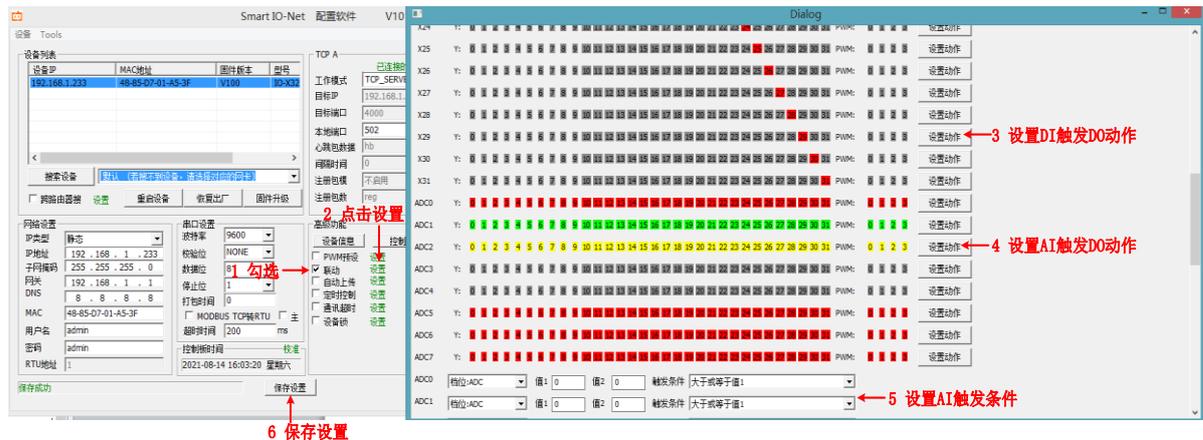


图 4.11 联动控制设置

举例 1: DI 输入电平控制

1) 输入输出一对一断开联动:

当 DI1 输入接口存在有效输入信号时，DO1 继电器常开触点与公共端触点断开；当 DI1 无有效输入信号时，DO1 继电器常开触点与公共端触点闭合。其他路 DI 和 DO 也是一样。

2) 输入输出一对一闭合联动:

当 DI1 输入接口存在有效输入信号时，DO1 继电器常开触点与公共端触点闭合；当 DI1 无有效输入信号时，DO1 继电器常开触点与公共端触点断开。其他路 DI 和 DO 也是一样。

举例 2: DI 输入按键控制

输入输出一对一翻转联动:

当 DI1 输入接口检测到输入信号的上升沿时，则将 DO1 继电器输出状态翻转。（当输入按键按下时，继电器输出状态翻转，按键松开，继电器输出状态不变。）

举例 3: AI 阈值控制

档位选择：首先根据设备背面拨码开关选择的 AI 档位，在配置软件上选择正确的 AI 档位：0~5V、0~10V、0~20mA。

触发条件：AI 的触发条件可以是“ \geq 值 1”、“在值 1 和值 2 的区间内”。

例如：AI1 的档位选择“0~5V”，触发条件为“ \geq 值 1”，值 1 设为 3.3V，Y1 继电器断开。当设备的 AI1 接口上的模拟量电压大于 3.3V 时，Y1 继电器断开，当电压再次小于 3.3V 时，Y1 继电器闭合。

举例 4: 一对多控制

一路 DI/AI，可以同时触发多路继电器动作，每路继电器的动作可以是“断开”、“闭合”“翻转”、“不动作”。

4.8 自动上报



图 4.12 自动上报设置

设备可将设备的 DI 输入状态、DO 输出状态、AI 模拟量值、脉冲计数值主动上报给服务器。

设备支持条件触发上报、定时上报。使用案例配置方法详见 1.4。自动上报案例详见 8.2 小节。

设备上报数据格式支持：JSON 格式、ASCII 格式、ModBus RTU 格式、自定义格式，用户可根据需要选择合适的数据格式。具体详见第 5 章。

4.9 RTC 定时

设备内置 RTC 时钟，可精准实现设备的定时控制，即使网络中断，也可以实现设备的定时控制。时间校准方法如所示。时间校准是按照电脑的时间来校准的，若电脑时间不准，请先调整电脑时间后，再校准。



图 4.13 RTC 时钟校准

4.10 ModBus TCP 转 RTU

设备支持 ModBus TCP 转 RTU 功能,用户可通过简单的配置即可实现 ModBus 网关功能。配置方法如图 4.14 所示。

设备支持主机模式和从机模式,用户可根据需求选择合适的模式。

主机模式:用户的网络设备作为主机,请求 RS485 总线上的从机数据时,需用主机模式。

从机模式:用户的 RS485 总线上的设备作为主机,请求网络上的从机数据时,需用从机模式。

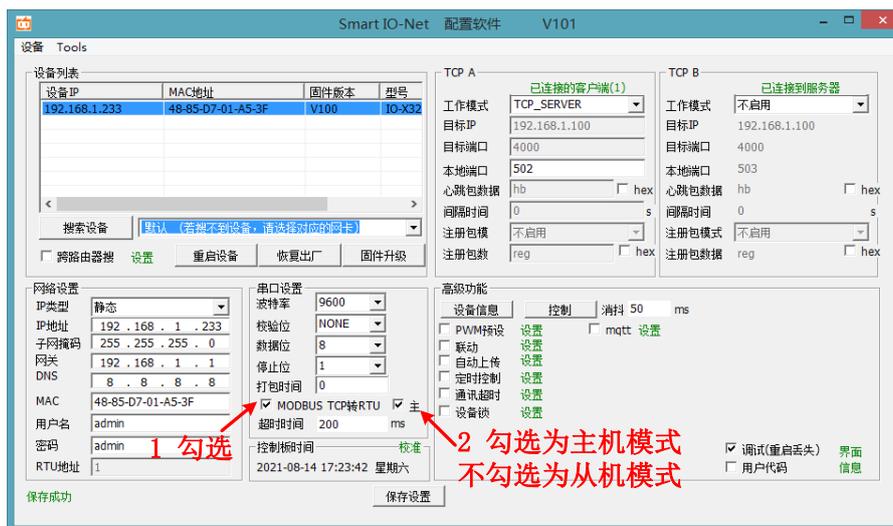


图 4.14 使能 ModBus TCP 转 RTU

4.11 二次开发

设备支持二次开发,提供丰富的函数库,方便满足用户的各种需求。具体介绍详见产品相关的二次开发文档。

5. 控制协议

设备支持 ASCII 格式、JSON 格式、ModBus RTU 格式、自定义指令格式控制及主动上报，对于符合这些控制协议的数据，设备会做出相应的动作，对于不符合这些协议的数据，设备会将数据透传。

本章主要介绍设备的控制协议。

5.1 ASCII 格式

通过 ASCII 格式协议，用户可设置 32 路 DO 输出状态，设置 DO 继电器延时断开时间，读取 DI 状态、DO 状态、脉冲计数值、AI 模拟量值。

5.1.1 设置 DO 继电器输出状态

设置 32 路 DO 继电器状态帧（主机）：

帧头	设备地址	命令码	第 1 路 DO	第 2 路 DO	第 3 路 DO	第 30 路 DO	第 31 路 DO	第 32 路 DO	帧尾
zq	0~63 其中 255 是广播地址	set	0: 断开 1: 闭合 2: 翻转 3: 不动作	0: 断开 1: 闭合 2: 翻转 3: 不动作	0: 断开 1: 闭合 2: 翻转 3: 不动作	0: 断开 1: 闭合 2: 翻转 3: 不动作	0: 断开 1: 闭合 2: 翻转 3: 不动作	0: 断开 1: 闭合 2: 翻转 3: 不动作	qz

应答帧：

帧头	设备地址	命令码	第 1 路 DO	第 2 路 DO	第 3 路 DO	第 30 路 DO	第 31 路 DO	第 32 路 DO	帧尾
zq	0~63 其中 255 是广播地址	ret	0: 断开 1: 闭合	0: 断开 1: 闭合	0: 断开 1: 闭合	0: 断开 1: 闭合	0: 断开 1: 闭合	0: 断开 1: 闭合	qz

例如：

```
发送：zq 1 set 1 1 1 1 1 1 1 1 qz //设置前 8 路 DO 继电器常开与公共端触点闭合
接收：zq 1 ret 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 qz //返回 32 路继电器状态
```

- ① 应答帧中的每一项以“空格”分开。
- ② 地址 255 是广播地址。

5.1.2 设置单路 DO 继电器状态

设置单路 DO 继电器状态：

帧头	设备地址	命令码	哪一路 DO 延时断开	DO 状态	帧尾
zq	0~63	set	y00~y31	0: 断开 1: 闭合 2: 翻转 3: 不动作	qz

应答帧：

帧头	设备地址	命令码	哪一路 DO 延时断开	DO 状态	帧尾
zq	0~63	ret	y00~y31	1: 闭合 0: 断开	qz

例如:

发送: zq 1 set y02 1 qz //控制第 2 路继电器闭合

应答: zq 1 ret y02 1 qz

5.1.3 设置 DO 继电器延时断开时间

从 DO 继电器常开触点闭合开始计时, 设置的延时时间到后, DO 继电器常开触点断开。延时时间范围 0~ 2147483647 毫秒。

设置单路 DO 继电器延时断开:

帧头	设备地址	命令码	哪一路 DO 延时断开	DO 状态	延时断开时间 (ms)	帧尾
zq	0~63	set	y00~y31	0: 断开 1: 闭合 2: 翻转 3: 不动作	0~2147483647	qz

应答帧:

帧头	设备地址	命令码	哪一路 DO 延时断开	DO 状态	延时断开时间 (ms)	帧尾
zq	0~63	ret	y00~y31	1: 闭合 0: 断开	0~2147483647	qz

例如:

发送: zq 1 set y02 1 5000 qz //控制第 2 路继电器闭合 5 秒后断开

应答: zq 1 ret y02 1 5000 qz

获取 DO 继电器还剩多长时间断开指令:

帧头	设备地址	命令码	哪一路 DO 延时断开	帧尾
zq	0~63	get	y00~y31	qz

应答帧:

帧头	设备地址	命令码	哪一路 DO 延时断开	DO 状态	延时断开时间 (ms)	帧尾
zq	0~63	ret	y00~y31	1: 闭合 0: 断开	0~2147483647	qz

例如:

发送: zq 1 get y02 qz //读取第 2 路 DO 继电器状态

应答: zq 1 ret y02 1 5000 qz //返回第 2 路继电器状态及还剩多长时间断开。

发送: zq 1 get x qz //读取所有 DI 状态
 应答: zq 1 ret x:00000000000000000000000000000000 qz //返回 DI 状态

5.1.6 只读 DO 继电器状态

查询设备 DO 状态帧:

帧头	设备地址	命令码	DO 输出状态	帧尾
zq	0~63	get	y	qz

应答帧:

帧头	设备地址	命令码	32 路 DO 输出状态	帧尾
zq	0~63	ret	y:y1~y32 0: DO 继电器断开 1: DO 继电器闭合	qz

例如:

发送: zq 1 get y qz //读取所有 DO 继电器状态
 应答: zq 1 ret y:00000000000000000000000000000000 qz //返回 DO 继电器状态

5.1.7 只读模拟量 AI 值

查询设备 ADC 值帧:

帧头	设备地址	命令码	ADC 值	帧尾
zq	0~63	get	adc	qz

应答帧:

帧头	设备地址	命令码	8 路 ADC 值	帧尾
zq	0~63	ret	adc:A11~A18	qz

例如:

发送: zq 1 get adc qz //读取 8 路 ADC 值
 应答: zq 1 ret adc:123 123 123 123 0 0 0 0 qz //返回 8 路 ADC 值

④ 模拟量计算公式: 0~5V 档位: $V = \text{adc} * 5 / 2027$ 。0~10V 档位: $V = \text{adc} * 10 / 2027$ 。
 0~20mA 档位: $I = \text{adc} * 20 / 2027$

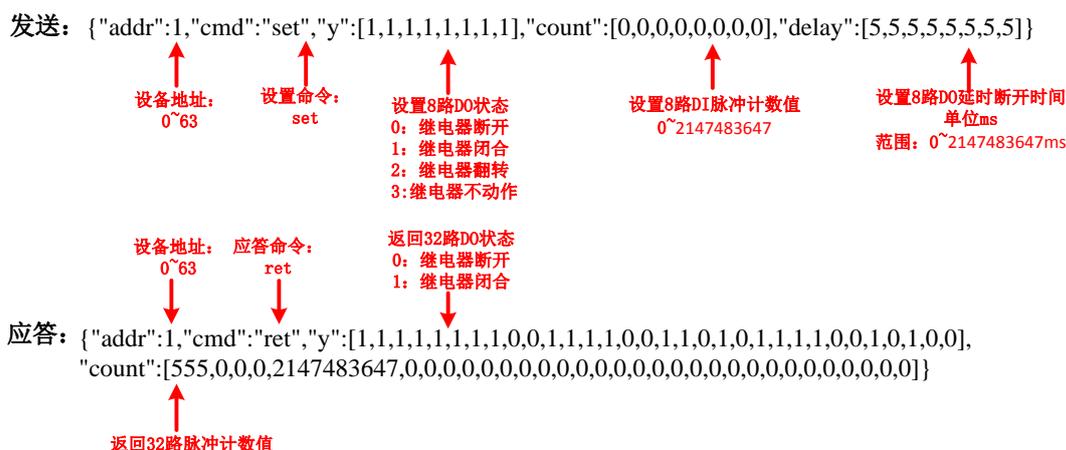
5.1.8 只读 DI 的脉冲计数值

查询设备脉冲计数值帧:

帧头	设备地址	命令码	脉冲计数值	帧尾
zq	0~63	get	count	qz

应答帧:

帧头	设备地址	命令码	28 路 DI 脉冲计数值	帧尾
zq	0~63	ret	count:00000000	qz



例如:

```
发送:
{"addr":1,"cmd":"set","y":[1,1,1,1,1,1,1,1],"count":[555,0,0,0,2147483647,0,0,0],"delay":[5000,5000,5000,5000,5000,5000,5000,5000]} //将 8 路继电器常开触点闭合，并延时 5 秒后，将常开触点断开；设置脉冲计数值。

应答:
{"addr":1,"cmd":"ret","y":[1,1,1,1,1,1,1,1,0,0,1,1,1,1,0,0,1,1,0,0,1,1,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,0],"count":[555,0,0,0,2147483647,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]} //返回状态
```

- ① 可单独控制某一项，如：{"addr":1,"cmd":"set","y":[1,1]} //控制 32 路继电器闭合。
- ② 设备地址取值范围为 0~63，其中地址 255 是广播地址。
- ③ 脉冲计数值范围：0~2147483647。
- ④ 继电器延时断开时间范围：0~2147483647。

5.2.2 读取设备所有接口状态

通过该指令可以读取 DI 状态、DO 状态、ADC 值、脉冲计数值。

- ① 设备地址范围 0~63，其中 255 是广播地址。
- ① “cmd”:后的数据为命令码，“upload”为设备数据主动上传服务器命令。
- ① “x”:后的数据为设备的 DI 输入状态，0 代表 DI 无有效信号，1 代表对应 DI 接口有有效信号。
- ① “y”:后的数据为设备的 DO 输出状态，0 代表 DO 继电器常开与公共端触点断开，1 代表 DO 继电器常开与公共端触点闭合。
- ① “adc”:后的数据为设备 AI 接口的模拟量数值。
- ① “count”:后的数据为设备的 DI 脉冲计数值，数值代表对应 DI 接口有有效信号的次数。
- ① “mac”:后的数据为设备的 MAC 地址，每台设备的 MAC 地址都是唯一的，可以用来识别数据来源。
- ① “id”: 后的数据，用户可以自定义 id，最大不超过 64 个字节，可用于识别数据来源。

5.3 ModBus RTU 格式

ModBus RTU 协议是工业上常用的通讯协议，采用查询+响应的通讯模式：主站设备向指定地址的从站设备发送请求报文，从站设备收到指令后做出相应的动作，并向主站发送响应报文。ModBus RTU 协议的通讯格式如下：



5.3.1 地址域（Additional address）

Additional address 为从站地址，取值范围 0x00~0xff。

当 Additional address 为 0xff 时，是广播地址，所有从机都能接收并处理，必要时要做回应。广播地址可以用于对控制板的编址以及获取控制板的地址。

5.3.2 功能码（Function code）

本设备支持的 ModBus RTU 功能码如表 5.1 所示。

表 5.1 ModBus RTU 功能码

功能码	含义	功能	属性
0x01	读多个线圈	Data: 2 字节起始地址+2 字节线圈个数，线圈个数不能超过 8	读
0x02	读离散量输入	Data: 2 字节起始地址+2 字节输入点个数，个数不能超过 8	读
0x03	读多个保持寄存器	Data: 2 字节起始地址+2 字节寄存器个数（寄存器含义见表 5.2）	读
0x05	写单个线圈	Data: 2 字节起始地址+2 字节线圈值	写
0x06	写单个保持寄存器	Data: 2 字节起始地址+2 字节寄存器值	写
0x10	写多个保持寄存器	Data: 2 字节起始地址+2 字节寄存器数量+1 字节数据长度+数据	写
0x0f	写多个线圈	Data: 2 字节起始地址+2 字节线圈个数+1 字节个数+数值	写

1. 功能码：0x01

读 DO 继电器输出状态：

字节数	1	1	1	1	1	1	1	1
名称	设备地址	功能码 0x01	起始地址 高	起始地址 低	读取数量 高	读取数量 低	CRC 高	CRC 低
取值范围	0x01~0xff	0x01	0x00	0x00~0x00	0x00	0x00~0x08	CRC 计算值	

2. 功能码：0x02

读 DI 输入状态：

字节数	1	1	1	1	1	1	1	1
名称	设备地址	功能码 0x02	起始地址 高	起始地址 低	读取数量 高	读取数量 低	CRC 高	CRC 低
取值范围	0x01~0xff	0x02	0x00	0x00~0x00	0x00	0x00~0x08	CRC 计算值	

3. 功能码：0x03

读取设备所有接口状态：

字节数	1	1	1	1	1	1	1	1
名称	设备地址	功能码 0x03	寄存器起始地址 高	寄存器起始地址 低	寄存器数量 高	寄存器数量 低	CRC 高	CRC 低
取值范围	0x01~0xff	0x03	见表 5.2		0x00		CRC 计算值	

应答数据帧：

字节数	1	1	1	1~60			1	1
名称	设备地址	功能码 0x03	数据长度	每项数据长度、DI 状态、DO 状态、ADC 值、脉冲计数值			CRC 高	CRC 低
取值范围	0x01~0xff	0x03					CRC 计算值	

ModBus 地址信息

表 5.2 ModBus 地址信息表

寄存器地址	项目	描述
0x0000	DI 输入状态的数据长度 (只读)	DI 输入状态的数据长度，2 个字节 如：0x00 20 代表 DI 输入状态有 32 个字节数据
0x0001~0x0010	DI 输入状态 (只读)	DI1~DI32 输入状态，每个字节代表 1 路 DI 状态，共 32 个字节 00：设备 DI 接口无有效信号 01：设备 DI 接口有有效信号
0x0011	DO 输出状态的数据长度 (只读)	DO 输出状态的数据长度，2 个字节 如：0x00 20 代表 DO 输出状态有 32 个字节数据
0x0012~0x0021	DO 输出状态 (可读，可写)	DO1~DO32 输出继电器状态，每个字节代表 1 路 DO 状态，共 32 个字节 00：设备 DO 继电器常开和公共端触点断开 01：设备 DO 继电器常开和公共端触点闭合

0x0022	AI 模拟量的数据长度 (只读)	AI 模拟量的数据长度, 2 个字节 如: 0x00 10 代表 AI 模拟量值有 16 个字节数据
0x0023~0x002A	AI 模拟量值 (只读)	AI1~AI8 模拟量值, 2 个字节代表 1 路 AI 值, 共 16 个字节
0x002B	DI 脉冲计数的数据长度 (只读)	DI 脉冲计数的数据长度, 2 个字节 如: 0x00 80 代表 DI 脉冲计数的数据有 128 个字节数据
0x002C~0x006B	DI 脉冲计数值 (可读, 可写)	DI1~DI32 输入脉冲计数值, 4 个字节代表 1 路 DI 的脉冲计数值, 共 128 个字节 脉冲计数值范围 0x00000000~0xffff ffff
0x03E8~0x03EA	第 1 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03EB~0x03ED	第 2 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03EE~0x03F0	第 3 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03F1~0x03F3	第 4 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03F4~0x03F6	第 5 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03F7~0x03F9	第 6 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03FA~0x03FC	第 7 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x03FD~0x03FF	第 8 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0400~0x0402	第 9 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0403~0x0405	第 10 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0406~0x0408	第 11 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff

0x0409~0x040B	第 12 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x040C~0x040E	第 13 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x040F~0x0411	第 14 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0412~0x0414	第 15 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0415~0x0417	第 16 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0418~0x041A	第 17 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x041B~0x041D	第 18 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x041E~0x0420	第 19 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0421~0x0423	第 20 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0424~0x0426	第 21 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0427~0x0429	第 22 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x042A~0x042C	第 23 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x042D~0x042F	第 24 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0430~0x0432	第 25 路 DO 继电器延时断开 (可读, 可写)	设置/读取继电器状态及延时断开时间, 共 6 个字节。 前两个字节代表输出状态: 0x0000 断开, 0x0001 闭合 后四个字节代表延时断开时间: 0x0000 0000~0xffff ffff
0x0433~0x0435	第 26 路 DO 继电器延时断开	设置/读取继电器状态及延时断开时间, 共 6 个字节。

62 4F //返回两路 AI 的 0~5V 档位的电压值

④ 模拟量计算公式：0~5V 档位： $V = \text{adc} * 5 / 2027$ 。0~10V 档位： $V = \text{adc} * 10 / 2027$ 。
0~20mA 档位： $I = \text{adc} * 20 / 2027$

5.3.3 ModBus 格式主动上报

首先用上位机软件使能主动上报功能，并配置主动上报触发条件、上报格式等参数，具体配置方法详见 1.4。自动上报案例详见 8.2 小节。

ModBus 格式主动上报是以 0x03 功能码的格式上传的，具体参考 ModBus 格式 0x03 功能码小节介绍。

5.4 自定义指令格式

通过智嵌物联提供的自定义协议，可以读取设备的 DI 状态（命令码：0x52）、读取 DO 状态（命令码：0x72、0x53）、控制 DO 输出（命令码：0x70、0x57）、读取设备所有相关数据（命令码：0x20）。

1. 集中控制指令

表 5.4 ZQWL-IO 集中控制指令表

项目	帧头		地址码	命令码	DI/DO 状态 8 字节数据	校验和	帧尾	
	Byte1	Byte2					Byte14	Byte15
字节数	Byte1	Byte2	Byte3	Byte4	Byte5~ Byte12	Byte13	Byte14	Byte15
读输入状态	0X48	0X3A	Addr	0X52	任意值	前 12 字节和（只取低 8 位）	0X45	0X44
应答“读输入状态”	0X48	0X3A	Addr	0X41	DATA1~DATA8	前 12 字节和（只取低 8 位）	0X45	0X44
写继电器状态	0X48	0X3A	Addr	0X57	DATA1~DATA8	前 12 字节和（只取低 8 位）	0X45	0X44
应答“写继电器状态”	0X48	0X3A	Addr	0X54	DATA1~DATA8	前 12 字节和（只取低 8 位）	0X45	0X44
读继电器状态	0X48	0X3A	Addr	0X53	任意值	前 12 字节和（只取低 8 位）	0X45	0X44
应答“读继电器状态”	0X48	0X3A	Addr	0X54	DATA1~DATA8	前 12 字节和（只取低 8 位）	0X45	0X44

说明：

上表中的“8 字节数据”即对应模块继电器状态，2 个 bit 表示 1 路继电器，每 1 个字节表示 4 路继电器。

8 字节数据(DATA1~DATA8)与继电器的对应关系如表：

DATA1								
Bit	7	6	5	4	3	2	1	0
含义	第 4 路继电器		第 3 路继电器		第 2 路继电器		第 1 路继电器	
DATA2								
Bit	7	6	5	4	3	2	1	0
含义	第 8 路继电器		第 7 路继电器		第 6 路继电器		第 5 路继电器	
DATA3								
Bit	7	6	5	4	3	2	1	0
含义	第 12 路继电器		第 11 路继电器		第 10 路继电器		第 9 路继电器	

DATA4								
Bit	7	6	5	4	3	2	1	0
含义	第 16 路继电器		第 15 路继电器		第 14 路继电器		第 13 路继电器	
DATA5								
Bit	7	6	5	4	3	2	1	0
含义	第 20 路继电器		第 19 路继电器		第 18 路继电器		第 17 路继电器	
DATA6								
Bit	7	6	5	4	3	2	1	0
含义	第 24 路继电器		第 23 路继电器		第 22 路继电器		第 21 路继电器	
DATA7								
Bit	7	6	5	4	3	2	1	0
含义	第 28 路继电器		第 27 路继电器		第 26 路继电器		第 25 路继电器	
DATA8								
Bit	7	6	5	4	3	2	1	0
含义	第 32 路继电器		第 31 路继电器		第 30 路继电器		第 29 路继电器	

每个继电器状态占用 2 个 bit 位，其含义如表：

2 bit 高位	2 bit 低位	含义
0	0	继电器断开
0	1	继电器闭合
1	0	继电器保持原状态
1	1	继电器保持原状态

2 个 bit 为 00 或 01 时分别代表该路继电器的两种不同的状态。在使用“写继电器状态”命令中，**如果不想改变某个继电器的状态，那么就可以将该路继电器代表的 bit 写为 10 或 11。**

集中控制命令码举例（十六进制）：

向地址为 1 的控制板写继电器状态：

发送：48 3a 01 57 01 00 01 00 00 00 00 00 dc 45 44 //第 1 个和第 9 个继电器闭合；其余继电器为断开。

注意继电器板只识别 0 和 1，其他数据不做任何动作，所以如果不想让某一路动作，可以将该路赋为其他值。例如不让第 5 个和第 9 个继电器改变状态，可以发如下指令：

发送：48 3a 01 57 01 02 01 02 00 00 00 00 e0 45 44

只需要将第 5 和第 9 路置为 02（或其他值）即可。

控制板收到以上命令后，会返回控制板继电器状态，如：

应答：48 3a 01 54 01 00 01 00 00 00 00 00 d9 45 44

① 地址码:为设备的地址，可以通过上位机软件进行配置，取值范围 0x01~0xff，设备默认地址为 0x01。地址 0xff 为广播地址。

② 校验和: 前 12 字节和（只取低 8 位）。

2. 单路继电器输出控制指令（延时断开时间单位：秒）

此类指令帧长为 10 字节，可以实现对单路继电器的控制（一帧数据只能控制一个继电器状态）。此类指令也可以实现继电器的延时断开功能，延时时间单位：秒。

详细指令如表 5.5 所示。

表 5.5 ZQWL-IO 单路控制指令表 1

项目	帧头		地址码	命令码	继电器状态 2 字节数据		继电器 延时断开时间 (s)		帧尾	
	Byte1	Byte2			Byte5	Byte6	Byte7	Byte8	Byte9	Byte10
写继电器状态	0X48	0X3A	Addr	0X70	继电器序号	继电器状态	时间 TH	时间 TL	0X45	0X44
应答“写继电器状态”	0X48	0X3A	Addr	0X71	继电器序号	继电器状态	时间 TH	时间 TL	0X45	0X44
读继电器状态	0X48	0X3A	Addr	0X72	继电器序号	继电器状态	时间 TH	时间 TL	0X45	0X44
应答“读继电器状态”	0X48	0X3A	Addr	0X71	继电器序号	继电器状态	时间 TH	时间 TL	0X45	0X44

说明：

Byte3 是控制板的地址，取值范围 0x00~0xfe，用户可以通过配置指令来设置地址码；

Byte5 是要操作的继电器序号，取值范围是 1 到 32（对应十六进制为 0x01 到 0x20）；

Byte6 为要操作的继电器状态：0x00 为常闭触点闭合常开触点断开，0x01 为常闭触点断开常开触点闭合，其他值为继电器保持原来状态；

Byte7 和 Byte8 为延时时间 T（收到 Byte6 为 0x01 时开始计时，延时结束后关闭该路继电器输出），延时单位为秒，Byte7 是时间高字节 TH，Byte8 是时间低字节 TL。例如延时 10 分钟后关闭继电器，则：

时间 T=10 分钟=600 秒，换算成十六进制为 0x0258，所以 TH=0x 02，TL=0x 58。

如果 Byte7 和 Byte8 都填 0x00，则不启用延时关闭功能（即继电器闭合后不会主动断开）。

① 命令码 0x70、0x71、0x72，这三个命令码设置的延时断开时间单位是秒（S）。

单路命令码举例（十六进制）：

将地址为 1 的控制板的第 1 路继电器打开：

发送：48 3a 01 70 01 01 00 00 45 44 //设备收到该命令后，将第 1 路继电器常闭触点断开，常开触点闭合
 应答：48 3a 01 70 01 01 00 00 45 44 //应答第一路继电器状态

将地址为 1 的控制板的第 1 个继电器关闭：

发送：48 3a 01 70 01 00 00 00 45 44 //设备收到该命令后，将第 1 路继电器常闭触点闭合，常开触点断开
 应答：48 3a 01 71 01 00 00 00 45 44 //应答第一路继电器状态

将地址为 1 的控制板的第 1 路继电器打开延时 10 分钟后关闭：

发送：48 3a 01 70 01 01 02 58 45 44

控制板收到以上命令后，将第 1 路的继电器常闭触点断开，常开触点闭合，并会返回控制板继电器状态，然后开始计时，10 分钟之后将第一路的继电器常闭触点闭合，常开断开。

将地址为 1 的控制板的第 1 路继电器打开延时 5 秒后关闭：

发送：48 3a 01 70 01 01 00 05 45 44

控制板收到以上命令后，将第 1 路的继电器常闭触点断开，常开触点闭合，并会返回控制板继电器状态，然后开始计时，5 秒之后将第一路的继电器常闭触点闭合，常开断开。

3. 单路继电器输出控制指令（延时断开时间单位：毫秒）

此类指令帧长为 12 字节，可以实现对单路继电器的控制（一帧数据只能控制一个继电器状态）。此类指令也可以实现继电器的延时断开功能，延时控制时间单位：毫秒。

表 5.6 ZQWL-IO 单路控制指令表 1

项目	帧头		地址码	命令码	控制继电器 4 字节数据		继电器 延时断开时间 (ms)				帧尾	
	Byte1	Byte2			Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
写继电器状态	0X48	0X3A	Addr	0X73	继电器序号	继电器状态	时间 TH1	时间 TL1	时间 TH2	时间 TL2	0X45	0X44
应答“写继电器状态”	0X48	0X3A	Addr	0X74	继电器序号	继电器状态	时间 TH1	时间 TL1	时间 TH2	时间 TL2	0X45	0X44
读继电器状态	0X48	0X3A	Addr	0X75	继电器序号	继电器状态	时间 TH1	时间 TL1	时间 TH2	时间 TL2	0X45	0X44
应答“读继电器状态”	0X48	0X3A	Addr	0X74	继电器序号	继电器状态	时间 TH1	时间 TL1	时间 TH2	时间 TL2	0X45	0X44

说明：

Byte3 是控制板的地址，取值范围 0x00~0xfe，用户可以通过配置指令来设置地址码；

Byte5 是要操作的继电器序号，取值范围是 1 到 32（对应十六进制为 0x01 到 0x20）；

Byte6 为要操作的继电器状态：0x00 为常闭触点闭合常开触点断开，0x01 为常闭触点断开常开触点闭合，其他值为继电器保持原来状态；

Byte7~Byte10 为延时时间 T（收到 Byte6 为 0x01 时开始计时，延时结束后关闭该路继电器输出），延时单位为毫秒，例如延时 10 分钟后关闭继电器，则：

时间 T=10 分钟=600 秒=600000 毫秒，换算成十六进制为 0x00 09 27 0C，所以 TH1=0x00，TL1=0x09，TH2=0x27，TL2=0x0C。

如果 Byte7~Byte10 都填 0x00，则不启用延时关闭功能(即继电器闭合后不会主动断开)。

④ 命令码 0x73、0x74、0x75，这三个命令码设置的延时断开时间单位是毫秒（ms）。

单路命令码举例（十六进制）：

将地址为 1 的控制板的第 1 路继电器打开：

发送：48 3a 01 73 01 01 00 00 45 44 //设备收到该命令后，将第 1 路继电器常闭触点断开，常开触点闭合
 应答：48 3a 01 74 01 01 00 00 45 44 //应答第一路继电器状态

将地址为 1 的控制板的第 1 个继电器关闭：

首先用上位机软件使能主动上报功能，并配置主动上报触发条件、上报格式等参数，具体配置方法详见 1.4。自动上报案例详见 8.2 小节。

自定义格式主动上报，是按自定义命令码 0x21 格式上报的，详情参考自定义命令 0x21 相关介绍。

6. 固件升级

升级前，请先下载最新的固件，可在官网下载或直接向公司销售索要。

通过网口对本设备进行升级，具体步骤如图 6.1 所示。

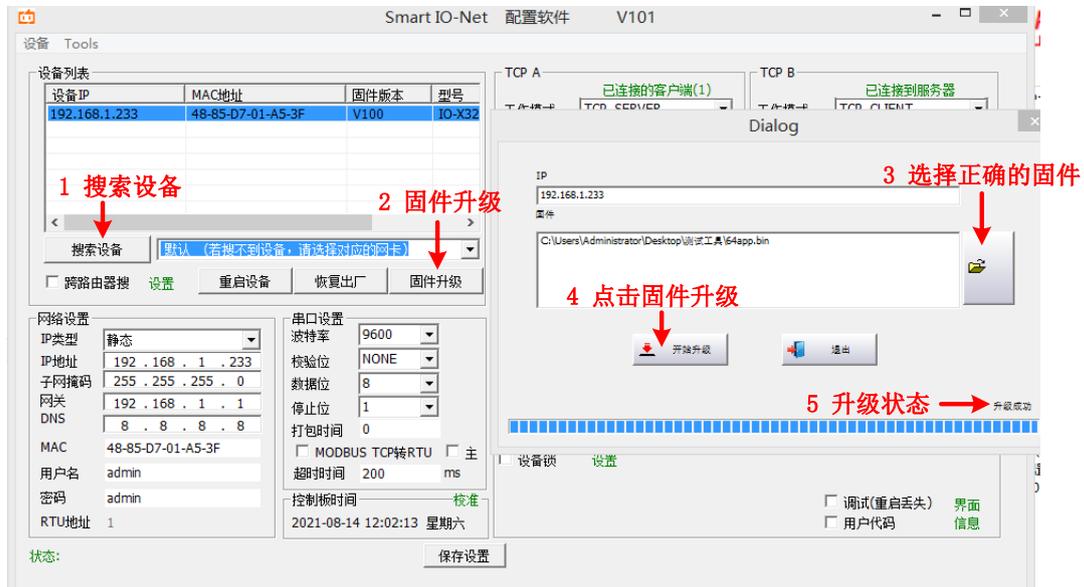


图 6.1 本地升级界面 1

7. 恢复出厂设置

设备恢复出厂后，所有的参数会重新设置到出厂时默认的参数。设备恢复出厂的方法有两种：通过配置软件恢复出厂、通过硬件恢复出厂。

表 7.1 设备默认参数

项目	默认参数
波特率	9600bps
数据位	8
校验位	NONE
停止位	1
工作模式	网络透传
设备 IP	192.168.1.253

1. 通过配置软件恢复出厂

使用“智嵌物联 4G DTU&RTU 配置软件”来恢复出厂。具体步骤如图 7.1 所示。

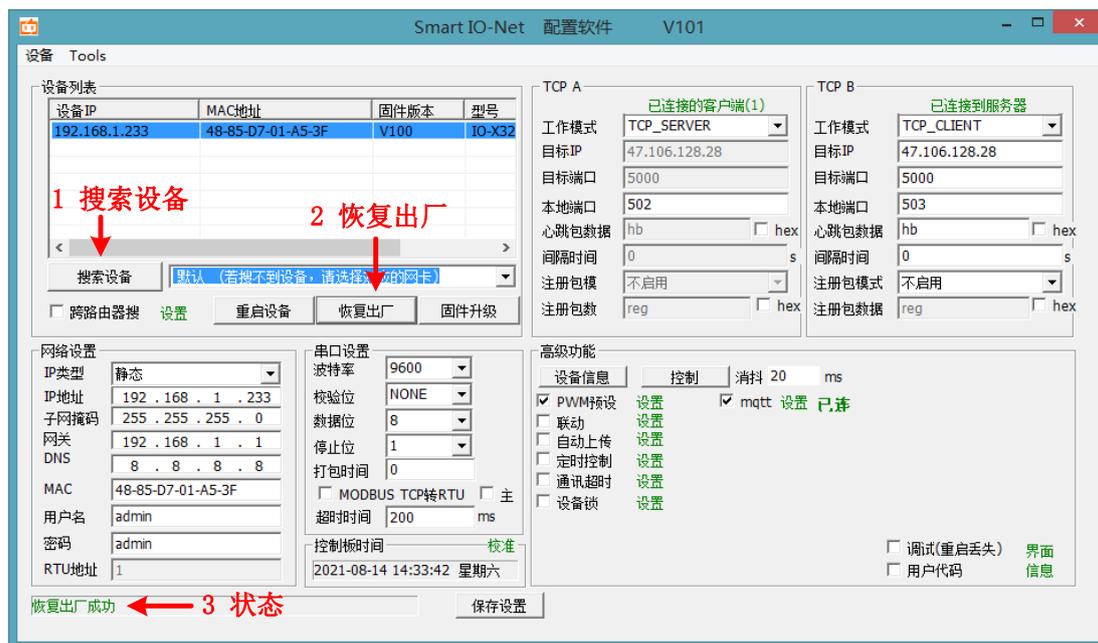


图 7.1 用配置软件恢复出厂步骤

2. 硬件恢复出厂

设备上有用于复位及恢复出厂的按键，面板丝印“Reset”。

短按“Reset”按键，复位设备；

长按“Reset”按键，并保持 6 秒以上（RUN 灯快闪），实现恢复出厂设置；

8. 应用案例

8.1 设备接入智嵌云平台实现远程控制案例

智嵌物联的网络 IO 控制器（云版本）支持接入智嵌云，实现设备的远程控制与检测。本文档以智嵌的 8 路网络 IO 控制器为例，介绍设备接入智嵌云的方法，以及实现网页控制及手机 APP 控制的方法。

3. 硬件连接

用交换机将 IO 控制器设备、路由器、电脑通过网线连接起来，如图 8.1 所示，用电源适配器为设备供电。供电后请先观察设备指示灯是否正常，如表 1 所示。

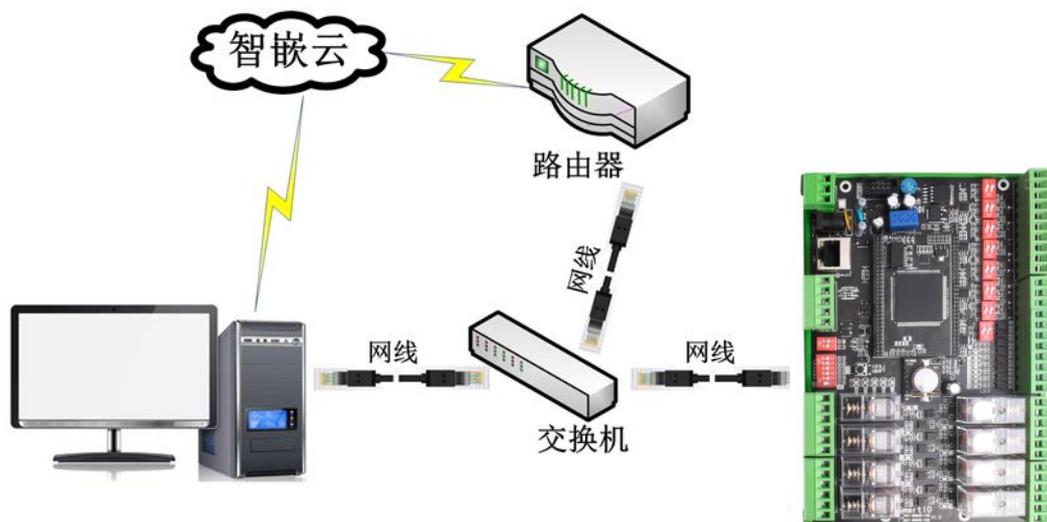


图 8.1 硬件连接

表 8.1 设备指示灯意义

指示灯	设备正常时
电源指示灯（PWR）	常亮
运行指示灯（RUN）	闪烁（频率约 1HZ）
网口灯	一个灯常亮，一个灯有数据时会闪烁

4. 登录智嵌云平台账号

浏览器进入智嵌云管理平台，网址：www.zqwliot.com，选择智嵌云控的新版本界面，如图 8.2 所示。进入登录界面，注册并登录智嵌云平台账号，若已注册，直接登录即可，如图 8.3 所示。



图 8.2 智嵌云管理平台



图 8.3 智嵌云登录界面

5. 创建项目

智嵌云平台是通过项目的方式来管理设备的，因此在用户添加设备之前要先创建一个项目及分组，如图 8.4 所示。



图 8.4 智嵌云平台创建项目步骤

6. 添加设备

此步骤的目的是在用户的账号下添加该设备，此步完成后，系统会自动生成 SN 号，该 SN 号会在绑定设备时用到。

在已创建的项目分组下添加继电器控制板设备，具体步骤如图 8.5 所示。



图 8.5 智嵌云平台添加设备步骤

设备添加完成后，在设备列表菜单下找到刚添加的设备，复制设备 SN 及通讯密码，后面备用。



图 8.6 添加设备完成

7. 配置设备参数

此步骤的目的是将网络设备与智嵌云平台建立连接。

打开“网络 IO 控制板配置软件”，按图 8.7 所示步骤进行配置。保存参数后，新参数即可生效。



图 8.7 设备参数配置

8. 网页控制

以上步骤完成后，设备已经接入到智嵌云上，如图 8.8 所示，设备已在线，此时可以通过网页或 APP 控制设备。



图 8.8 设备在线

点击“进入”，可跳转到设备的控制界面，如图 8.9、图 8.10 所示。鼠标点击每一路继电器的“开”、“关”、“翻转”，继电器会做出相应的动作。点击“刷新状态”，可实时获取设备的开关量输入状态。



图 8.9 设备控制界面



图 8.10 设备开关量输入检测界面

9. 手机 APP 控制

下载“智嵌云控”APP。用手机浏览器扫面以下二维码下载 APP 安装包。



图 8.11 “智嵌云控”APP 下载二维码

打开“智嵌云控”APP，进入登录界面（第一次使用需先注册智嵌云平台账号）。如图 8.12 所示。

① 若已通过网页方式注册账号，无需再注册，可以直接登录。



图 8.12 “智嵌云控” APP 登录界面

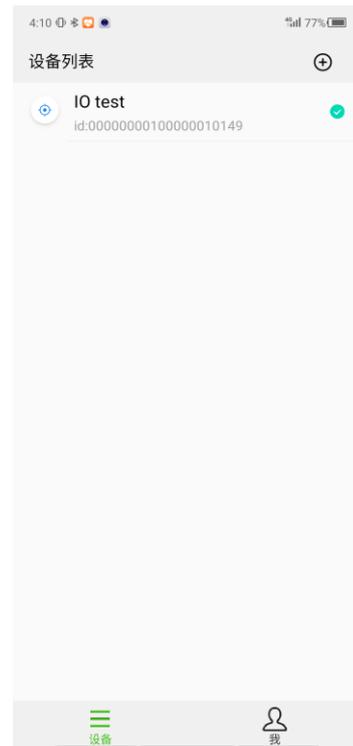


图 8.13 已添加的 IO test 设备

登录后，就可以看到之前添加的设备“IO test”了，如图 8.12 所示。

下面就可以对 IO 控制器进行远程控制了。如图 8.12、图 8.14、图 8.16 所示。



图 8.14 远程控制继电器输出



图 8.15 远程检测开关量输入状态

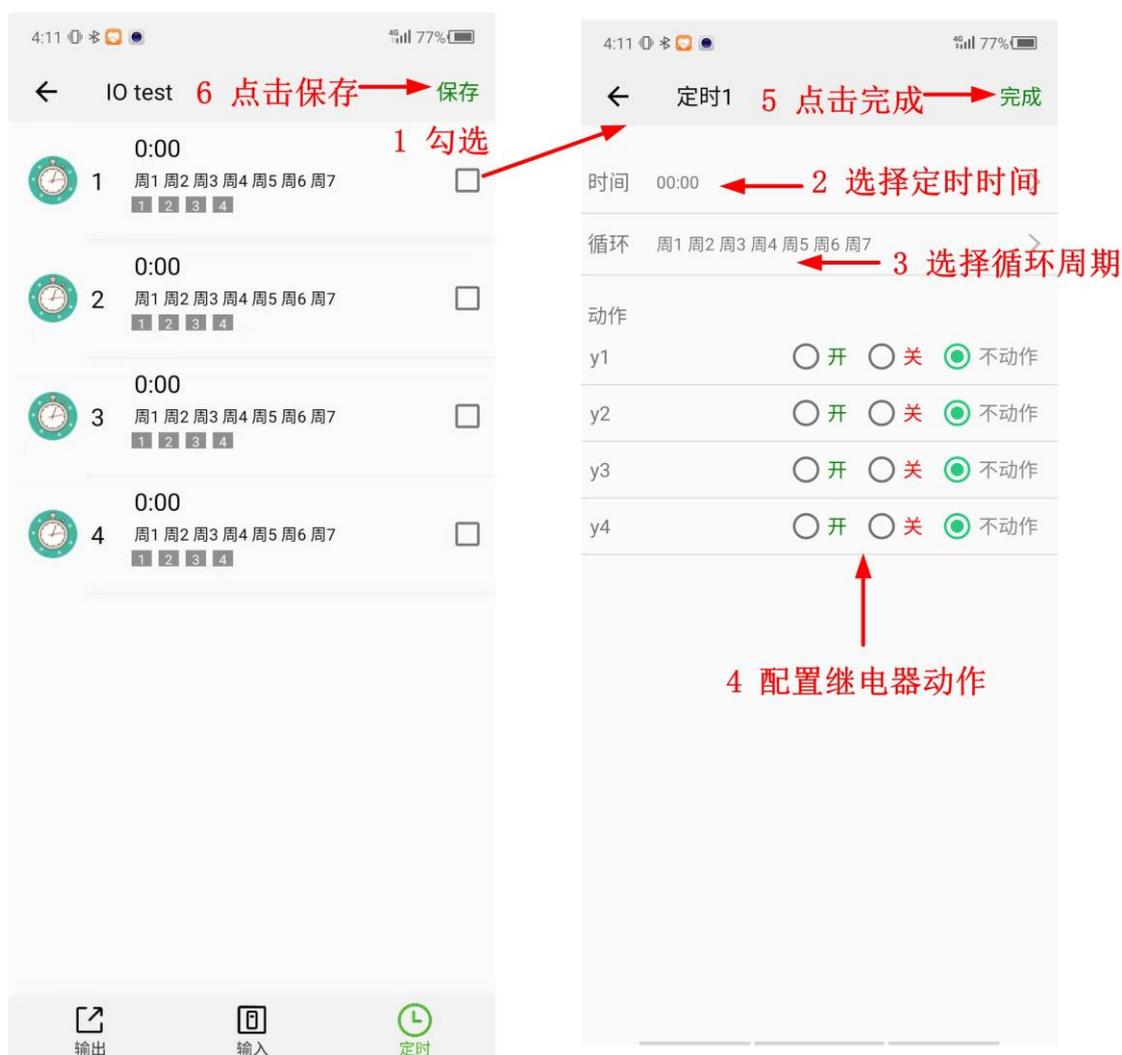


图 8.16 定时控制继电器输出

① 当设备的数字量输入接口上有有效信号时，用户可通过刷新手机输入信号界面，查看输入信号状态（当用户开启设备的自动上报功能后，可不用手动刷新）

8.2 设备 DI、DO、AI、脉冲计数主动上报

1. 配置设备参数

主动上传支持条件触发上传和定时上传，用户可根据需求设置，具体设置方法如图 8.17。

上传格式支持：JSON 格式、ASCII 格式、ModBus 格式、自定义格式，用户可根据需求设置。第一次使用上报功能，建议用户每种格式都尝试一遍，以便找到适合自己应用的协议格式。

④ 第一次使用设备的自动上报功能，建议每种上报格式都试一下，以便选一种最适合自己的应用的格式。

8.3 设备与组态王等组态软件通信

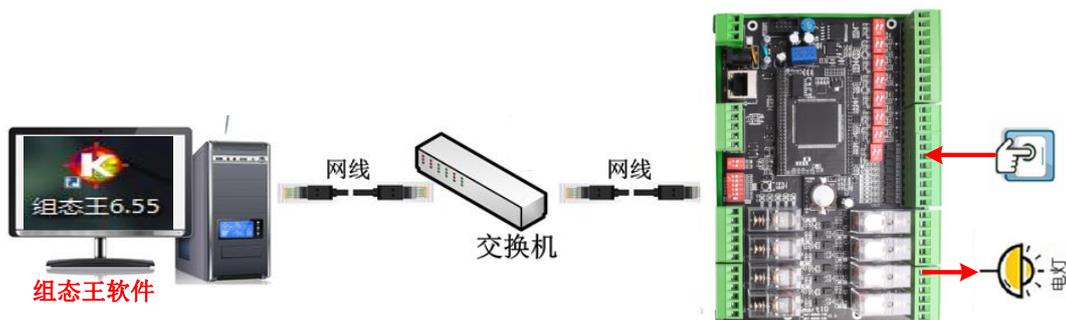


图 8.23 设备与组态王通信硬件连接

1. 硬件连接

设备上电；用网线连接设备和电脑。

2. 创建组态王工程软件

打开组态王软件 6.55，新建工程“MODBUS RTU”。

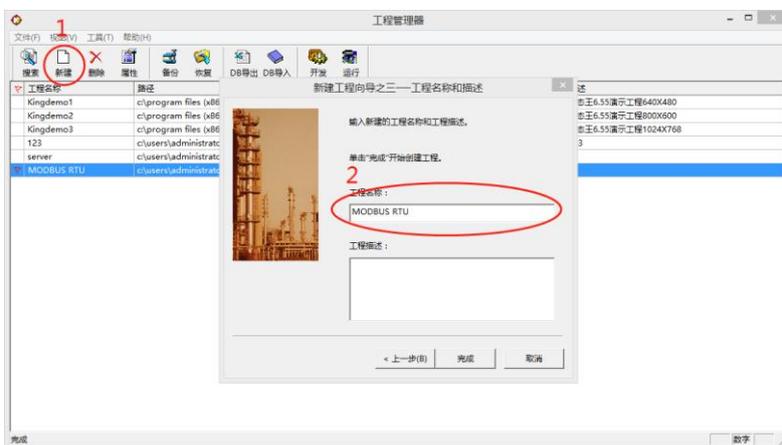


图 8.24 组态王新建工程

3. 组态王新建 ModBus TCP 设备

本此演示设备选择莫迪康的 ModBus TCP 设备，具体设置步骤如图 8.25、图 8.26、图 8.27、图 8.28 所示。

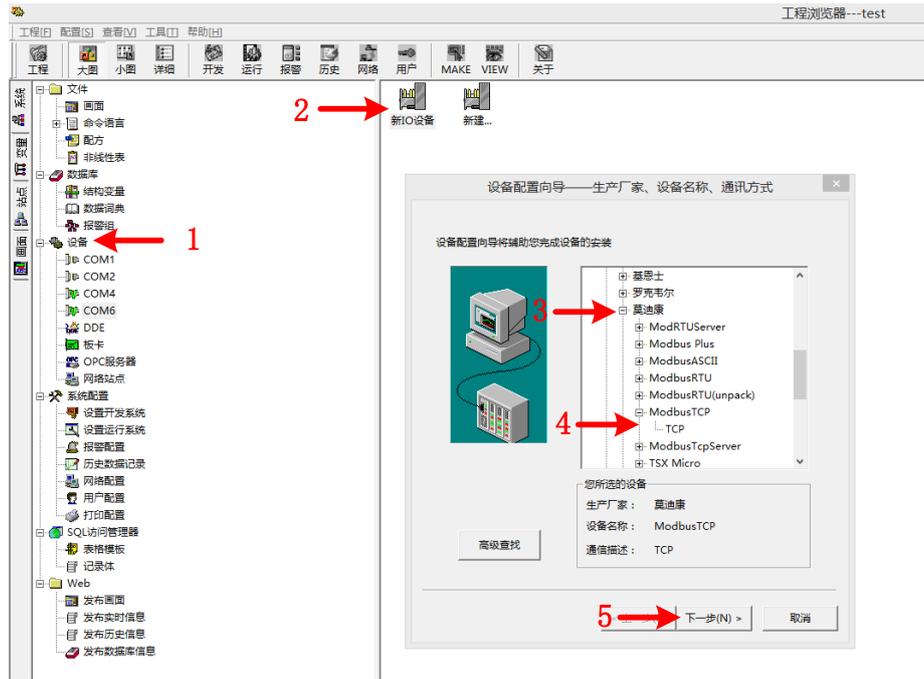


图 8.25 组态王新建设备步骤 1



图 8.26 组态王新建设备步骤 2

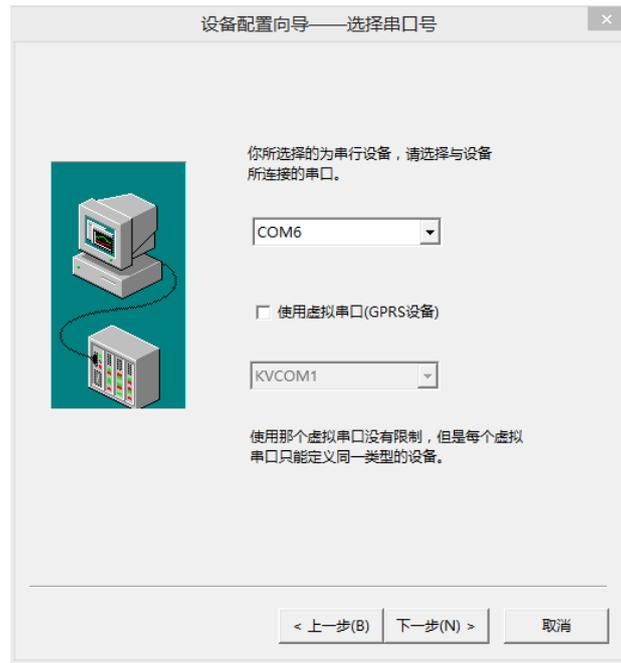


图 8.27 组态王新建设备步骤 3



图 8.28 组态王新建设备步骤 4

4. 添加自定义变量

按图 8.29、图 8.30、图 8.31 所示步骤分别新建变量：

“DI1” ~ “DI8”：功能码为 0x02，寄存器地址分别为 0~7，对应组态王软件的寄存器分别为 100001~100008。

“DO1” ~ “DO8”：功能码为 0x01，寄存器地址分别为 0~7，对应组态王软件的寄存

器分别为 000001~000008。

“AI1”~“AI8”：功能码为 0x04，AI1 的起始寄存器地址为 100，AI8 的起始地址为 114，对应组态王软件的寄存器分别为 300101、300103、300105、300107、300109、300111、300113、300115。

COUNT1~COUNT4：功能码为 0x03，COUNT1 的起始寄存器地址是 44；COUNT2 的起始寄存器地址是 46；COUNT1 的起始寄存器地址是 48；COUNT2 的起始寄存器地址是 50；对应组态王软件的寄存器分别为 40045、40047、40049、40051。



图 8.29 组态王添加 DI 变量

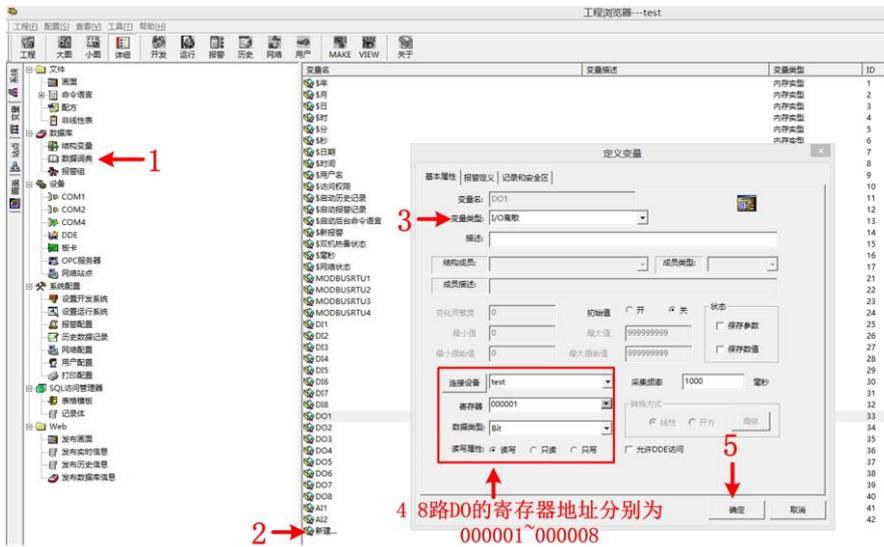


图 8.30 组态王添加 DO 变量

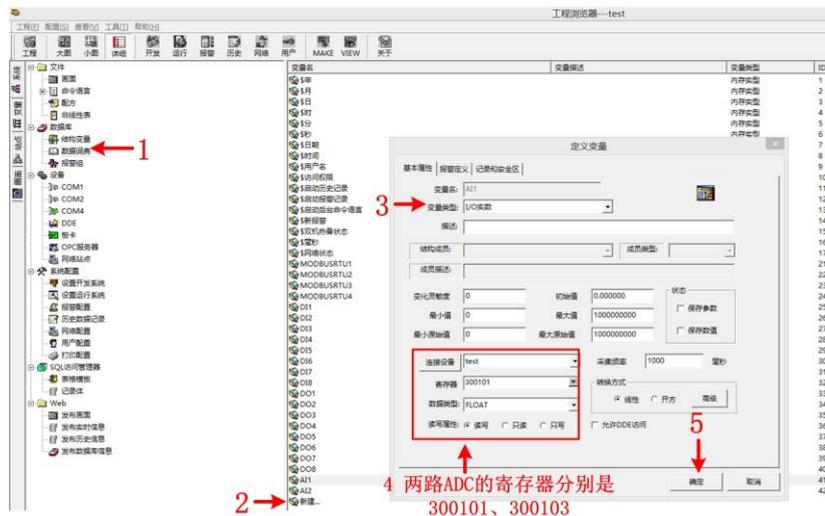


图 8.31 组态王添加 AI 变量

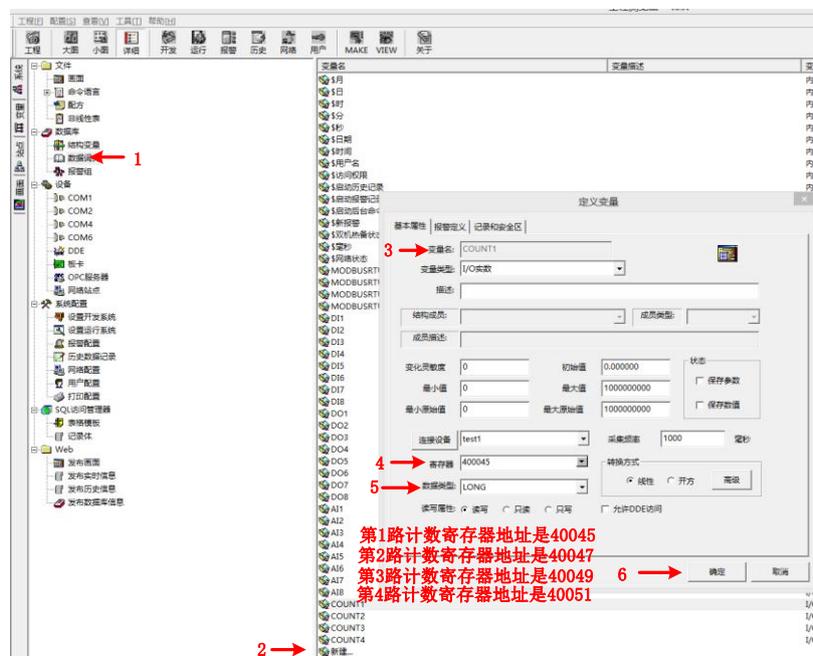


图 8.32 组态王添加脉冲计数变量

5. 创建新画面

新画面取名“MODBUSRTU”，如图 8.33 所示。



图 8.33 组态王创建新画面

6. 分别双击界面中的按钮及文本，在弹出的对话框中，按图 8.34、图 8.35、图 8.36 步骤设置。

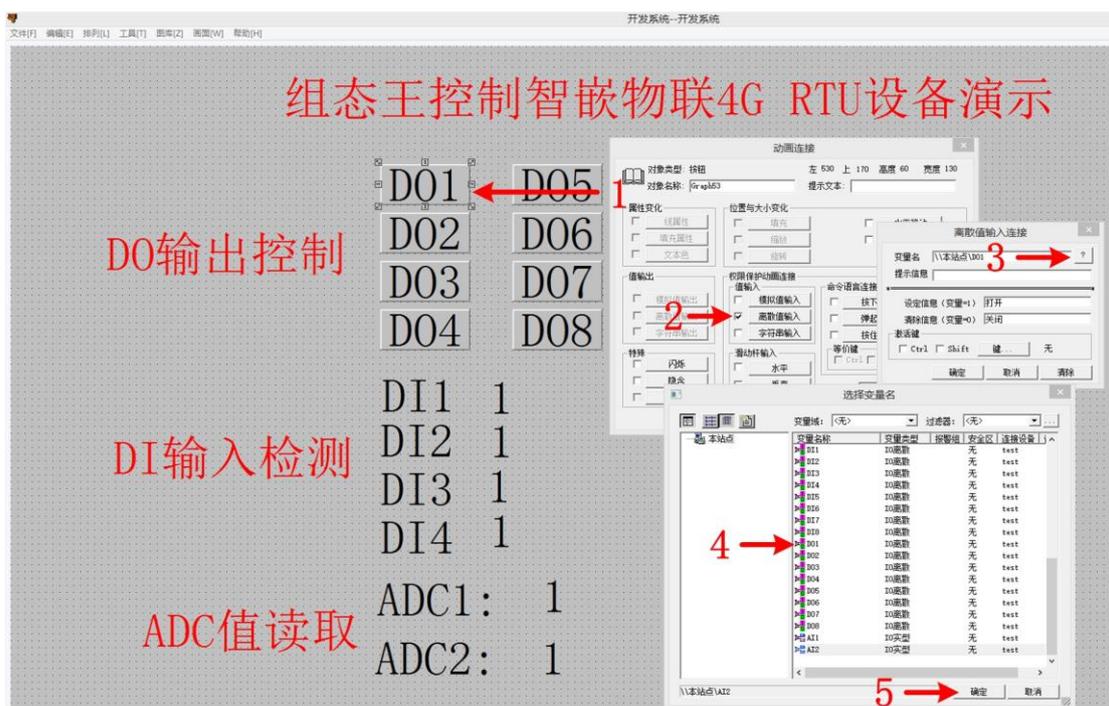


图 8.34 组态王变量关联 DO 输出

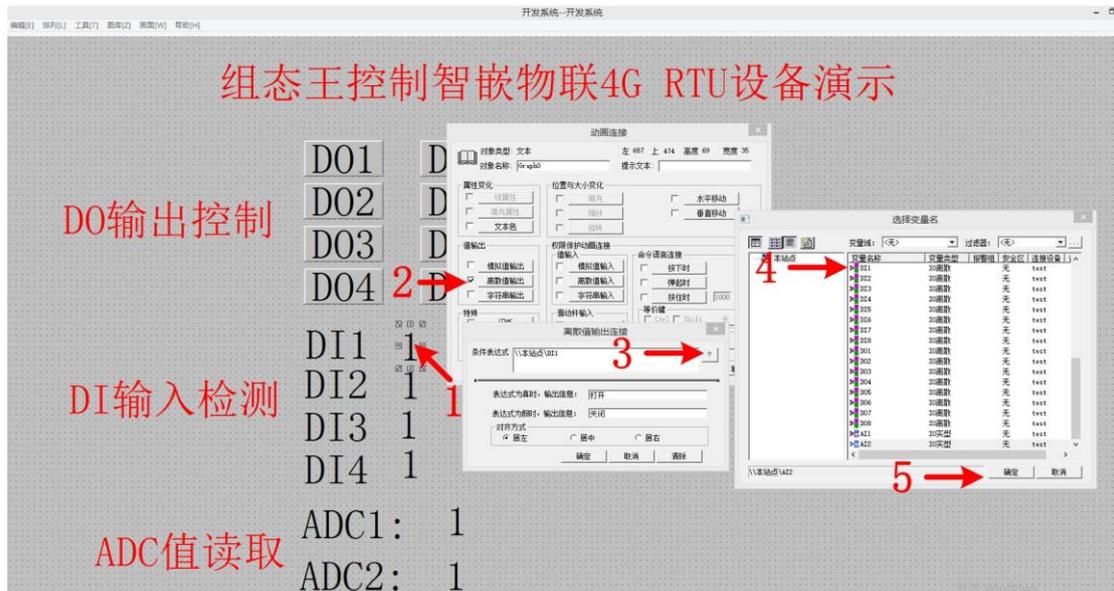


图 8.35 组态王变量关联 DI 输入

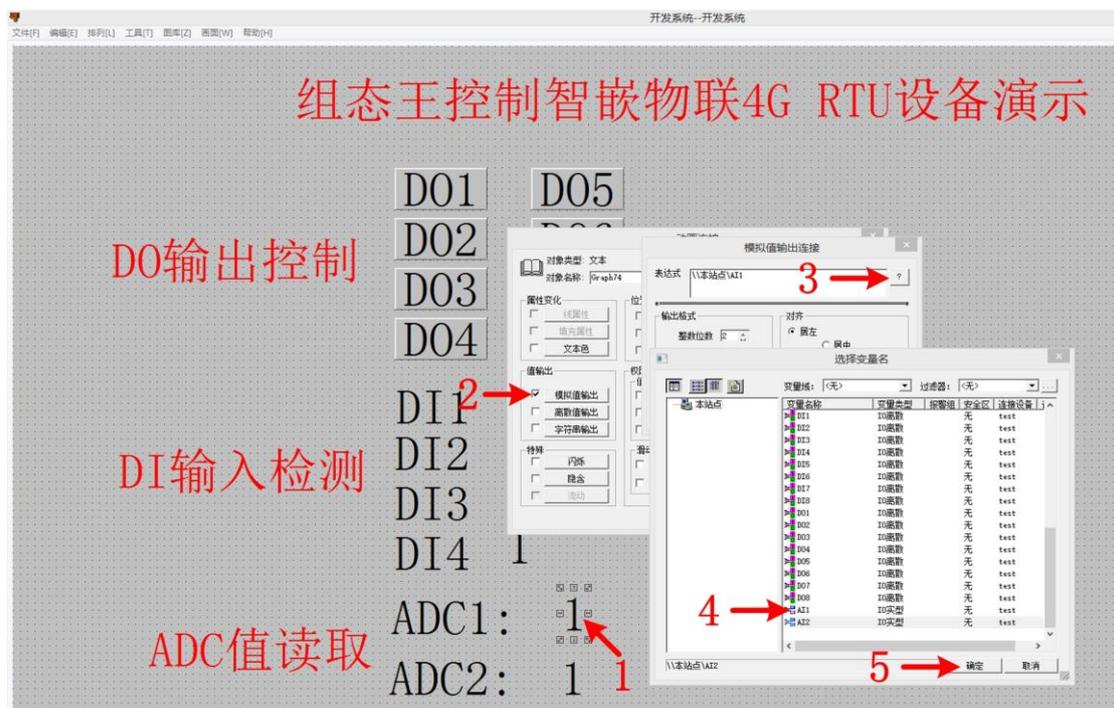


图 8.36 组态王变量关联 ADC

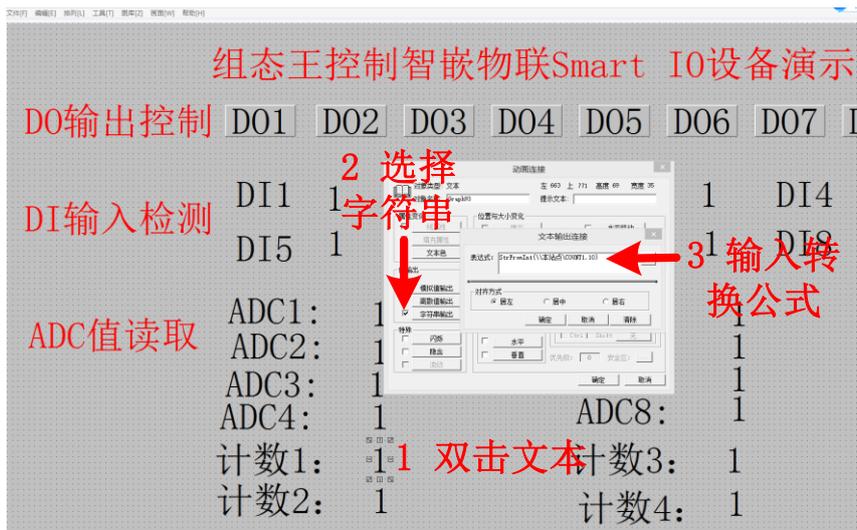


图 8.37 组态王变量关联脉冲计数

图中的转换公式为：StrFromInt(\\本站点\COUNT1,10)

7. 设置好后，保存画面，并切换到 View，如图 8.38 所示。

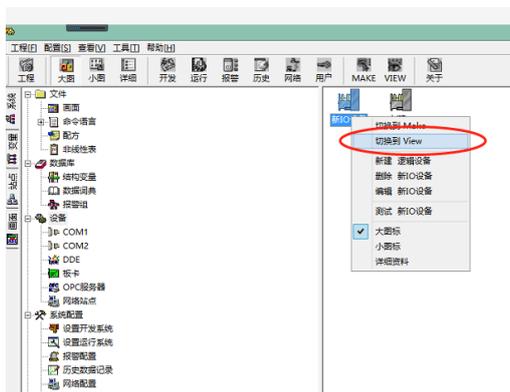


图 8.38 组态王切换到 View 界面

8. 远程控制结果

按照以上步骤设置完成后，组态王软件即可获取 Smart IO 设备设备的 AI 值、DI 状态，也可控制 DO 继电器输出。如图 8.39 所示。

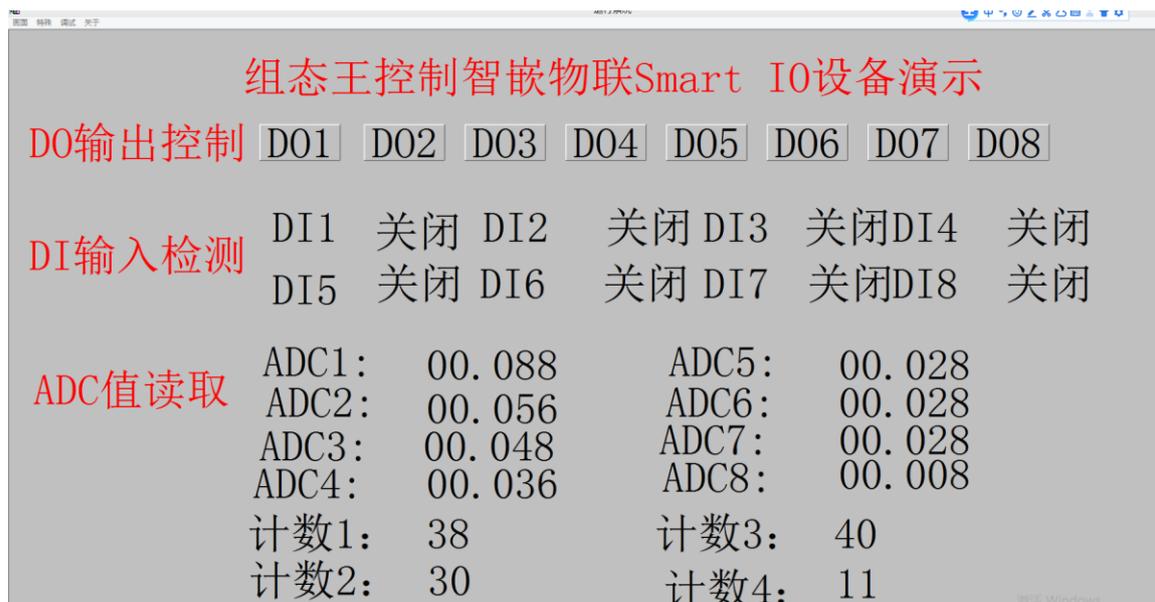


图 8.39 组态王软件读取 Smart IO 设备数据

8.4 设备 MQTT 协议测试案例

MQTT 协议是一种消息队列传输协议，采用订阅、发布机制，订阅者只接收自己已经订阅的数据，非订阅数据则不接收，既保证了必要的数据的交换，又避免了无效数据造成的储存与处理。因此在工业物联网中得到广泛的应用。

MQTTX 是一款功能强大的 MQTT5.0 开源桌面测试软件，能够帮助用户轻松管理多个 MQTT 客户端，为用户的 MQTT/TCP、MQTT/TLS 等测试提供了帮助。

本案例利用 MQTTX 软件来实现 4G RTU 设备的控制及 RS485 数据的透明传输，网络拓扑如图 8.40 所示。

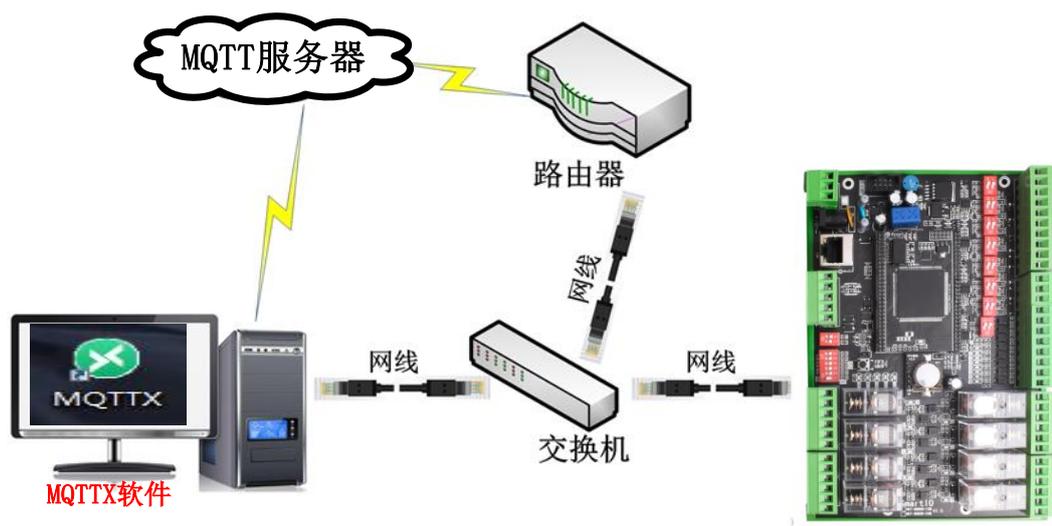


图 8.40 MQTTX 与 4G RTU 设备通信拓扑

可通过该案例实现如下功能：

- 设备的 RS485 总线与 MQTTX 软件之间的数据透传；
- ASCII 码控制设备；
- JSON 格式指令控制设备；
- ModBus 指令格式控制设备；
- 自定义指令格式控制设备；

1. 硬件连接

通过交换机连接电脑、设备、路由器；用适配器为设备供电。供电后请先观察设备指示灯是否正常，如图 8.40 所示。

2. 下载并安装 MQTTX 软件

MQTTX 软件下载地址：<https://mqttx.app/>

安装后，打开软件，并按图 8.41、图 8.42 所示步骤设置 MQTT 服务器、端口号、发布主题、订阅主题等参数。

📌 智嵌物联自建 MQTT 服务器，供用户调试产品使用。MQTT 服务器 IP:39.108.220.80；端口号：8883

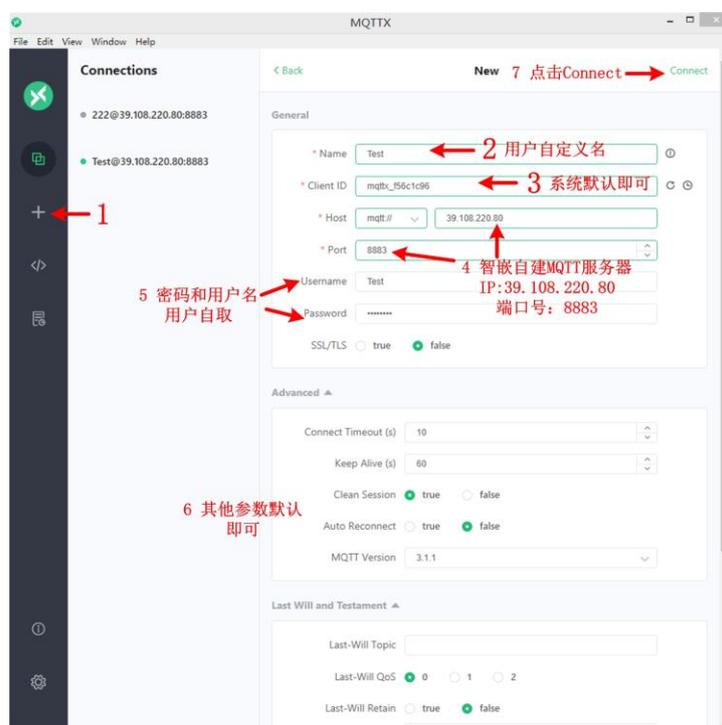


图 8.41 MQTTX 软件连接 MQTT 服务器

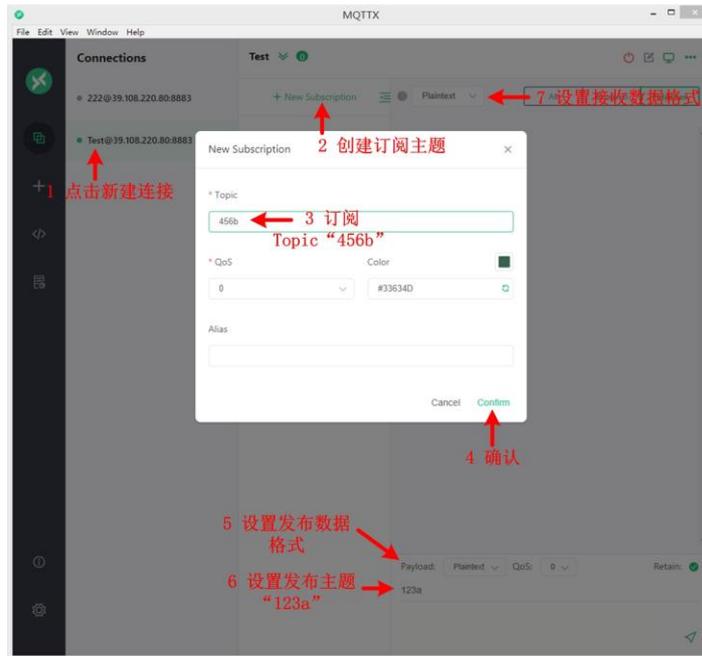


图 8.42 设置订阅主题、发布主题

3. 配置 Smart IO 设备参数

此步骤的目的是将 Smart IO 设备连接到 MQTT 服务器。

在“Smart IO-net 配置软件”中点击【MQTT】配置按钮，如图 8.43 所示，会弹出 MQTT 配置框，如图 8.44 所示。

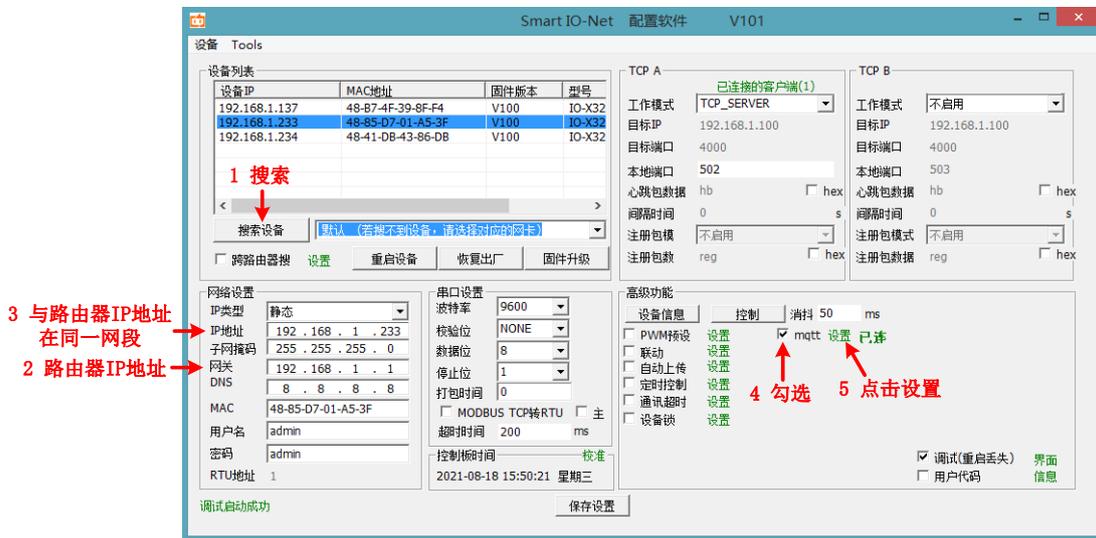


图 8.43 智嵌云参数配置 1



图 8.44 智嵌云参数配置 2

按照图 8.44 所示步骤设置，点击界面中的【?】图标，界面会弹出相应的帮助信息。

4. 透传测试

配置完成之后，保存参数并重启设备。打开串口调试助手，连接 USB 转 RS485 工具的串口号，然后互相发送数据，如图 8.45 所示。

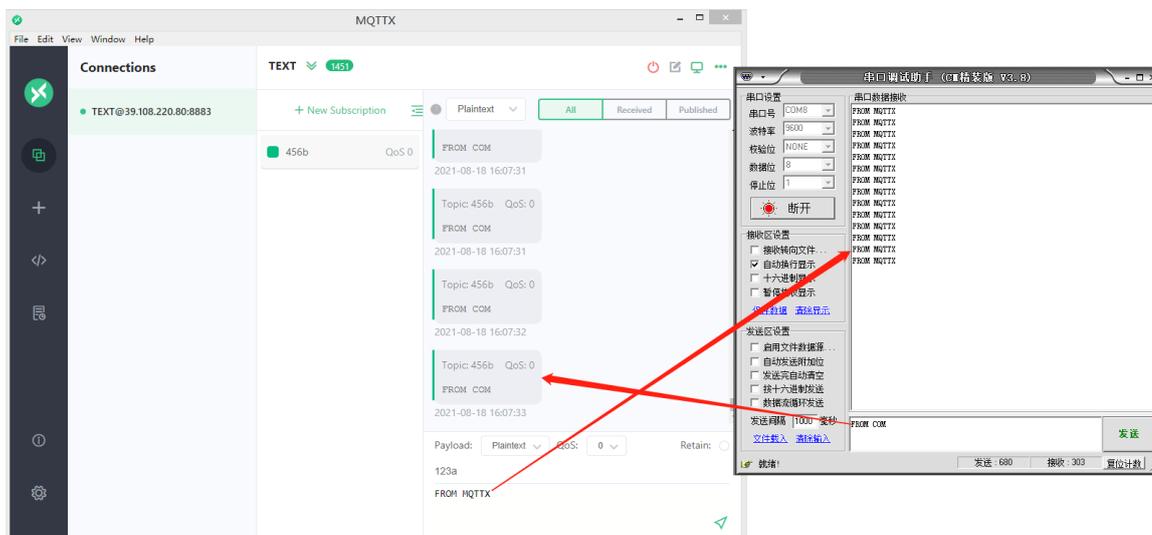


图 8.45 数据透传成功

5. ASCII 格式控制 Smart IO 设备接口

ASCII 格式指令介绍详见第 5.1 节。

可复制第 5.1 节举例中的指令到 MQTTX 软件的发送框，点击发送，即可控制/获取设备的接口状态。如图 8.46 所示。

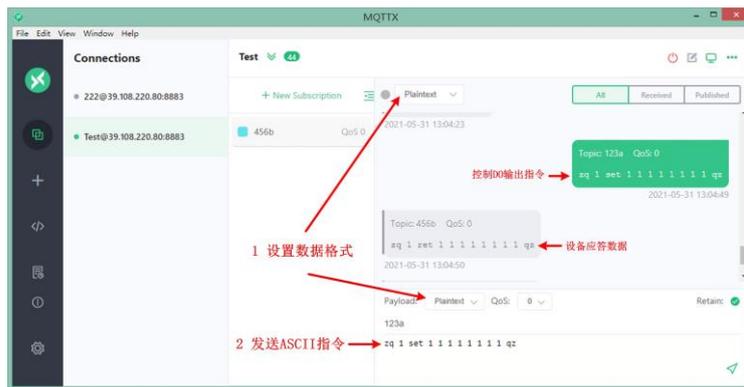


图 8.46 MQTTX 发送 ASCII 格式指令

6. JSON 格式控制 Smart IO 设备

JSON 格式指令介绍详见第 5.2 节。

可复制以上指令到 MQTTX 软件的发送框，点击发送，即可控制/获取设备的接口状态。如图 8.47 所示。

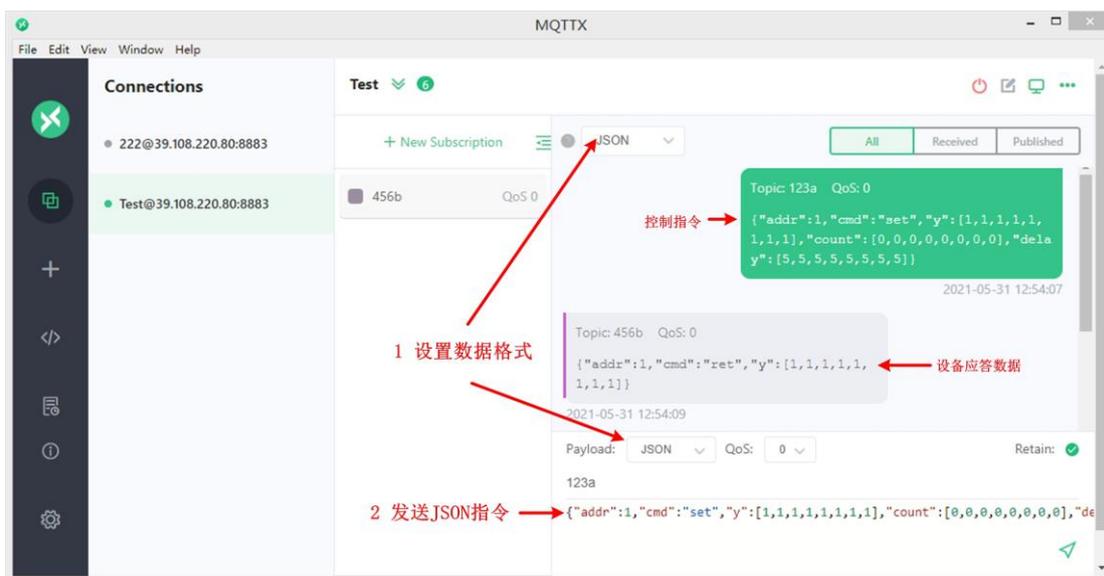


图 8.47 MQTTX 发送 JSON 格式指令

7. ModBus RTU 指令控制 Smart IO 设备

ModBus RTU 格式指令介绍详见第 5.3 节。

将以上举例中的 ModBus 发送指令复制到 MQTTX 的发送框（需将数据间的空格删掉），发送数据即可得到设备的应答，如图 8.48 所示。

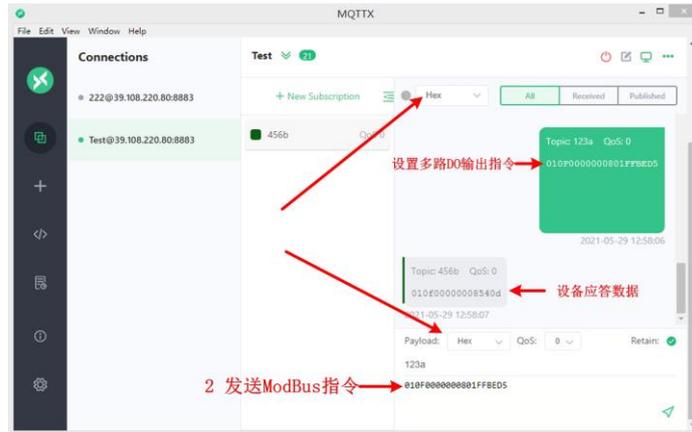


图 8.48 MQTTX 发送 ModBus 格式指令

8. 自定义指令控制 Smart IO 设备

自定义格式指令介绍详见第 5.4 节。

将举例中的发送指令复制到 MQTTX 的发送框（需将数据间的空格删掉），发送数据即可得到设备的应答，如图 8.49 所示。

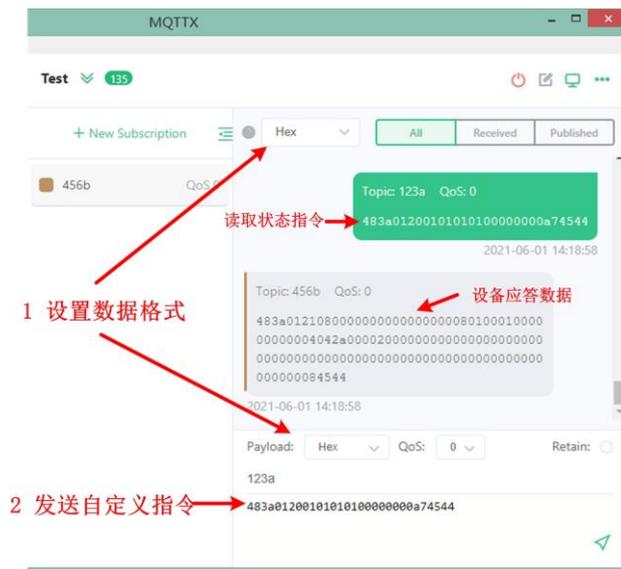


图 8.49 MQTTX 发送自定义格式指令

销售网络

智嵌物联，让连接更稳定！

企业愿景：成为国内物联网设备首选品牌！

企业使命：为客户利益而努力创新，为推动工业物联网发展而不懈奋斗！

产品理念：稳定！稳定！还是稳定！

服务理念：客户在哪里，我们就在哪里！



深圳总部

地址：广东省深圳市宝安区新桥街道新桥社区
新和大道 6-18 号 1203

网址：www.zhiqwl.com

电话：0755-23203231

北京办事处

地址：北京市房山城区德润街6号院8号楼3层

电话：18210365439

更多销售网络正在紧张筹备中……



天猫店铺



淘宝店铺



京东店铺



微信公众号



公司官网