

Modbus / Modbus TCP 转 MQTT

MQT-804

产品手册

V3.1

Rev A



上海泗博自动化技术有限公司

SiboTech Automation Co., Ltd.

技术支持热线: 021-3126 5138

E-mail: support@sibotech.net



目录

一、产品概述	3
1.1 产品分类	3
1.2 产品功能	3
1.3 产品特点	3
1.4 技术指标	4
二、快速应用指南	6
三、硬件说明	7
3.1 产品外观	7
3.2 指示灯	8
3.3 按键	8
3.4 接口	9
3.4.1 电源接口	9
3.4.2 RS485 接口	9
四、配置软件使用说明	11
4.1 配置前注意事项	11
4.2 用户界面	12
4.3 设备视图操作	14
4.3.1 设备视图界面	14
4.3.2 设备视图操作方式	15
4.3.3 设备视图操作种类	16
4.4 配置视图操作	18
4.4.1 无线互联网	18
4.4.2 高级功能	19
4.4.3 MQTT Topic	21
4.4.4 Modbus TCP 主站（仅 MQT-804A 支持）	22
4.4.5 Modbus TCP 从站（仅 MQT-804B 支持）	23
4.4.6 串口	25
4.4.7 GPS	31
4.5 冲突检测	32
4.5.1 命令列表操作	33
4.5.2 内存映射区操作	34
4.6 上载下载配置	35
4.6.1 上载配置	36
4.6.2 下载配置	37
4.7 加载和保存配置	38
4.7.1 保存配置工程	38
4.7.2 加载配置工程	39
4.8 EXCEL 文档输出	39



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

4.9 自动映射	40
4.10 属性名称排序	41
4.11 属性名称冲突检测	41
4.12 远程更新配置文件	42
4.13 自定义 JSON 格式	42
五、数据传输	46
5.1 连接流程	46
5.2 数据交换	46
六、典型应用	49
七、安装	50
7.1 机械尺寸	50
7.2 安装方法	50
八、运行维护及注意事项	51
九、版权信息	53
十、相关产品	54
十一、修订记录	55
附录 A: Modbus 协议	56
附录 B: 数据通讯格式示例	60
发布:	60
订阅:	63
附录 C: 切换产品工作模式操作步骤	66

一、产品概述

1.1 产品分类

MQT-804 网关共有两种产品，分别支持不同的协议，只需下载不同的 Bin 文件即可完成切换，对应的 Bin 文件可在配置软件 MQT-123 的安装目录中获取（默认安装路径例如：C:\Program Files (x86)\Sibotech\MQT-123 V4.0.x\mqt-804-bin），或者联系技术支持获取。下图为 MQT-804 两种工作模式对应的具体功能，关于工作模式切换操作步骤，请见[附录 C：切换产品工作模式操作步骤](#)。

简称	产品模式	支持协议
MQT-804A	Modbus TCP Client / Modbus Master	Modbus TCP 主站，Modbus 主站
MQT-804B	Modbus TCP Server/ Modbus Master-Slave	Modbus TCP 从站，Modbus 主/从站

1.2 产品功能

MQT-804 是基于 RS485、网口通讯，支持 Modbus 总线协议、Modbus TCP 协议、GPS 定位和 eMTC、NB-IoT 无线数据通讯网络的一款远程监控终端，主要针对需要无人值守和远程监控的工业监控现场，可用于 Modbus 主/从站设备、Modbus TCP 主/从站设备通过无线网络传输数据，支持 MQTT 协议连接至各大云平台（阿里云 IoT、百度天工、EMQ 等），可广泛应用于水质监控、环境检测、智能楼宇、智慧农业、能源、电力等有设备远程监控需求的领域。

1.3 产品特点

- ◆ RS485 串口支持 Modbus 主/从站
 - ◇ 串口作为 Modbus 主站，对接各种仪器仪表设备，进行数据采集和控制，通过无线网络与云平台交换数据
 - ◇ 串口作为 Modbus 从站，对接 PLC、DCS 等主站设备，通过无线网络传输并在云平台展示相关数据
- ◆ 网口支持 Modbus TCP 主/从站
 - ◇ 网口作为 Modbus TCP 主站，对接支持 Modbus TCP 通讯的从站，如：智能仪表
 - ◇ 网口作为 Modbus TCP 从站，对接支持 Modbus TCP 通讯的主站，如：SCADA 软件等
- ◆ 支持 GPS 定位



- ◆ 支持永远在线（TCP 主动连接远端云服务器），具有连接探测、自动重连功能
- ◆ 支持 URL 连接
- ◆ 支持 TLS/DTLS 加密，支持自定义 Client ID 功能
- ◆ 支持的 QoS 等级为 QoS 0 和 QoS 1
- ◆ 支持新数据推送机制或周期发送机制，数据发送间隔可设置，以节省客户流量费用
- ◆ 支持 Modbus 寄存器发送格式或属性名称发送格式，属性名称发送格式更加通俗易懂，以缩短设备投运时间
- ◆ 支持随时通过云平台监控设备端运行状态
- ◆ 自动监测无线信号，具有信号强度指示功能
- ◆ 工业级的可靠性，实时监控模块状态，故障自复位

1.4 技术指标

[1] eMTC / NB-IoT 网络

- ◆ 支持频段：
 - eMTC（3Gpp Release 13 CAT-M）：
B1/B2/B3/B4/B5/B8/B12/B13/B14/B18/B19/B20/B25/B26/B27/B28/B66/B85
 - NB-IoT（3Gpp Release 13 CAT-NB）：
B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66/B71/B85
- ◆ 网络带宽：
 - CAT-M: 上行:1119Kbps; 下行:589Kbps
 - CAT-NB: 上行:150Kdps; 下行:136Kdps

[2] 串行端口

支持 RS485 接口，半双工，奇偶校验位支持无校验、奇校验、偶校验、标记、空格，停止位支持 1 位和 2 位

波特率：600 - 115200bps

支持的协议类型：Modbus 协议，支持作 **Modbus RTU/ASCII** 主/从站

支持功能码：01、02、03、04、05、06、15、16

接口类型：3.81 三针端子



隔离设计：1kV 光电隔离

[3] 以太网口

网口类型：一个 10M/100M 自适应的网口，半双工、全双工

支持的协议类型：

Modbus TCP 协议，支持作 **Modbus TCP 主/从站**（Modbus TCP Client/Server）

支持功能码：01、02、03、04、05、06、15、16

接口类型：RJ45*2

其他：支持 DHCP

[4] 最大连接数：

Modbus RTU 连接数：

最大输入字节数 $\leq 512\text{Bytes}$

最大输出字节数 $\leq 512\text{Bytes}$

最大连接站点数：31（不带中继）

最大命令条数：100 条

Modbus TCP 连接数：

最大输入字节数 $\leq 512\text{Bytes}$

最大输出字节数 $\leq 512\text{Bytes}$

最大连接站点数：4 个，可以连接四个 Modbus TCP 从站

命令条数：100 条

注：RTU 和 TCP 字节数和命令数独立计数，互不影响

最大属性个数：Modbus RTU+Modbus TCP 共 1000 个属性

[5] 整机供电及防护安装

供电：DC 24V（11-30V）

工作环境温度：-20℃~60℃，相对湿度：5%~95%（无凝露）

机械尺寸：25mm（宽）×100mm（高）×90mm（深）

安装方式：35mm 导轨

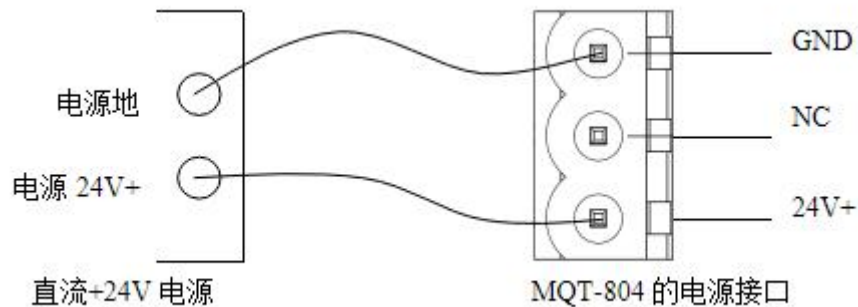
防护等级：IP20



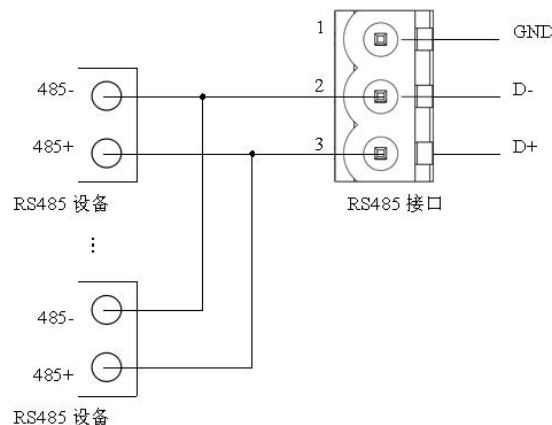
二、快速应用指南

1. 将 Nano SIM 卡正确插入 MQT-804 正面的卡托中，并将卡托插入卡座中。
2. 连接 NB-IoT/eMTC 网络天线（长直的天线）和 GPS 天线（方头的天线）。
3. 按如下方式连接电源，输入电压范围：10V~30V。

在完成所有连接之前，请勿打开设备电源。



4. 连接 RS-485 接口的串行设备。



5. 通过以太网线将 MQT-804 网关与 Modbus TCP 主/从站（服务端/客户端）连接。
6. 打开 MQT-804 网关电源的供电。

请从上海泗博自动化公司官网 www.sibotech.net 进入 MQT-804 产品页面获取 MQT-123 配置软件并安装，打开软件选择当前 MQT-804 模式并进行配置（网关默认处于 Modbus TCP S / Modbus M/S 模式，默认为 DHCP 模式，由路由器分配 IP 地址）。



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

三、硬件说明

3.1 产品外观



注：产品实拍图。



3.2 指示灯

状态		说明
STA	长灭	无线模块未启动
	绿灯慢闪（亮 200ms/灭 1800ms）	无线模块搜网状态
	绿灯慢闪（亮 1800ms/灭 200ms）	无线模块待机状态
	绿灯快闪（亮 125ms/灭 125ms）	无线模块传输数据
SG	绿灯长灭	SIM 卡无效
	绿灯长亮	正在建立连接或者无法建立连接
	绿灯慢闪（亮 1000ms/灭 1000ms）	信号强度一格
	绿灯闪（亮 500ms/灭 500ms）	信号强度二格
	绿灯快闪（亮 200ms/灭 200ms）	信号强度三格
TX/RX	绿灯闪烁	RS485 口无数据通讯
	绿灯长亮	RS485 口有数据收发
ENS	绿灯闪烁（慢）	无 Modbus TCP 主/从站连接
	绿灯闪烁（快）	IP 地址为 192.168.0.10
	红灯闪烁	配置软件定位 或有 Modbus TCP 从站连接断开
	绿灯长亮	有 Modbus TCP 主/从站连接

3.3 按键

在设备上电后，长按按钮开关 3s 后 IP 地址恢复到缺省值 192.168.0.10，同时 ENS 橙灯闪烁一次，可以在搜索不到设备时使用。再次轻触按钮开关，设备自动重启并正常运行。

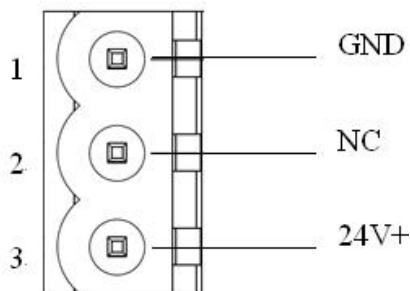
在设备上电后，连续快速双击按钮开关，此时设备可以被搜索到，但无法进行上下载配置，主要用于多台设备连接时，防止被他人误改配置。再次轻触按钮开关，设备允许被配置。



按钮	功能
长按 3 秒	配置模式，IP 地址恢复到缺省值 192.168.0.10。再次单击按钮网关重启并回复运行模式。
双击按钮	运行禁止配置模式，此时设备可以被搜索到，但无法进行上下下载配置，主要用于多台设备连接时，防止被他人误改配置。再次轻触按钮开关，设备允许被配置
按住按钮后再上电	进行网关模式的切换，用网线将产品接入路由器或直连电脑，打开浏览器，输入 192.168.0.10 网址，进入产品功能更新页面，输入用户名：user，密码：sibotech，登录后选择需要更新切换的模式文件，点击下载，完成更新后重启网关及完成模式切换。

3.4 接口

3.4.1 电源接口



引脚	功能
1	GND，电源地
2	NC，无连接
3	24V+，直流正 24V

3.4.2 RS485 接口

MQT-804 产品的 RS485 接口是标准的 RS485 接口，以下简述本产品 RS485 特性及安装要点：

3.4.2.1 RS485 传输技术基本特征

① 网络拓扑：线性总线，两端有有源的总线终端电阻；



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

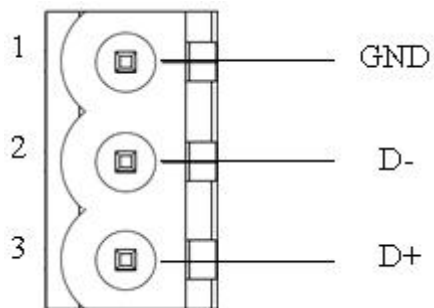
- ② 传输速率：600 bit/s~115.2Kbit/s;
- ③ 介质：屏蔽双绞电缆，也可取消屏蔽，取决于环境条件（EMC）;
- ④ 站点数：每分段 31 个站（不带中继），可多到 127 个站（带中继）;
- ⑤ 插头连接：3/5 针可插拔端子。

3.4.2.2 RS485 传输设备安装要点

- ① 全部设备均与 RS485 总线连接;
- ② 每个分段上最多可接 31 个站;
- ③ 总线的最远两端各有一个总线终端电阻，120Ω 1/2W 确保网络可靠运行。

串行接口采用开放式 3 针可插拔端子，用户可以根据面板上的指示进行接线。

三针端子：



引脚	功能
1	GND
2	D-, RS485-
3	D+, RS485+



四、配置软件使用说明

4.1 配置前注意事项

MQT-123 是一款基于 Windows 平台，用来配置 MQT-804 相关参数及命令的配置软件。

本说明书描述了网关配置软件 MQT-123 的具体使用方法和注意事项，方便工程人员的操作运用。在使用本软件前，请仔细阅读本说明书。

本软件通过网口通讯，上载或下载配置文件。

双击软件图标即可打开配置软件 MQT-123，选择 MQT-804 产品，确认后再选择对应的网关模式即可进入配置主界面



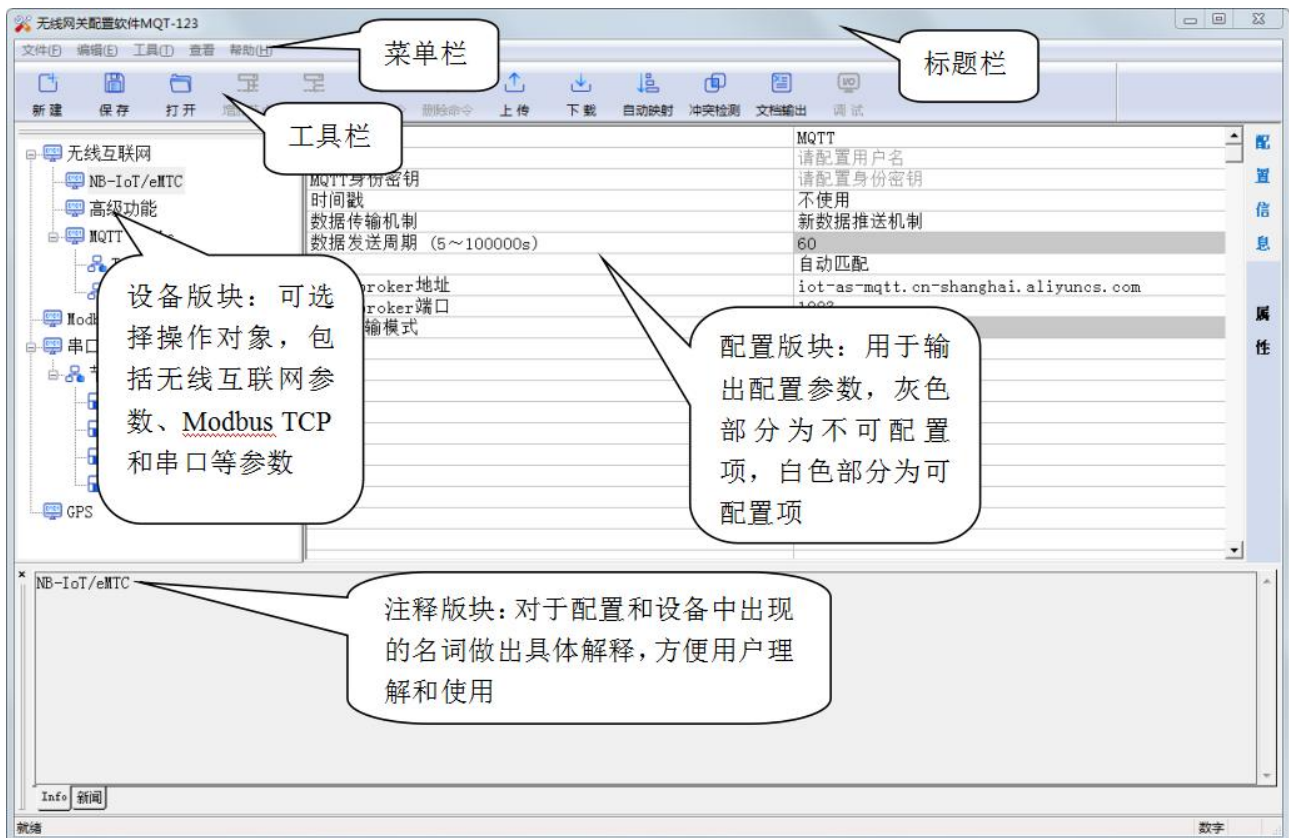


4.2 用户界面

界面包括：标题栏、菜单栏、工具栏、状态栏、设备板块、配置板块、和注释板块。

备注：在该软件中，所有的灰色部分为不可更改项。

804A 用户界面



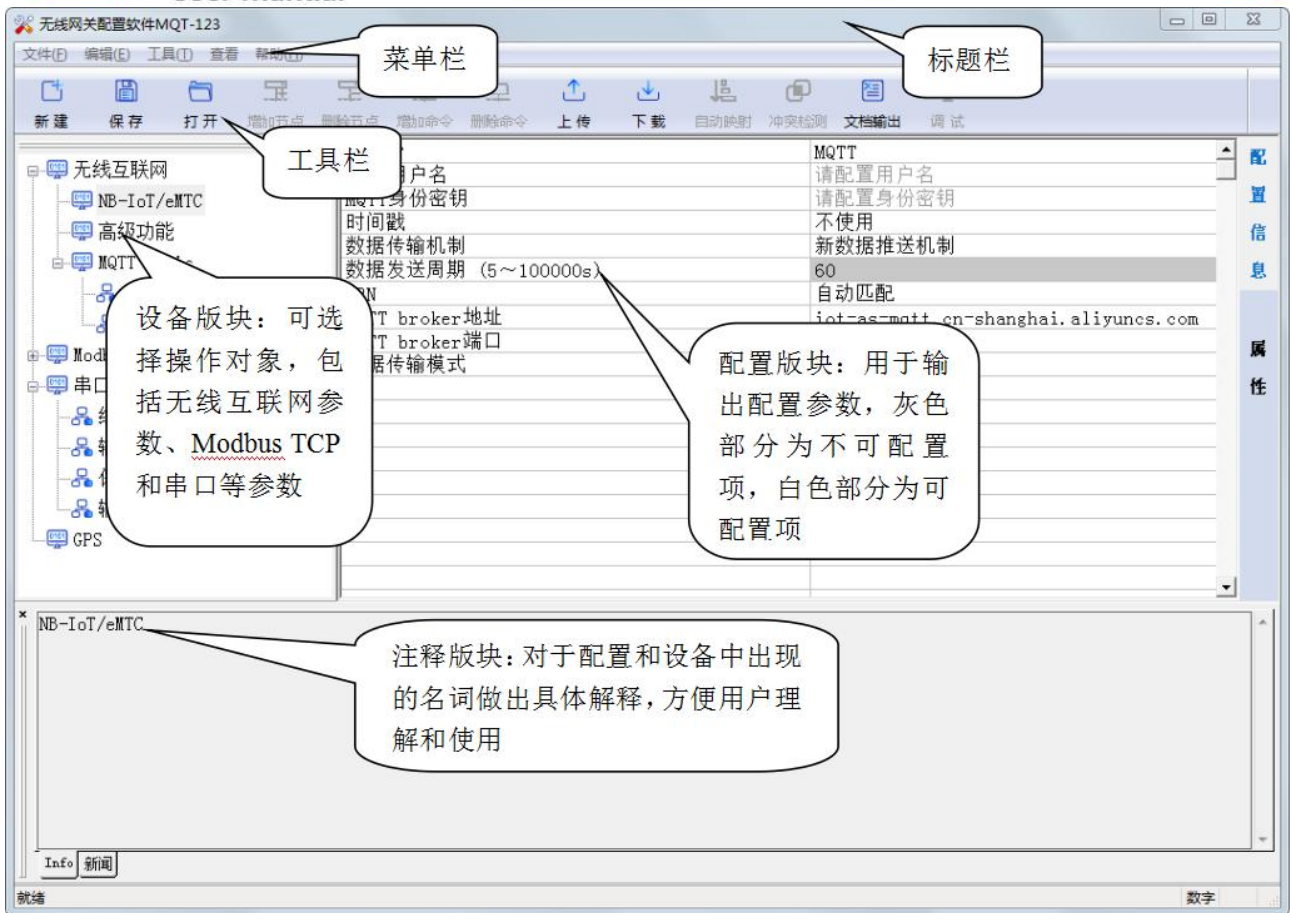
804B 用户界面



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual



工具栏:

工具栏如下图所示:



从左至右的功能分别是：新建、保存、打开、增加节点、删除节点、增加命令、删除命令、上载配置信息、下载配置信息、自动映射、冲突检测、Excel 配置文档输出、通信调试。



: 新建一个配置工程



: 保存当前配置



打开：打开一个配置工程



增加节点：增加一个 Modbus /Modbus TCP 主站节点



删除节点：删除一个 Modbus /Modbus TCP 主站节点



增加命令：增加一条 Modbus /Modbus TCP 主站命令



删除命令：删除一条 Modbus /Modbus TCP 主站命令



上传：将配置信息从模块中读取上来，并且显示在软件中



下载：将配置信息从软件中下载到模块



自动映射：自动将每条命令的数据映射到输入输出缓冲区



冲突检测：用于检测“内存映射数据”是否有冲突

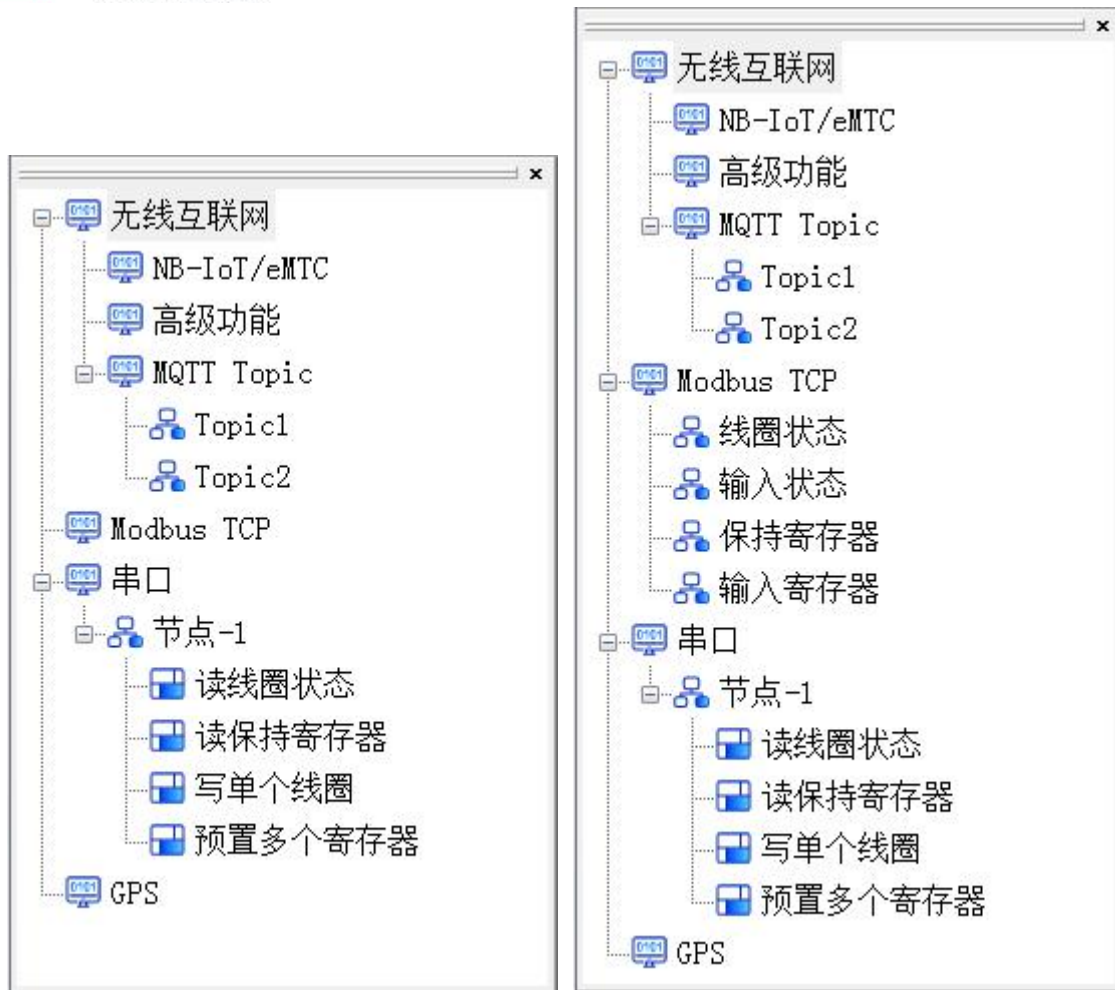


文档输出：将当前配置输出到本地硬盘，以.xls 文件格式保存

4.3 设备视图操作

4.3.1 设备视图界面

从左到右分别为 MQT-804A/B 对应的设备视图界面



MQT-804A

MQT-804B

4.3.2 设备视图操作方式

对于设备视图，Topic、GPS 部分均可通过鼠标左键和右键的组合方式进行操作，串口 Modbus 主站部分支持如下三种操作方式：编辑菜单、快捷工具栏和右键编辑菜单。



编辑菜单



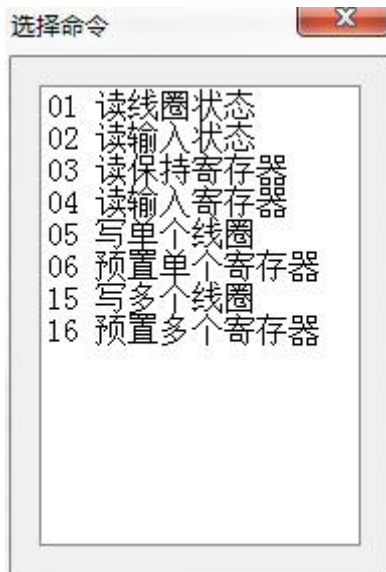
4.3.3 设备视图操作种类

1) 增加节点操作：在串口 Modbus 主站模式下，在串口选项上单击鼠标左键，选中该节点，然后执行增加节点操作。在串口下增加一个名字为“节点-n”的节点。注意：节点数最多为 100 个。

2) 删除节点操作：单击鼠标左键，选中待删除节点，然后执行删除节点操作。该节点及其下所有命令全部删除。

3) 增加命令操作：在节点上单击鼠标左键，然后执行增加命令操作，为该节点添加命令。注意：命令数最多为 100 个。

串口协议类型为 Modbus 主站时，弹出如下选择命令对话框，供用户选择，如下图所示：



目前支持的命令号：01，02，03，04，05，06，15，16 号命令

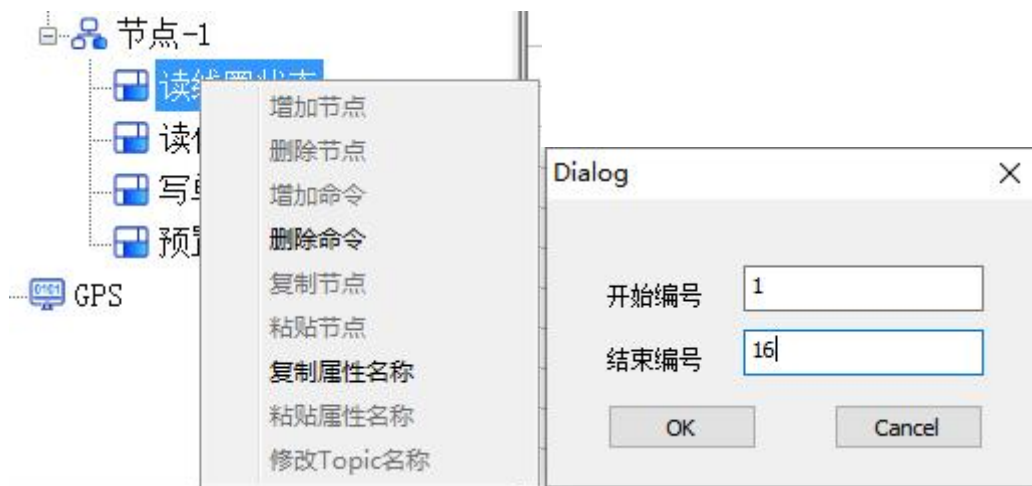
选择命令：双击命令条目即可将选中的命令添加到当前选中节点下

4) 删除命令操作：单击鼠标左键，选中待删除命令，然后执行删除命令操作。该命令即被删除。

5) 复制节点操作：在节点上单击鼠标左键，然后执行复制节点操作，对该节点进行复制。

6) 粘贴节点操作：执行复制节点操作，对被复制的节点进行粘贴。

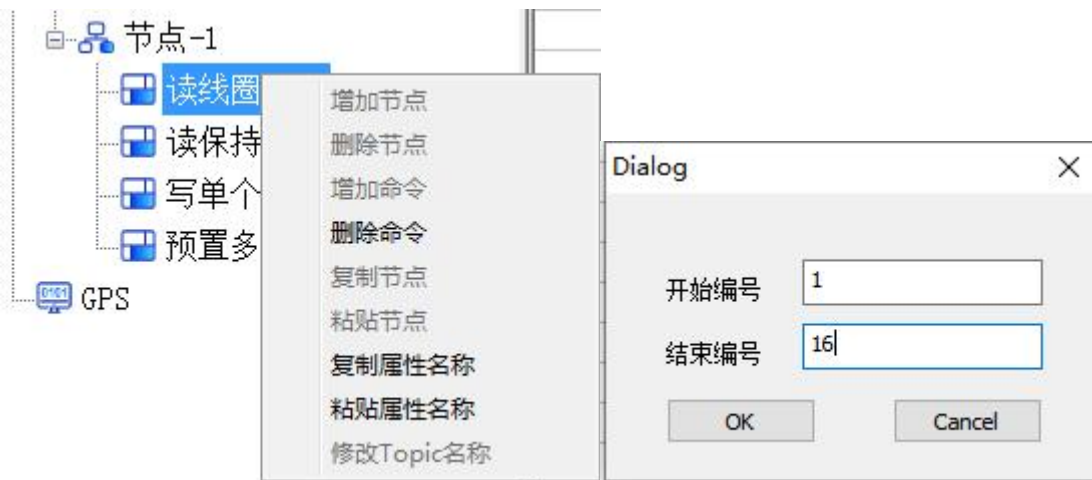
7) 复制属性名称操作：在任一命令选项上单击鼠标右键，再选择复制属性名称选型，单击鼠标左键，在 Dialog 提示框中填入需要复制属性名称的开始编号和结束编号即可。



8) 粘贴属性名称操作：在任一命令选项上单击鼠标右键，再选择粘贴属性名称选型，单击鼠标左键，在 Dialog 提示框中填入需要粘贴属性名称的开始编号和结束编号即可完成属性名称的粘贴（该操作还可在



多个 MQT-123 软件之间完成属性名称的复制和粘贴)。



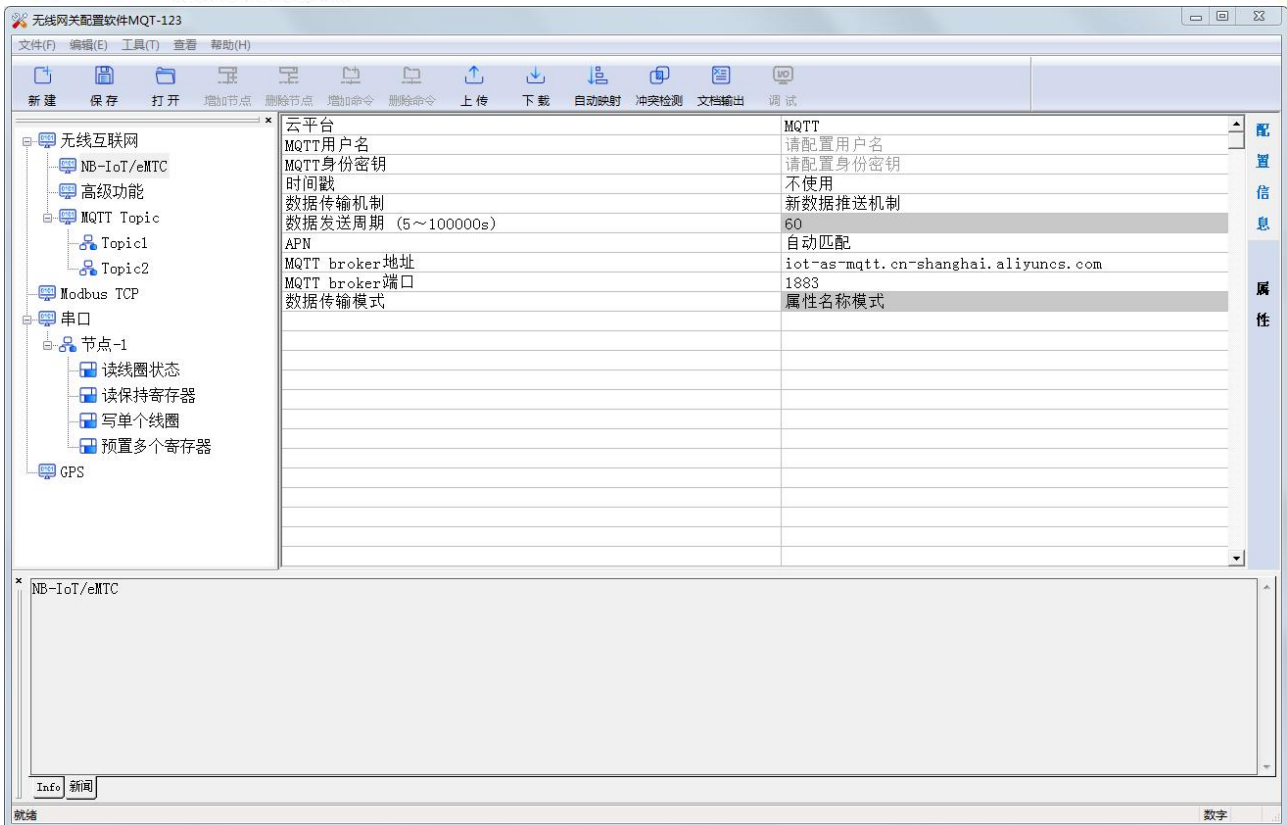
4.4 配置视图操作

以 MQT-804B 对接阿里云 IoT 为例介绍。

4.4.1 无线互联网

4.4.1.1 NB-IoT/eMTC

可配置参数：云平台、MQTT 用户名、MQTT 身份秘钥、时间戳、数据传输机制、数据发送周期、APN、MQTT 客户端地址、MQTT 客户端端口、数据传输格式。



云平台：MQTT 平台

MQTT 用户名、MQTT 身份密钥：连接通用 MQTT 平台的用户名和身份密钥，可从所要连接的云平台获取

时间戳：用于开启和关闭发送数据中是否包含时间信息的选型。

数据传输机制：提供新数据推送机制和周期发送机制两种选项，新数据推送机制是指有数据变化时，将变化的数据发送至云；周期发送机制是指按照配置的周期发生数据至云端，无论数据是否发生变化

数据发送周期：周期发送数据机制的数据发送周期时间，范围 1~100000s，默认为 60s。当数据传输机制为“周期发送机制”时，此项有效。

APN：APN(Access Point Name) 用来标识所属无线网络的业务种类，默认为“自动匹配”，设备可自行搜索 APN 并直接入网；若为专网或其他特殊应用，可配置成“自定义”输入 APN 信息。

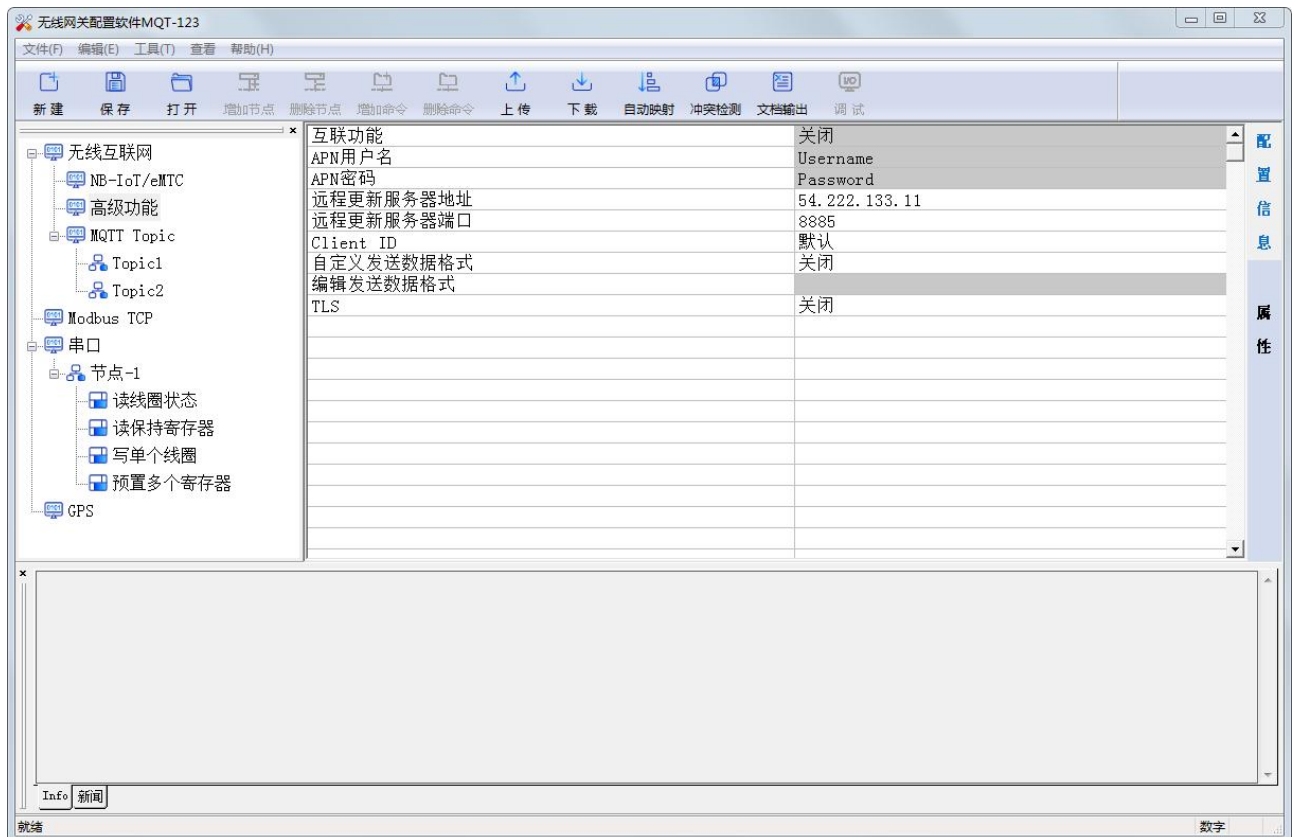
MQTT broker 地址：所连接云平台的 IP 或 URL 地址

MQTT broker 端口：所连接云平台的端口号

4.4.2 高级功能

可配置参数：互联功能、APN 用户名、APN 密码、远程更新服务器地址、远程更新服务器端口、Client

ID、自定义发送数据格式、编辑发送数据格式和 TLS 加密传输。



互连功能：开启后可通过配置 2 台 MQT-804 设备 topic 与属性名称，实现 2 台设备的数据互传、共享（还可与泗博公司其他支持互连功能的物联网产品实现数据共享。另 MQT-804A 不支持互连功能）。

APN 用户名：（专网或其他应用）根据运营商提供的相关参数设置拨号过程中用于验证身份的用户名。仅在“APN”配置为“自定义”状态下才可修改。

APN 密码：（专网或其他应用）根据运营商提供的相关参数设置拨号过程中用于验证身份的密码。仅在“APN”配置为“自定义”状态下才可修改。

远程更新服务器 IP 默认设置为：54.222.133.11

远程更新服务器端口默认设置为：8885

（远程更新功能用于对设备进行远程、无线升级产品功能、远程更新配置，及远程重启设备所使用，服务器的 IP 和端口号请勿修改，以免影响远程更新功能的正常运行。）

物模型通信：仅在连接阿里云平台时支持，开启后可实现与阿里云的物模型模式通信；在开启此选项后，连接阿里云的 topic 要填写与物模型相关的 topic 全称

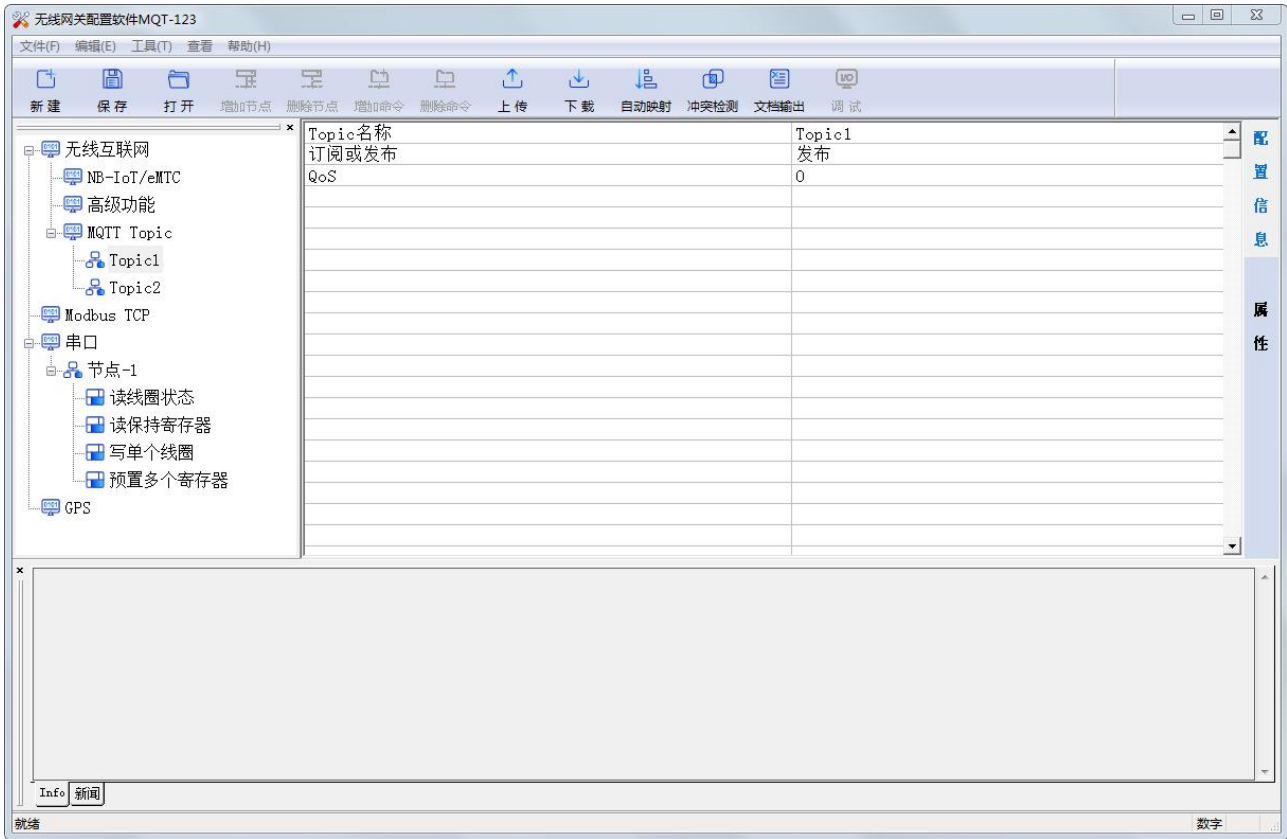


Client ID: 可配置连接云平台的 Client ID，默认采用设备序号；在连接各大云平台时，每个客户端连接时的 Client ID 都必须唯一，故建议在无特殊需求时，此选项配置为默认

TLS 加密传输: 开启后可通过 MQTTS 加密协议连接云平台，实现数据的安全传输，默认支持 TLS 1.2。

4.4.3 MQTT Topic

添加 topic 节点后，可配置参数为：名称、QoS、订阅或发布。



Topic 名称:可由英文字符、数字、下划线、反斜杠、横线组成，且长度应不超过 128 字节，最多支持 25 个 Topic

发布: 网关数据发送到 MQTT 云平台。

订阅: MQTT 云平台数据发送到网关。

QOS:

0: 对于网关而言，有且仅发一次数据包，是否收到完全不管，适合那些不是很重要的数据

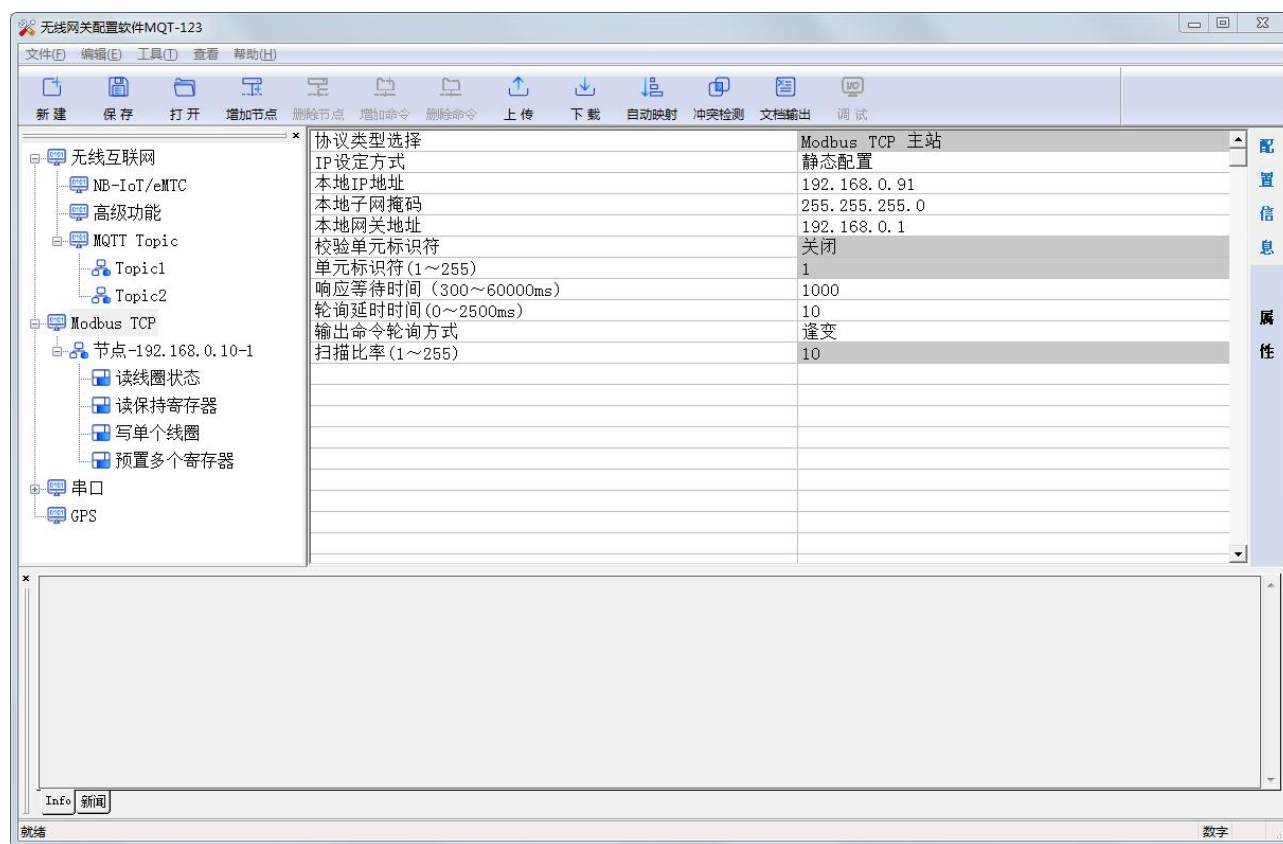
1: 对于网关而言，会有一次 ack 来确保数据发送至客户端或服务器，但不能保正只发送一次，

也就是说如果没有收到 ack，就会重发数据，最多发送 3 次

4.4.4 Modbus TCP 主站（仅 MQT-804A 支持）

可配置参数：IP 设定方式、本地 IP 地址、本地子网掩码、本地网关地址、响应等待时间、轮询延长时间、输出命令轮询模式。

其他参数：协议类型选择、校验单元标识符、单元标识符、扫描比率。



协议类型选择：显示您当前选择的协议类型。

IP 设定方式：支持静态配置/DHCP（默认）。

（注意：产品出厂或从 A 模式切换到 B 模式也会恢复到默认配置，都是处于 DHCP 状态，ENS 指示灯红灯闪烁，需要支持 DHCP 功能的路由器分配 IP；若没有成功分配 IP，30s 后 IP 地址恢复默认值：192.168.0.10；也可以通过长按按钮 3s，使得 IP 地址恢复默认值，即可搜索设备并下载配置。）

本地 IP 地址：设置网关的 IP 地址，只有当此地址与电脑处于同一网段时，设备才能被搜索到。

本地子网掩码：设置网关的子网掩码。



本地网关地址：设置网关的网关地址。

校验单元标识符：暂不支持配置。

单元标识符：暂不支持配置。

响应等待时间：当 Modbus TCP 主站发送命令后，等待从站响应的的时间，范围：300 ~ 60000ms

轮询延时时间：每条 Modbus TCP 命令发完并收到正确响应或响应超时之后，下一次发送此条 Modbus TCP 命令之前，延迟的时间，每条 Modbus TCP 命令的轮询延时时间独立，互不影响，范围：10 ~ 2500ms。实际轮询延时时间为输入值除以 10 之后取整，再乘以 10，单位 ms。假如输入值为 19，则实际轮询延时时间为 10ms；假如输入值为 25，则实际轮询延迟时间为 20ms。

输出命令轮询模式：Modbus TCP 主站写命令（输出命令），有三种输出模式：连续输出，禁止输出，逢变输出：

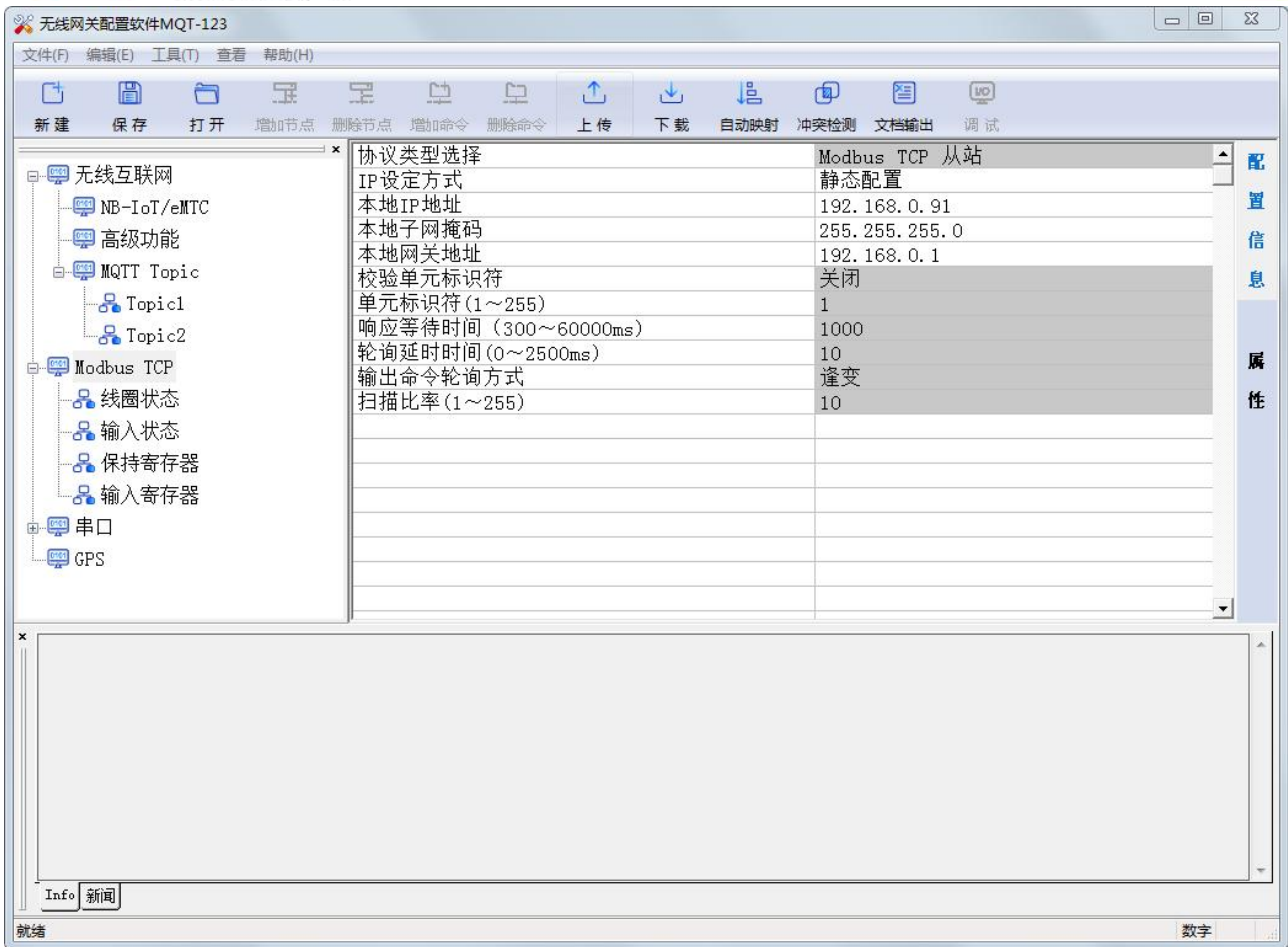
- 1) 连续输出：与 Modbus TCP 主站读命令输出方式相同
- 2) 禁止输出：禁止输出 Modbus TCP 主站写命令
- 3) 逢变输出：输出数据有变化时，输出 Modbus TCP 写命令，并在接收到正确响应后停止输出

扫描比率：暂不支持配置

4.4.5 Modbus TCP 从站（仅 MQT-804B 支持）

可配置参数：IP 设定方式、本地 IP 地址、本地子网掩码、本地网关地址、线圈状态起始地址、线圈状态块大小、输入状态位起始地址、输入状态位块大小、保持寄存器起始地址、保持寄存器块大小、输入寄存器起始地址、输入寄存器块大小。

其他参数：协议类型选择、校验单元标识符、单元标识符、响应等待时间、轮询等待时间、轮询等待时间、轮询延时时间、输出命令轮询方式、扫描比率。



IP 设定方式：支持静态配置/DHCP（默认）。

（注意：产品出厂或从 A 模式切换到 B 模式也会恢复到默认配置，都是处于 DHCP 状态，ENS 指示灯红灯闪烁，需要支持 DHCP 功能的路由器分配 IP；若没有成功分配 IP，30s 后 IP 地址恢复默认值：192.168.0.10；也可以通过长按按钮 3s，使得 IP 地址恢复默认值，即可搜索设备并下载配置。）

本地 IP 地址：设置网关的 IP 地址，只有当此地址与电脑处于同一网段时，设备才能被搜索到。

本地子网掩码：设置网关的子网掩码。

本地网关地址：设置网关的网关地址。

协议类型选择：显示您当前选择的协议类型。

校验单元标识符：暂不支持配置。

单元标识符：即从设备地址，默认为 1，暂不可支持配置。

线圈状态起始地址：0XXXX 区的起始地址，范围为 0 ~ 511。

线圈状态块大小：0XXXX 区的块大小，范围为 0 ~ 512。线圈状态起始地址加上线圈状态块大小的



范围为 0 ~ 512。

输入状态位起始地址：1XXXX 区的起始地址，范围为 0 ~ 511。

输入状态位块大小：1XXXX 区的块大小，范围为 0 ~ 512。输入状态起始地址加上输入状态块大小的范围为 0 ~ 512。

保持寄存器起始地址：4XXXX 区的起始地址，范围为 0 ~ 255。

保持寄存器块大小：4XXXX 区的块大小，范围为 0 ~ 256。保持寄存器起始地址加上保持寄存器块大小的范围为 0 ~ 256。

输入寄存器起始地址：3XXXX 区的起始地址，范围为 0 ~ 255。

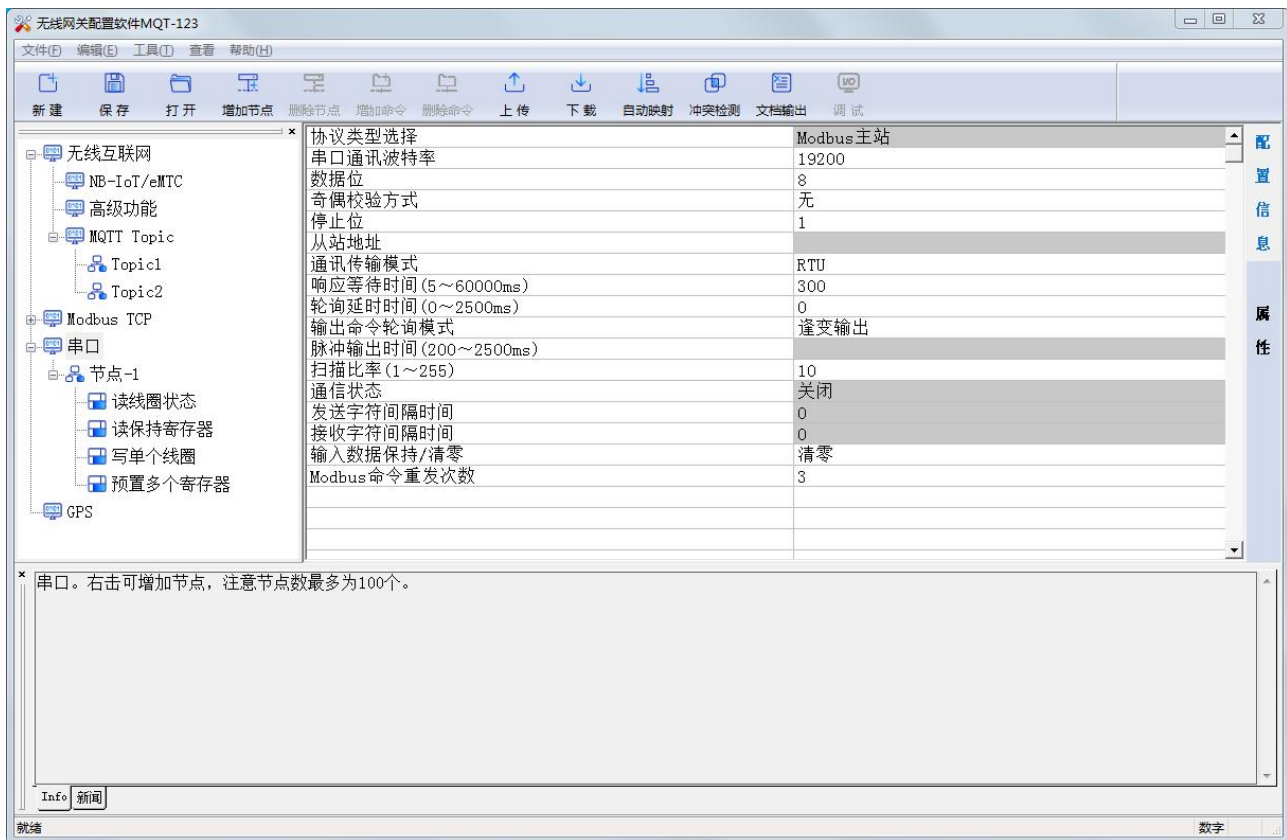
输入寄存器块大小：3XXXX 区的块大小，范围为 0 ~ 256。输入寄存器起始地址加上输入寄存器块大小的范围为 0 ~ 256。

4.4.6 串口

协议类型：Modbus 主站

可配置参数为：串口通信波特率、数据位、奇偶校验方式、停止位、通讯传输模式、响应等待时间、轮询延时时间、输出命令轮询模式、扫描比率、输入数据保持/清零、Modbus 命令重发次数。

配置视图界面显示如下：



串口通信波特率：Modbus 端口的通讯波特率。

数据位：每个字节的数据位个数。

奇偶校验方式：奇、偶、无、标记和空格五种校验方式可选。

停止位：停止位个数，1、2 可选。

通讯传输模式：数据符合 Modbus 协议时，数据的传输格式，支持 RTU、ASCII 模式可选。

响应等待时间：当 Modbus 主站发送命令后，等待从站响应的的时间，范围：5 ~ 60000ms

轮询延时时间：一条 Modbus 命令发完并收到正确响应或响应超时之后，发送下一条 Modbus 命令之前，延迟的时间，范围：0 ~ 2500ms。实际轮询延时时间为输入值除以 10 之后取整，再乘以 10，单位 ms。假如输入值为 9，则实际轮询延时时间为 0ms；假如输入值为 15，则实际轮询延迟时间为 10ms。

输出命令轮询模式：Modbus 主站写命令（输出命令），有三种输出模式：连续输出，禁止输出，逢变输出：

- 1) 连续输出：与 Modbus 主站读命令输出方式相同，根据扫描比率进行扫描输出
- 2) 禁止输出：禁止输出 Modbus 主站写命令

3) 逢变输出：输出数据有变化时，输出写命令，并在接收到正确响应后停止输出

扫描比率：慢速扫描周期与快速扫描周期的比值，如果该值设为 10，那么快速扫描命令发出 10 次，慢速扫描命令发出 1 次。

输入数据保持/清零：当 Modbus 命令响应错误次数达到 Modbus 命令重发次数，对应的 Modbus 输入数据是否被清零。

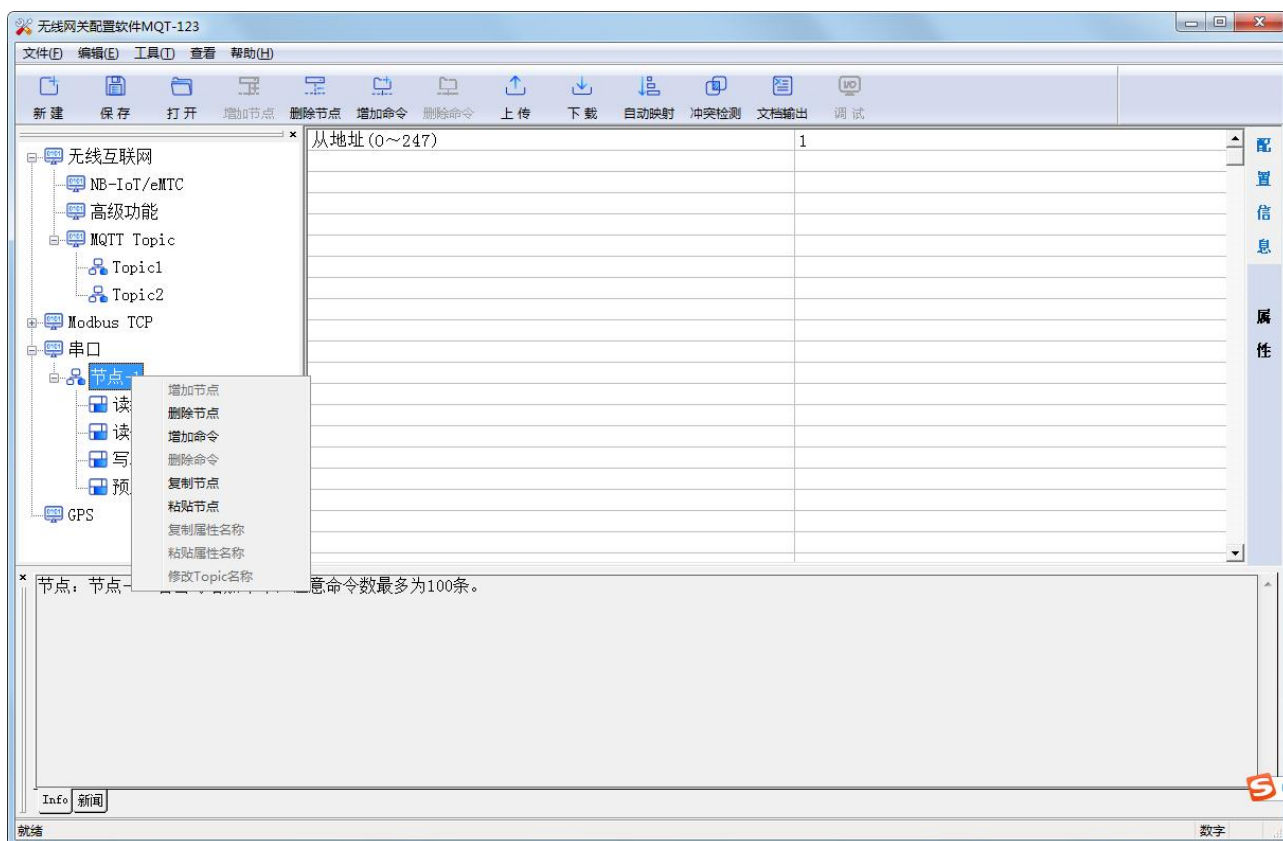
选择“清零”，Modbus 输入数据清零；

选择“保持”，Modbus 输入数据保持最后一次接收到的正确数据。

Modbus 命令重发次数：当 Modbus 命令响应错误时，该命令重发的次数，输入范围 2~254，默认值 3。

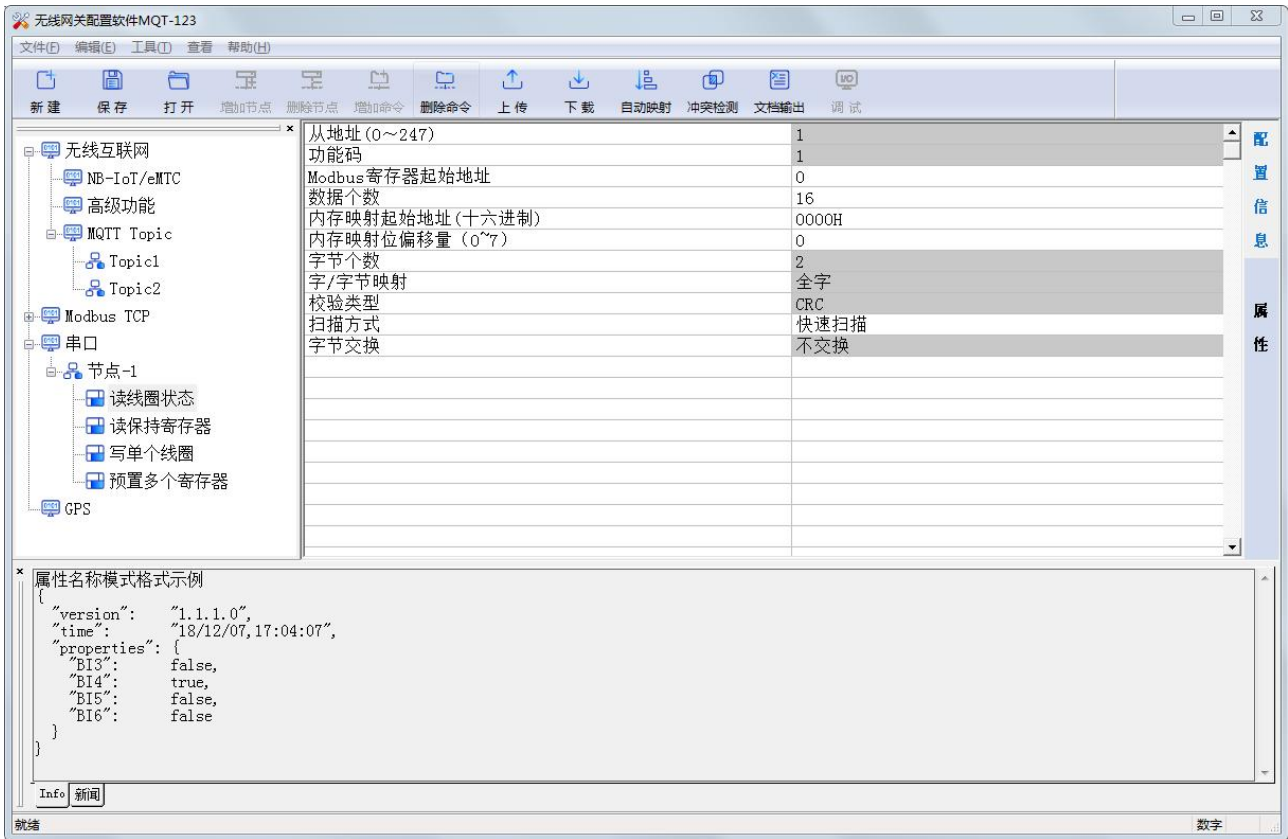
节点配置视图界面

在“Modbus 主站”模式下，在设备视图界面，单击新建的节点，配置视图界面显示如下：



命令配置视图界面

在设备视图界面，协议类型选择 Modbus 主站时，单击新建的命令，配置视图界面显示如下：



Modbus 寄存器起始地址：Modbus 从站设备中寄存器/开关量/线圈等起始地址，范围是 0 ~ 65535

注：配置软件 MQT-123 中该条目指的是协议地址，当用户输入 PLC 地址时，确定后会自动弹出如下图所示的对话框，点击确定后，用户输入的 PLC 地址会被转换成协议地址。



PLC 地址与对应的协议地址举例如下表所示：

命令	PLC 地址举例	对应的协议地址
线圈状态	00001~00010	00000~00009
输入状态	10001~10010	00000~00009



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

保持寄存器	40001~40010	00000~00009
输入寄存器	30001~30010	00000~00009

例如：当配置的 Modbus 主站命令为 03H（读保持寄存器），当用户在这一条目中（Modbus 寄存器起始地址）输入 40001，确定后会弹出上图所示的对话框，当点击确定后，输入的 PLC 地址 40001 会被转换成协议地址 0。

数据个数： Modbus 主站设备中寄存器/开关量/线圈的个数

内存映射起始地址（十六进制）： 在模块内存缓冲区中数据的起始地址

数据在模块内存中映射的地址范围

读命令： 0x0000~0x01FF

写命令： 0x4000~0x41FF

写命令作为本地数据交换也可使用区域： 0x0000~0x01FF

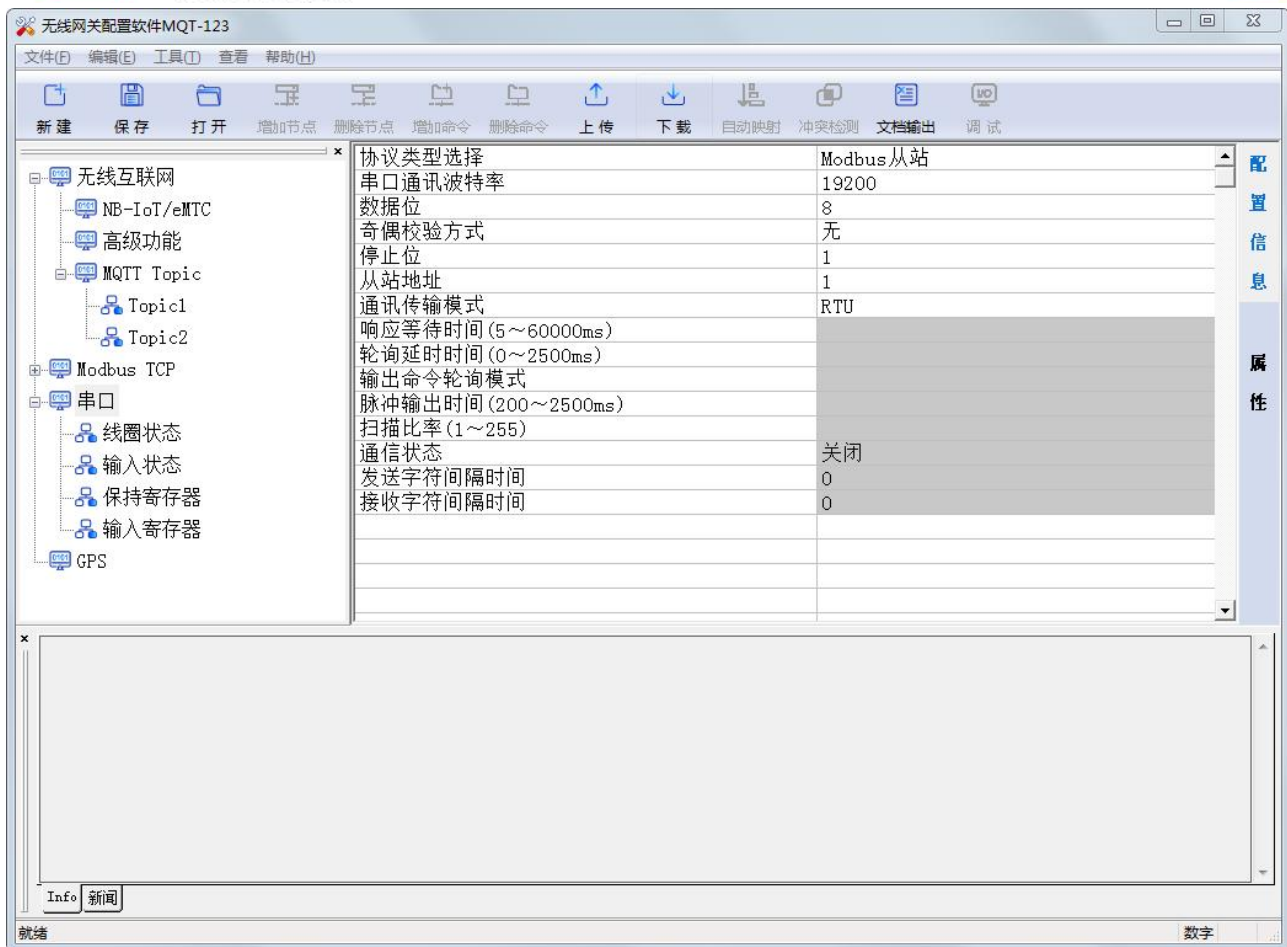
内存映射位偏移量（0 ~ 7）： 对于位操作指令，起始位在字节中的位置，范围是 0 ~ 7

扫描方式： 有两种扫描方式，快速扫描和慢速扫描，适应用户对不同的命令的快速扫描或慢速扫描的要求。慢速扫描周期等于快速扫描周期乘以扫描比率，一般扫描比率为 10（在“串口”设置界面中设置）

协议类型： Modbus 从站

可配置参数为： 串口通信波特率、数据位、奇偶校验方式、停止位、从站地址、通讯传输模式。

配置视图界面显示如下：



串口通信波特率：Modbus 端口的通讯波特率。

数据位：每个字节的数据位个数。

奇偶校验方式：奇、偶、无、标记和空格五种校验方式可选。

停止位：停止位个数，1、2 可选。

从站地址：范围是 0 ~ 247 可选。

通讯传输模式：数据符合 Modbus 协议时，数据的传输格式，支持 RTU、ASCII 模式可选。

线圈状态块大小：0XXXX 区的块大小，范围为 0 ~ 512。线圈状态起始地址加上线圈状态块大小的范围为 0 ~ 512。

输入状态位起始地址：1XXXX 区的起始地址，范围为 0 ~ 511。

输入状态位块大小：1XXXX 区的块大小，范围为 0 ~ 512。输入状态起始地址加上输入状态块大小的范围为 0 ~ 512。

保持寄存器起始地址：4XXXX 区的起始地址，范围为 0 ~ 255。



(1) 地球坐标 (WGS84):

国际标准，从 GPS 设备中取出的数据的坐标系

国际地图提供商使用的坐标系

(2) 火星坐标 (GCJ-02)也叫国测局坐标系

中国标准，从国行移动设备中定位获取的坐标数据使用这个坐标系

国家规定：国内出版的各种地图系统（包括电子形式），必须至少采用 GCJ-02 对地理位置进行首次加密。

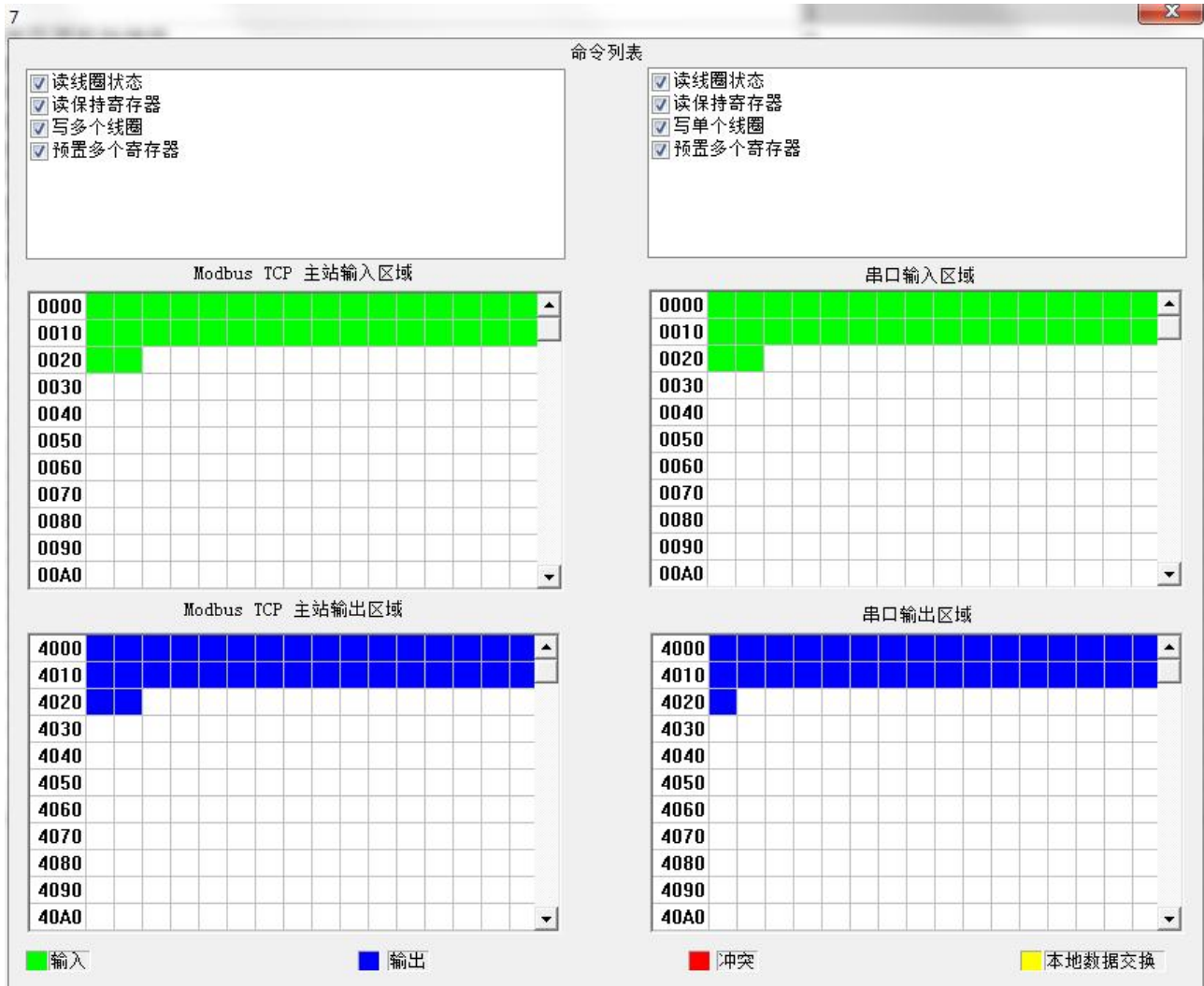
(3) 百度坐标 (BD-09)

百度标准，百度 SDK，百度地图，Geocoding 使用

坐标系转换工具：<https://tool.lu/coordinate/>

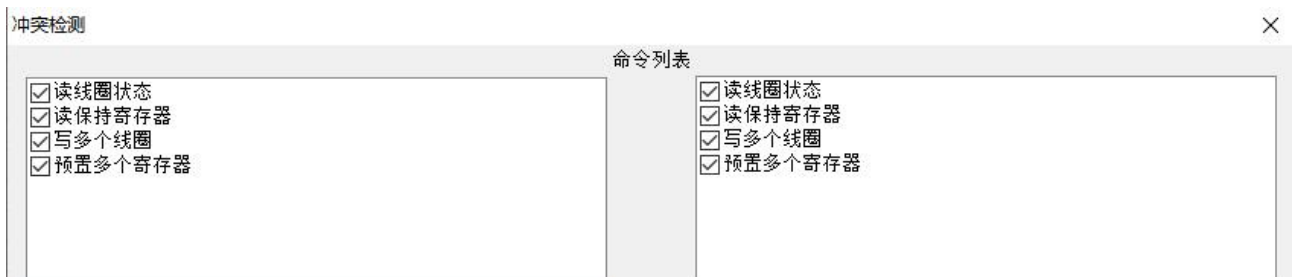
4.5 冲突检测

用于检测“内存映射数据”是否有冲突，若发现冲突的情况，可及时做调整。视图显示如下：



4.5.1 命令列表操作

在命令列表视图显示所有配置的命令，每条命令前的选中框，用于在内存映射区检查该条命令所占内存映射位置。单击某条命令，使选中框打勾，在内存映射区会显示相应命令所占空间位置，再次单击该命令，去掉选中框勾，命令不在映射区显示所占空间。该功能可用于命令间内存映射区的冲突检测。





4.5.2 内存映射区操作

内存映射区分输入区域和输出区域。

输入映射地址从 0x0000~0x3FFF；

输出映射地址从 0x4000~0x7FFF。

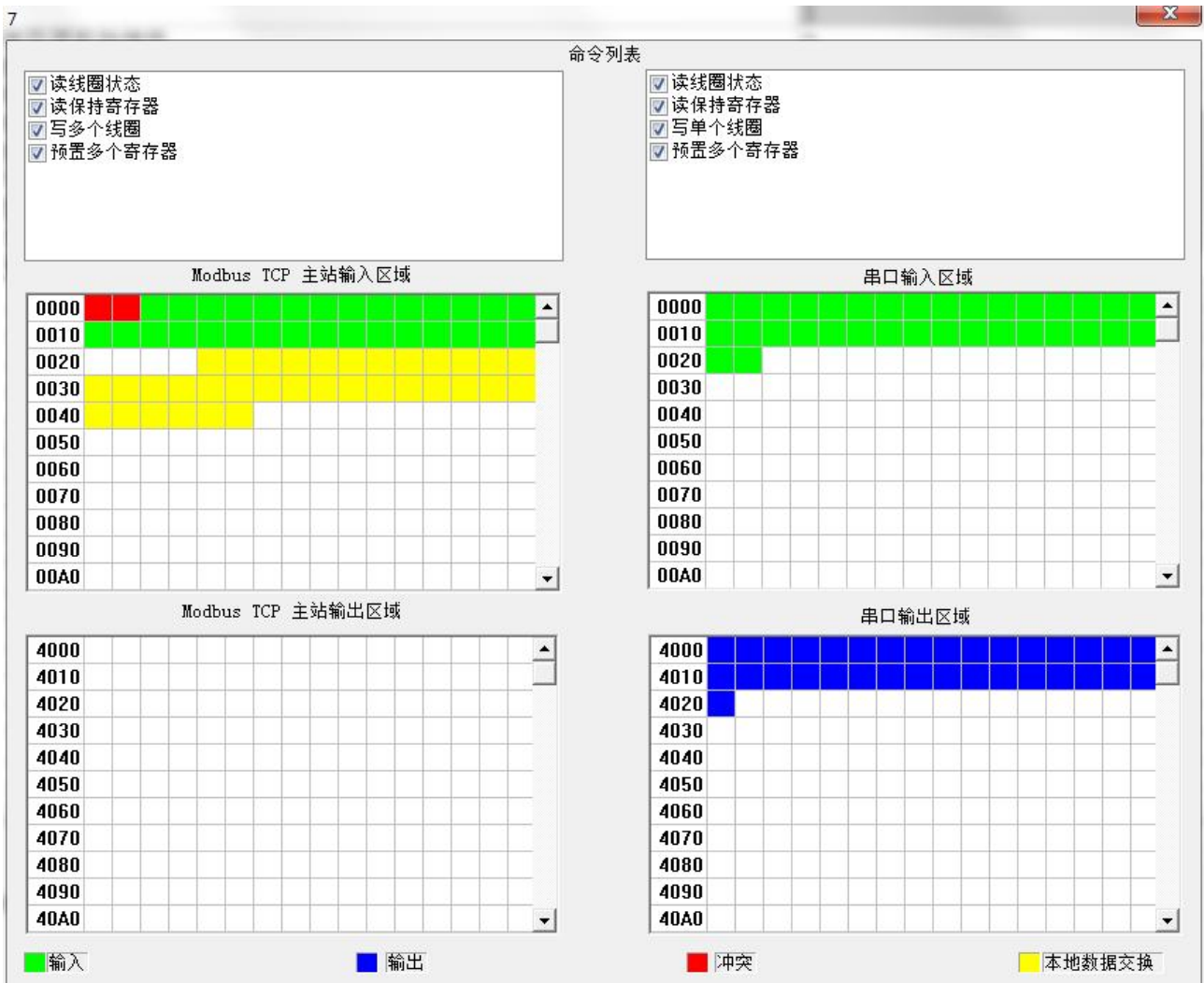
每个方格代表一个字节地址。

绿色：读命令在输入映射区显示，无冲突时呈绿色；

黄色：写命令当地址映射区位于输入区，无冲突时呈黄色；

蓝色：当地址映射区位于输出区，无冲突时呈蓝色。

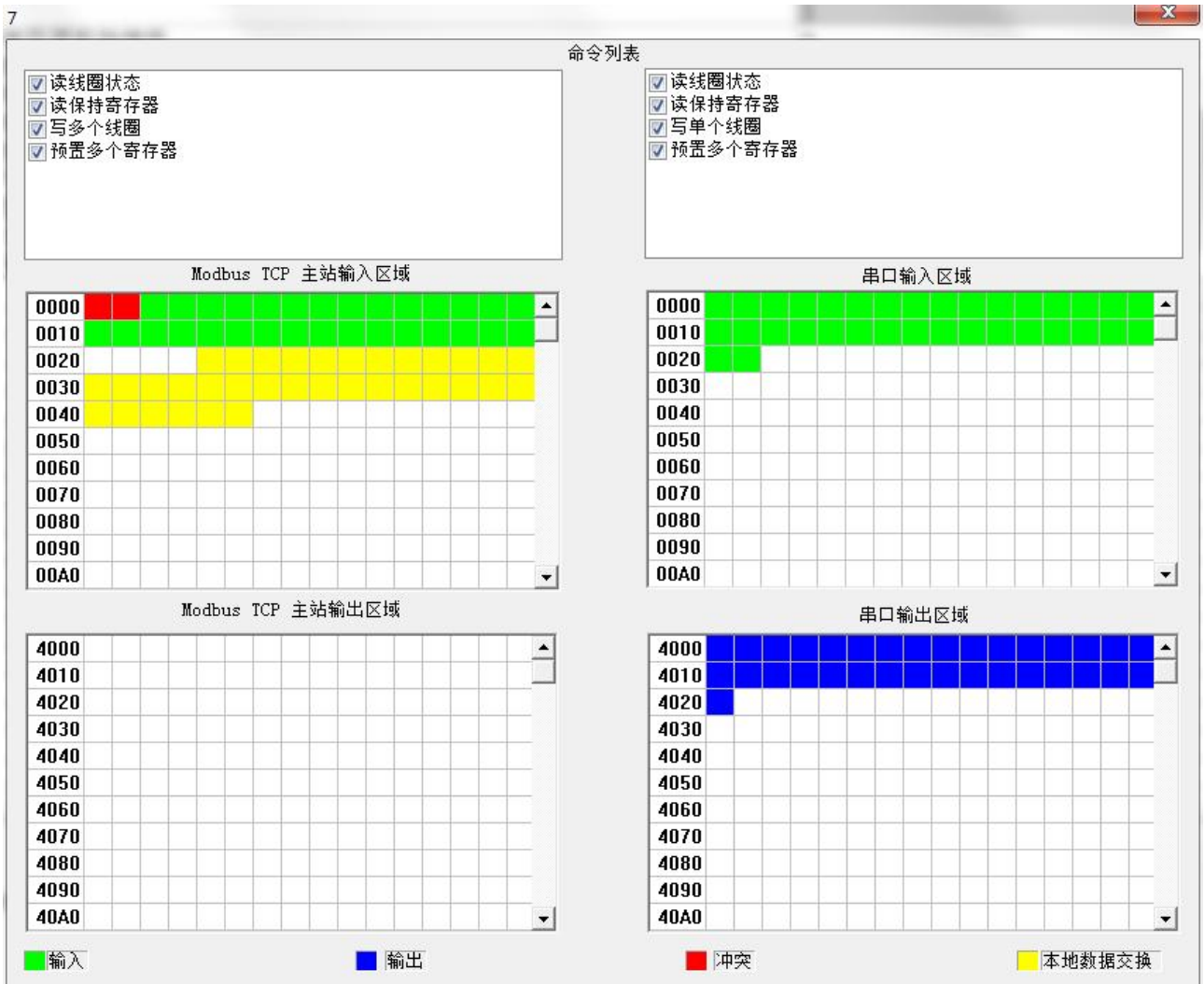
红色：在输入区或输出区，不同命令占用同一字节地址，该字节区域呈红色。



对于位操作指令，以上色格显示含义同样适用。



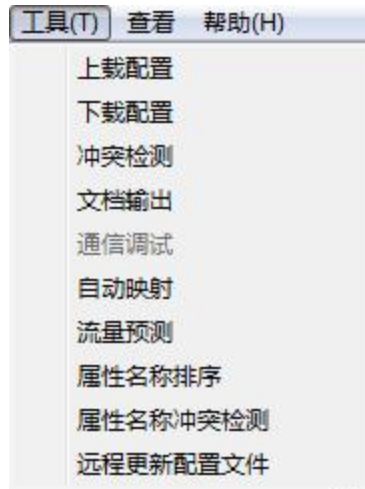
单击输入输出区域方格，该方格对应字节的各个位显示是否被占用，如下图所示：



上图指的是前两个字节数据有冲突。

4.6 上载下载配置

工具菜单项如下：



4.6.1 上载配置

选择上载配置，显示界面如下：

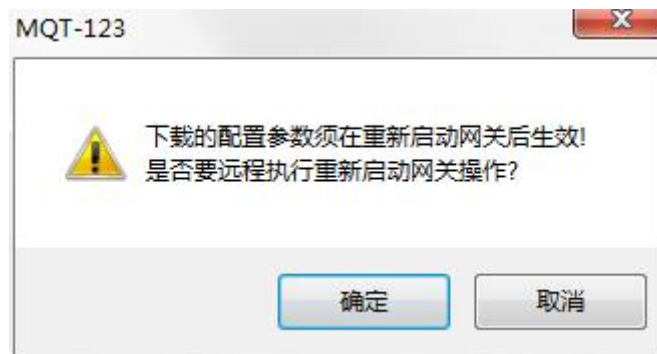




4.6.2 下载配置

选择下载配置，显示界面如下：





4.7 加载和保存配置

4.7.1 保存配置工程

选择“保存”，可以将配置好的工程以.json 文档保存。



4.7.2 加载配置工程

选择“打开”，可以将以保存的.json 文件打开。



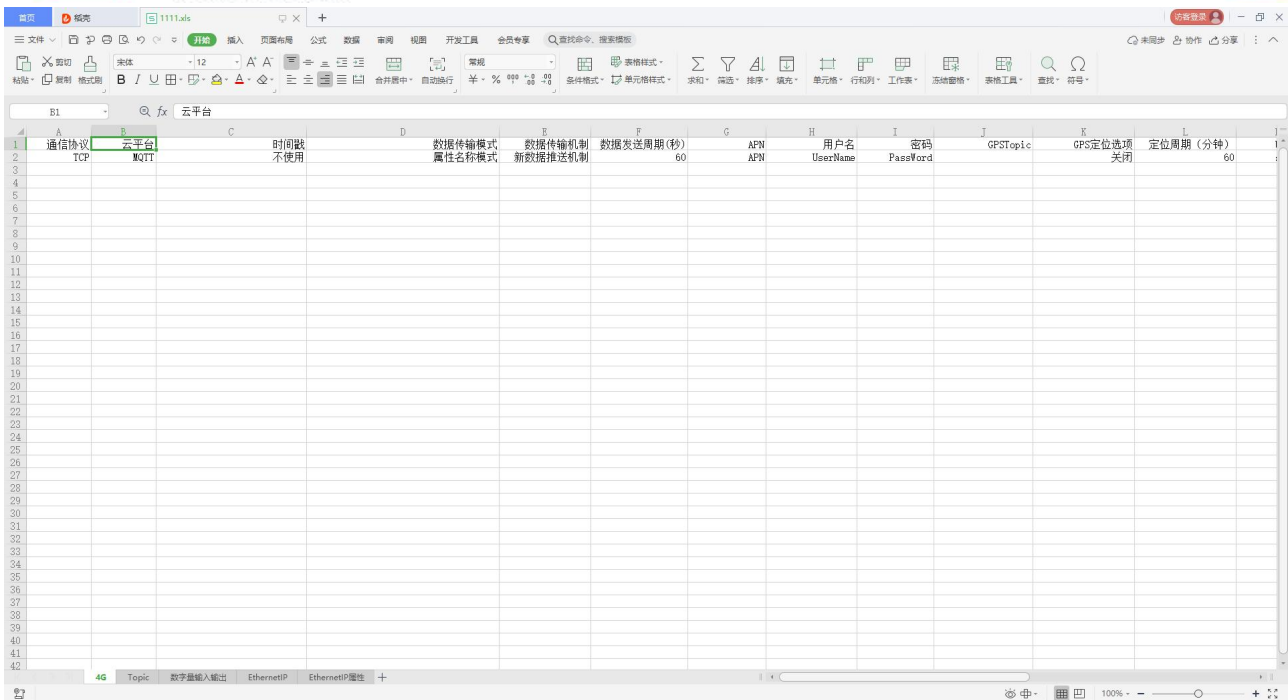
4.8 EXCEL 文档输出

配置文档输出有助于用户查看相关配置,使用此功能前请确保电脑上安装有 Microsoft Excel。



保存成功后将自动打开,如下所示:

MQT-123 导出的 Excel:



保存的 xls 有 NB-IoT/eMTC、Topic、数字量输入输出、Modbus TCP、Modbus TCP Slave 属性、串口、主站命令、Modbus 主站属性、Modbus 从站属性几个部分

NB-IoT/eMTC: 无线参数参数与 GPS 参数

Topic: Topic 参数

Modbus TCP: Modbus TCP Slave 参数;

Modbus TCP Slave 属性: Modbus TCP Slave 属性参数;

串口: 串口参数

主站命令: Modbus 命令参数

Modbus 主站属性: Modbus 主站属性参数

Modbus 从站属性: Modbus 从站属性参数

4.9 自动映射

自动映射功能是软件自动计算并填写每条命令的内存映射起始地址参数，自动映射后内存映射起始地址没有冲突。



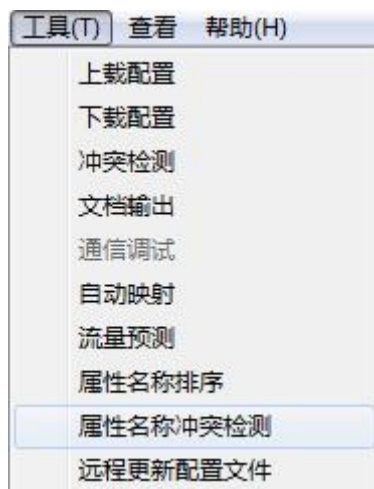
4.10 属性名称排序

属性名称排序功能是软件自动为一个 Topic 下的属性排序，使该 Topic 下的属性名称不冲突。



4.11 属性名称冲突检测

属性名称冲突检测功能是软件自动检测当前所做的配置中，是否存在同一个 Topic 下的属性名称重复，如果存在，软件会提示重复属性名称所在的位置。



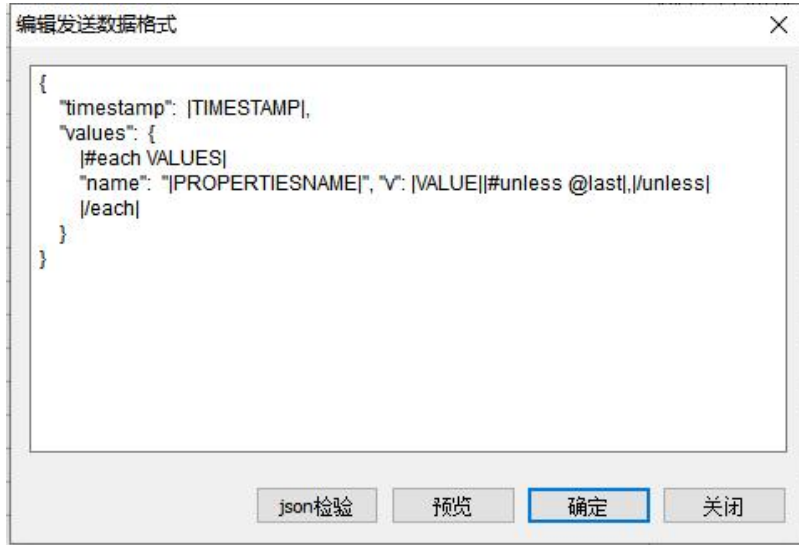
4.12 远程更新配置文件

本产品支持远程升级产品版本，和远程更新产品配置的功能。而远程更新产品配置则需要有专用的 Bin 类型的配置文件，与 MQT-123 软件保存配置生成的 JSON 类型的配置文件不同。“远程更新配置文件”的功能就是生成 Bin 类型的配置文件，将该文件提交到 GAPS 系统中（可向技术支持申请账号），经由管理员确认后即可实现远程更新产品配置。



4.13 自定义 JSON 格式

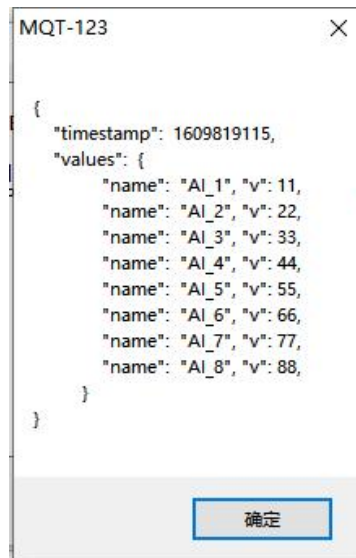
只在属性名称模式下支持自定义 JSON 格式，要编辑 JSON 格式在“自定义数据发送格式”选择开启即可。在弹出的窗口中编辑 JSON 格式，编辑完成后需进行 JSON 校验只有通过 JSON 校验后才能确认。



编辑完成后需进行 JSON 校验只有通过 JSON 校验后才能确认，单击“json”校验开始校验



校验通过后还可单击“预览”进行预览





(1) 变量说明:

| TAGNAME |

标记的名称

| VALUE |

标记的值

| TIMESTAMP |

读取标记的时间，表示为自 1970 年 1 月 1 日午夜以来的毫秒数

(2) 语法说明:

|#each VALUES|

|TAGNAME|, |VALUE|, |TIMESTAMP|,

|/each|

“each”关键字允许为每个属性更新生成文本。介于“|#each VALUES|”和“|/each|”之间的模板在发布时，会为每个属性复制出同样的模板，每次最多发布 8 个属性，超过 8 个属性分包发送

模板实例:

{

|#each VALUES|

"|TAGNAME|_value": |VALUE|,

"|TAGNAME|_timestamp": |TIMESTAMP|,

|/each|

}

{

"AI1_value": 23,

"AI1_timestamp": 1456150184825,

"AI2_value": 17,



...

}

(3) 附加语法说明:

```
|#unless @last|,|/unless|
```

这可以理解为：“除非这是列表中的最后一项，否则请插入逗号。”

这有效地消除了列表中最后一项的尾随逗号。虽然这在大多数情况下可以省略，但如果有效负载的使用者将尾部逗号视为语法错误，则有必要这样做。

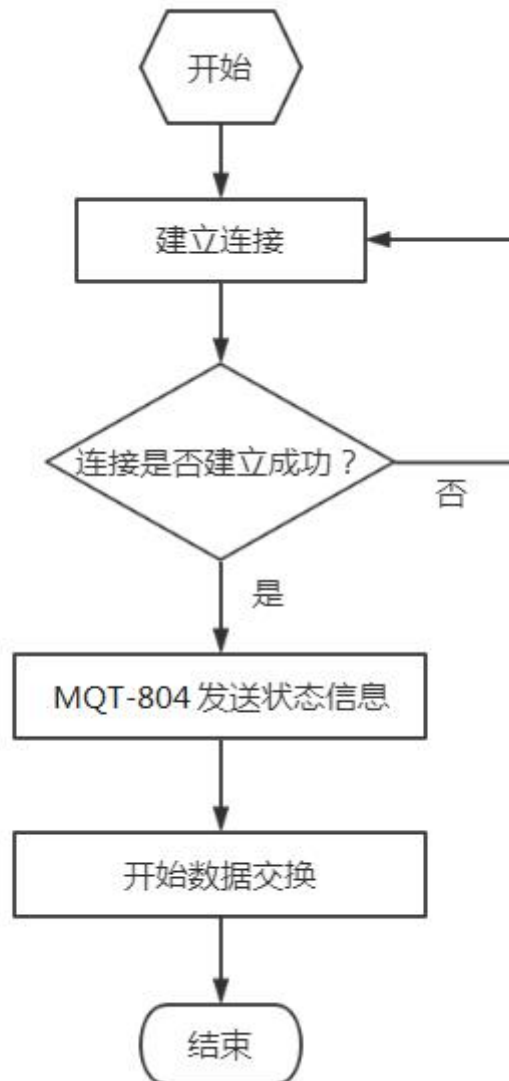
(4) 默认 JSON 格式如下

```
{  
  "version": "1.1.6.0",  
  "time": "19/03/27,17:10:03",  
  "properties": {  
    "AI2": 999999.500000,  
    "AI3": 888888.500000,  
    "AI4": 777777.500000,  
    "AI5": 666666.500000  
  }  
}
```



五、数据传输

5.1 连接流程



- ① MQT-804 与阿里云 IoT 等云平台的 Broker 建立通信
- ② 建立通信后，MQT-804 就会发送状态信息

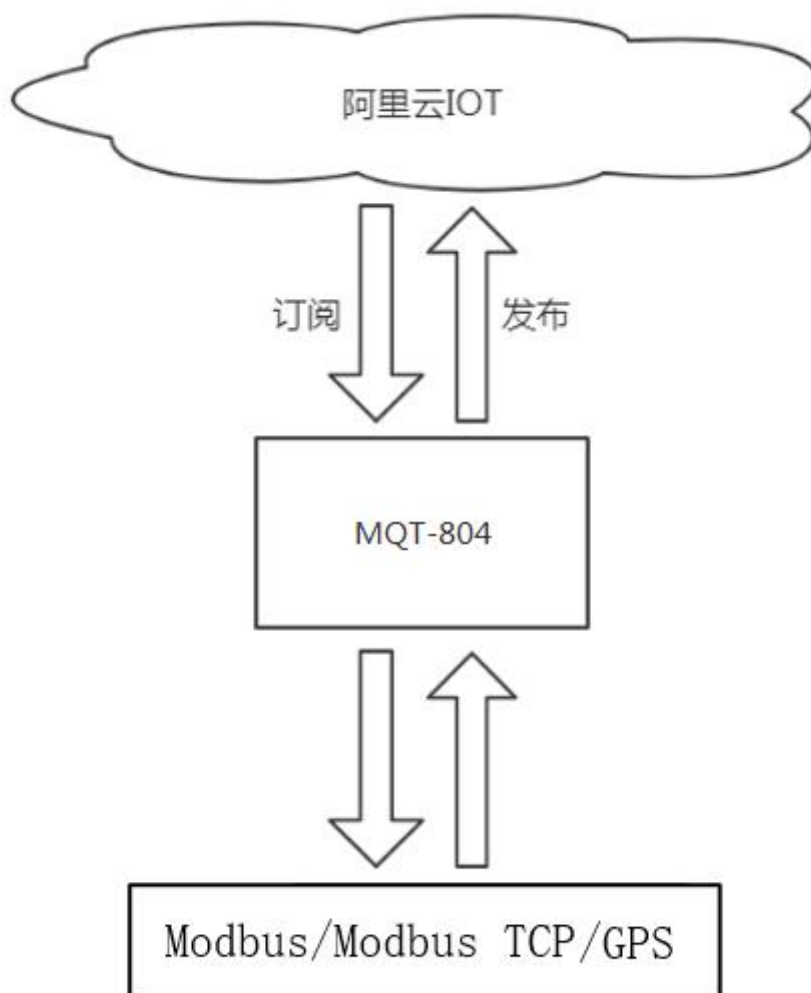
5.2 数据交换

本网关实现串口（Modbus）、Modbus TCP、GPS 通过 MQTT 与各大云平台（阿里云 IoT 等）进行数



据通讯，它们之间的数据是双向的转换和传递。

数据交换如下图所示：（以阿里云 IoT 为例）



- 1) 阿里云 IoT 对 MQT-804 写数据请求
- 2) 当阿里云 IoT 写数据成功之后，MQT-804 就会对阿里云 IoT 发送写命令响应

备注：

发布数据格式有 2 种，举例如下：

寄存器模式格式示例：

```
{  
  "version": "1.1.6.0",
```




User Manual

```
"type": "MODBUS_TCP_SLAVE",  
"block": 4,  
"addr": 1,  
"npoint": 4,  
"data": "11,22,33,44",  
"time": "19/03/27,16:48:39"  
}
```

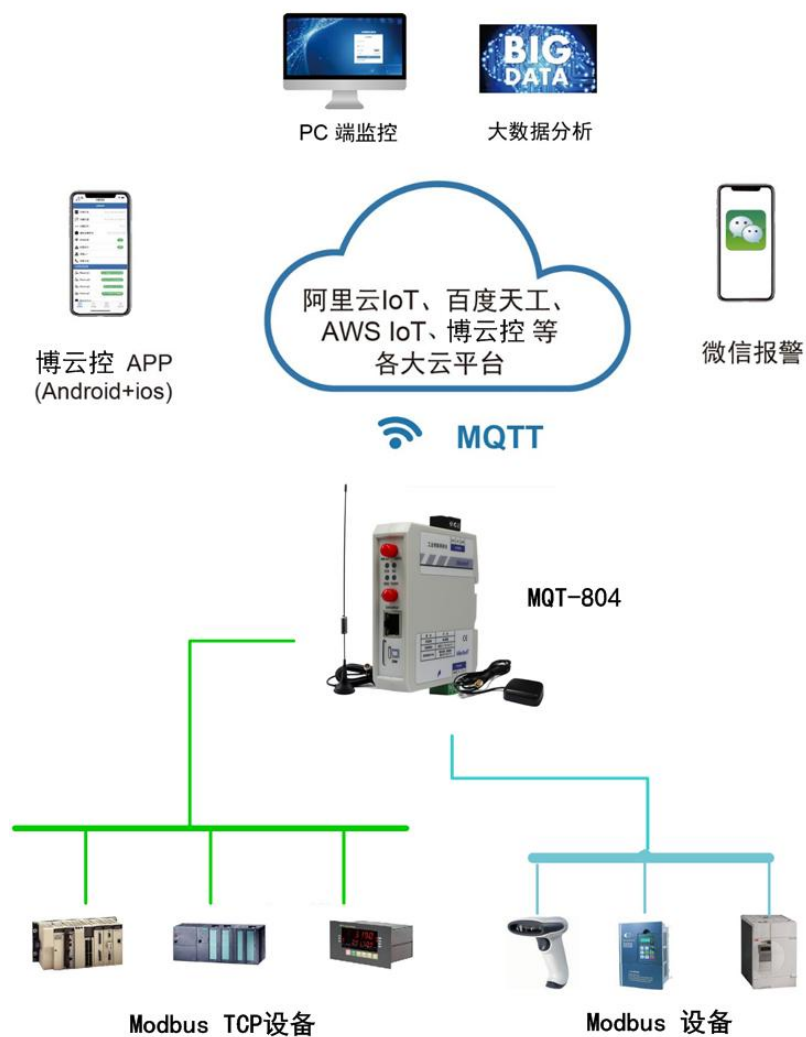
属性名称模式格式示例:

```
{  
  "version": "1.1.6.0",  
  "time": "19/03/27,17:10:03",  
  "properties": {  
    "AI2": 999999,  
    "AI3": 888888,  
    "AI4": 777777,  
    "AI5": 666666  
  }  
}
```



六、典型应用

MQT-804A 支持采集 Modbus 从站设备的数据、与 Modbus TCP 从站设备通信，MQT-804B 支持采集 Modbus 主/从站设备的数据、与 Modbus TCP 主站设备通信，通过 MQTT 协议将采集的数据及 GPS 定位信息上传到各大云平台。

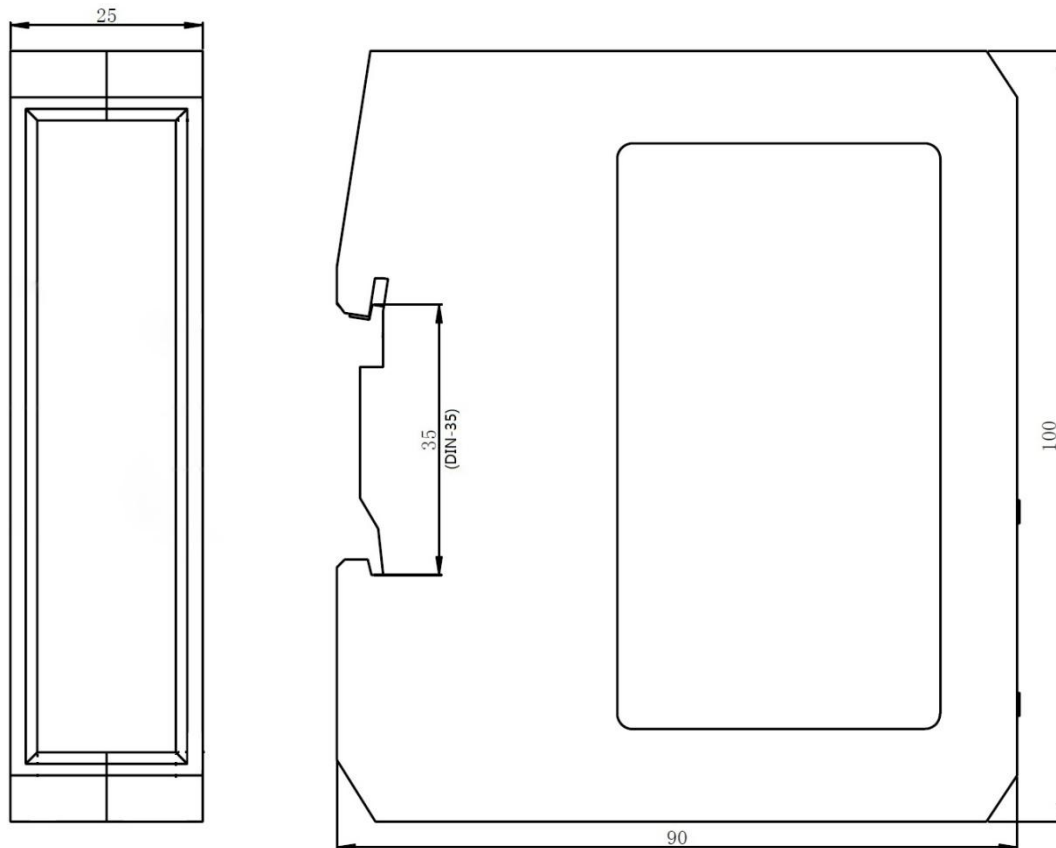




七、安装

7.1 机械尺寸

尺寸：25mm（宽）×100mm（高）×90mm（深）

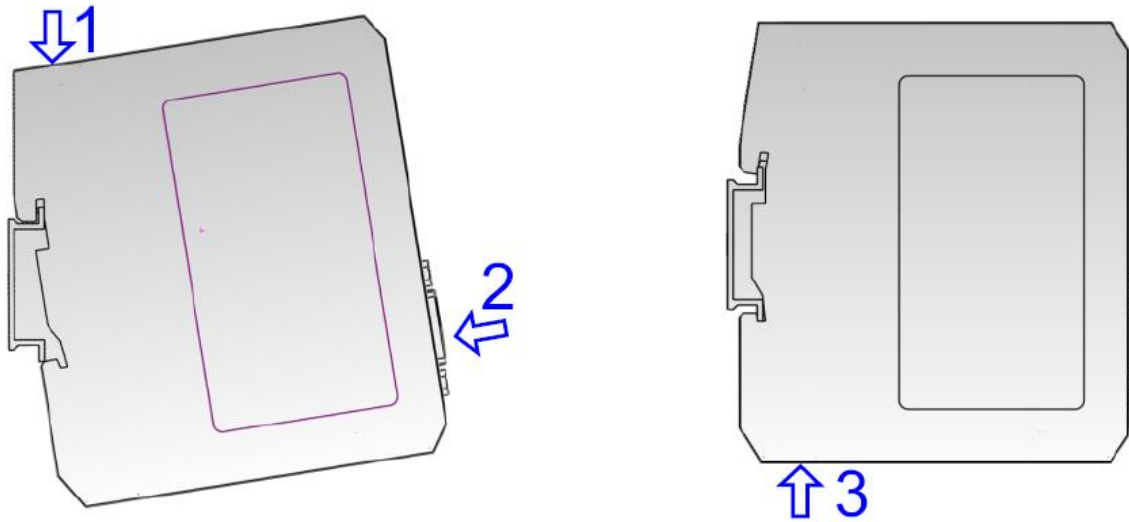


7.2 安装方法

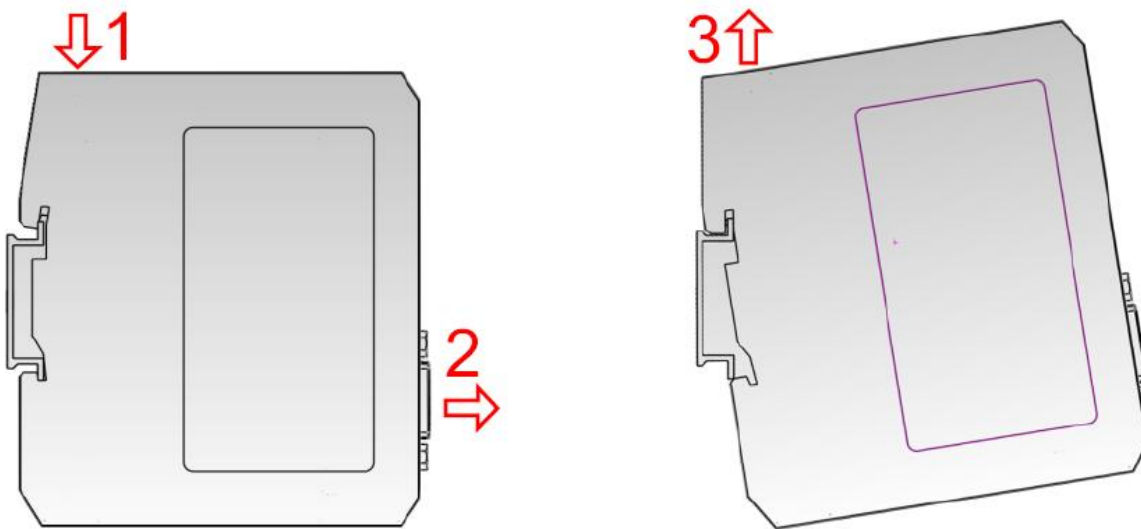
35mm DIN 导轨安装



安装网关



拆卸网关





八、运行维护及注意事项

- ◆ 模块需防止重压，以防面板损坏；
- ◆ 模块需防止撞击，有可能会损坏内部器件；
- ◆ 供电电压控制在说明书的要求范围内，以防模块烧坏；
- ◆ 模块需防止进水，进水后将影响正常工作；
- ◆ 上电前请检查接线，有无错接或者短路。



九、版权信息

本说明书中提及的数据和案例未经授权不可复制。泗博公司在产品的发展过程中，有可能在不通知用户的情况下对产品进行改版。

SiboTech[®] 是上海泗博自动化技术有限公司的注册商标。

该产品有许多应用，使用者必须确认所有的操作步骤和结果符合相应场合的安全性，包括法律方面，规章，编码和标准。



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

十、相关产品

本公司其它相关产品包括：

MGS-801、MGS-803B、MQT-805、IOT-860 等

获得以上几款产品的说明，请访问公司网站 www.sibotech.net，或者拨打技术支持热线：021-3126 5138。



十一、修订记录

时间	修订版本	修改内容
2022-12	A	V3.1 初始版本说明书

上海泗博自动化技术有限公司
SiboTech Automation Co., Ltd.
技术支持热线：021-3126 5138
E-mail: support@sibotech.net
网址： www.sibotech.net



附录 A: Modbus 协议

Modbus RTU 协议:

说明: 与本产品通讯的设备必须带有 Modbus 接口, 同时设备 Modbus 协议必须符合下面的规定, 本公司提供用户定制服务。

1. 协议概述

物理层: 传输方式: RS485

通讯地址: 0-247

通讯波特率: 可设定

通讯介质: 屏蔽双绞线

传输方式: 主从半双工方式。

协议在一根通讯线上使用应答式连接(半双工), 这意味着在一根单独的通讯线上信号沿着相反的两个方向传输。首先, 主计算机的信号寻址到一台唯一的终端设备(从机), 然后, 在相反的方向上终端设备发出的应答信号传输给主机。

协议只允许在主计算机和终端设备之间, 而不允许独立的设备之间的数据交换, 这就不会在使它们初始化时占据通讯线路, 而仅限于响应到达本机的查询信号。

一个数据帧格式:

1 位起始位, 8 位数据, 1 位停止位。

一个数据包格式

地址	功能码	数据	校验码
8-Bits	8-Bits	N x 8-Bits	16-Bits

协议详细定义了校验码、数据序列等, 这些都是特定数据交换的必要内容。

当数据帧到达终端设备时, 它通过一个简单的“口”进入寻址到的设备, 该设备去掉数据帧的“信封”(数据头), 读取数据, 如果没有错误, 就执行数据所请求的任务, 然后, 它将自己生成的数据加入到取得的“信封”中, 把数据帧返回给发送者。返回的响应数据中包含了以下内容: 终端从机地址(Address)、被执行了的命令(Function)、执行命令生成的被请求数据(Data)和一个校验码(Check)。发生任何错误都不会有成功的响应。

地址 (Address) 域

地址域在帧的开始部分, 由 8 位 (0~255) 组成, 这些位标明了用户指定的终端设备的地址, 该设备将接收来自与之相连的主机数据。每个终端设备的地址必须是唯一的, 仅仅被寻址到的终端会响应包含了该地址的查询。当终端发送回一个响应, 响应中的从机地址数据便告诉了主机哪台终端正与之进行通信。

功能 (Function) 域

功能域代码告诉了被寻址到的终端执行何种功能。表 1-1 列出了所有的功能码、它们的意义及它们的初始功能。



表 1-1 功能码

代码	意义	行为
03	读数据	获得一个或多个寄存器的当前二进制值
06	预置单寄存器	放置一个特定的二进制值到一个单寄存器中
16	预置多寄存器	放置特定的二进制值到一系列多寄存器中

数据域

数据域包含了终端执行特定功能所需要的数据或者终端响应查询时采集到的数据。这些数据的内容可能是数值、参考地址或者极限值。例如：功能域码告诉终端读取一个寄存器，数据域则需要指明从哪个寄存器开始及读取多少个数据，内嵌的地址和数据依照类型和从机之间的不同能力而有所不同。

错误校验域

该域允许主机和终端检查传输过程中的错误。有时，由于电噪声和其它干扰，一组数据在从一个设备传输到另一个设备时在线路上可能会发生一些改变，出错校验能够保证主机或者终端不去响应那些传输过程中发生了改变的数据，这就提高了系统的安全性和效率，出错校验使用了 16 位循环冗余的方法。

[注] 发送序列总是相同的 – 地址、功能码、数据和与方向相关的出错校验。

错误检测

循环冗余校验（CRC）域占用两个字节，包含了一个 16 位的二进制值。CRC 值由发送设备计算出来，然后附加到数据帧上，接收设备在接收数据时重新计算 CRC 值，然后与接收到的 CRC 域中的值进行比较，如果这两个值不相等，就发生了错误。

CRC 运算时，首先将一个 16 位的寄存器预置为全 1，然后连续把数据帧中的 8 位字节与该寄存器的当前值进行运算，仅仅每个字节的 8 个数据位参与生成 CRC，起始位和终止位以及可能使用的奇偶位都不影响 CRC。

在生成 CRC 时，每个 8 位字节与寄存器中的内容进行异或，然后将结果向低位移位，高位则用“0”补充，最低位（LSB）移出并检测，如果是 1，该寄存器就与一个预设的固定值进行一次异或运算，如果最低位为 0，不作任何处理。

上述处理重复进行，直到执行完了 8 次移位操作，当最后一位（第 8 位）移完以后，下一个 8 位字节与寄存器中的当前值进行异或运算，同样进行上述的另一个 8 次移位异或操作，当数据帧中的所有字节都作了处理，生成的最终值就是 CRC 值。

生成一个 CRC 的流程为：

预置一个 16 位寄存器为 0FFFFH（全 1），称之为 CRC 寄存器。

把数据帧中的第一个 8 位字节与 CRC 寄存器中的低字节进行异或运算，结果存回 CRC 寄存器。将 CRC 寄存器向右移一位，最高位填以 0，最低位移出并检测。

如果最低位为 0：重复第三步（下一次移位）。

如果最低位为 1：将 CRC 寄存器与一个预设的固定值（0A001H）进行异或运算。

重复第三步和第四步直到 8 次移位。这样处理完了一个完整的八位。

重复第 2 步到第 5 步来处理下一个八位，直到所有的字节处理结束。

最终 CRC 寄存器得值就是 CRC 的值。



2. 应用层功能详解

第一章已经简述了协议和数据帧，使用此软件的程序员可以使用下述的方法以便通过协议正确的建立他们的特定应用程序。

本章所述协议将尽可能的使用如图 2-1 所示的格式，（数字为 16 进制）。

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03H	03H	00H	01H	00H	03H	55H	E9H

图 2-1 协议例述

2.1 读数据（功能码 03）

查询

图 2-2 的例子是从 03 号从机读 3 个采集到的基本数据 U1,U2,U3, U1 的地址为 0001H, U2 的地址为 0002H, U3 的地址为 0003H,

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03H	03H	00H	01H	00H	03H	55H	E9H

图 2-2 读 Uca 和 Ia 的查询数据帧

响应

响应包含从机地址、功能码、数据的数量和 CRC 错误校验。

图 2-3 的例子是读取 U1,U2,U3 的响应。

地址	功能码	变量的总字节数	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	03H	06H	01H	7CH	01H	7DH	01H	7CH	F9H	9BH

图 2-3 读 U1,U2,U3 的响应数据帧

2.2 预置多寄存器（功能码 10）

查询

功能码 10H 允许用户改变多个寄存器的内容，设备可从任何地址开始设置最多 16 个变量的值。控制器是以动态扫描方式工作的，任何时刻都可以改变寄存器内容。

图 2-4 是修改 3 号从站设备的负载监控 1 和负载监控 2 的动作及延时时间的设定值，其中负载监控 1 的动作设定值寄存器地址为 00 2AH，设定值为 0x07 D0，延时时间的设定值寄存器地址为 00 2BH，设定值为 0x00 0A，负载监控 2 的动作设定值寄存器地址为 00 2CH，设定值为 0x07 0D，延时时间的设定值寄存器地址为 00 2DH，设定值为 0x00 0A。



地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	变量的总字节数	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	10H	00H	2AH	00H	04H	08H	07H	D0H	00H	0AH	07H	0D0H	00H	0AH	25H	7CH

图示 2-4 修改负载监控 1 和负载监控 2 的动作值及延时时间的设定值

响应

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量的个数高字节	变量的个数低字节	校验码低字节	校验码高字节
03	10H	00H	2AH	00H	04H	EBH	8DH

图示 2-5 修改负载监控 1 和负载监控 2 的动作值及延时时间的设定值的响应

2.3 预置单寄存器（功能码 06）

查询

功能码 06 允许用户改变单个寄存器的内容，DAE 系统内部的任何单寄存器都可以使用此命令来改变其值。既然仪器是以动态扫描方式工作的，任何时刻都可以改变单寄存器内容。

下面的例子是请求 03 号从机修改过载动作设定值 Ir1，Ir1 地址是 002EH。

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	06H	00H	2EH	07H	0D0H	EBH	8DH

图示 2-6 修改过载动作设定值 Ir1

响应

对于预置单寄存器请求的正常响应是在寄存器值改变以后将接收到的数据传送回去。

地址	功能码	变量起始地址高字节	变量起始地址低字节	变量值高字节	变量值低字节	校验码低字节	校验码高字节
03H	06H	00H	2EH	07H	0D0H	EBH	8DH

图示 2-7 修改过载动作设定值 Ir1



附录 B：数据通讯格式示例

发布：

注：“//”后为注释部分

属性名称模式：

GPS：

```
{  
"version": "1.1.1.0",  
"GPS": {  
    "Latitude": 31.057020, //纬度  
    "Longitude": 121.189590 //经度  
}  
}
```

Modbus 主站/ Modbus 从站/Modbus TCP 主站：

读线圈状态/读输入状态/线圈状态：

```
{  
"version": "1.1.1.0", //版本，以便不同版本的格式数据管理  
"time": "19/03/07,17:04:07", //时间戳  
"properties": {  
"BI3": false, //属性名称模式发送主站或从站数据时，若一次性要传输超过 8 个属性的  
数据时，那么 MQT-804 网关会将数据每 8 个数据为一包，分多包发送，直至将所有数据发送完毕。  
"BI4": true,  
"BI5": false,  
"BI6": false  
}  
}
```

读保持寄存器/读输入寄存器/保持寄存器：

```
{  
"version": "1.1.1.0",  
"time": "19/03/07,17:10:03",
```



```
"properties": {
  "AI2": 999999.500000, //数据与配置软件中设置的一致
  "AI3": 888888.500000,
  "AI4": 777777.500000,
  "AI5": 666666.500000
}
```

寄存器模式:

GPS:

```
{
  "version": "1.1.1.0",
  "type": "GPS",
  "Latitude": 31.057243,
  "Longitude": 121.188759
}
```

Modbus 主站:

读线圈状态/读输入状态:

实例（读线圈状态）:

```
{
  "version": "1.1.1.0",
  "type": "MODBUS_MASTER",
  "slave id": 1, //slave id 是从站的地址
  "fc": 1, //fc 是 Modbus 的功能码
  "addr": 0, // addr 是 Modbus 寄存器起始地址，从 0 开始
  "npoint": 2, //npoint 是 data 中数据的个数
  "data": "42,81", //data 中的数据与 Modbus 协议一致
  "time": "19/02/27,16:05:14" //时间戳，如不配置则没有
}
```

读输入寄存器/读保持寄存器:

实例（读保持寄存器）:

```
{
  "version": "1.1.1.0",
  "type": "MODBUS_MASTER",
  "slave id": 1,
  "fc": 3,
```



```
"addr": 0,  
"npoint":32,  
"data":"11,11,00,00,33,33,00,00,55,55,00,00,77,77,00,00,99,99,00,00,11,11,00,00,13,13,00,00,15,15,00,00"  
}
```

Modbus 从站:

线圈状态:

只改变一个线圈的值:

```
{  
  "version": "1.1.1.0",  
  "type": "MODBUS_TCP_SLAVE",  
  "block": 0, //block 代表区  
  "addr": 2, //addr 是 Modbus 寄存器起始地址, 从 0 开始  
  "npoint": 1, //npoint 为 1 时, 说明只改变了一个寄存器, 那么 data 中为 00 或 FF,  
  分别对应 0 或 1  
  "data": "FF",  
  "time": "19/02/27,16:13:40"  
}
```

改变连续多个线圈的值:

实例 1:

```
{  
  "version": "1.1.1.0",  
  "type": "MODBUS_TCP_SLAVE",  
  "block": 1,  
  "addr": 3, //addr 是 Modbus 寄存器起始地址, 从 0 开始  
  "npoint": 2, //npoint 不为 1, 说明连续改变多个线圈的值, 此例子需要去读取 data  
  中的第 1 个字节的第 4 位和第 5 位  
  "data": "18",  
  "time": "19/02/27,16:23:35"  
}
```

实例 2:

```
{  
  "version": "1.1.1.0",  
  "type": "MODBUS_TCP_SLAVE",  
  "block": 0,  
  "addr": 6, //addr 是 Modbus 寄存器起始地址, 从 0 开始  
  "npoint": 4, //npoint 不为 1, 说明连续改变多个线圈的值, 此例子需要去读取 data  
  中第 1 个字节的第 1 位和第 2 位以及第 2 个字节的第 7 位和第 8 位  
  "data": "c0,03", "time": "19/02/27,16:35:01"  
}
```



```
}
```

保持寄存器:

实例:

```
{  
"version": "1.1.1.0",  
"type": "MODBUS_TCP_SLAVE",  
"block": 4,  
"addr": 1,  
"npoint": 4,  
"data": "11,22,33,44",  
"time": "19/02/27,16:48:39"  
}
```

订阅:

注：“//”后为注释部分

属性名称模式：（一次最多只能改变 8 个属性）

Modbus 主站/ Modbus 从站/Modbus TCP 从站:

写线圈/输入状态:

```
{  
"properties": {  
"BO3": false,  
"BO4": true,  
"BO5": false  
}  
}
```

预置寄存器/输入寄存器:

```
{  
"properties": {  
"AO3": 123.5,  
"AO4": 321.5,  
"AO5": 555.6  
}  
}
```




寄存器模式:

Modbus 主站/Modbus TCP 主站:

写线圈:

写单个线圈:

```
{  
"type": "MODBUS_MASTER",  
"slave id": 1,  
"fc": 5, //fc 是 Modbus 的功能码  
"addr ": 0,  
"npoint": 1, //npoint 为 1 时, 说明只改变了一个寄存器, 那么 data 中为 00 或 FF,  
分别对应 0 或 1  
"data": "FF"  
}
```

写多个线圈:

```
{  
"type": "MODBUS_MASTER",  
"slave id": 1,  
"fc": 15, //fc 是 Modbus 的功能码  
"addr ": 0,  
"npoint": 2, //npoint 为 2 时, 与发布时的位对应关系一致  
"data": "ff,aa"  
}
```

预置寄存器:

预置单个寄存器:

```
{  
"type": "MODBUS_MASTER",  
"slave id": 1,  
"fc": 6,  
"addr ": 0,  
"npoint": 2,  
"data": "aa,bb"  
}
```

预置多个寄存器:

```
{  
"type": "MODBUS_MASTER",  
"slave id": 1,  
"fc": 16,  
"addr": 0,  
"npoint": 32,
```



```
"data": "11,11,00,00,22,22,00,00,33,33,00,00,44,44,00,00,00,00,00,00,00,00,00,00,88,88,00,00,99,99"
}
```

Modbus 从站:

输入状态:

改变单个输入状态位:

```
{
"type": "MODBUS_SLAVE",
"block": 1, //block 代表区
"addr": 0,
"npoint": 1,
"data": "FF"
}
```

改变多个输入状态位:

```
{
"type": "MODBUS_SLAVE",
"block": 1,
"addr": 0,
"npoint": 8, //npoint 不为 0, 说明改变多个, 这个例子是改变了 8 位
"data": "aa"
}
```

输入寄存器:

```
{
"type": "MODBUS_SLAVE",
"block": 3,
"addr": 1,
"npoint": 12,
"data": "02,22,03,33,04,44,05,55,06,66,07,77"
}
```

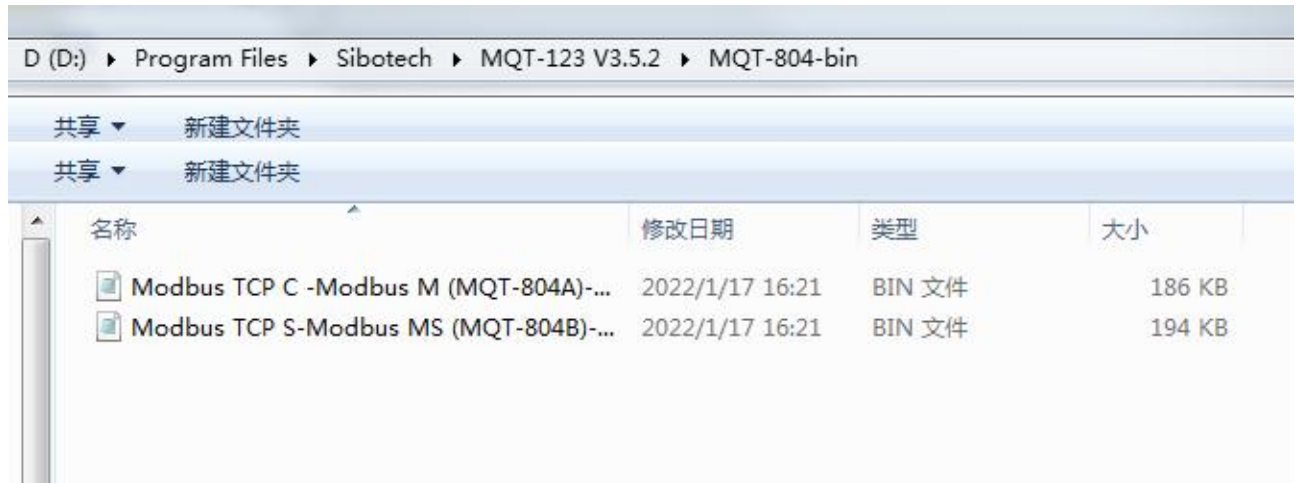


附录 C：切换产品工作模式操作步骤

MQT-804 两种工作模式的固件在 MQT-123 软件安装目录中；

例如 MQT-123 安装路径为：D:\Program Files\Sibotech\MQT-123 V3.5.2；

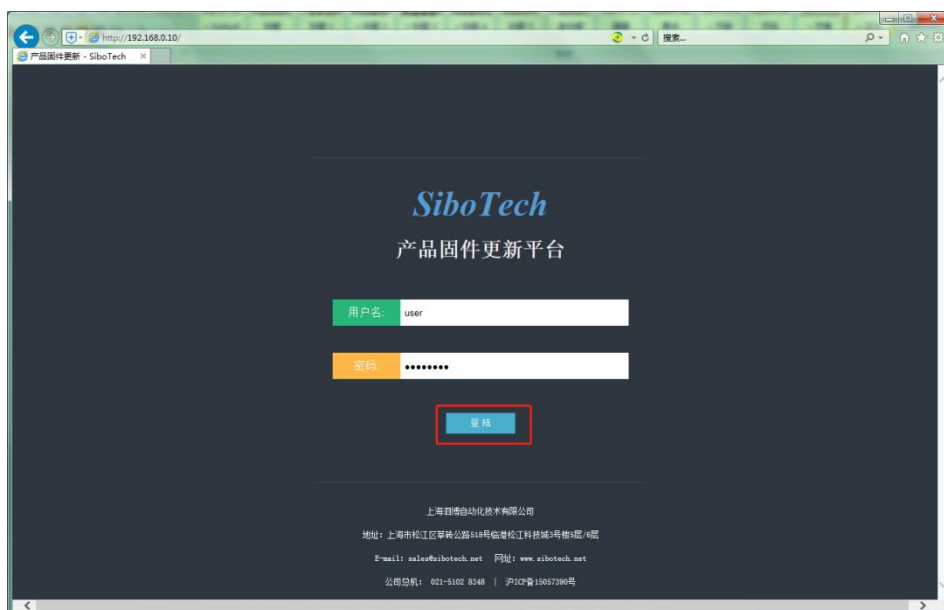
则 Bin 文件的目录为：D:\Program Files\Sibotech\MQT-123 V3.5.2\MQT-804-bin。



工作模式切换步骤：

1. 按住前面板按钮后再给网关上电，设备 ENS 和 TX/RX 指示灯绿灯闪烁，设备进入本地固件更新模式

2. 打开浏览器（支持 IE、谷歌和 360 等浏览器），输入 192.168.0.10 网址，输入用户名：user，密码：sibotech 后点击“登录”





3. 点击“浏览”，进入到存放固件的目录中，选择需要更新的固件后，点击“下载”即可。



4. 网页显示固件更新成功后，点击“重启设备”



5. 显示设备重启完成后，表示已成功切换网关模式，可结合配置软件使用新模式的功能



MQT-804

Modbus/Modbus TCP 转 MQTT

User Manual

