

积木式边缘网关 USR-M100

边缘计算应用案例



联网找有人，靠谱

可信赖的智慧工业物联网伙伴

目录

| | |
|-----------------------|----|
| 1. 产品简介 | 3 |
| 2. 边缘计算功能 | 3 |
| 3. 边缘计算应用案例 | 3 |
| 3.1. 环境准备 | 3 |
| 3.2. 边缘采集应用 | 4 |
| 3.2.1. 设备配置 | 4 |
| 3.2.2. 从机配置 | 7 |
| 3.2.3. 主机配置 | 8 |
| 3.2.4. 数据验证 | 9 |
| 3.3. 边缘计算 | 10 |
| 3.3.1. 设备配置 | 10 |
| 3.3.2. 数据验证 | 11 |
| 3.4. 主动上报 | 11 |
| 3.4.1. IO 点位添加 | 11 |
| 3.4.2. 数据上报配置 | 13 |
| 3.4.3. 网络调试助手配置 | 14 |
| 3.4.4. 从机配置 | 14 |
| 3.4.5. 周期上报测试 | 15 |
| 3.4.6. 定时上报测试 | 17 |
| 3.4.7. 变化上报 | 18 |
| 3.5. 点位批量导入导出 | 19 |
| 3.5.1. 导出 | 20 |
| 3.5.2. 编辑 | 20 |
| 3.5.3. 导入 | 20 |

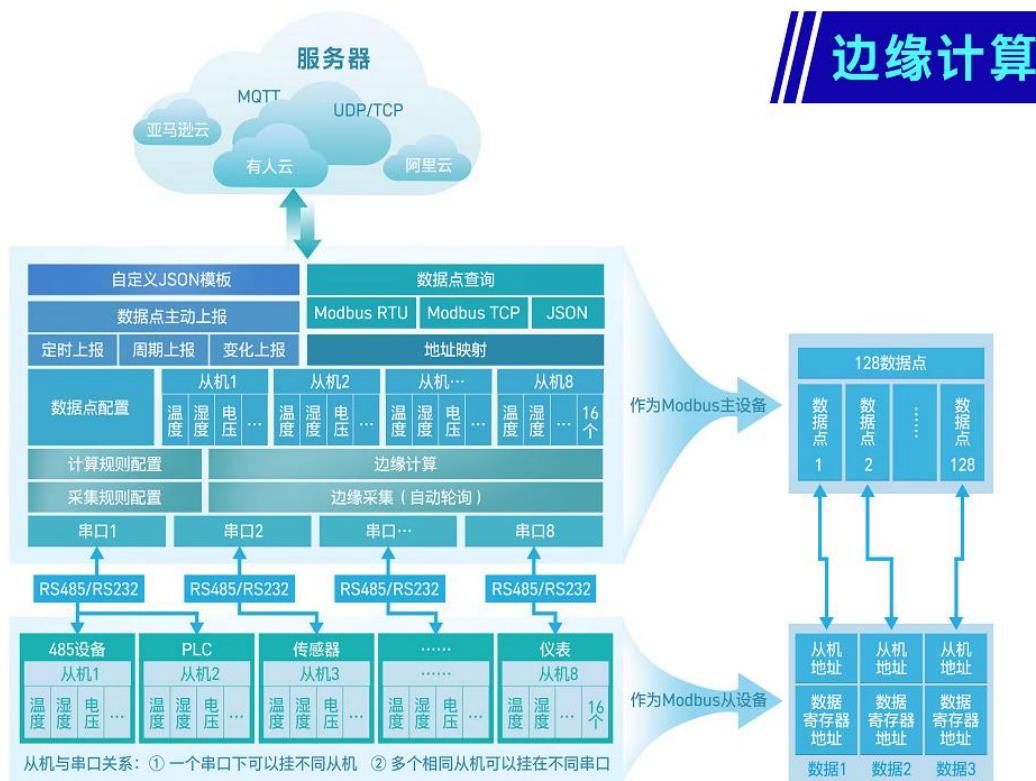
1. 产品简介

USR-M100 是一款高性价比综合性可拓展数传网关，集成了边缘采集和计算，IO 采集和控制，点位联动，点位读写，数据透传，快速上云和数据加密等多种核心功能。产品采用 Cortex-M7 内核，主频高达 400Mhz；工业级设计，数据传输稳定，同时支持网口和 LTE Cat1 网络，网络通信方式更加多样化。产品支持 TCP/UDP/MQTT(S)/HTTP(S)等协议通信,支持 modbusRTU/TCP 协议转换和 modbus/Json 协议数据上报。产品更是嵌入了有人云，阿里云和 AWS 等常用平台的接入，方便客户和平台的对接。产品在硬件上集成了 2 路 DI，2 路 DO 和 2 路 AI，不仅能实现工业现场控制和采集的需求，还能实现根据各种采集点数据或状态进行联动控制。可以广泛应用在智慧养殖，智慧工厂等多种工业智能化方案中。

产品在结构上采用可拓展设计，未来用户使用过程中，如果发现当前设备存在部分功能无法满足需求时，可以根据缺少的功能选择满足相应的拓展面板进行拼接即可实现功能的满足，无需再去购买整机，节省成本的同时还能方便客户接入当前现场环境。

2. 边缘计算功能

边缘网关包括数据采集、数据计算、数据上报和数据读写等功能，支持 Modbus RTU 转 Json，Modbus RTU 转 Modbus TCP 等通用工业协议转换。边缘计算网关功能中，网关设备作为轮询的发起方，自主地、周期性地从终端设备读取、解析、计算用户预置的数据信息，同时网关设备可以根据用户预设的上报逻辑，有选择性地数据进行上报，无需远端服务器主动参与，即可完成数据采集任务，极大地提升了云端服务器资源利用率，可以接入更多的设备进行管理。



3. 边缘计算应用案例

3.1. 环境准备

- (1) USR-M100 一台
- (2) USB 转 RS485 串口线一根

- (3) 网线一根
- (4) 12V/1A 电源适配器一个
- (5) Modbus Slave 软件：模拟 Modbus 从机设备。
- (6) Modbus Poll 软件：模拟云端主动采集
- (7) 网络调试助手

3.2. 边缘采集应用

本例主要介绍边缘采集功能的应用，引导大家使用 Modbus Slave 软件模拟串口 Modbus 设备，从机地址为 01，40001 寄存器为 16 位温度数据待采集，40002 寄存器为 16 位湿度数据待采集，并将该从机数据点信息添加到 M100 中，实现串口的边缘采集。

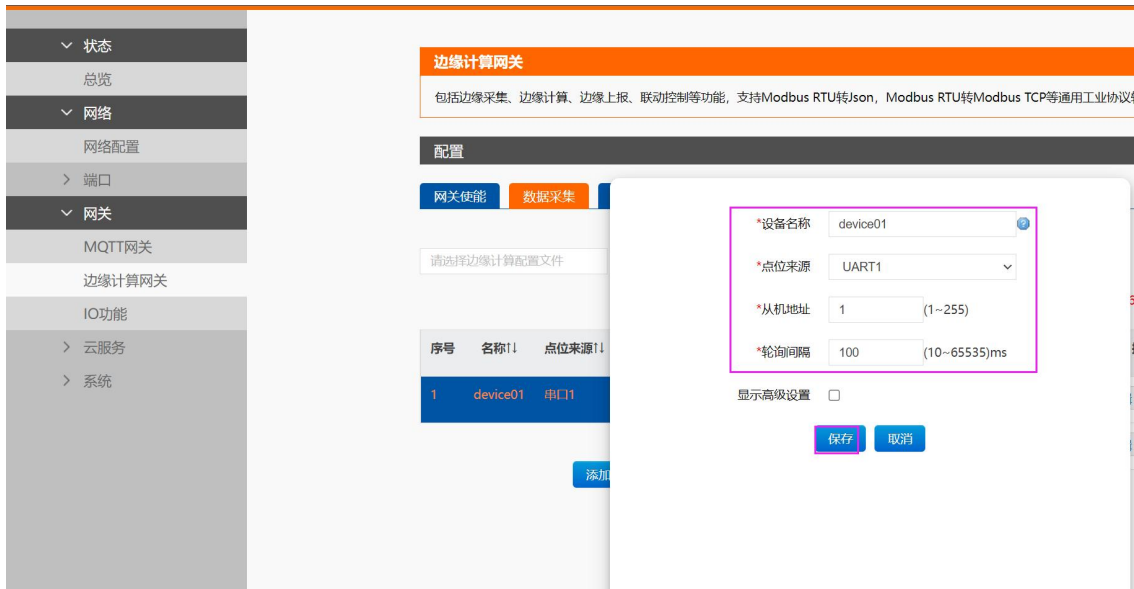
3.2.1. 设备配置

- (1) 进入设备的配置界面，选择“网关”下面的“边缘计算网关”
- (2) 在“网关使能”配置项中将“网关使能”开启
- (3) 点击“下一步”跳转到“数据采集”



- (4) 在“数据采集”配置页选中默认从机，点击“编辑”
- (5) 根据实际待采集的从机设备配置参数，本例中配置串口号 1，从机地址 1，轮询间隔 100
- (6) 点击“保存”该从机。





(7) 配置数据点位参数，本应用直接编辑点位 1 和点位 2 即可，如果需要更多点位可以直接点击添加。

(8) 点位 1 设置数据点名称 node0101，功能码选择 03，寄存器地址设置 0，数据类型选择“16 位无符号”，超时时间设置 200

(9) 点击“保存”，保存该数据点配置。

(10) 点位 2 设置数据点名称 node0102，功能码选择 03，寄存器地址设置 1，数据类型选择“16 位无符号”，超时时间设置 200

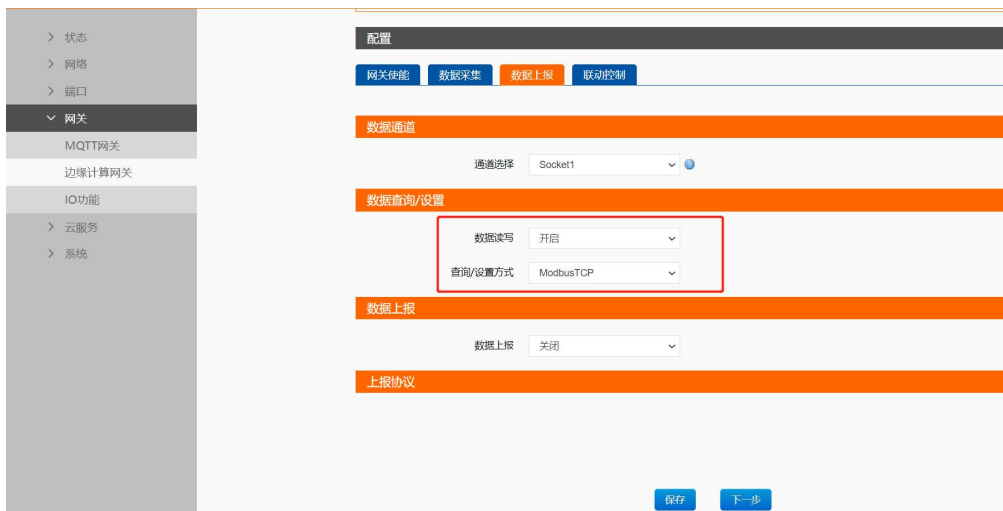
(11) 点击“保存”，保存该数据点配置。



(12) 点位参数配置完成后，点击保存按钮，然后选择继续配置。配置数据上报。

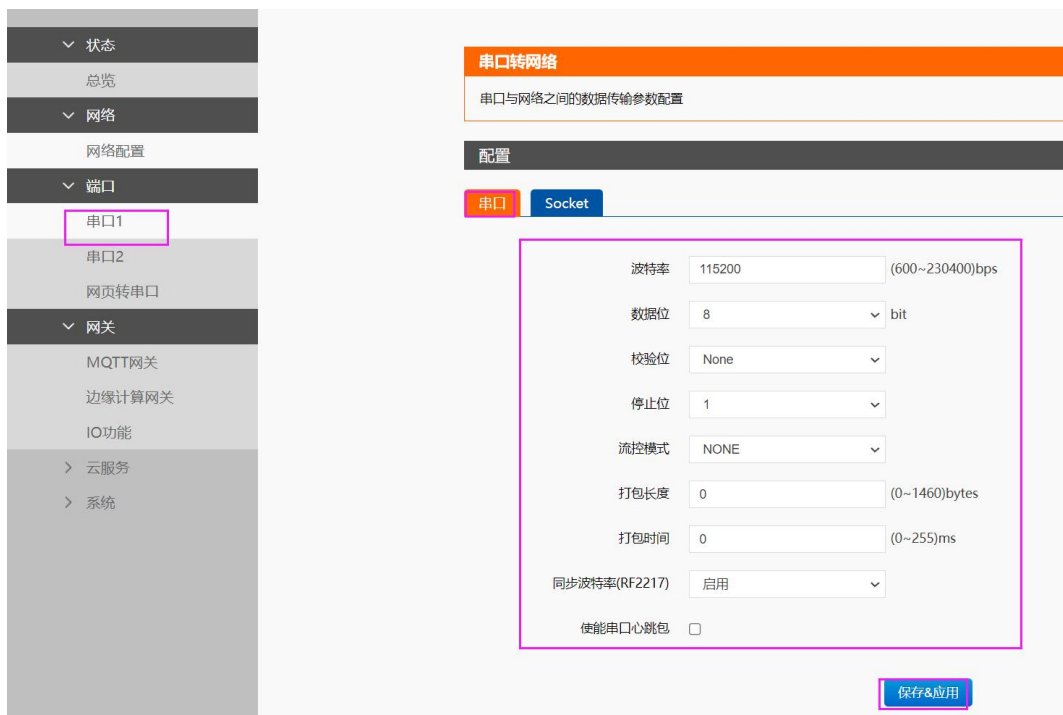
(13) 通道选择 socket1，数据读写开启并选择 Modbus TCP，数据上报关闭。

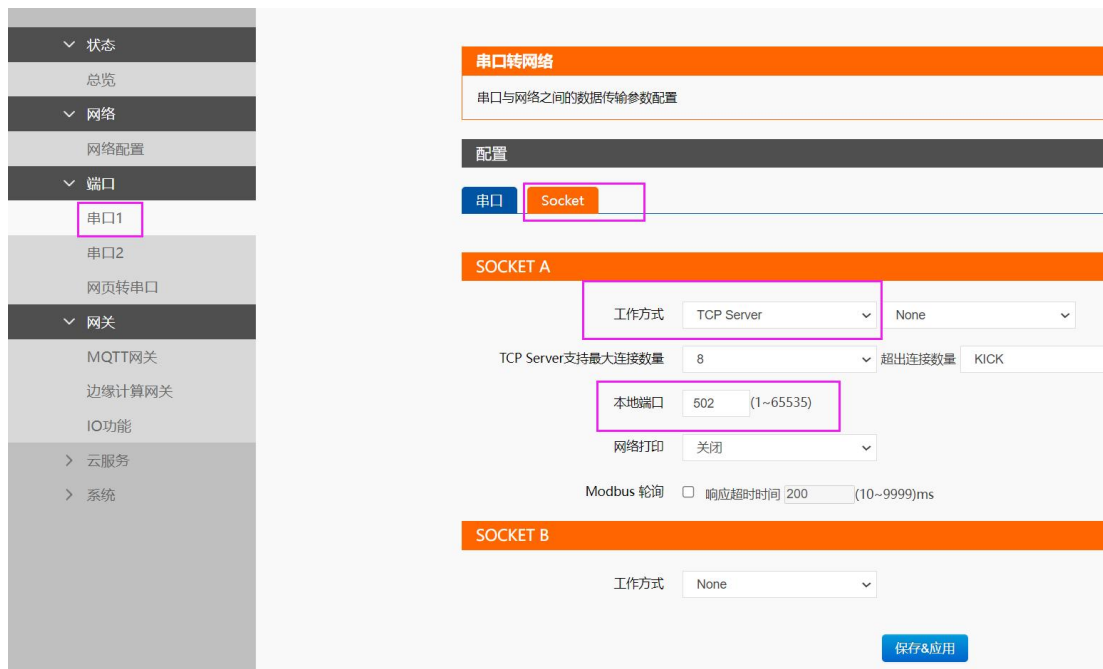
(14) 配置完成后，点击保存并选择继续配置。



(15) 配置串口参数，根据实际选用的串口进行配置，本应用使用串口 1 并使用默认配置，所以在端口->串口 1 界面后，直接点击保存并选择继续配置。

(16) 切换到 socket 界面，配置 socketA 参数位 TCP Server，端口改为 502，保存并重启设备即可。

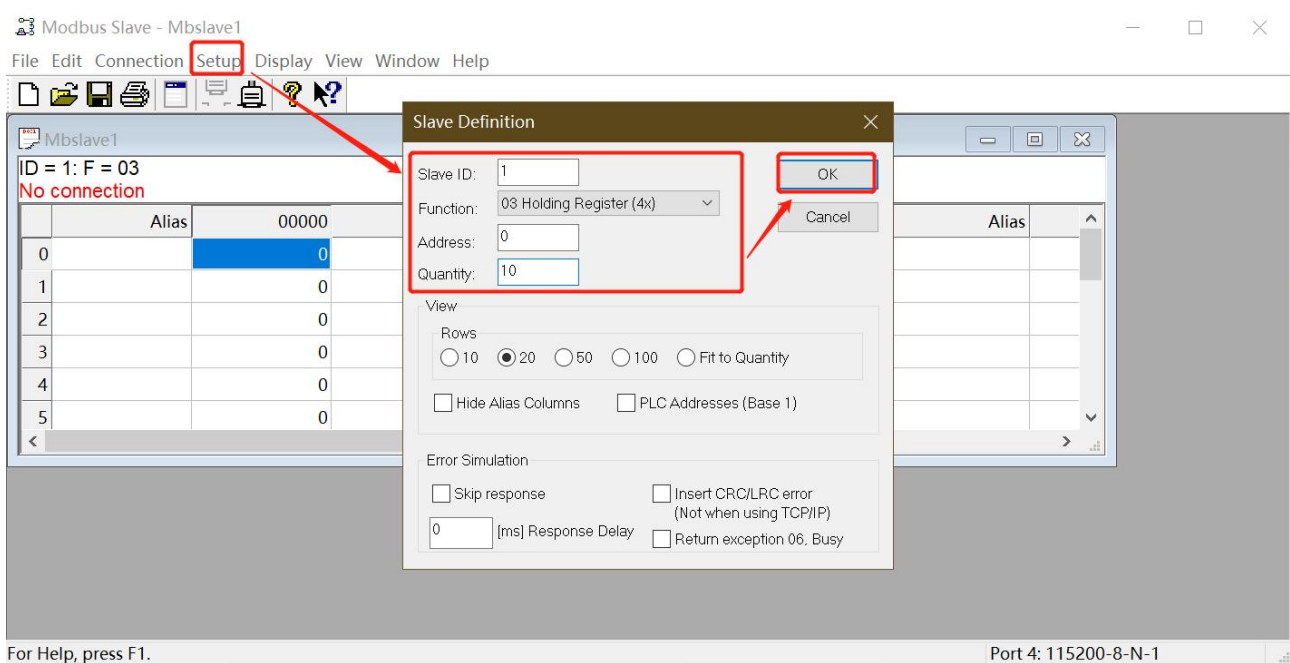




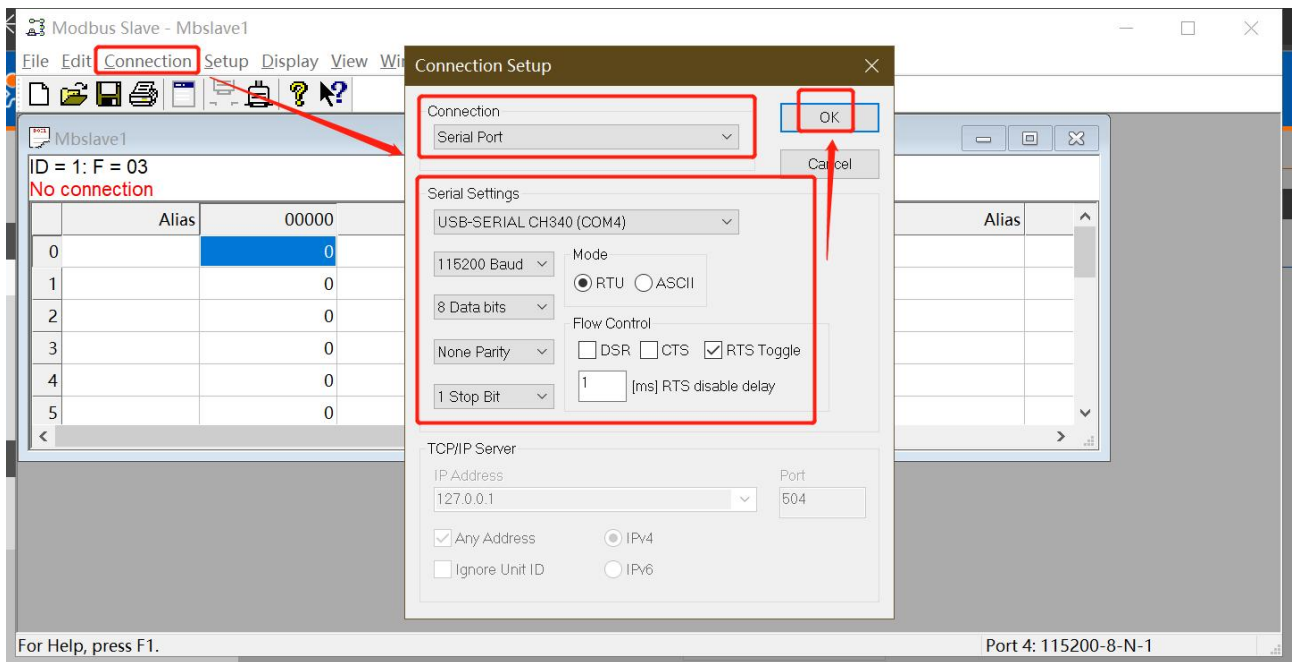
3.2.2. 从机配置

本应用采用 Modbus Slave 模拟从机设备，配置步骤如下：

- (1) “Setup->Slave Definition” 界面,从机地址 “Slave ID” 填 1，功能码 “Function” 选择 03，寄存器起始地址 “Address” 填 0，寄存器数量 “Quantity” 填 10 即可，配置完成后点击 “OK”

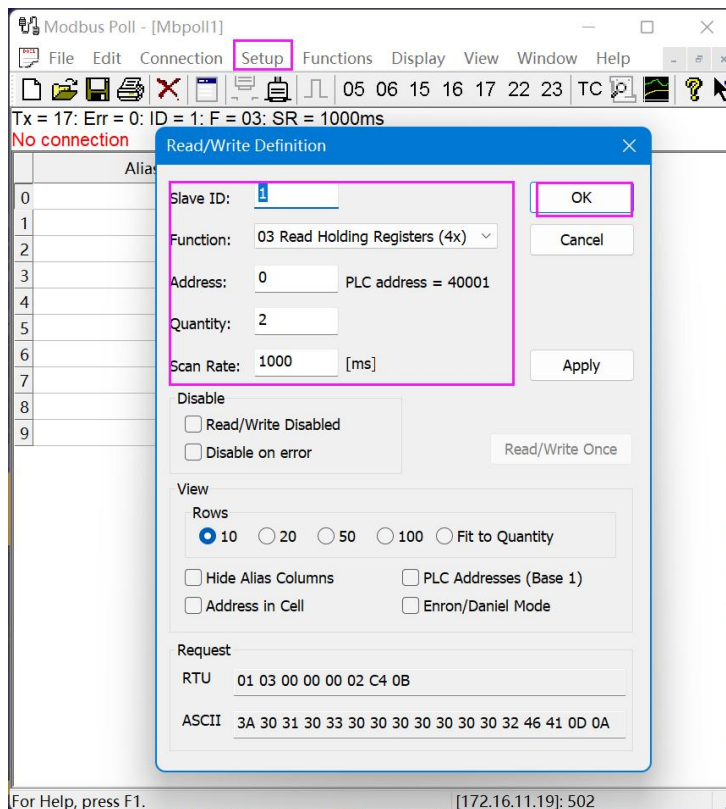


- (2) 在 “connection” -- “connect” 页面配置连接参数，“connection” 选择 “Serial Port”，“Serial Settings” 选择对应的串口（可在电脑设备管理器界面查看），波特率、数据位、校验位、停止位配置和 M100 的串口参数一致，“Mode” 选择 RTU，配置完成后点击 “OK”

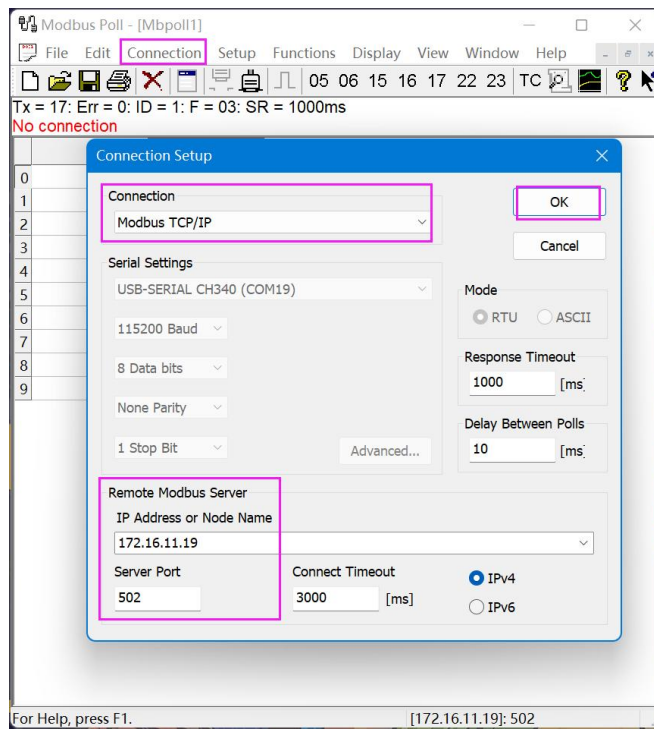


3.2.3. 主机配置

- (1) “Setup->Read/Write Definition” 界面,从机地址 “Slave ID” 填 1, 功能码 “Function” 选择 03, 寄存器起始地址 “Address” 填 0, 寄存器数量 “Quantity” 填 2 即可, 配置完成后点击 “OK”

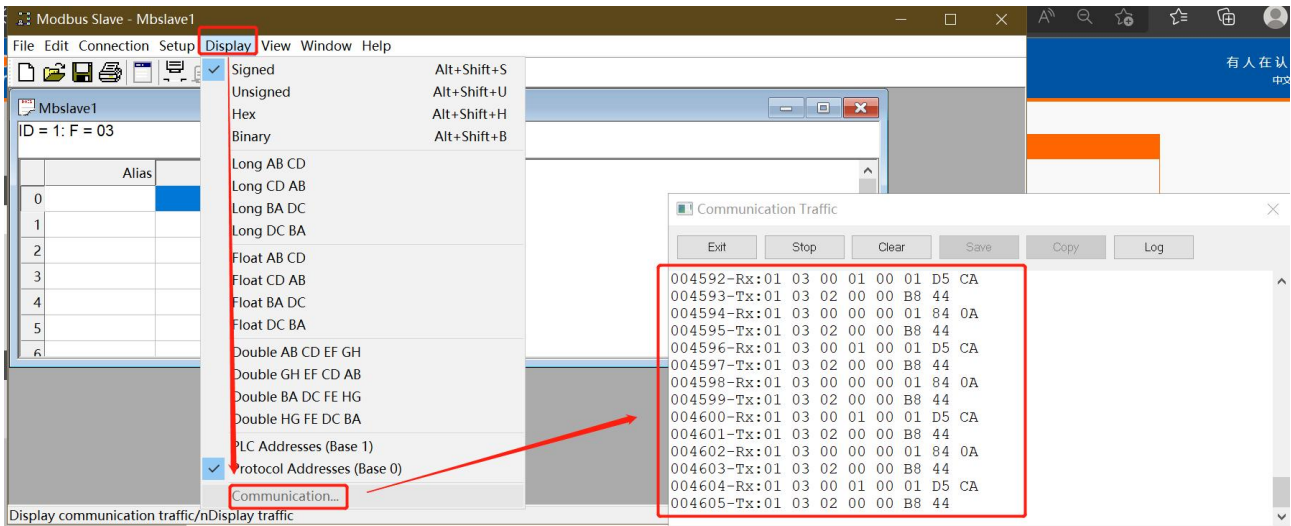


- (2) 在 “connection” -- “connect” 页面配置连接参数, “connection” 选择 “Modbus TCP/IP”, serverIP 填入设备的本机地址, 端口填入 502, 配置完成后点击 “OK”。

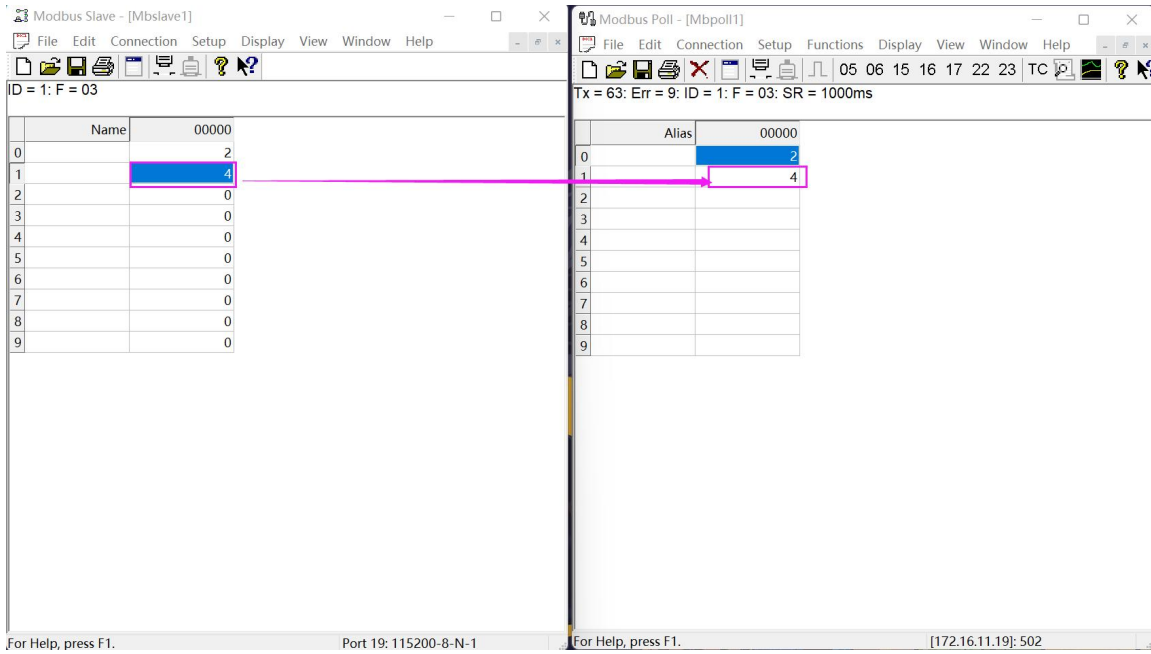


3.2.4. 数据验证

设备和 Modbus 工具配置好后，会看到设备的数据指示灯频繁闪烁，表明设备串口有数据交互，然后通过 Modbus Salve 的调试窗口观察数据收发。设备获取的数据，会在设备内进行存储和更新。



通过 Modbus Poll 界面可以看到 Modbus Salve 的寄存器数据，并且修改 Modbus Salve 寄存器数据，Poll 相应的寄存器数据也会随之发生变化。

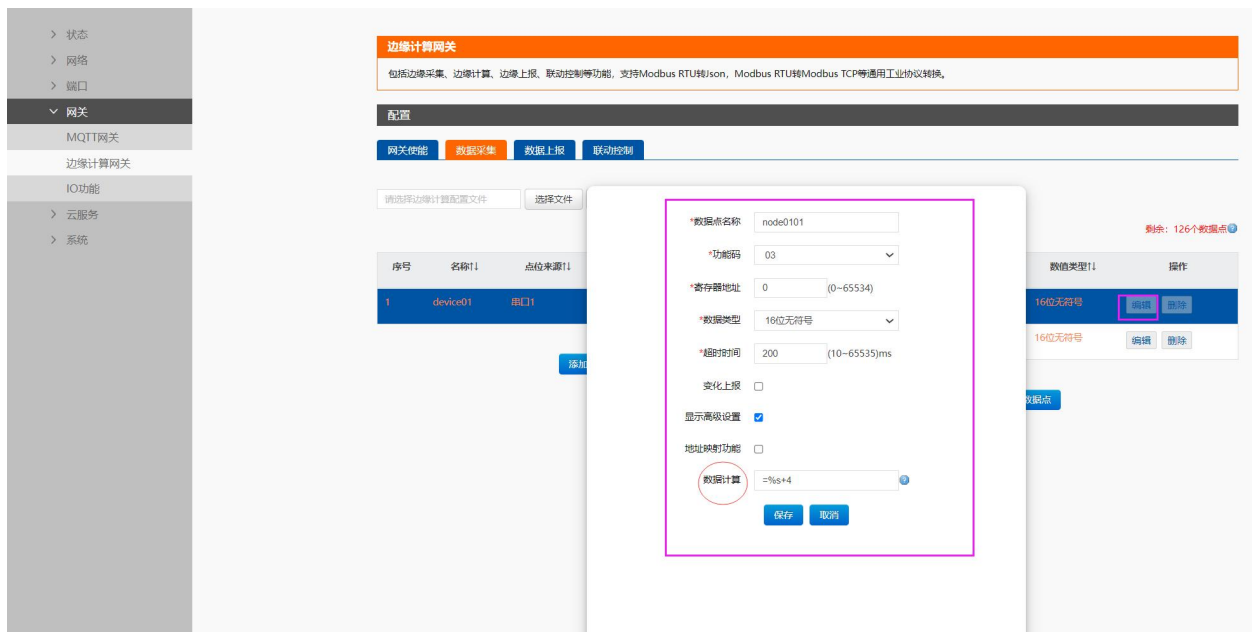


3.3. 边缘计算

本应用主要介绍边缘计算功能，引导大家使用 Modbus 工具进行功能验证。

3.3.1. 设备配置

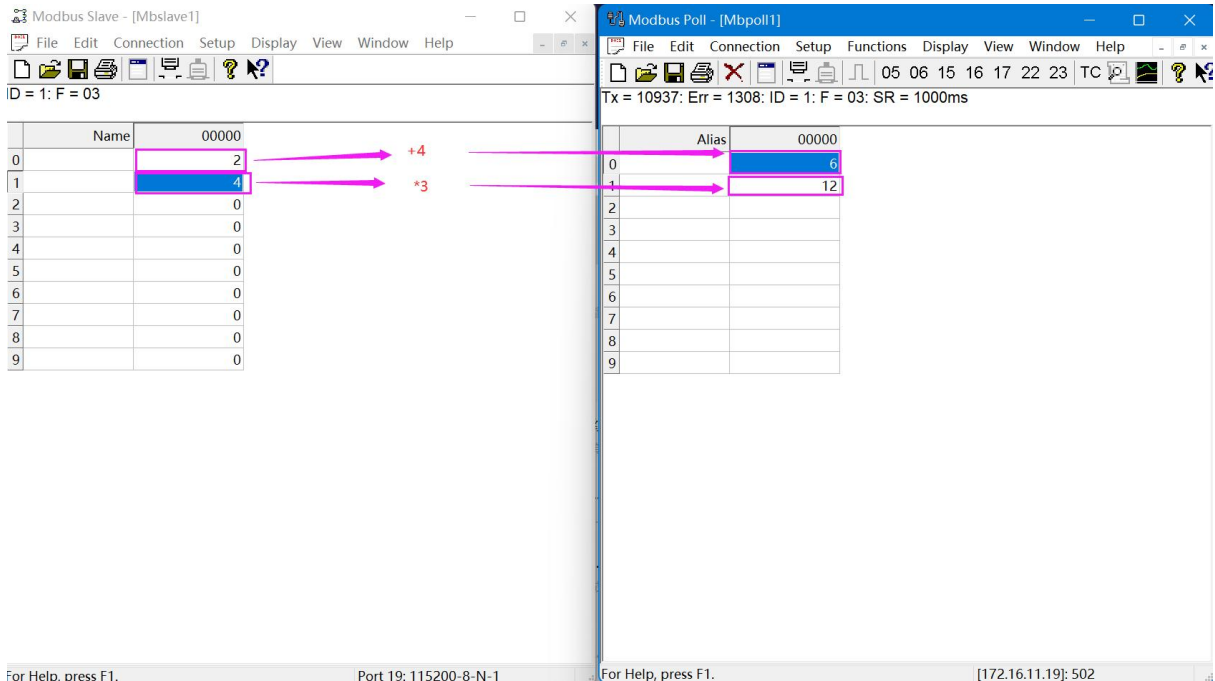
- 1、基于边缘采集的配置进行操作，配置数据点位信息，从机和点位添加，参考“边缘采集章节”，边缘计算需要在点位配置界面增加计算公式。点位 1 计算公式为： $=\%s+4$ ；点位 2 的计算公式为： $=\%s*3$
- 2、保存点位并重启设备。



3.3.2. 数据验证

参考边缘采集的主从机配置，然后进行验证即可。

通过 Modbus Poll 界面可以看到 Modbus Slave 的寄存器数据计算后的结果，并且修改 Modbus Slave 寄存器数据，Poll 相应的寄存器数据也会随之发生变化。



3.4. 主动上报

本应用主要结合边缘采集的配置和相应工具，测试数据主动上报功能，快速实现配置和应用。

3.4.1. IO 点位添加

M100 的支持 AI, DI 采集量进行主动上报，也支持 DO 状态主动上报。IO 的主动上报功能是以点位的形势结合边缘计算功能进行设计。所以 IO 的主动上报功能要开启边缘计算并设置好相应的 IO 从机和点位，开启边缘上报并配置相关参数。

- 1、**IO 从机添加**：在“边缘计算网关—>数据采集”界面添加 IO 从机，点位来源选择“IO”，从机地址直接代入。IO 从机地址可以在 IO 功能界面进行修改。



2、IO 点位配置：选中 IO 从机，在点位界面进行 IO 点位添加，IO 点位可以直接代入，下拉选择即可，DO 和 DI 可以选择变化上报，AI 可以在高级设置中增加添加计算公式，进行单位变换计算，本示例直接将 uA 转换为 mA，输入公式为=%s/100。



3.4.2. 虚拟数据点添加

虚拟数据点主要用于多点位间计算功能，当客户有多个点位需要合并计算的时候，可以创建一个虚拟点位，在虚拟点位中增加多点位计算公式，并将结果存储到虚拟点位中，以供客户主动上报和服务器的读取。

点位从机添加如下图，点位来源选择 VIRTUAL。



虚拟从机下的虚拟点位的添加，多点位计算公式中，需要计算的点位代入点位名称即可，依然支持加减乘除运算。例如两个点位进行求和处理，公式为：=%s+%s,node0101,node0102，其中 node0101,node0102 为两个点位的名称。

*数据点名称

*功能码

*寄存器地址 (0~65534)

*数据类型

*超时时间 (10~65535)ms

变化上报

显示高级设置

地址映射功能

数据计算

3.4.3. 数据上报配置

- 1、边缘采集配置，参考边缘采集应用进行配置并保存。
- 2、边缘计算功能可根据实际情况自行选择参考，本应用不做要求
- 3、主动上报配置，在设备内置网页中“网关->边缘计算网关->数据上报”界面进行配置
- 4、选择通道，本应用采用 TCP SOCKET1 进行通信。客户可以根据实际情况选择其他通道。
- 5、数据上报开启，上报方式可以选择周期上报，定时上报和变化上报三种，三种方式可以并行上报，互不影响。
- 6、上报协议配置，默认位 json 格式，设备支持 json 模板自定义，符合 json 格式即可。本应用采用 `{"Current": "node0101", "Voltage": "node0102", "DI1": "DI1"}`。
- 7、保存并选择继续配置。

配置

网关使能 数据采集 数据上报 联动控制

数据通道

通道选择

数据查询/设置

数据读写

数据上报

数据上报

周期上报

上报周期 (1~36000)s

定时上报 (需要先开启NTP)

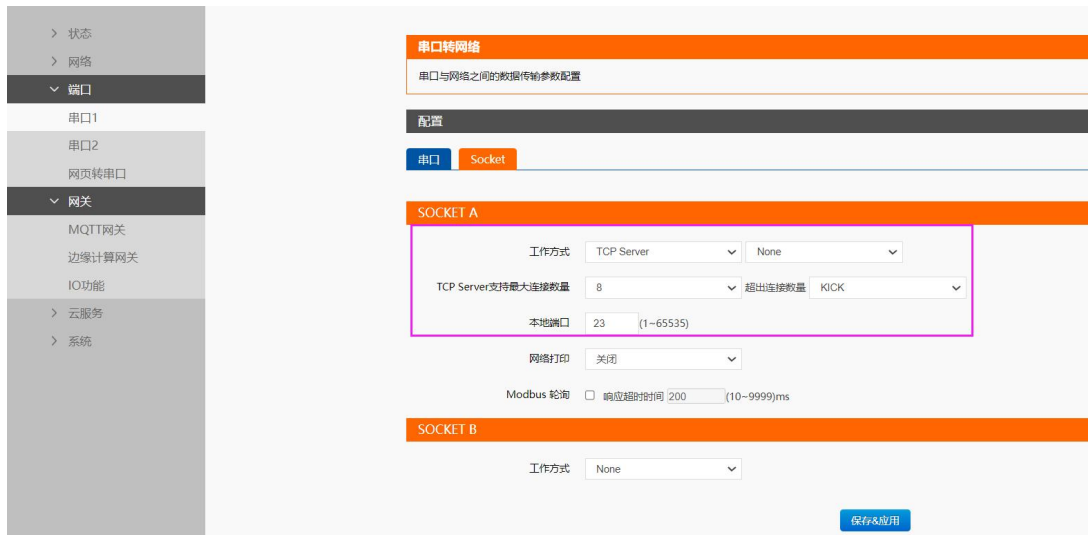
失败填充

引导包含

上报协议

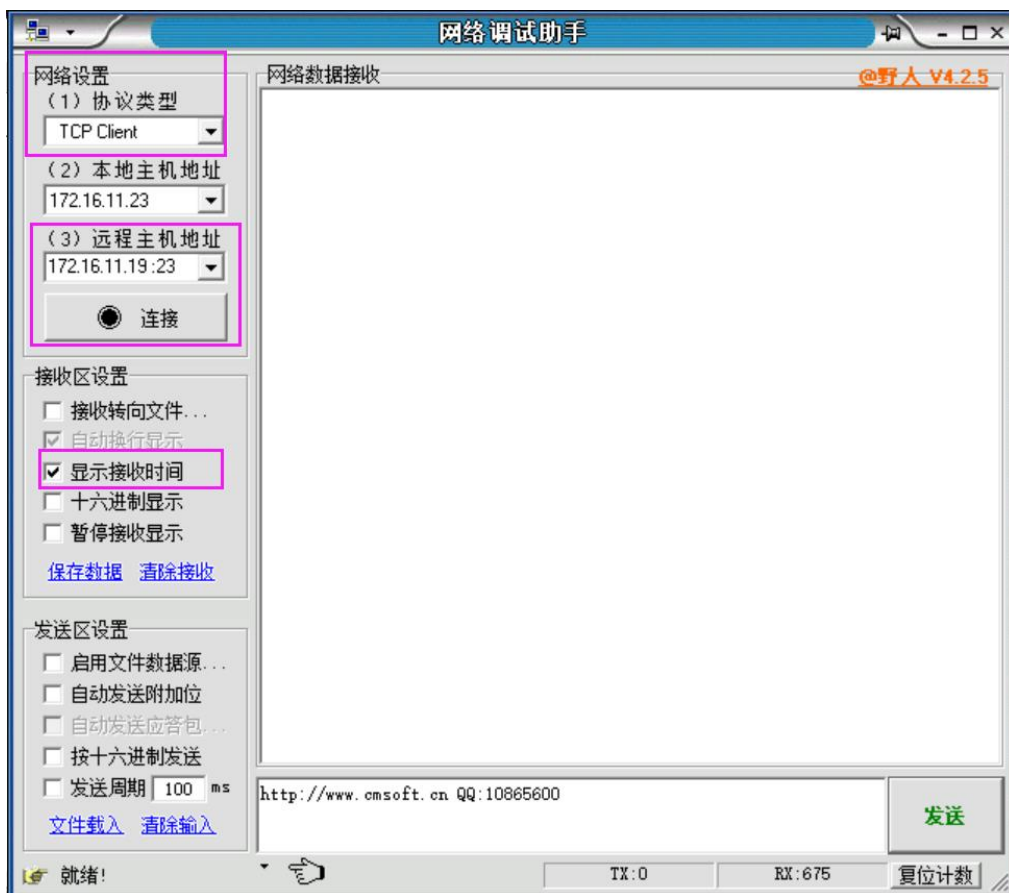
Json模板 (<2048 bytes) (55 byte)

8、串口和 socket 配置，本应用直接使用默认值，直接保存并重启设备即可。



3.4.4. 网络调试助手配置

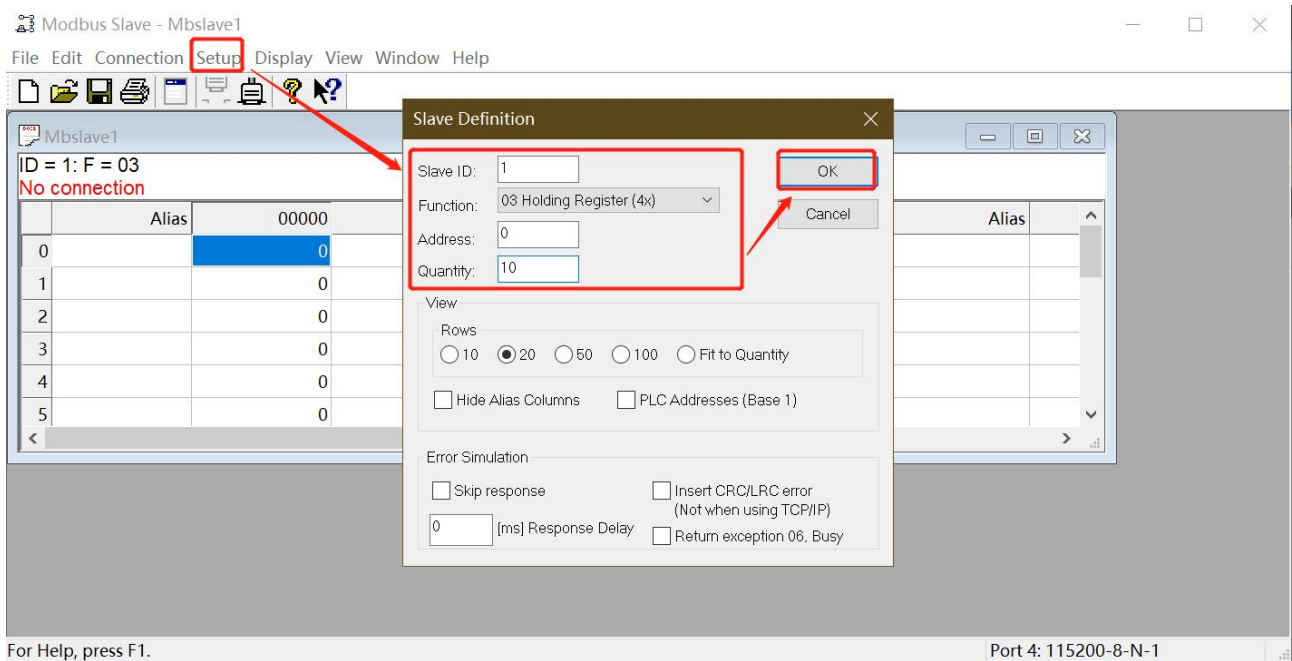
网络调试助手主要用于模拟服务器，接收上报的信息。协议类型选择 TCP Client，远程主机地址填写设备的本机地址和本地端口，接收区选项勾选“显示接收时间”，然后点击链接，网络助手配置完成，这样就可以接收设备主动上报的信息。



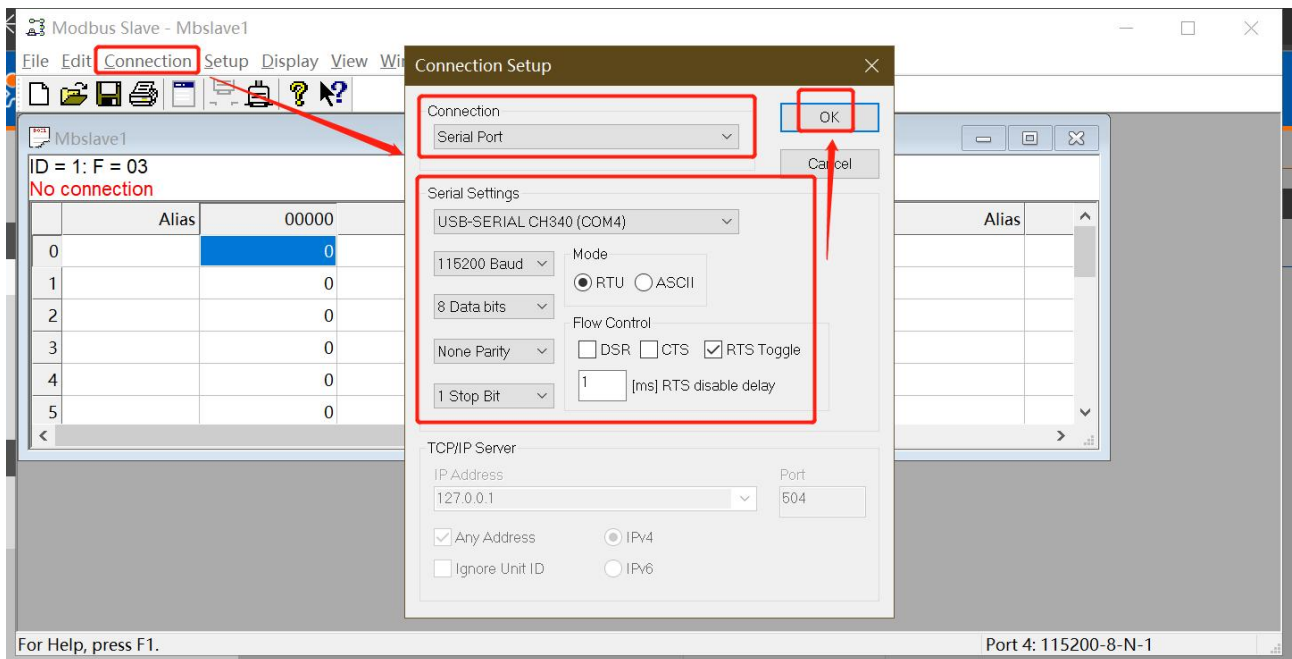
3.4.5. 从机配置

本应用采用 Modbus Slave 模拟从机设备，配置步骤如下：

- (3) “Setup->Slave Definition” 界面,从机地址 “Slave ID” 填 1, 功能码 “Function” 选择 03, 寄存器起始地址 “Address” 填 0, 寄存器数量 “Quantity” 填 10 即可, 配置完成后点击 “OK”



- (4) 在 “connection” -- “connect” 页面配置连接参数, “connection” 选择 “Serial Port”, “Serial Settings” 选择对应的串口 (可在电脑设备管理器界面查看), 波特率、数据位、校验位、停止位配置和 M100 的串口参数一致, “Mode” 选择 RTU, 配置完成后点击 “OK”



3.4.6. 周期上报测试

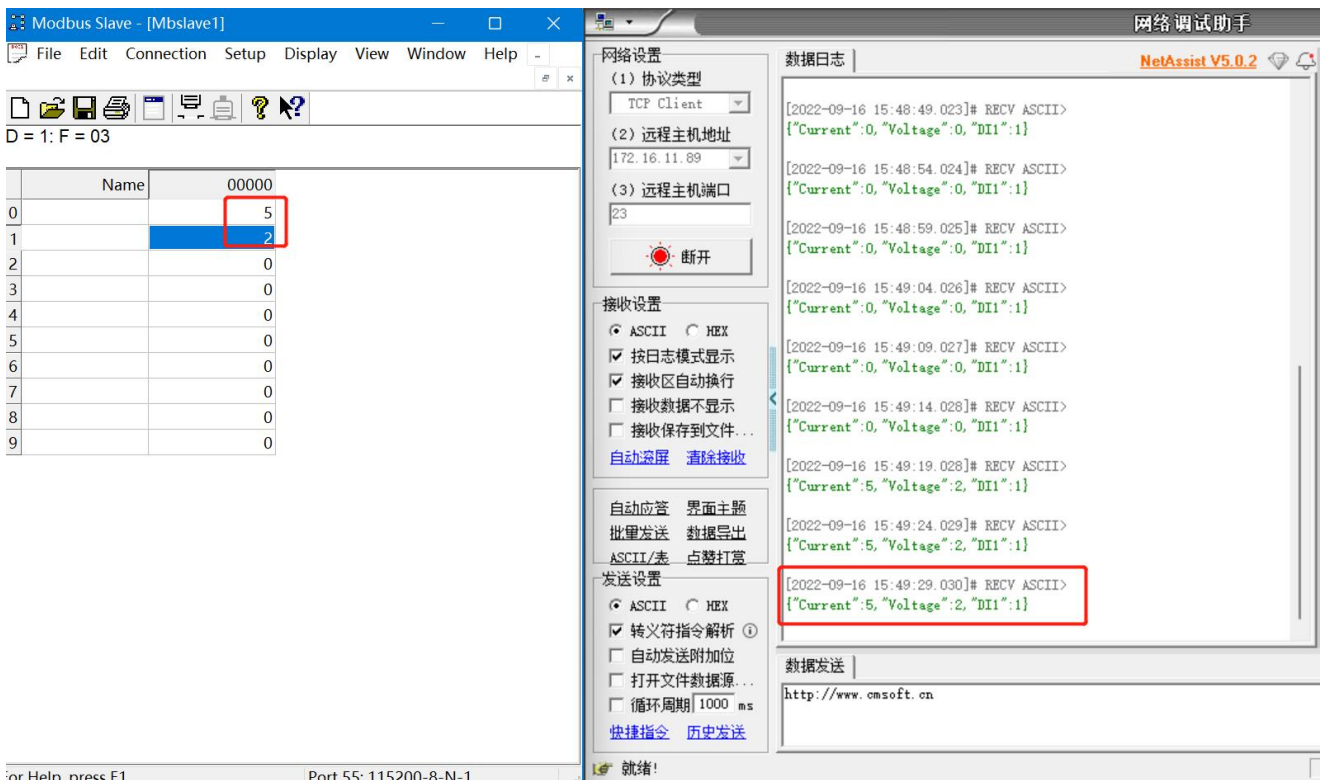
3.4.6.1. 周期上报配置

在数据上报界面, 使能数据上报功能后, 勾选周期上报, 并设置上报周期时间, 本应用默认使用 5s。保存参数并重启设备。



3.4.6.2. 周期上报数据测试

设备重启后，调试助手连接设备，通过改变 Modbus Slave 寄存器的数值，可以看到数据周期上报，并改为从机当前数据，同时 DI1 的状态变量也进行了主动上报展示。



3.4.7. 定时上报测试

3.4.7.1. 定时上报配置

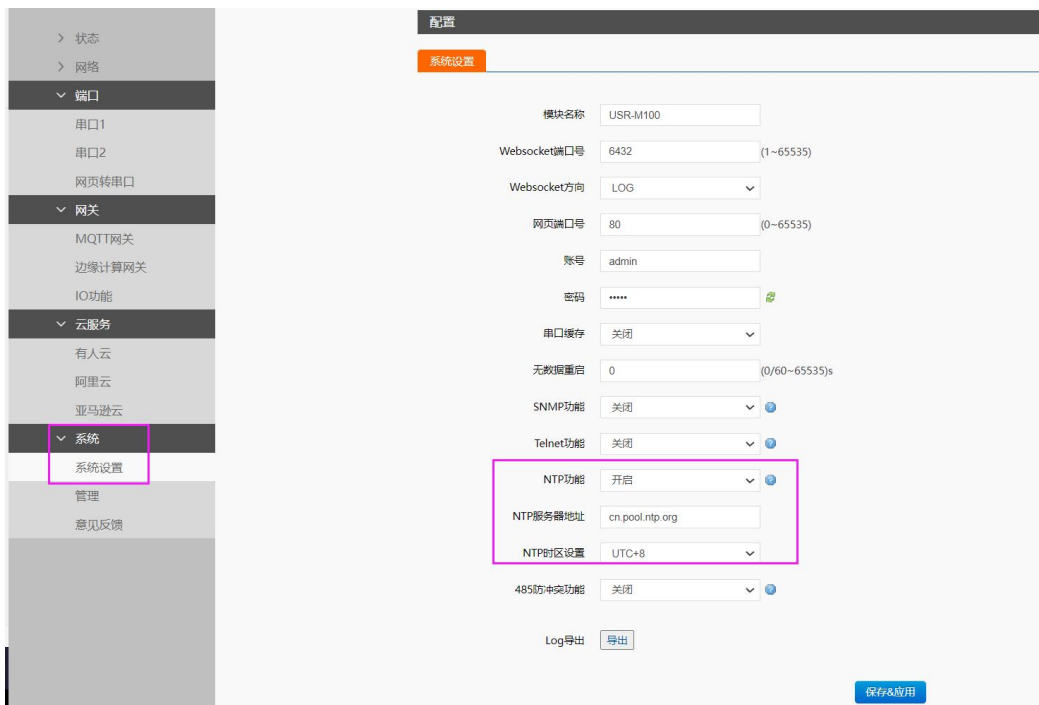
定时上报配置需要先开启 NTP 功能，然后再进行功能测试。在数据上报界面勾选定时上报，并选择定时时间，本应用案例选择整分钟上报，既每分钟都上报一次。（需要关掉周期上报进行验证）

定时上报和 NTP 配置参考下图：

The screenshot shows the '数据上报' (Data Reporting) configuration page. The interface includes several sections:

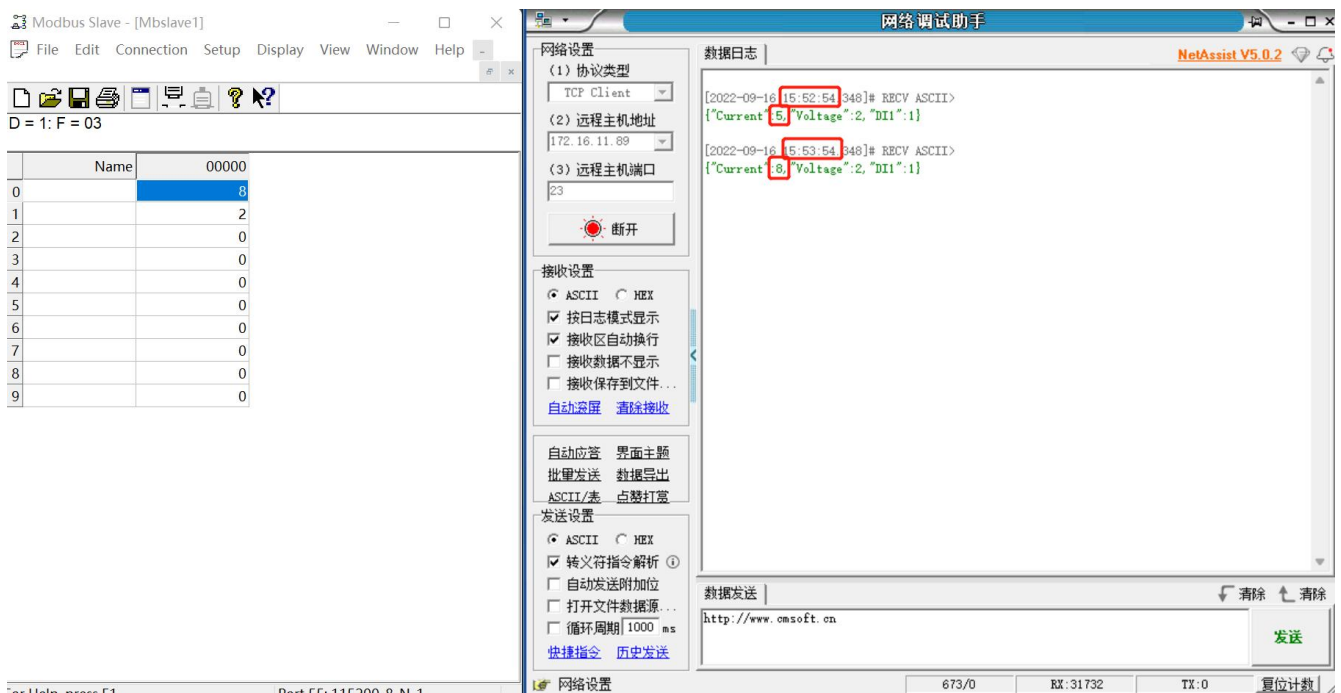
- 数据通道** (Data Channel): Channel selection is set to 'Socket1'.
- 数据查询/设置** (Data Query/Settings): Data read/write is set to '关闭' (Closed).
- 数据上报** (Data Reporting):
 - '数据上报' (Data Reporting) is set to '开启' (Enabled).
 - '周期上报' (Periodic Reporting) is unchecked.
 - '定时上报' (Scheduled Reporting) is checked, with a note '(需要先开启NTP)'. This section is highlighted with a red box.
 - '上报时间' (Reporting Time) is set to '整小时上报' (Whole hour reporting).
 - '失败填充' (Failure Filling) and '引导包含' (Guide Inclusion) are unchecked.
- 上报协议** (Reporting Protocol):
 - 'Json模板' (Json Template) is set to: `{"Current": "node0101", "Voltage": "node0102", "DI1": "DI1"}`. The size is shown as (<2048 bytes) (55 byte).

At the bottom, there are '保存' (Save) and '下一步' (Next Step) buttons.



3.4.7.2. 定时上报数据测试

设备重启后，调试助手连接设备，通过改变 Modbus Slave 寄存器的数值，可以看到数据整分钟上报，并改为从机当前数据。



3.4.8. 变化上报

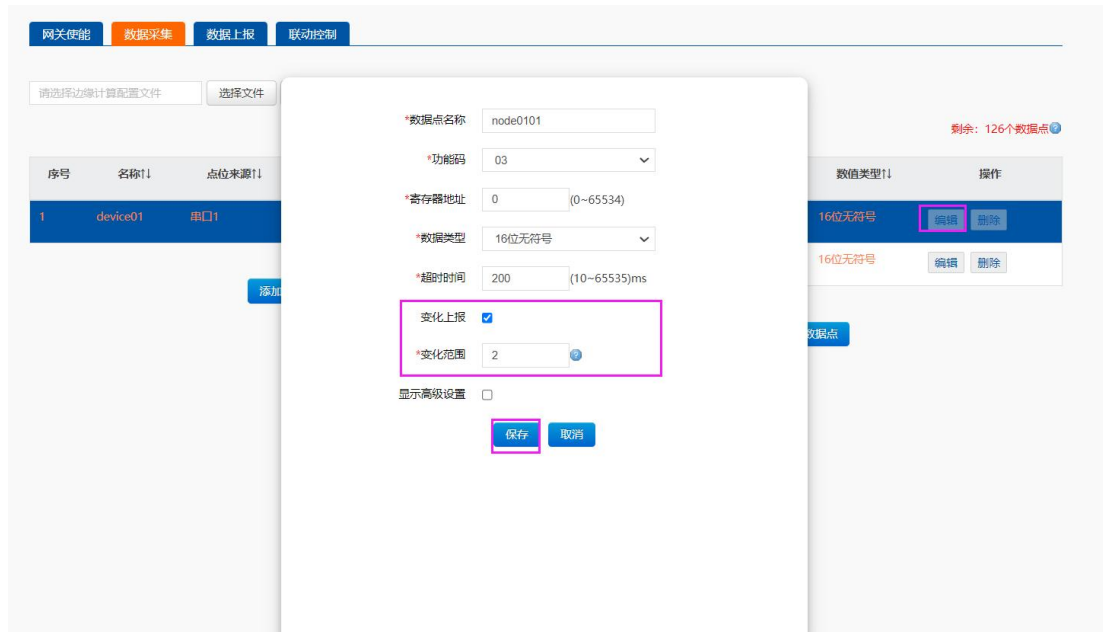
变化上报主要是数据变化后，根据变化条件判断数据是否要立即上报。例如变化范围为 2，如果数据采集后对比上一次数据，实际变化差值大于等于 2，本数据则会立即上报到服务器。

注：变化立即上报点位会按照 json 模板上报，所以其他未变化点位也会跟随上报一次。

3.4.8.1. 变化上报配置

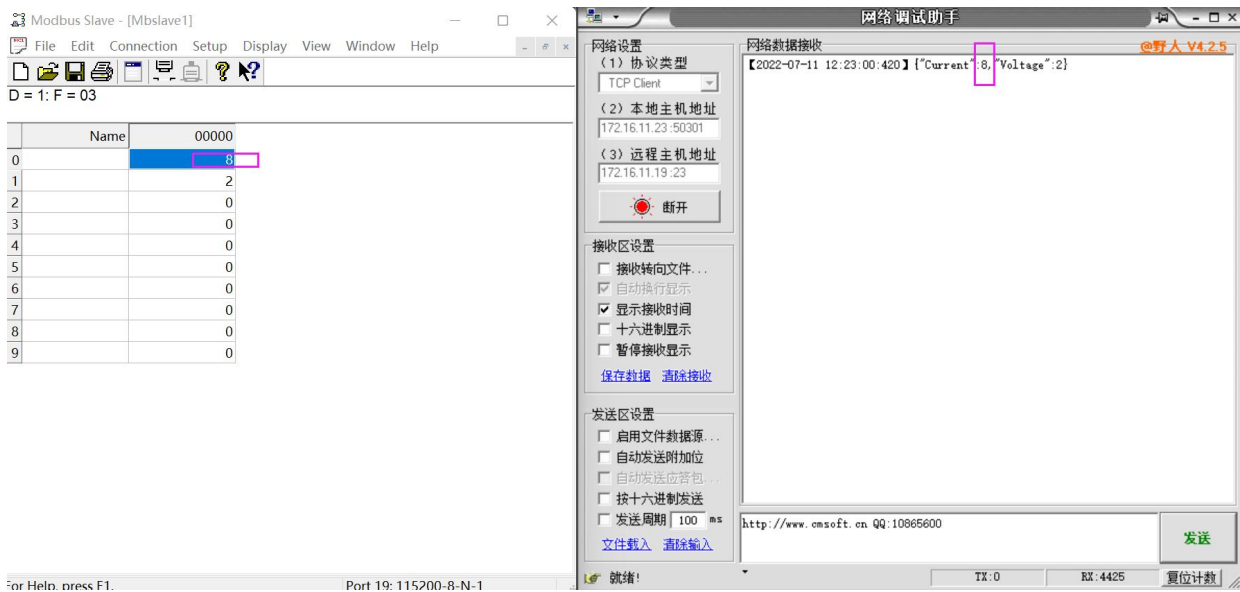
变化上报配置需要在每个点位上单独配置，仅对配置的点位起作用，点位之间相互不影响。

在数据采集界面，找到已经配置好的点位或者新增点位，勾选变化上报，并设置变化范围。本应用选择使用变化范围为 2，设置完成后保存点位配置



3.4.8.2. 变化上报数据测试

通过改变 Modbus Slave 的寄存器数值，可以看到网络调试助手立即收到上报的信息，如图，数据从 5 变到 8，数据立即上报。



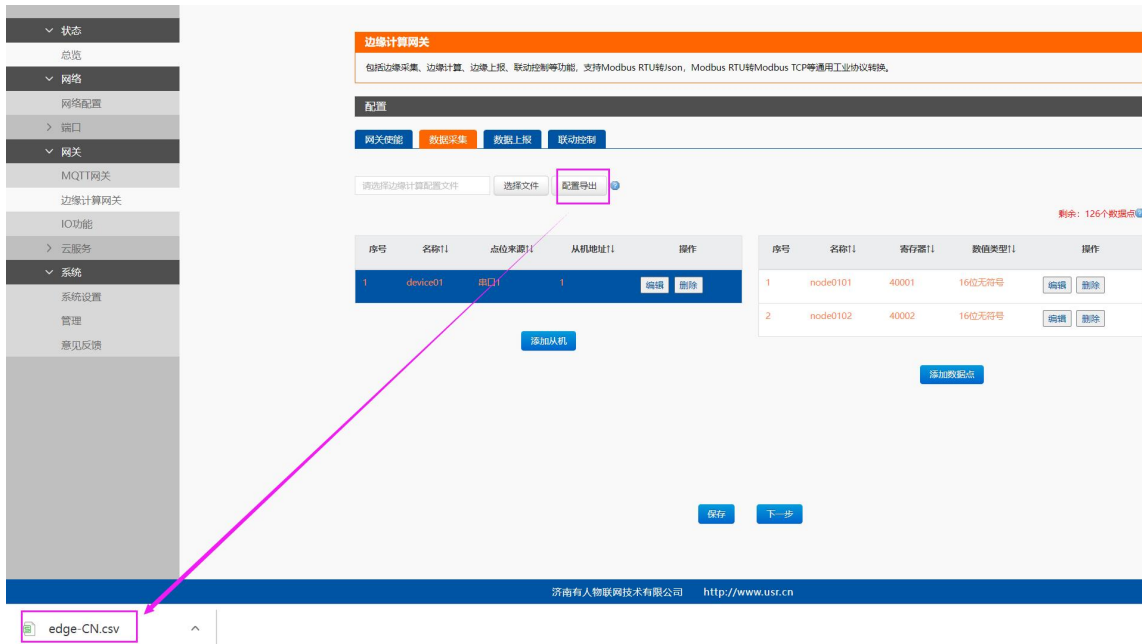
3.5. 点位批量导入导出

边缘采集的数据点位可以通过内置网页界面添加，也可以通过导入导出的方式进行批量添加。

当采集点位较多时，可以通过导出，导入的方式实现快速添加。整体操作逻辑是先导出设备模板，然后按照模板进行点位添加，完成后再将点位列表进行导入，即可完成多点位的快速添加。

3.5.1. 导出

“网关” -- “边缘计算网关” -- “数据采集” 点击“配置导出”按键，将以.csv 文件格式导出当前“数据采集”中的配置。



3.5.2. 编辑

打开的.csv 文件如图所示。

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|----------|-----|------|------|------|------|---|----------|-----|-------|--------|------|------|------------|---|
| 1 | 设备名称 | 串口号 | 从机地址 | 轮询间隔 | 映射地址 | 合并采集 | | 数据点名 | 功能码 | 寄存器地址 | 数据类型 | 超时时间 | 映射地址 | 数据计算 | |
| 2 | device01 | 1 | 1 | 0 | N/A | OFF | | node0101 | 3 | 0 | 16位无符号 | 100 | N/A | =(%s+1)*10 | |
| 3 | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | |

根据实际需求按照原有格式，添加如下数据点，保存文件。

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|----|----------|-----|------|------|------|------|---|----------|-----|-------|--------|------|------|------------|---|
| 1 | 设备名称 | 串口号 | 从机地址 | 轮询间隔 | 映射地址 | 合并采集 | | 数据点名 | 功能码 | 寄存器地址 | 数据类型 | 超时时间 | 映射地址 | 数据计算 | |
| 2 | device01 | 1 | 1 | 0 | N/A | OFF | | node0101 | 3 | 0 | 16位无符号 | 100 | N/A | =(%s+1)*10 | |
| 3 | device01 | 1 | 1 | 0 | N/A | OFF | | node0102 | 3 | 1 | 16位无符号 | 100 | N/A | | |
| 4 | device01 | 1 | 1 | 0 | N/A | OFF | | node0103 | 3 | 2 | 16位无符号 | 100 | N/A | | |
| 5 | device01 | 1 | 1 | 0 | N/A | OFF | | node0104 | 3 | 3 | 16位无符号 | 100 | N/A | | |
| 6 | device01 | 1 | 1 | 0 | N/A | OFF | | node0105 | 3 | 4 | 16位无符号 | 100 | N/A | | |
| 7 | device01 | 1 | 1 | 0 | N/A | OFF | | node0106 | 3 | 5 | 16位无符号 | 100 | N/A | | |
| 8 | device01 | 1 | 1 | 0 | N/A | OFF | | node0107 | 3 | 6 | 16位无符号 | 100 | N/A | | |
| 9 | device01 | 1 | 1 | 0 | N/A | OFF | | node0108 | 3 | 7 | 16位无符号 | 100 | N/A | | |
| 10 | device01 | 1 | 1 | 0 | N/A | OFF | | node0109 | 3 | 8 | 16位无符号 | 100 | N/A | | |
| 11 | device01 | 1 | 1 | 0 | N/A | OFF | | node0110 | 3 | 9 | 16位无符号 | 100 | N/A | | |
| 12 | device02 | 2 | 2 | 0 | N/A | OFF | | node0201 | 3 | 0 | 16位无符号 | 100 | N/A | | |
| 13 | device02 | 2 | 2 | 0 | N/A | OFF | | node0202 | 3 | 1 | 16位无符号 | 100 | N/A | | |

未开启的功能会以 N/A 做填充

3.5.3. 导入

点击“选择文件”按键，选择修改后的配置文件，导入成功后界面中将直接显示配置文件中增、删、改的从机和数据点配置：



注意：

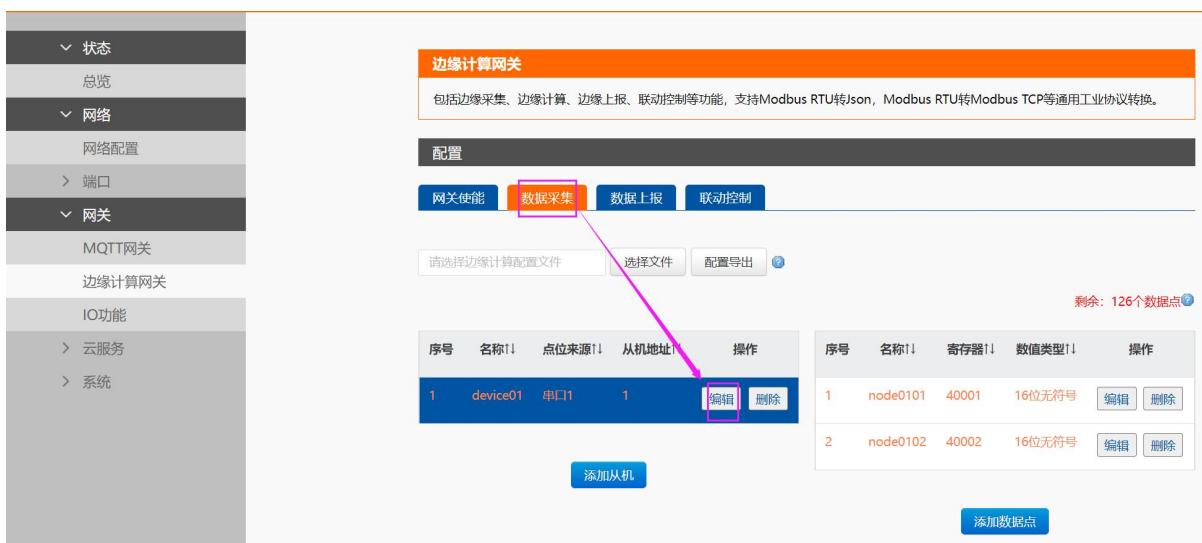
1. 中文页面下导出的点表文件只能在中文界面下导入，反之亦然。
2. 数据点名称需要全表内唯一。
3. 若导入失败，请检查配置文件中修改的参数类型、数值等是否超出正常范围。

3.6. 数据读写

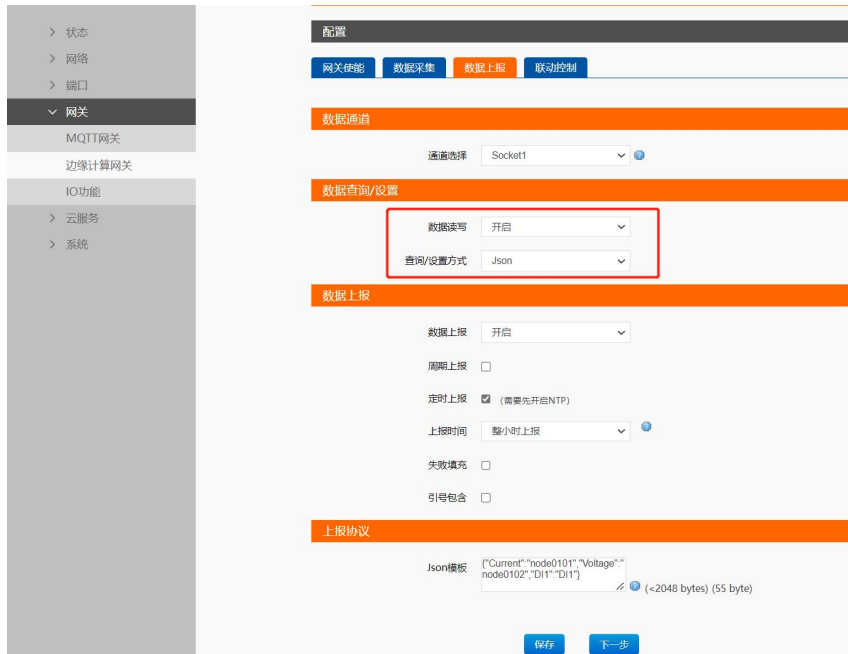
边缘计算模式下，点位数据除了主动上报功能外，服务器还可以通过响应的协议对点位进行读取和写入，实现状态的获取和控制。

3.6.1. 参数配置

在设备内置网页，配置好响应的点位。



边缘计算网关→数据上报界面，开启数据读写功能，选择相应的读写协议。本示例采用Json来实现数据的读写。参数配置完成后，保存重启。



3.6.2. Json 读写格式

Json 读写命令格式：（可以同时获取多个点位数据）

```
{"rw_prot": {"Ver": "协议版本", "dir": "数据走向", "id": "信息编号", "r_data": [{"name": "点位名称"}], "w_data": [{"name": "点位名称", "value": "data"}]}}
```

Json 读写命令字段说明：

| 字段名称 | 字段说明 | 字段选择 |
|---------|----------------------------|----------------|
| rw_prot | 协议包头 | |
| ver | 协议版本 | 1.0.1 |
| dir | 数据走向，服务器下发命令填入 down | down: 服务器下发 |
| id | 服务器下发数据的编码，可以作为序列识别使用 | 客户自定义，设备回复不做改变 |
| r_data | 数据读取字段 | |
| w_data | 数据控制字段 | |
| name | 点位名称，和点表中的点位名称一致即可代入点位 | |
| value | 读写命令中仅有写有 value 字段，为写入的有效值 | |

Json 读写回复格式：

```
{"rw_prot": {"Ver": "协议版本", "dir": "数据走向", "id": "信息编号", "r_data": [{"name": "点位名称", "value": "data", "err": "错误码"}], "w_data": [{"name": "点位名称", "value": "data", "err": "错误码"}]}}
```

Json 读写回复字段说明：

| 字段名称 | 字段说明 | 字段选择 |
|---------|------------------|------------------------------------|
| rw_prot | 协议包头 | |
| ver | 协议版本 | 1.0.1 |
| dir | 数据走向，设备回复内容填入 up | up: 设备回复 |
| id | 信息识别码，保持和下发命令一致 | |
| r_data | 数据读取字段 | |
| w_data | 数据控制字段 | |
| name | 点位名称，对应点表中的点位 | |
| value | 点位对应的有效数据 | 读错误，value 有效值为空 写错误，value 值为历史值 |
| err | 错误码 | 0: 数据正常执行 1: 数据错误执行 |

Json 字段错误回复：

- 1、Json 格式错误：设备不做回复
- 2、ver, dir, id 三个字段，任意一个错误，则按照错误协议进行回复。
- 3、其他字段正确，r_data 或 w_data 字段仅错误一个，则错误的字段舍弃，正确字段进行回复；如果两个字段都错误，按照错误协议回复。
- 4、错误协议: {"rw_prot": {"Ver": "1.0.1", "dir": "up", "err": "1"}}

| 字段名称 | 字段说明 | 字段选择 |
|---------|--------------|--------------------|
| rw_prot | 协议包头 | |
| ver | 协议版本 | 1.0.1 |
| dir | 数据走向，上报和下发两种 | up: 设备回复 |
| err | 错误码 | 0: 正常执行 1: 错误执行 |

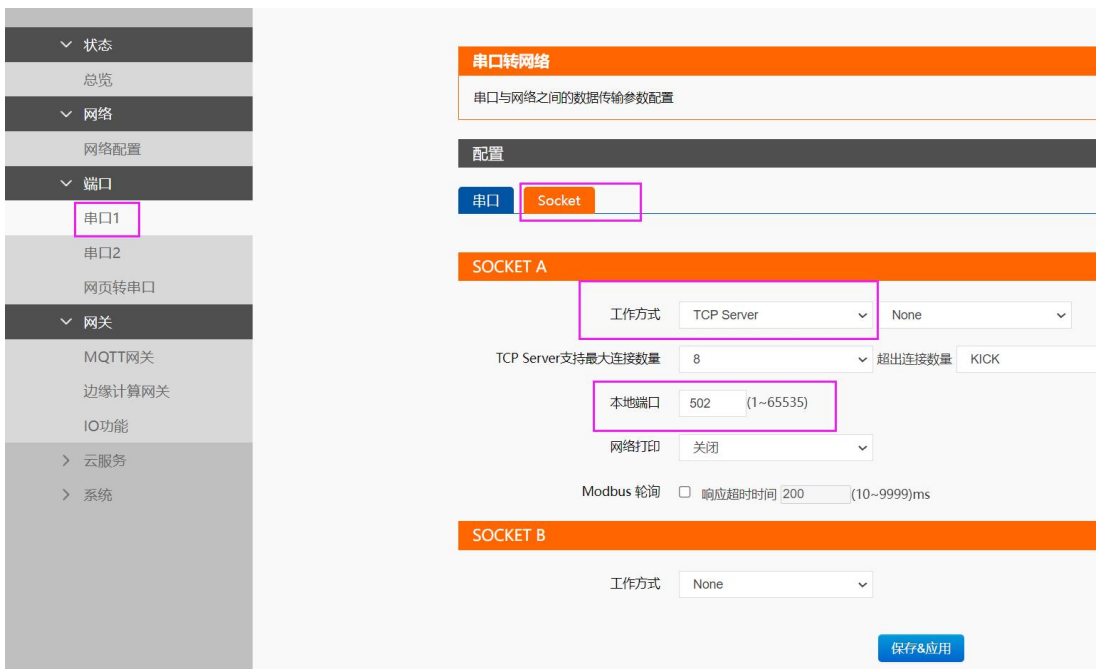
说明：

- 1、读写命令错误时，读命令回复内容的 value 值为空，写命令的回复内容的 value 值为历史数据值。

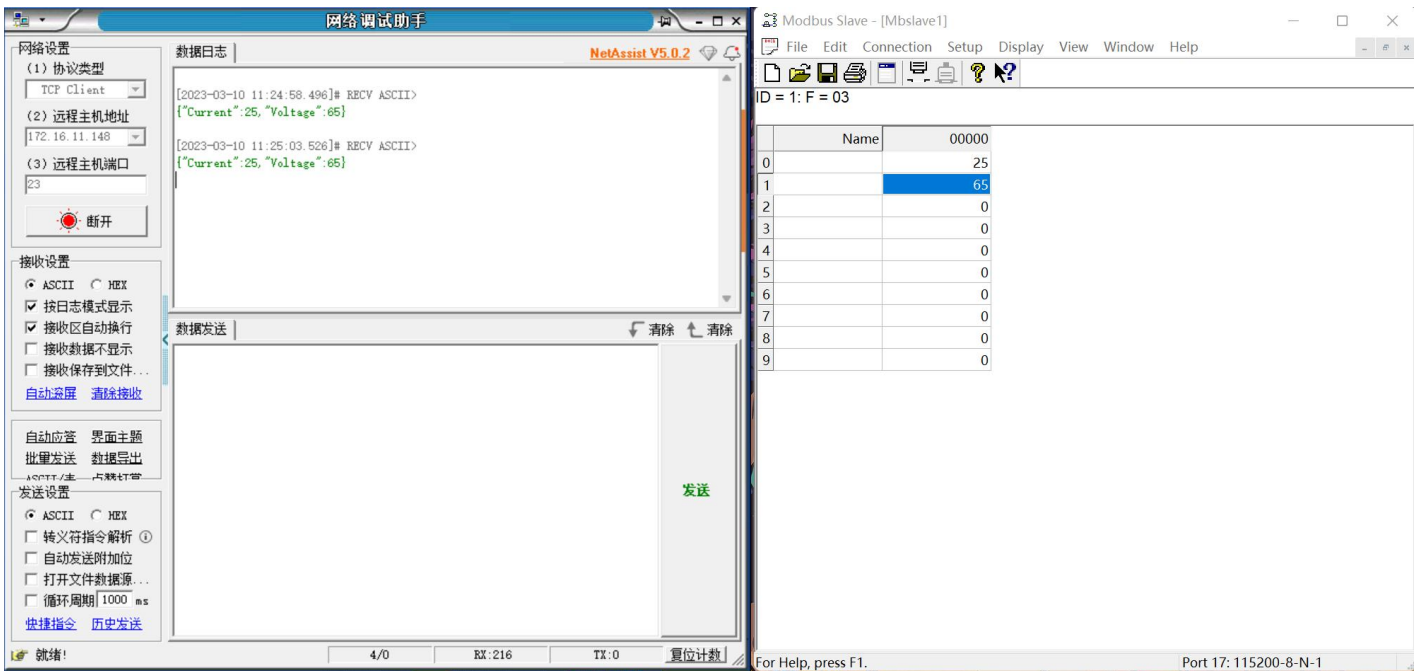
读写操作最大上限为读写各 5 个数据点位同时操作。

3.6.3. 读写实例

读写和上报共用一个通道，本示例依然采用 socket1 来实现数据读写。Socket1 参数配置如下图，配置完成后，保存重启。



设备正常运行后，打开串口调试助手和 modbus slave，分别接入串口和网络，实现数据采集和上报。



依据 Json 读写模板对设备的点位进行配置，按照示例进行 Json 读写命令组建：读取点位 1 的数据，改写点位 2 的数据从 65 改为 70。

下发命令：

```

{"rw_prot": {"Ver": "1.0.1","dir": "down","id": "12345","r_data": [{"name": "node0101"}],"w_data": [{"name": "node0102","value": "70"}]}}
    
```

回复内容：

```

{"rw_prot":{"Ver":"1.0.1","dir":"up","id":"12345","r_data":[{"name":"node0101","value":"25","err":"0"}],"w_data":[{"name":"node0102","value":"70","err":"0"}]}}
    
```

The image shows two software windows. The left window is '网络调试助手' (NetAssist V5.0.2) and the right window is 'Modbus Slave - [Mbslave1]'.

NetAssist V5.0.2 Settings:

- 网络设置: (1) 协议类型: TCP Client; (2) 远程主机地址: 172.16.11.148; (3) 远程主机端口: 23.
- 接收设置: ASCII, HEX; 换日志模式显示; 接收区自动换行; 接收数据不显示; 接收保存到文件...
- 发送设置: ASCII, HEX; 转义指令解析; 自动发送附加位; 打开文件数据源...; 循环周期: 1000 ms.

数据日志 (Data Log):

```

[Current":25,"Voltage":70]
[2023-03-10 11:33:40.962]# SEND ASCII>
{"rw_prot":{"Ver":"1.0.1","dir":"down","id":"12345","r_data":[{"name":"node0101"},"w_data":[{"name":"node0102","value":"70"}]}}
[2023-03-10 11:33:41.014]# RECV ASCII<
{"rw_prot":{"Ver":"1.0.1","dir":"up","id":"12345","r_data":[{"name":"node0101","value":"25","err":"0"}],"w_data":[{"name":"node0102","value":"70","err":"0"}]}}
[2023-03-10 11:33:41.413]# RECV ASCII>
{"Current":25,"Voltage":70}
    
```

数据发送 (Data Send):

```

{"rw_prot":{"Ver":"1.0.1","dir":"down","id":"12345","r_data":[{"name":"node0101"},"w_data":[{"name":"node0102","value":"70"}]}}
    
```

Modbus Slave - [Mbslave1] Table:

| Name | 00000 |
|------|-------|
| 0 | 25 |
| 1 | 70 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |

StatusBar: 就绪! 102/2 RX: 3315 TX: 274 复位计数 For Help, press F1. Port 17: 115200-8-N-1

可信赖的智慧工业物联网伙伴

天猫旗舰店: <https://youren.tmall.com>

京东旗舰店: <https://youren.jd.com>

官方网站: www.usr.cn

技术支持工单: im.usr.cn

战略合作联络: ceo@usr.cn

软件合作联络: console@usr.cn

电话: 0531-66592361

地址: 山东省济南市历下区茂岭山三号路中欧校友产业大厦 13 楼



关注有人微信公众号



登录商城快速下单