

类别	内容
关键词	MQTT、Modbus、边缘计算、网关
摘要	此文档旨在为客户提供快速入门指南、系统功能配置及其他功能说明等。

修订历史

版本	日期	原因
V1.04	2023/09/07	更新数据类型转换说明
V1.03	2023/08/30	增加 reg_batch 函数说明
V1.02	2023/08/21	更新自定义数据点章节的描述
V1.01	2023/08/18	更新数据类型转换和寄存器组合说明
V1.00	2023/06/29	创建文档

目 录

1. 产品简介.....	1
1.1 产品特点.....	1
1.2 应用领域.....	2
1.3 接口实物图.....	2
1.4 接口说明.....	2
1.4.1 接口.....	2
1.4.2 指示灯.....	3
2. 快速入门.....	4
2.1 软件准备.....	4
2.1.1 安装配置上位机.....	4
2.2 硬件准备.....	5
2.3 搜索设备.....	5
2.4 以太网配置.....	6
2.5 串口配置.....	7
2.6 MQTT 配置.....	7
2.7 配置 Modbus 轮询数据点.....	8
2.8 使用 Modbus Slave 模拟从机.....	9
2.9 使用 MQTT 客户端查看数据.....	11
3. 网关配置.....	12
3.1 设备管理.....	12
3.1.1 设备搜索.....	12
3.1.2 读取配置.....	12
3.1.3 导入配置、导出配置和保存配置.....	13
3.2 查看网关状态.....	13
3.3 串口配置.....	14
3.4 以太网配置.....	14
3.5 MQTT 配置.....	15
3.5.1 基本配置.....	15
3.5.2 订阅主题.....	16
3.5.3 发布主题.....	16
4. 边缘计算.....	18
4.1 配置导入与导出.....	18
4.2 添加从机和数据点.....	19
4.2.1 添加从机.....	19
4.2.2 添加数据点.....	19
4.2.3 数据报文格式.....	20
4.3 自定义数据格式.....	21
4.3.1 基本用法.....	23
4.3.2 高级用法.....	24
4.3.3 注意事项.....	26
4.4 上报模式.....	26

4.4.1	立即上报.....	26
4.4.2	变化上报.....	26
4.4.3	阈值上报.....	27
5.	远程控制.....	29
5.1	查询数据点.....	29
5.2	设置数据点.....	29
6.	系统配置.....	31
6.1	系统信息.....	31
6.2	升级.....	31
6.3	重启与恢复.....	32
6.4	修改账户密码.....	32
6.5	设置时间.....	33
7.	FAQ.....	34
8.	免责声明.....	35

1. 产品简介

GCOM80-2NET 是广州致远电子股份有限公司专为工业领域数据采集、现场设备接入而开发的工业级网关产品。采用紧凑型设计，数据通道全隔离设计，为数据传输稳定性带来了足够的保障，同时具备 Modbus 数据采集与边缘计算、传输、时间同步等功能，极大缓解客户数据上云压力。

此产品具备 8 路隔离 RS485 端口，2 路百兆以太网口，支持设备远程配置/远程监控，支持 Modbus RTU 与 MQTT 上下行数据格式转换，可运用于复杂的工作环境中，可广泛满足各类工业现场设备数据传输应用需求。



1.1 产品特点

- 支持 MQTT 网络协议；
- 支持 Modbus RTU 与 JSON 互转；
- 支持 bit、uint16/int16、int32/uint32、float、BCD 等多种 ModbusRTU 数据类型及大小端转换；
- 支持对采集 ModbusRTU 数据定时上报、变化上报、阈值上报；
- 支持中文数据点上报；
- 支持强大的寄存器数据自定义处理，包括简单的倍率转换、位运算、寄存器间组合运算、幂次运算、条件判断 if 等，真正实现数据边缘计算，极大减轻服务器处理压力及流量焦虑；
- 支持 MQTT 远程读取及下发修改数据点；
- 支持静态 IP、DHCP 方式获取 IP；

- 支持 NTP 时间校准；
- 支持看门狗及复位；
- 支持上位机软件配置；
- 可通过 LED 查看各数据通道状态，如网络连接、数据收发等；

1.2 应用领域

- 动环监测系统；
- 工业现场设备数据采集系统；
- 储能电站系统；
- 工商储能系统；
- 自动化控制领域；
- 储能 BMS 系统；
- 远程抄表

1.3 接口实物图

如图 1.1、图 1.2 所示，GCOM80-2NET 面板分指示灯区域和接口区域，接口区域含有电源、8 路 RS485、2 路网口、USB 接口和 RESET 按键。



图 1.1 GCOM80-2NET 产品正面接口图



图 1.2 GCOM80-2NET 产品背面图

1.4 接口说明

1.4.1 接口

GCOM80-2NET 接口功能说明如表 1.1 所示。

表 1.1 接口说明

接口名称	丝印标号	描述
电源接口	DC 9~36V	DC 电源接口，DC 9~36V
RS485 接口 ^①	RS485	用于 RS485 通信，引出三根线分别是 A(DATA+)、B(DATA-)、GND，连接 RS485 设备是 A(+)接 A(+), B(-)接 B(-)
按键	RESET	1.复位：按键短按（3 秒以内）松开网关会复位；

		2.恢复出厂：长按 3s 以上，SYS 灯由慢闪变快闪（周期 100ms）后松开按键，系统自动恢复出厂设置，约 2S 后自动重启
网口 1	NET1	可用于以太网通信和 网关配置 ^②
网口 2	NET2	可用于以太网通信和 网关配置 ^③
USB Type-C	USB	可用于 网关配置

注①：上电过程中，RS485-1 串口间隔 50ms 持续发送“gcom80_2net_upgrade”，即可进入 Ymodem 升级，默认波特率 115200bps；

注②、注③：如果网口配置成 DHCP，但网口无法从路由器获得有效 IP 地址，那么上位机需要等待约 10s 后再配置。

1.4.2 指示灯

GCOM80-2NET 指示灯功能说明如表 1.2 所示。

表 1.2 指示灯说明

名称	说明
PWR	电源指示灯，单色灯（红色）
SYS	1. 系统运行灯，单色灯（绿色）； 2. 正常运行时，周期 1000ms 闪烁； 3. 恢复出厂时，周期 100ms 快闪 2S；
RS485-1~RS485-8	1. RS485 通信指示灯，单色灯（绿色）； 2. 正常运行时常亮， 数据采集成功 一次则闪烁一次
NET1、NET2	1. MQTT 连接指示灯，单色灯（绿色）； 2. 当 MQTT 未连接 NET 灯不亮，当 MQTT 连接成功 NET 灯亮； 3. 当有数据收发时 NET 灯闪烁

2. 快速入门

本章节将指引用户如何使用网关通过从 RS485-1 端口 Modbus 采集从机数据并上报至 MQTT 服务器。

流程如下：

1. 安装配置上位机；
2. 硬件准备；
3. 上位机搜索并登录设备；
4. 配置以太网参数；
5. 配置串口参数；
6. 配置 MQTT 参数；
7. 配置从机和数据点；
8. 配置 Modbus Slave 模拟从机设备；
9. 查看网关上报数据。

2.1 软件准备

2.1.1 安装配置上位机

上位机 GXCOM-Tool 可从致远电子官网下载（www.zlg.cn），下载完之后即可安装，安装过程如下。



图 2.1 点击“下一步”

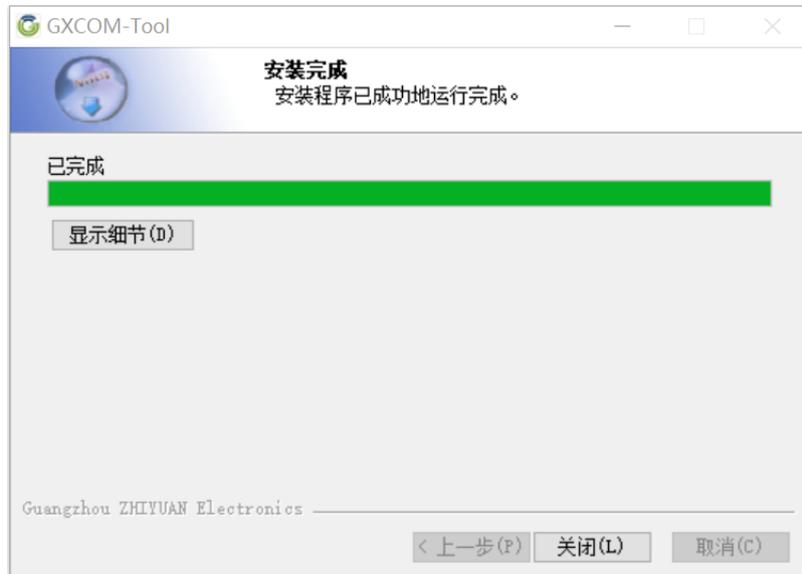


图 2.2 成功安装，点击“关闭”即可

2.2 硬件准备

给网关供电，然后用数据线将网关的 **USB** 口连接到电脑，然后使用 **USB 转 RS485** 模块将网关的 **RS485-1** 端口连接到电脑，如图 2.3 所示，硬件准备就完成了。



图 2.3 硬件连接示意图

2.3 搜索设备

打开上位机，如图 2.4 所示，然后点击左上角的“设备搜索”，等待约 2 秒后可以看到搜索到的设备，然后双击设备会弹出登录窗口，如图 2.5 所示，输入密码“admin”即可进入配置页面。

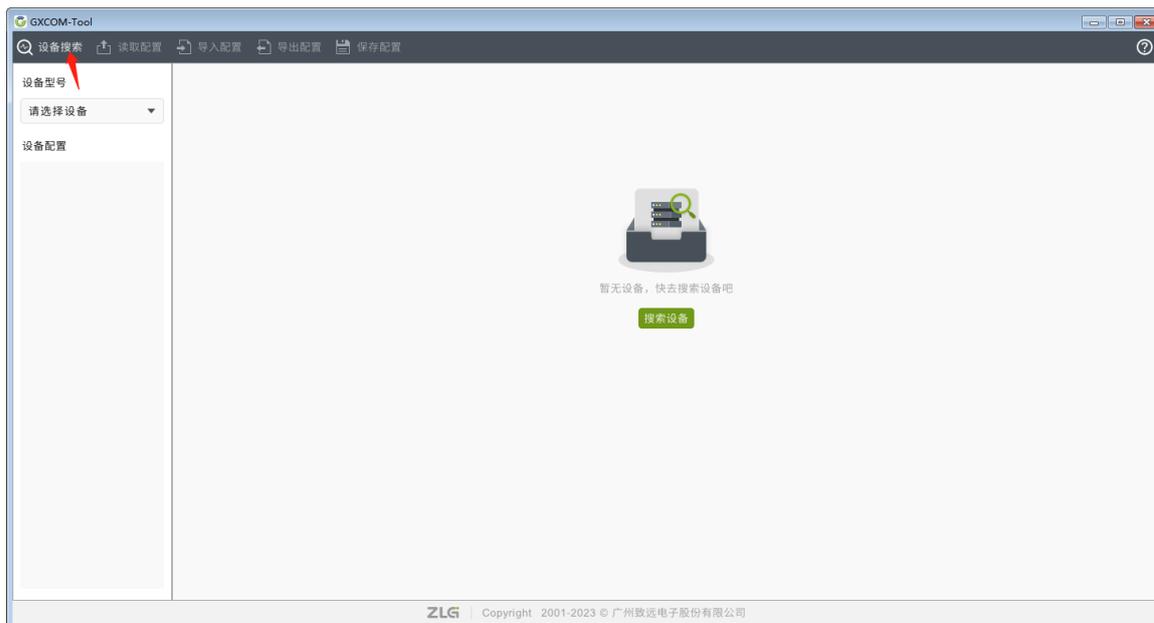


图 2.4 配置软件主页面



图 2.5 登录设备

2.4 以太网配置

如图 2.6 所示, 将以太网 NET1 配置成 DHCP, 用户需要注意路由器是否已经开启了 DHCP 服务功能, 否则无法为网关分配 IP 地址。

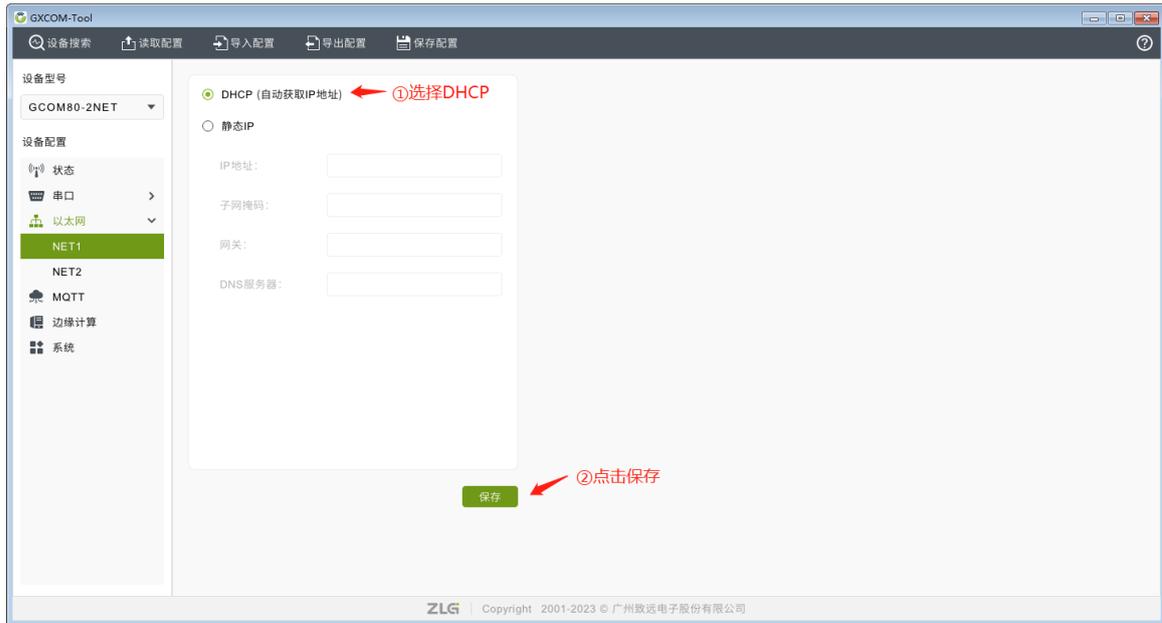


图 2.6 NET1 开启 DHCP

2.5 串口配置

用户按如图 2.7 所示参数配置串口 1（即 RS485-1 端口），包括波特率及 Modbus 从机响应超时时间。



图 2.7 配置串口

2.6 MQTT 配置

如图 2.8 所示，将服务器地址配置成“test.zlgcloud.com”（支持域名及 IP），端口号配置为 1893，其它配置保持默认。

配置好服务器地址后，可通过观察网关的 NET1 指示灯状态查看是否连上服务器，如果

有亮，说明已成功连上服务器。



图 2.8 MQTT 客户端参数配置

2.7 配置 Modbus 轮询数据点

如图 2.9 所示，点击下方的“添加从机”按钮添加从机，然后按截图内容配置从机。

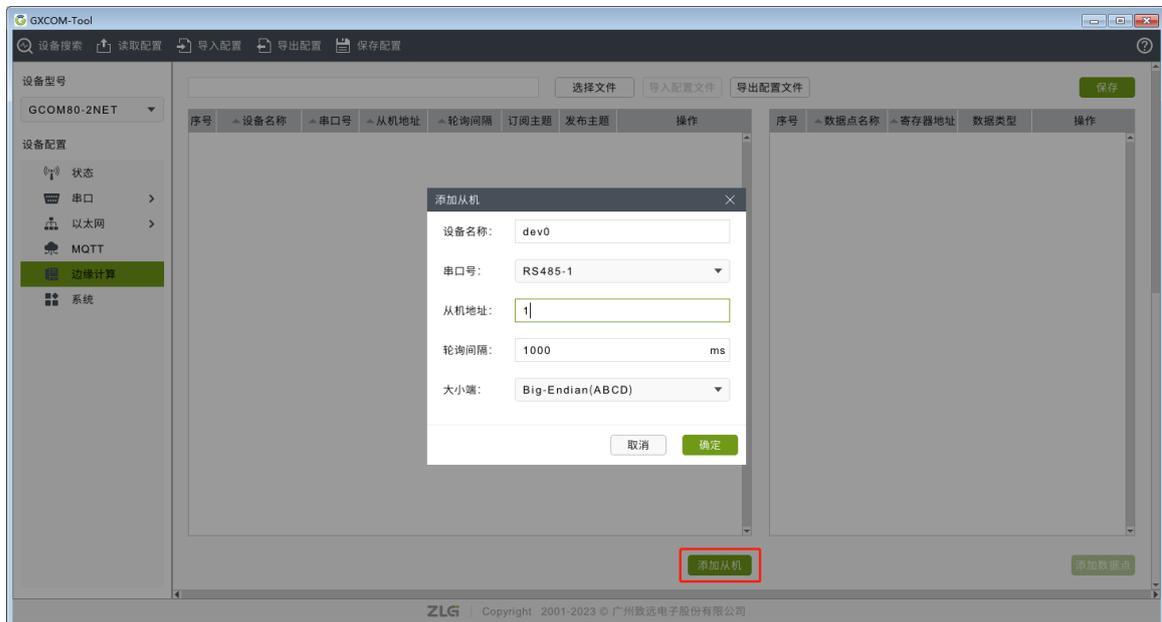


图 2.9 添加从机

然后鼠标选中左边窗口的从机，然后再点右下角的“添加数据点按钮”，在弹窗中按图 2.10 所示内容配置数据点。最后点击右上角的保存按钮，保存成功后会有弹窗提示是否重启设备，这里点击确定。

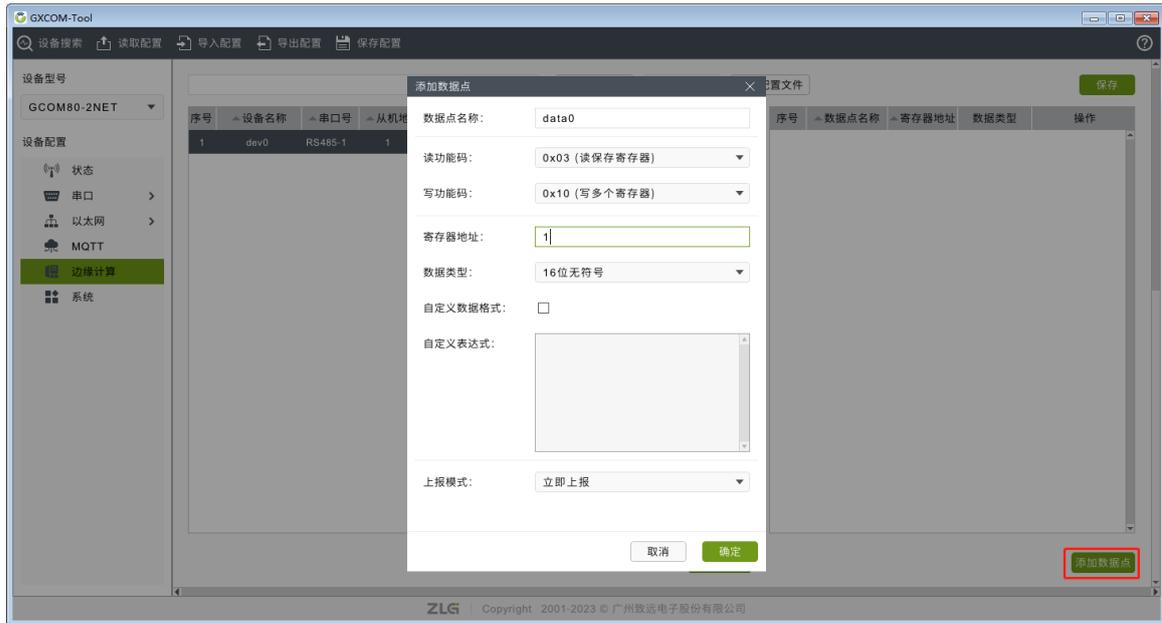


图 2.10 添加数据点

2.8 使用 Modbus Slave 模拟从机

为了方便模拟 Modbus 从机设备，用户可以自行搜索下载并安装“Modbus Slave”软件，打开软件，如图 2.11 点击“Connection->Connect”后配置串口相关参数，这里以波特率 115200，数据位 8bit，停止位 1bit，模式为 RTU 为例，然后点击 OK。

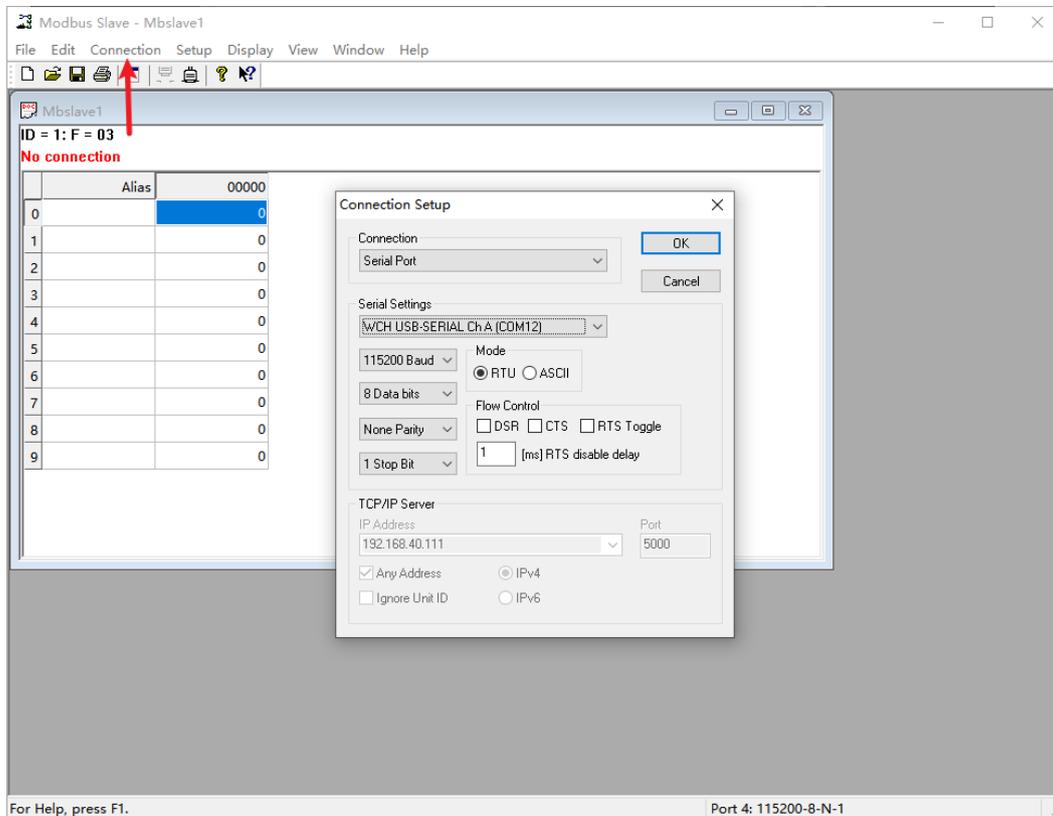


图 2.11 配置从机波特率

然后配置从机寄存器，如图 2.12 所示点击箭头的图标，在弹窗中配置从机地址为 1，功能码为 03，寄存器地址为 0，数量为 10，其它参数按图片设置，然后点击 OK。

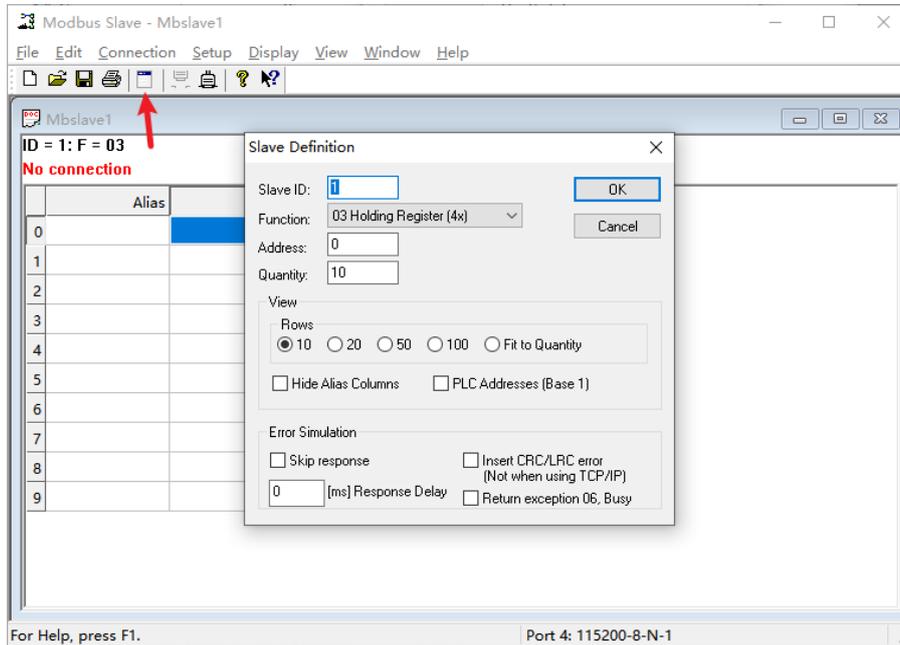


图 2.12 配置寄存器

然后点击如图 2.13 的图标打开串口。

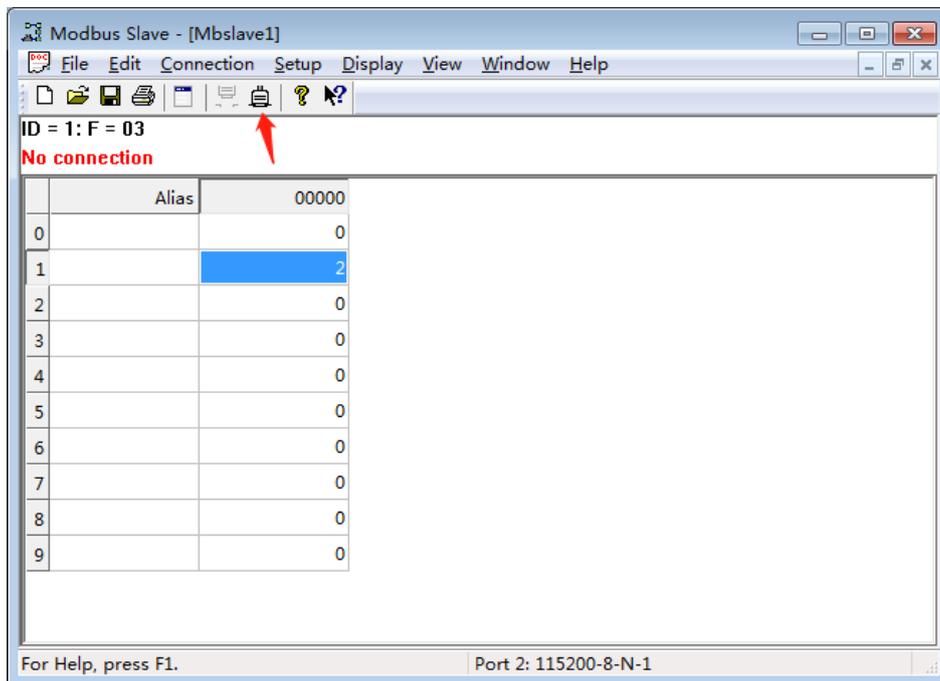


图 2.13 打开串口

如果上述数据点及从机配置正常，可看到“RS485-1”绿灯会按照轮询周期闪烁，说明网关已轮询正常。

2.9 使用 MQTT 客户端查看数据

然后打开 MQTT 客户端，这里以 MQTXX 为例，用户可以使用其它软件代替。如图 2.14 所示，配置好服务器地址 test.zlgcloud.com 和端口 1893，然后点击 Connect。

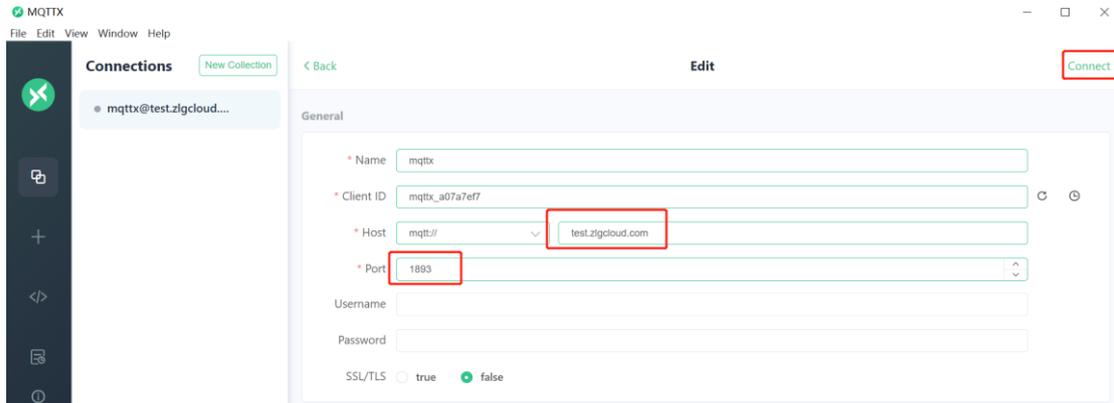


图 2.14 配置 MQTT 客户端

然后如图 2.15 所示订阅主题 “/mqtt_pub”。

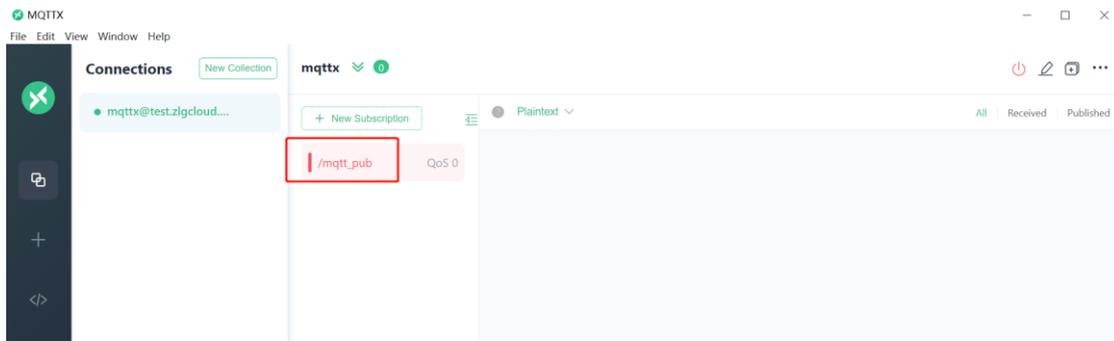


图 2.15 订阅主题

如图 2.16 所示，查看 mqtt 客户端订阅主题上报的数据内容，说明网关采集数据并上报成功。

其中 dev0 为设备名，data0 为寄存器数据点数据。



图 2.16 订阅主题数据

3. 网关配置

3.1 设备管理

如图 3.1 所示，上位机的菜单栏为设备管理按钮。



图 3.1 设备管理

3.1.1 设备搜索

点击设备搜索按钮，等待约 2 秒后，如图 3.2 所示可以看到窗口上显示搜索到的网关信息。



图 3.2 设备搜索

3.1.2 读取配置

因为网关的状态页和系统信息页不是实时刷新的，所以用户查看时需要先点击“读取配

置”按钮刷新一下状态。

3.1.3 导入配置、导出配置和保存配置

有些场景中，我们需要把一台网关的配置批量导入到其他网关使用，我们可以用这个功能。

如图 3.3 所示，用户点击“导出配置”将网关所有配置导出到文件进行保存，用户点击“导入配置”将配置文件导入到上位机，然后点击“保存配置”可以将所有配置下发给网关，下发完配置后会上位机提示是否重启网关，建议立即重启。



图 3.3 网关导入、导出配置

3.2 查看网关状态

如图 3.4 所示，该界面主要显示网关的一些状态，用户通过这些状态可以了解整个网关的运行情况。



图 3.4 网关状态

3.3 串口配置

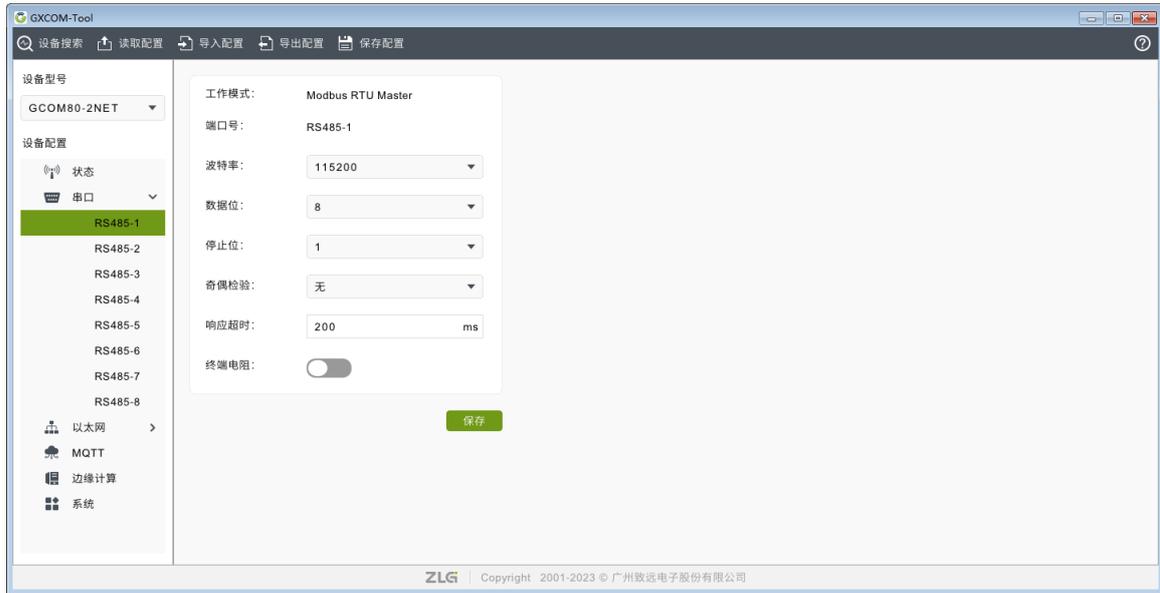


图 3.5 串口配置

串口配置页面如图 3.5 所示，下面详细介绍串口配置：

表 3.1 串口参数说明

串口参数	说明
波特率	支持 2400、4800、9600、19200、38400、57600、115200、230400
数据位	仅支持 8bits
奇偶校验	支持无、奇和偶
停止位	支持 1bits 和 2bits
响应超时	表示 Modbus 主机发完轮询报文后等待从机应答报文的超时时间
终端电阻	是否使能 120 Ω 终端电阻

3.4 以太网配置

如图 3.6 所示，用户可以配置以太网的 IP 地址、DHCP 等功能。填写或修改配置内容之后，点击保存即可生效。



图 3.6 以太网配置

在配置以太网时需要注意选择静态 IP 时需要配置 IP 地址、子网掩码和网关；

3.5 MQTT 配置

3.5.1 基本配置

如图 3.7 所示为 MQTT 基本配置。



图 3.7 MQTT 基本配置

下面详细介绍基本配置参数：

表 3.2 MQTT 配置参数说明

MQTT 配置参数	说明
服务器地址	设置需要连接的服务器地址，支持域名和 IP
端口号	设置连接的服务器端口，范围为 1~65535
重连间隔	当 MQTT 连接掉线时，网关支持自动重连，这里可以配置掉线后重连的时间间隔
客户端 ID	配置 MQTT 客户端 ID
用户名	配置 MQTT 客户端用户名
密码	配置 MQTT 客户端密码
保活时间	用户配置 MQTT 的保活时间，网关会根据保活时间周期发送 MQTT 心跳包
清除会话	开启则 MQTT 客户端掉线期间 MQTT 服务器将不保留该客户端订阅的主题数据，关闭则保留订阅的主题数据
上线主题	配置 MQTT 上线主题
上线消息	MQTT 客户端成功连接服务器后会向上线主题发布该消息
上线服务质量	配置上线主题的 Qos
遗嘱主题	配置 MQTT 遗嘱主题
遗嘱信息	MQTT 客户端出现掉线的情况则服务器向遗嘱主题发布该消息，订阅了该主题的客户端可以获得客户端的掉线通知
遗嘱服务质量	配置遗嘱主题的 Qos

3.5.2 订阅主题

如图 3.8 所示，用户可配置 8 路串口的订阅主题和服务质量。



图 3.8 MQTT 订阅主题

3.5.3 发布主题

如图 3.9 所示，用户可配置 8 路串口的发布主题、服务质量与保留主题。



图 3.9 MQTT 发布主题

4. 边缘计算

网关边缘计算支持周期轮询 Modbus 从机、字节序自动转换、寄存器数值组合运算、自定义数据格式、变化上报和阈值上报等功能，在网关侧即可完成数据的简单处理。

如图 4.1 所示，简单介绍网关数据上报流程，首先 Modbus 主机采集从机数据，如果该数据点是普通数据点（没有使能自定义数据格式）则需要先经过大小端转换，如果使能了自定义数据格式则对数据先进行表达式运算，然后根据配置的上报模式判断是否需要上报，最后网关会把数据打包成 json 格式上报给服务器。

下面介绍网关数据下发流程，首先网关接收到 MQTT 报文，然后进行解析，生成 Modbus 请求，最后由 Modbus 主机去查询或设置从机寄存器。



图 4.1 边缘计算功能示意图

下面详细介绍边缘计算页面的配置。

4.1 配置导入与导出

往往待轮询的 Modbus 从机及寄存器配置可能会很多，并且可能需要将这些配置批量在其他网关使用，因而可以使用导入及导出功能。

特别需要注意：首行中的“导入配置”与“边缘计算->导入配置文件”的作用不一致，首行的“导入配置”是导入导出设备整体配置文件，若需要导入导出边缘计算配置文件，则需使用“边缘计算->导入配置文件/导出配置文件”。

如图 4.2 所示，用户可以将当前的数据点配置导出为文件保存，也可以选择从外部导入数据点文件。

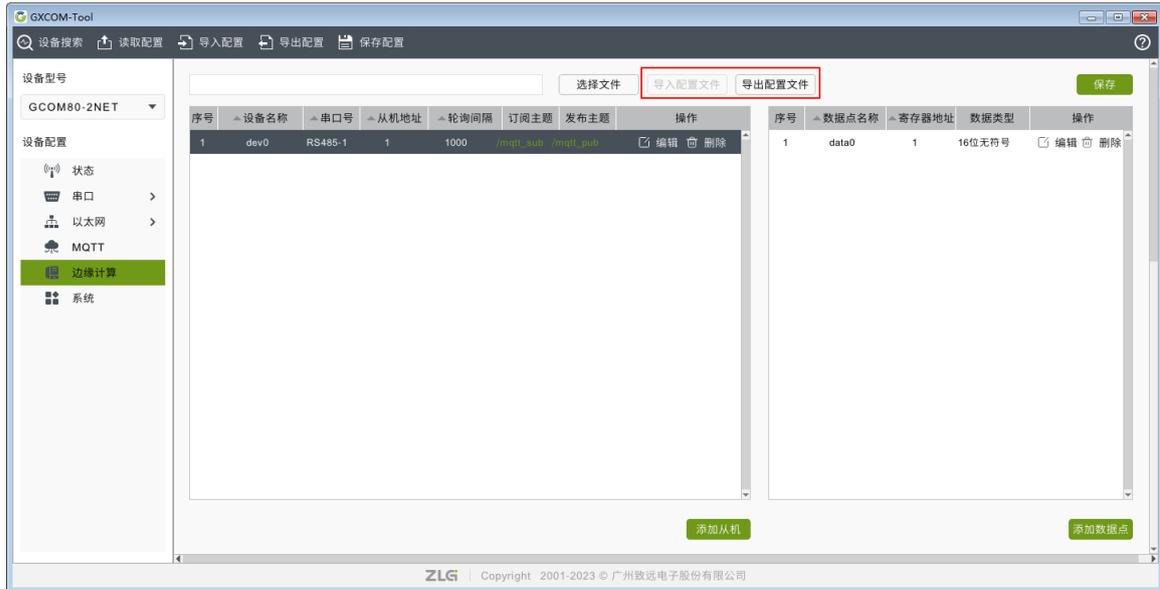


图 4.2 边缘计算配置导入与导出

4.2 添加从机和数据点

4.2.1 添加从机

点击“添加从机”按钮，可以看到如图 4.3 弹窗，用户需要设置设备名称，配置该设备通信的串口号，modbus 从机地址，轮询的间隔以及大小端。

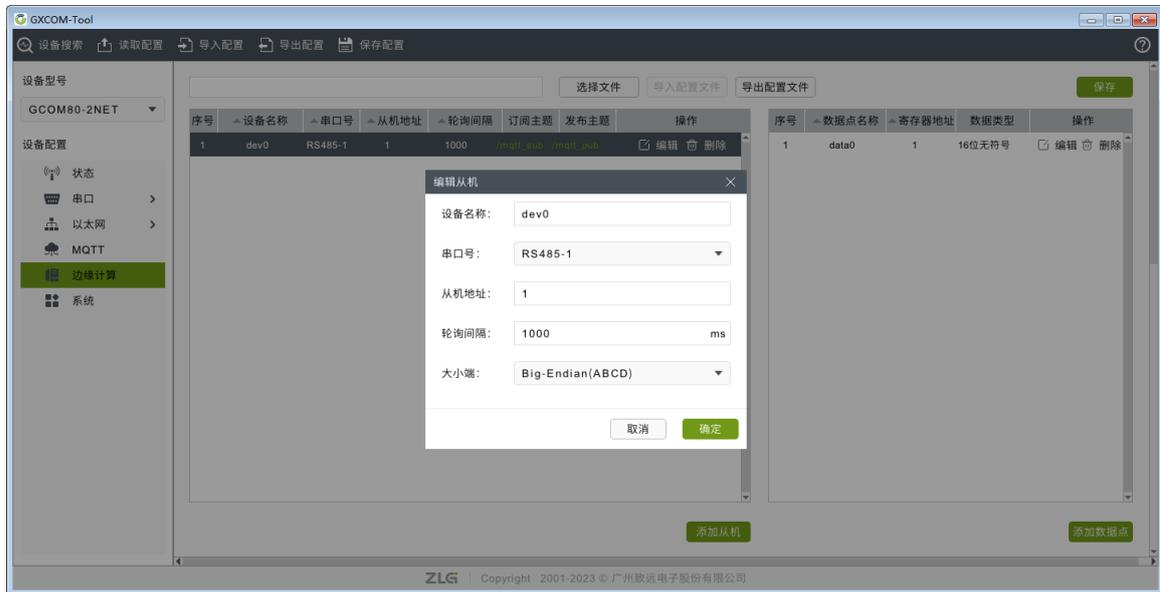


图 4.3 添加从机

4.2.2 添加数据点

如图 4.4 所示鼠标点击选中左边窗口中设备，然后点击右下角“添加数据点”按钮，用户需要输入数据点名称，读写的 Modbus 功能码，寄存器地址和数据类型。编辑完成后点击右上角的保存。

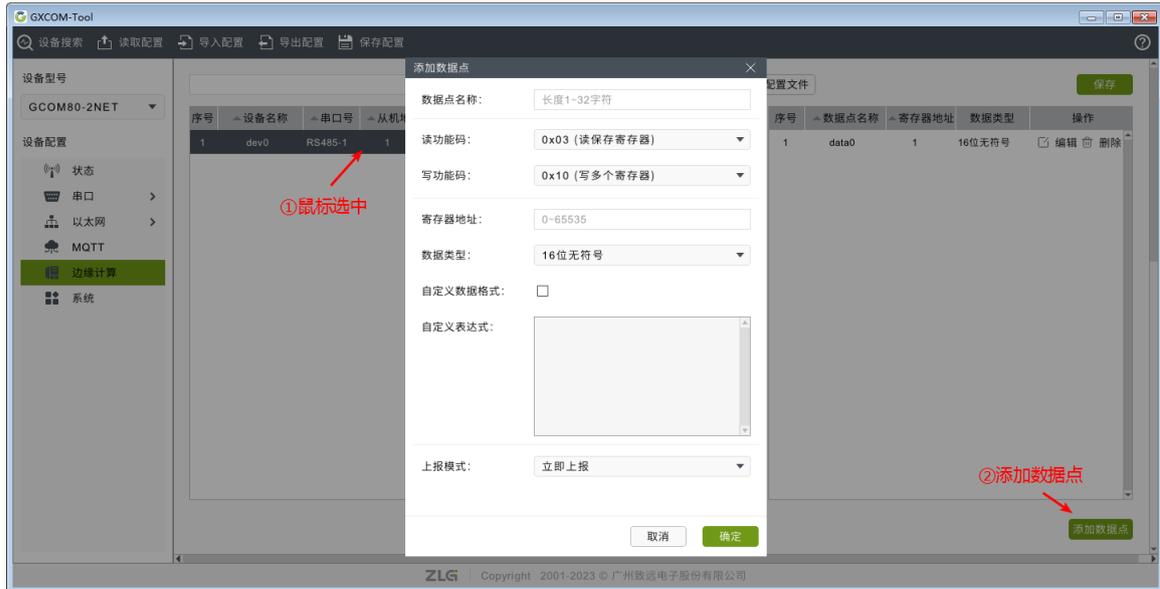


图 4.4 添加数据点

如图 4.5 所示，网关支持中文的设备名和数据点名。



图 4.5 中文的从机名和数据名

4.2.3 数据报文格式

如图 4.6 所示为从机和数据点的配置，对应实际上报的报文格式如图 4.7 所示。

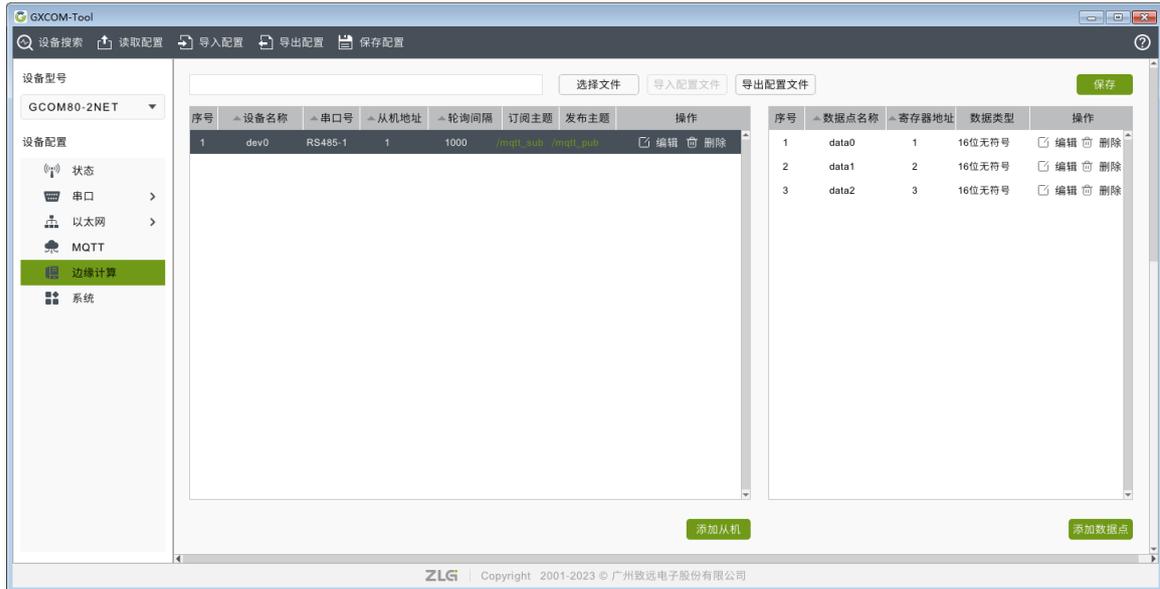


图 4.6 从机和数据点示例

```
{
  "ver" : "1.0.0",
  "gwid" : "gw0014970f007c00144aae",
  "time" : "1970-1-1 9:39:15",
  "seq" : "23.1.1",
  "dev0" : {
    "data0" : 2,
    "data1" : 0,
    "data2" : 0
  }
}
```

图 4.7 报文格式

报文字段说明如表 4.1 所示：

表 4.1 报文字段说明

字段名	说明
ver	报文版本号，主要用于报文格式的版本管理，目前版本为 1.0.0
gwid	网关 ID
time	报文的上报时间
seq	当前串口报文序号，格式为“报文序号.当前报文分包序号.当前报文的分包总数”，目前 1.0.0 版本没有分包，所以格式都固定为“x.1.1”
dev0	从机名
data0、data1、data2	数据点名，每个数据点名后面为数据点的值

4.3 自定义数据格式

网关除了支持普通的 modbus 寄存器上报，还支持单个或多个寄存器组合运算，如图 4.8 所示，用户可以把“自定义数据格式”选项勾选上，然后在下面的输入框输入“自定义表达式”。

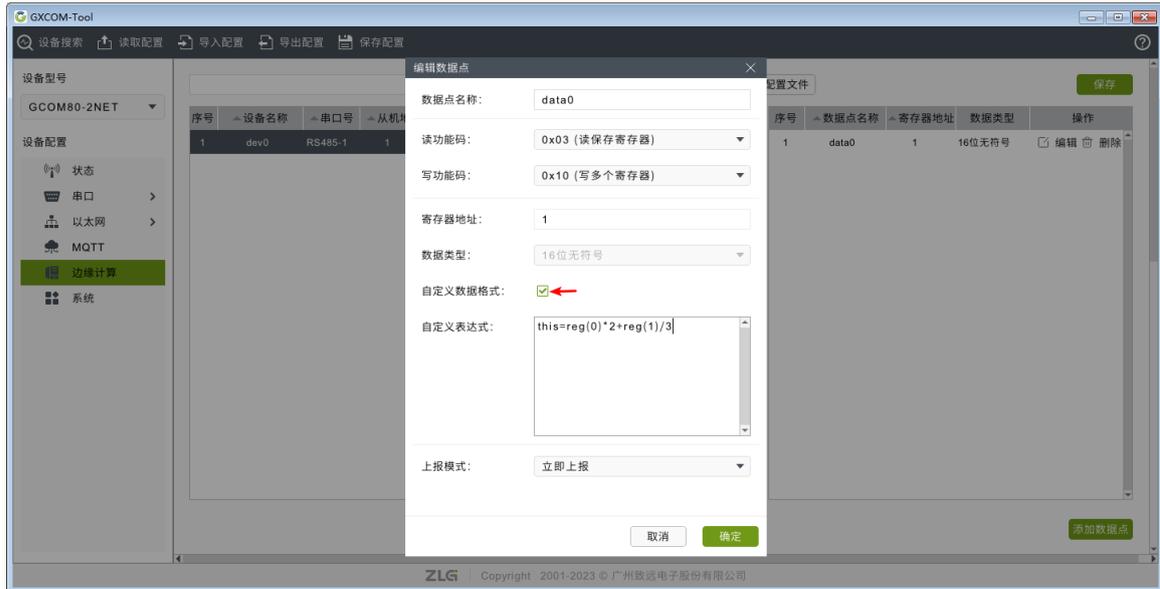


图 4.8 自定义数据格式

自定义表达式最终计算结果都必须通过 `this` 变量返回。自定义表达式有 `reg` 和 `reg_batch` 两个函数支持读取 Modbus 寄存器，`reg` 函数用于读取单个寄存器的值，`reg_batch` 函数用于读取多个寄存器的值，两个函数读取的寄存器类型由上面的“读功能码”类型决定。

1. reg 函数说明

声明：

```
uint16_t reg(uint16_t addr)
```

参数：

`addr` 为寄存器地址，支持十进制和十六进制，范围为 0~65535

返回值：

寄存器值

实例：

```
this=reg(100)
```

```
this=reg(0x26)
```

2. reg_batch 函数说明

声明：

```
char* reg_batch(uint16_t addr, uint16_t size, char *endian, char *type)
```

参数：

`addr` 为寄存器地址，支持十进制和十六进制，范围为 0~65535；

`size` 为寄存器个数，范围为 1~100；

`endian` 为寄存器数值的大小端顺序，支持“big”、“little”、“big_swap”和“little_swap”，分别是“大端”、“小端”、“大端交换”和“小端交换”；

`type` 为数值类型，函数读取寄存器值之后会根据数值类型做转换，支持“u16”、“i16”、“u32”、“i32”和“float”；如果是“u32”、“i32”和“float”类型，函数会将 2 个寄存器拼接在一起，所以当 `size` 为 100 个，最终函数返回的数值只有 50 个。

返回值：

寄存器值组合字符串，每个寄存器值用逗号分隔，例如：111,222,333,444

实例：

```
this=reg_batch(0, 50, "big", "u16")
this=reg_batch(0, 40, "little", "u32")
```

更多自定义表达式规则和格式如下：

4.3.1 基本用法

1. 读单寄存器

表 4.2 读单寄存器用法说明

用法	说明
this=reg(0)	表示读取地址为 0 的寄存器值
this=reg(1)	表示读取地址为 1 的寄存器值
this=reg(0x12)	表示读取地址为 0x12 的寄存器值

2. 读多寄存器

表 4.3 读多寄存器用法说明

用法	说明
this=reg_batch(0, 10, "big", "u16")	表示以大端的方式读取地址 0~9 的 10 个寄存器的值并转换成 10 个无符号 16 位数值
this=reg_batch(0, 10, "little", "u32")	表示以小端的方式读取地址 0~9 的 10 个寄存器的值并转换成 5 个无符号 32 位数值
this=reg_batch(0, 2, "big", "float")	表示以大端的方式读取地址为 0 和 1 的 2 个寄存器值并转换成浮点型数值

3. 加减乘除运算

表 4.4 加减乘除运算用法说明

用法	说明
this=reg(0)+reg(1)	表示读取地址为 0 和 1 的寄存器值然后相加
this=reg(0)-reg(1)	表示读取地址为 0 和 1 的寄存器值然后相减
this=reg(0)*reg(1)	表示读取地址为 0 和 1 的寄存器值然后相乘
this=reg(0)/reg(1)	表示读取地址为 0 和 1 的寄存器值然后相除
this=reg(0)*2+reg(1)/3	表示读取地址为 0 的寄存器值然后乘以系数 2，然后读取地址为 1 的寄存器值除以系数 3，然后将前后两部分相加

4. 数据类型转换

表 4.5 数据类型转换用法说明

用法	说明
this=i8(value)	表示将变量 value 的值转成 int8 类型
this=u8(value)	表示将变量 value 的值转成 uint8 类型
this=i16(value)	表示将变量 value 的值转成 int16 类型
this=u16(value)	表示将变量 value 的值转成 uint16 类型

<code>this=i32(value)</code>	表示将变量 <code>value</code> 的值转成 <code>int32</code> 类型
<code>this=u32(value)</code>	表示将变量 <code>value</code> 的值转成 <code>uint32</code> 类型

注：reg 函数返回值默认为 `uint16` 类型，如果需要其它类型数值请通过上面的方式转换。

实例 1:

```
value=reg(12);
this=i16(value); //将 uint16_t 转为 int16_t 类型
```

实例 2:

```
value0=reg(10);
value1=reg(11);
this=u32(value1)<<16 | value0; //reg 返回值默认为 uint16_t 类型，为了使左移不丢失高位数据需要将其转为 uint32_t 类型
```

4.3.2 高级用法

1. 临时变量

脚本支持变量，不需要定义，直接使用。变量名必须以字母或下划线开头，后面可用数字、英文或下划线。

```
value0=reg(0)*2;
value1=reg(1)+3;
this=value0-value1;
```

2. 条件判断

```
if ((reg(1) > 80) && (reg(2) == 10)) {
    this="异常";
} else {
    this="正常";
}
```

3. 循环

```
this=reg(0);
for (i = 0; i < 10; i = i + 1) {
    this=this+1;
}
```

4. 寄存器组合

```
val=reg(0)+'';
val=val+ reg(1)+'';
val=val+ reg(2)+'';
this=val;
```

上报的报文格式如：10,23,45。

注：当“数字+字符串”的情况下“+”将不再是数学运算符，而是字符串拼接，数字会自动转成字符串，类似 python 的用法。网关只支持同一个式子连续最多 24 个“+”、“-”、“*”和“/”号。

5. 数据处理

表 4.6 数据处理用法说明

用法	说明
<code>this=abs(i16(reg(0)))</code>	表示读取地址为 0 的寄存器值然后将它转换成 int16 类型再取它的绝对值
<code>this=max(reg(0), reg(1))</code>	表示分别读取地址为 0 和 1 的寄存器值然后取它们 2 个中的最大值
<code>this=min(reg(0), reg(1))</code>	表示分别读取地址为 0 和 1 的寄存器值然后取它们 2 个中的最小值
<code>this=round(3.5)</code>	表示对 3.5 做四舍五入运算，结果为 4
<code>this=floor(3.2)</code>	表示对 3.2 向下取整，结果为 3
<code>this=ceil(3.2)</code>	表示对 3.2 向上取整，结果为 4
<code>this=random(1, 100)</code>	表示返回 1 至 100 之间的随机数

实例 1:

```
value=reg_batch(0, 2, "big", "float");//读取一个浮点数
this= round(value);
```

实例 2:

```
value=reg_batch(0, 2, "big", "float");//读取一个浮点数
this=ceil(value);
```

实例 3:

```
value=reg(0);
rnd=random(1, 100);
this=value*rnd;
```

6. 数学运算

表 4.7 数学运算用法说明

用法	说明
<code>this=pow(reg(0), n)</code>	表示读取地址为 0 的寄存器值然后计算它的 n 次方
<code>this=sqrt(reg(0))</code>	表示读取地址为 0 的寄存器值然后计算它的开平方
<code>this=sin(0.5)</code>	表示计算 0.5 的正弦值
<code>this=cos(0.5)</code>	表示计算 0.5 的余弦值
<code>this=tan(0.5)</code>	表示计算 0.5 的正切值

实例 1:

```
value=reg(0);
this=pow(value, 2);//计算 value 的平方
```

实例 2:

```
value=reg_batch(0, 2, "big", "float");//读取一个浮点数
this=sin(value);
```

7. 位操作

用法	说明
----	----

<code>this=(reg(1) >> 8) & 0xFF</code>	表示读取地址为 1 的寄存器值然后取高 8 位的值
<code>this=(reg(1) >> 4) & 0x01</code>	表示读取地址为 1 的寄存器值然后取第 4 位的值
<code>this=reg(1)<<8</code>	表示读取地址为 1 的寄存器值然后左移 8 位，这种方式会导致高 8 位丢失；如 0x1234，左移 8 位后结果为 0x3400。
<code>this=u32(reg(1))<<8</code>	表示读取地址为 1 的寄存器值然后强转为 u32 类型再左移 8 位，这种方式高 8 位会保留，如 0x1234，左移 8 位后结果为 0x123400。
<code>this=reg(1) 0xC1</code>	表示读取地址为 1 的寄存器值然后或上 0xC1

注：reg 函数返回值默认为 uint16 类型，左移会导致高位丢失，需要通过上面的“3.数据类型转换”为更大的数值类型再做移位操作。

4.3.3 注意事项

下面列举不支持的自定义数据格式：

- `reg(1)`: 缺少 `this` 变量；
- `this=reg(1+2)`、`this=reg_batch(1+2, 10+3, "big", "u16")`: 不支持参数是运算表达式；
- `this=reg(reg(0))`、`this=reg_batch(reg_batch(0, 1, "big", "u16"), 1, "big", "u16")`: 不支持 `reg` 和 `reg_batch` 函数嵌套；
- `this=reg(addr)`、`this=reg_batch(addr, 1, "big", "u16")`: 不支持函数参数是变量。

4.4 上报模式

4.4.1 立即上报

当数据点模式设置为立即上报，网关采集数据点数据后立即上报。

4.4.2 变化上报

变化上报就是当前数据点值和上次数据点值比较，如果超过变化范围则上报，如果不超过变化范围则不上报。

如图 4.9 所示，当数据点模式设置为变化上报，用户还需要设置变化范围。

注：如果是自定义数据点 `this` 是字符串的情况，那么变化范围只能填 0。

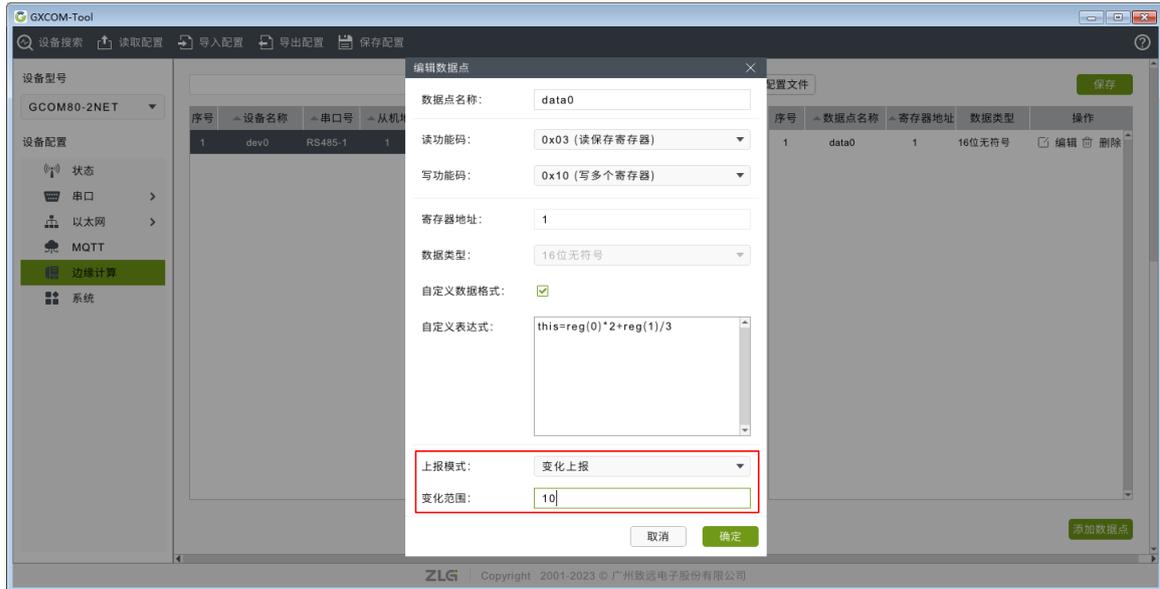


图 4.9 变化上报

4.4.3 阈值上报

阈值上报即：网关判断数据点值如果处于条件范围内则上报，如果在条件范围外则不上报。条件表达式判断支持“大于”、“大于等于”、“等于”、“不等于”、“小于”和“小于等于”；还支持多个条件组合，如“或”或者“与”。

注：阈值上报模式不支持自定义数据点 this 是字符串的情况，这种情况网关将不会上报该数据点。

具体操作如图 4.10 所示，上报模式选择“阈值上报”，然后鼠标点击下面“阈值上报”输入框。

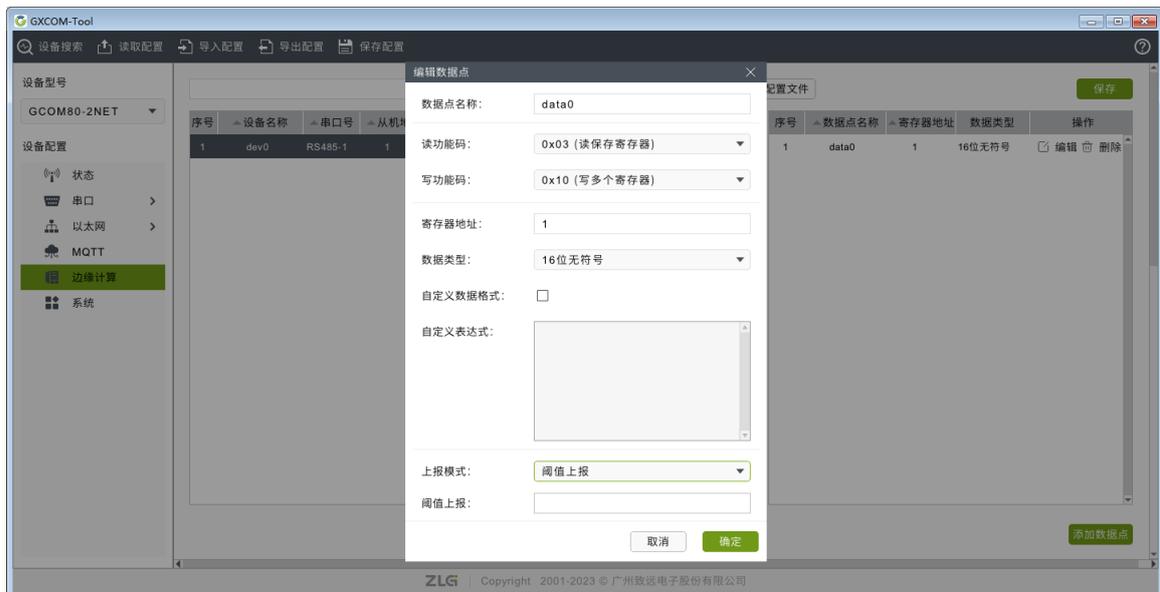


图 4.10 阈值上报配置

如图 4.11 所示会看到一个阈值表达式的输入窗口，举例如果用户需要输入一个“数据点值大于 1 并且小于 10”的条件，则可以按如图 4.11 所示的方式编辑。

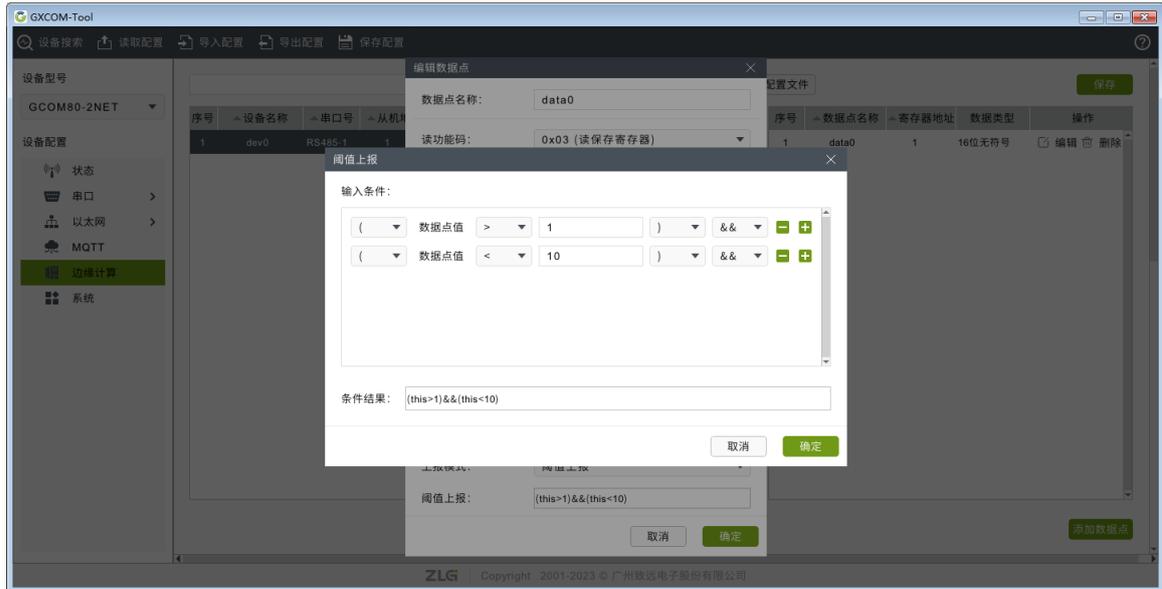


图 4.11 阈值上报表达式例 1

如果用户需要输入一个“数据点值小于 10 或大于 20”的条件，则可以按图 4.12 所示的方式编辑。

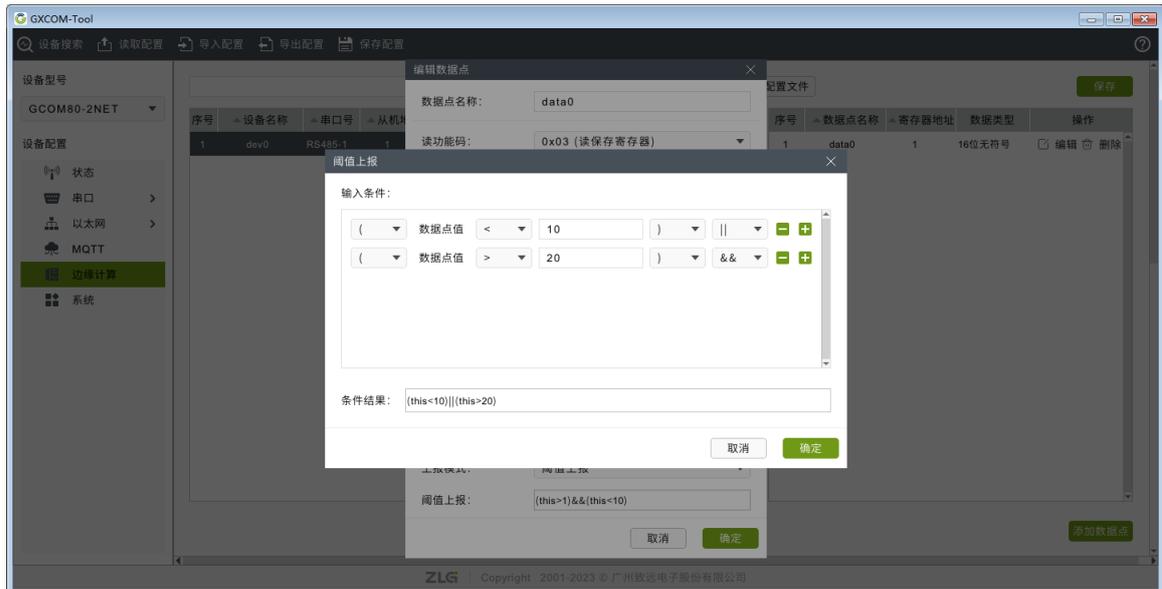


图 4.12 阈值上报表达式例 2

5. 远程控制

用户可以通过 MQTT 向网关发送远程控制报文实现查询数据点值和设置数据点值。

5.1 查询数据点

用户可以下发查询数据点报文，网关收到后会返回查询的数据点值，字段说明如表 5.1 所示，如程序清单 5.1 所示为查询数据点报文格式。

表 5.1 查询字段说明

字段名	说明
ver	报文版本，主要为了版本兼容，目前用户直接填 1.0.0 就可以了
gwid	目标网关 ID，网关 ID 可以在上位机的系统信息页面查看。也可以使用广播 ID “gwffffffffffffffff800” 来查询所有网关
cmd	操作命令，查询数据点的命令为 “data_point_get”
cmd_param	命令参数，这里表示分别读取从机 dev0 的 data0、data1、data2 和从机 dev1 的 data3、data4、data5 三个数据点

程序清单 5.1 查询数据点

```
{
  "ver": "1.0.0",
  "gwid": "gw0014970f006a001438c0",
  "cmd": "data_point_get",
  "cmd_param": {
    "dev0": ["data0", "data1", "data2"],
    "dev1": ["data3", "data4", "data5"]
  }
}
```

如果下发查询报文后没有收到网关的上报，请参考 FAQ 章节的“下发远程控制报文无反应”排查问题。

5.2 设置数据点

用户可以下发设置数据点报文（只支持非自定义数据格式的数据点），网关收到后会修改数据点的寄存器值，字段说明如表 5.2 所示，如程序清单 5.2 所示为设置数据点报文格式。

表 5.2 字段说明

字段名	说明
ver	报文版本，主要为了版本兼容，目前用户直接填 1.0.0 就可以了
gwid	目标网关 ID，网关 ID 可以在上位机的系统信息页面查看。也可以使用广播 ID “gwffffffffffffffff800” 来配置所有网关
cmd	操作命令，设置数据点的命令为 “data_point_set”
cmd_param	命令参数，这里表示分别设置从机 dev0 的 data0 为 1、data1 为 2、data2 为 3 和从机 dev1 的 data3 为 3、data4 为 4、data5 为 5 三个数据点。如果同一个从机里有多个相

同的数据点那么网关只会设置第一个数据点的值

程序清单 5.2 设置数据点

```
{
  "ver" : "1.0.0",
  "gwid" : "gw0014970f006a001438c0",
  "cmd" : "data_point_set",
  "cmd_param":{
    "dev0" : {"data0":1, "data1":2, "data2":3},
    "dev1" : {"data3":3, "data4":4, "data5":5}
  }
}
```

如果下发设置数据点报文后没有成功修改从机的寄存器，请参考 FAQ 章节的“下发远程控制报文无反应”排查问题。

6. 系统配置

6.1 系统信息

如图 6.1 所示，切换到“系统”->“系统信息”标签页即可查看到系统相关信息。系统信息主要有设备型号、设备 ID、系统时间、系统运行时长、固件版本和 boot 版本信息内容。



图 6.1 系统信息

6.2 升级

如图 6.2 所示，用户可以通过上位机对网关固件进行升级，点击“浏览”按钮可以选择要升级的固件，选择完成后点击“升级”按钮开始升级，升级过程约 1~2 分钟。



图 6.2 固件升级

6.3 重启与恢复

如图 6.3 所示，点击重启可以对网关进行重启，点击恢复出厂则会将网关当前配置清除恢复到出厂时的状态。



图 6.3 重启与恢复

6.4 修改账户密码

如图 6.4 所示，用户可以在此页面修改网关的登录密码。

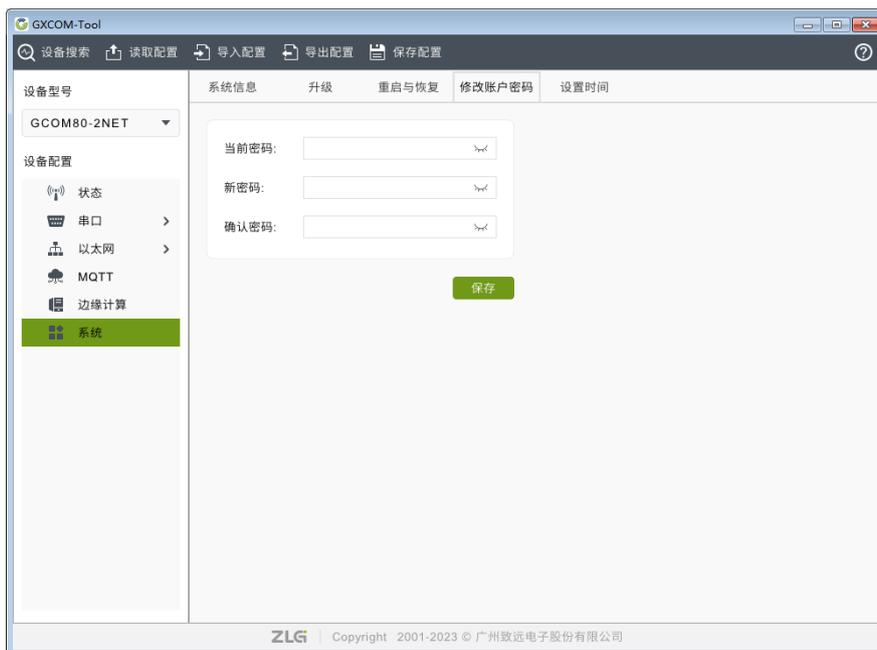


图 6.4 修改账户密码

6.5 设置时间

如图 6.5 所示，网关支持从网络自动同步时间也支持手动配置。当开启自动同步时，网关会自动从网络获取时间，此功能要求网关能连接外网。



图 6.5 设置时间

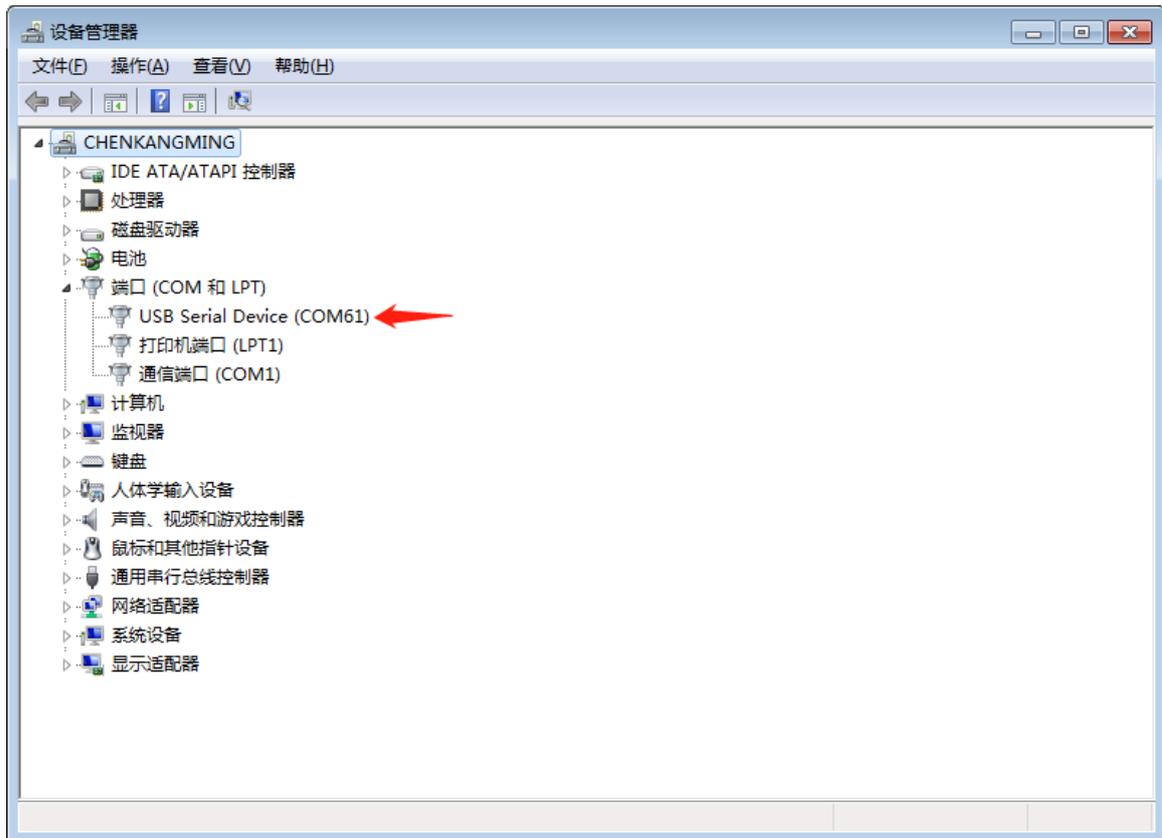
7. FAQ

1. 网关启动后只有 sys 灯闪烁，其它指示灯不亮

网关数据点数量较多，网关需要时间处理。如果网关恢复出厂设置，网关在删除数据点文件的时候也需要一定时间。网关在处理数据点时上位机无法配置网关，当听到网关蜂鸣器鸣叫则表示网关已处理完成。

2. 上位机搜索不到设备

查看电脑的设备管理器，看电脑是否正确识别到了网关的串口，如果没识别到或 USB 设备显示黄色感叹号就需要重新插拔一下 USB 口。



3. 下发远程控制报文无反应

此处的无反应是指下发查询数据点报文网关没有上报要查询的数据点，下发设置数据点报文没有成功修改从机的寄存器。出现该问题可能的原因如下：

- 下发的 json 报文格式是否正确，是否有漏写括号或逗号等符号；
- 下发报文的 gwid 是否和目标网关的 id 一致；
- 从机名是否正确；
- 数据点名是否正确；
- MQTT 订阅主题是否正确。

8. 免责声明

本着为用户提供更好服务的原则，广州致远电子股份有限公司（下称“致远电子”）在本手册中将尽可能地向用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，致远电子不能完全保证该文档在任何时段的时效性与适用性。致远电子有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问致远电子官方网站或者与致远电子工作人员联系。感谢您的包容与支持！

诚信共赢，持续学习，客户为先，专业专注，只做第一

广州致远电子股份有限公司

更多详情请访问
www.zlg.cn

欢迎拨打全国服务热线
400-888-4005

