

ZLAN6408 系列

4 路 DI/AI

IO 控制器/4G 网关

RS485 转 4DI/4AI

CAT1 4G 转 RS485/4DI/4AI



版本信息

对该文档有如下的修改：

修改记录				
日期	版本号	文档编号	保密等级	修改内容
2024-10-30	Rev.1	ZL DUI 20241030.1.0	公开	发布版本
2025-08-18	Rev.2	ZL DUI 20241030.2.0	公开	增加 6408-1

所有权信息

未经版权所有者同意，不得将本文档的全部或者部分以纸面或者电子文档的形式重新发布。

本文档只用于辅助读者使用产品，上海卓岚公司不对使用该文档中的信息而引起的损失或者错误负责。本文档描述的产品和文本正在不断地开发和完善中。上海卓岚信息科技有限公司有权利在未通知用户的情况下修改本文档。

目 录

1. 概述	5
2. 功能特点	6
3. 技术参数	6
4. 硬件说明	7
5. DI/AI 功能说明	10
5.1 使用 Vircom 工具连接设备	10
5.2 Modbus 寄存器	11
5.3 DI 使用说明	12
5.4 DI 计数使用说明	13
5.5 DI 逻辑反转	14
5.6 AI 使用说明	14
5.7 AI 高精度使用	15
5.8 DI 主动上报	16
5.9 AI 主动上报	18
5.10 DI 和 AI 同时上传	19
5.11 DI 触发电平再保持 2 秒功能	19
6. 4G 配置方式 (6408-8)	20
6.1 串口 AT 指令	20
6.2 固件/配置文件模式	26
7. 配置与使用方法 (6408-8)	28
7.1. 基础参数设置	28
7.2. MODBUS RTU 转 JSON 配置	29
7.3. 服务器透明传输测试	33
7.4. Modbus 协议转化测试	34
7.5. MQTT 协议测试	35
附录 1：参数总表	39
附录 2：AI 校准	42

售后服务和技术支持 43

1. 概述

ZLAN6408 是上海卓岚新推出的小体积 IO 控制器系列，支持采集开关量输入和模拟量输入信号。分为 6408-1 和 6408-8 两个子型号。

6408-1 支持 4 路 DI/AI 采集。可通过 RS485 接口读取 IO 数据。

6408-8 支持 4 路 DI/AI 采集，可通过 4G/RS485 接口读取 IO 数据。同时还具有 4G 数据采集网关的功能，可在 RS485-4G 接口上挂载第三方的 RS485 采集器、控制器，实现远程采集控制。

其中 DI 输入支持干节点和湿节点，带光耦隔离；AI 输入支持 4-20mA 电流输入，ADC 精度为 12 位，可以根据需求修改 AI 的属性为 0-5V 电压、0-10V 电压型、电阻型等属性。

。



图 1 ZLAN6408 网关外观图

其中 ZLAN6408 做为一款高性能的 4G 数据传输与处理设备，同时支持 TCP、HTTP 和 MQTT 协议，确保数据传输的安全性与高效性。网关不仅能够轻

松实现数据的远程采集与传输，还具备强大的数据处理能力。特别是网关支持串口 DLT645 协议、Modbus RTU 协议到网络部分 JSON 格式的数据转换，使得不同系统间的数据交互变得更为便捷。

2. 功能特点

1. 支持 4 路 DI 数字量输入，AI 模拟量输入。模拟量输入精度为 12 位。
2. 串口支持 1200~115200 波特率，支持 5~8 位数据位，支持无校验、奇校验、偶校验，支持 1~2 位停止位
3. 6408-8 支持 3 种制式模式，TD-LTE/ FDD-LTE/ GSM，包含有联通 4G，2G，移动 4G，2G 以及电信 4G 网络。
4. 6408-8 支持自定义 Modbus RTU 转 JSON 功能。
5. 6408-8 支持边缘计算功能：包括数据超限报警、数据平移缩放计算、数据变化上传、设备离线报警等。
6. 6408-8 支持串口 AT 命令配置，支持 ZLVircom 软件查看部分参数。
7. 6408-8 可以通过串口对设备进行固件更新。

3. 技术参数

外形	
尺寸：	长×宽×高=9.4cmx6.5cmx2.5cm
串口参数	
6408-1:485-IO	
6408-8:485-IO、485-4G	
波特率：	默认波特率为 115200bps，通过软件或指令修改。
数据位：	8 位。
校验位：	无校验、奇校验、偶校验。
停止位：	1 位
软件	
网络协议：	MODBUS TCP/MQTT/JSON/HTTP
RS485 协议：	MODBUS RTU

AI 输入形式	
电流输入: 4~20mA	
电压输入: 0~5V, 0~10V (需要定制)	
电阻输入: 0~10K, 电阻型的温湿度传感器等 (需要定制)	
电源	
9-24VDC,1W	
6408-8 (4G CAT1) 参数	
支持模式	4G CAT1 支持 3 种模式: B1/B3/B5/B8@FDD LTE B34/B38/B39/B40/B41@TDD-LTE B3/B8@GSM 包含有联通 4G, 2G, 移动 4G, 2G 以及电信 4G 网络。
传输速率	LTE: mA 10Mbps (下行) /mA 5 Mbps (上行) GPRS: 85.6Kbps (下行) /mA 85.6Kbps (上行)
SIM 卡	电压: 3V, 1.8V; 大小: 大卡 (小卡可以购买卡套使用)
天线接口	50 Ω/SmA 胶棒天线或吸盘天线可选。
环境要求	
操作温度:	-40~85 °C
储存温度:	-45~165 °C
湿度范围:	5~95% 相对湿度

4. 硬件说明

数据采集网关的正视图如图 2 所示。



图 2 ZLAN6408 正视图

网关采用抗辐射金属外壳，两边两个挂耳，可以使用螺丝固定。

面板灯：

指示灯	颜色	备注
POWER	红色	设备正常上电
4G/TCP	绿色/蓝色	绿色代表 4G 拨上号/蓝色代表链接已建立
DI	绿色	绿色代表 DI 有效，值为 1



图 3 接口图 1

采集网关前面接口如图 3 所示：

复位按钮：短按 3S 以上可以复位 4G 网关的参数。

1. RS485 接口：2 路 RS485 信号输入，-IO,+IO 接 Modbus RTU 主站可以读取 AI 和 DI 的数据。-4G,+4G 可以接 485 仪表从站,注意不要接电源(6408-1 只支持-IO,+IO)。
2. SIM 接口：安装 SIM 卡时应确保设备未上电。使用笔头、螺丝刀，将 SIM 卡槽顶出来，SIM 金属面朝下推入卡槽(6408-1 不支持)。
3. 天线：采集网关的天线接口采用 50Ω/SmA (公头)，外接天线必须使用适合 4G 工作波段的天线。

采集网关后面接口如图 4 所示：



图 4 接口图 2

1. 电源：接线端子式输入，输入电压 DC+9V~+24VDC。
2. GND：同电源负极,干节点输入时，将此端子和 DI1~DI4 之间跨接开关，可以采集到开关状态。
3. COM: 公共端,内部接口，外部不接。
4. DI1~DI4:4 路数字量输入。
5. AI1~AI4:4 路模拟量输入。

开关量模拟量输入：

1. 4 路数字量输入 DI1~DI4。

支持无源开关量(干节点)和有源电平(湿节点)。干节点只需要将其和 GND 短接即采集到 1 信号。湿节点时，有源电平和 GND 之差的范围如下：

VCC 电压	低电平范围	高电平范围
24V	0~17V	17~24V
9V	0~3V	3~9V

2. 4 路模拟量输入：精度为 12 位，默认为 4-20mA 电流输入(其他需要定制)：

- 1) 电流信号输入: 4~20mA。
- 2) 电压信号输入: 0~5V。
- 3) 电压信号输入: 0~10V。
- 4) 电阻阻抗输入: 如 0~10k 或电阻型的温湿度传感器等。

电压和电流都是相对于 GND。

5. DI/AI 功能说明

5.1 使用 Vircom 工具连接设备

网关通过 RS485 接口配置 IO 参数。将设备上电，连接 RS485 中的-IO 和+IO 接口。使用 VIRCOM 软件打开主界面设备管理，直接点击“IO 控制器”按钮。

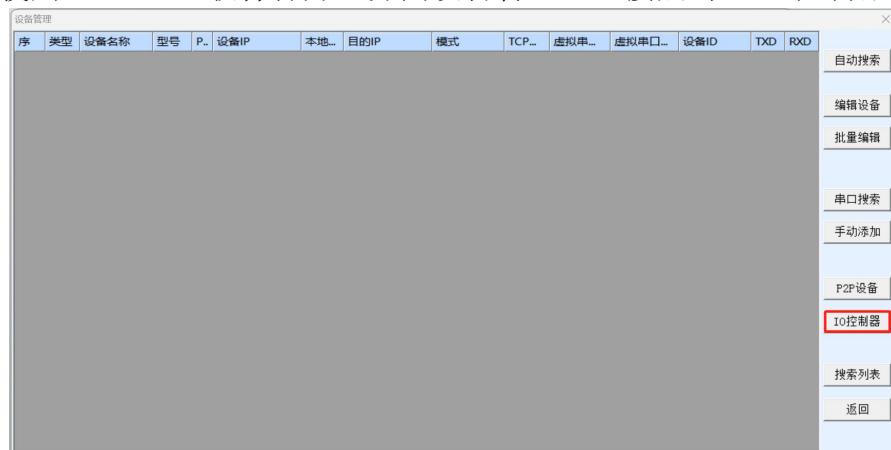


图 5 如何进入 IO 控制器对话框



图 6 IO 控制器对话框

选择正确的 COM 口点击“打开并搜索”来和设备通讯，不必选择波特率。如果之前设置过校验位，请选择对应的校验位后点击“打开并搜索”。打开 com 口之后通过软件 Modbus RTU 指令获得设备的参数。

设备获得参数之后就会显示在对话框中。后面可以对参数进行修改、DI 读取、AI 读取等测试。

5.2 Modbus 寄存器

串口支持 Modbus RTU 指令。具体寄存器和地址范围如下：

表 1. Modbus 寄存器总表

功能码	功能	地址范围
01/02	读 DI	0~3
04	读 AI	0~3
04	读 AI 高精度值	32~35
03	读基础参数	63~67
03	读扩展参数	68~162
03	读 DI 16 位计数	0~3

03	读 DI 32 位计数	256~263
06	设置参数	63~67
06	设置扩展参数	68~162
06	设置 DI 16 位计数	0~3
06	设置 DI 32 位计数	256~263
16	设置多 DI 16 位计数	0~3
16	设置多 DI 32 位计数	256~263
16	设置基础参数	63~67
16	设置扩展参数	68~162

5.3 DI 使用说明

读 DI 使用 01 指令，地址范围 0~3，对应 DI1 到 DI4。指令格式如下：

字节数	1	1	1	1	1	1	1	1
名称	设备地址	01	起始地址高	起始地址低	长度高	长度低	CRC 高	CRC 低

例如读取 4 个 DI 的 Modbus RTU 指令为：

发送-> 01 01 00 00 00 04 3D C9

返回-> 01 01 01 01 90 48

当 DI 输入为低电平的时候（注意设备供电电压 12V 以上时，5V 电压输入被认为低电平），返回对应的 bit 为 1，返回指令中的第 4 个字节为 0x01 表示第 1 路为闭合（低电平）状态。

IO 控制器对话框控制演示：

图 7 IO 控制器对话框中的 DI 读取

当 Vircom 连接设备成功之后，点击“查询 DI 状态”开就可以查询 DI 的状态。DI 为低电平的时候，对应的指示灯点亮，返回的对应的 bit 为 1。图中所示 DI4 打勾，说明 DI4 是低电平状态。

点击“自动”选择框，可以每隔1秒钟自动查询DI的状态，并显示出来。

5.4 DI 计数使用说明

当DI从高电平变为低电平、再变回高电平的一个周期算是一次计数。DI计数分为三种：16位无存储计数、32位无存储计数、32位有存储计数。无存储是指点掉后从0开始，有存储会掉电后保持计数。其中32位无存储计数、32位有存储计数是同一个寄存器位置，只是设置配置不同而已。

DI计数已经自动增加了消抖处理，消抖时间为10ms。

通过Modbus的03功能码，读取0~3寄存器位置就可以读到16位无存储计数，数据为大端格式。通过03功能码，读取256~263位置可以读取32位计数，数据位大端格式。

字节数	1	1	1	1	1	1	1	1
名称	设备地址	03	起始地址高	起始地址低	长度高	长度低	CRC高	CRC低

例如读取DI4的16位计数的Modbus RTU指令为：

发送-> 01 03 00 03 00 01 74 0a

返回-> 01 03 02 01 0a 39 d3

Modbus TCP指令为：

发送-> 00 00 00 00 00 06 01 03 00 03 00 01

返回-> 00 00 00 00 00 05 01 03 02 01 0a

这里读取寄存器3，返回数据的01 0a表示数值266。

例如读取DI4的32位计数的Modbus RTU指令为：

发送-> 01 03 01 06 00 02 25 F6

返回-> 01 03 04 00 00 01 14 fb ac

Modbus TCP指令为：

发送-> 00 00 00 00 00 06 01 03 01 06 00 02

返回-> 00 00 00 00 00 07 01 03 04 00 00 01 14

这里的00 00 01 14表示数值276。

通过ZLVircom配置软件中的“32位DI计数保存”功能来让32位计数为保

存或者不保存。如果希望对保存的数据清零，重新开始计数。只需要将“32位 DI 计数保存”功能设置为 0，即可清空计数。

5.5 DI 逻辑反转

正常时当 DI 输入为低电平的时候，返回对应的 bit 为 1。默认 DI 输入为高电平，低电平有效。假如现在需要 DI 输入为高电平有效，即默认的 bit 为 1，如果接低电平，则 bit 为 0，此时可以选择“DI 逻辑反转功能”。

DI 反转也会影响 DI 计数，DI 计数都是在 DI 从 0 变成 1 增 1，即高电平变为低电平。如果 DI 逻辑反转，则改成低电平变为高电平则计数增加一。

DI 逻辑反转的设置方法如下。

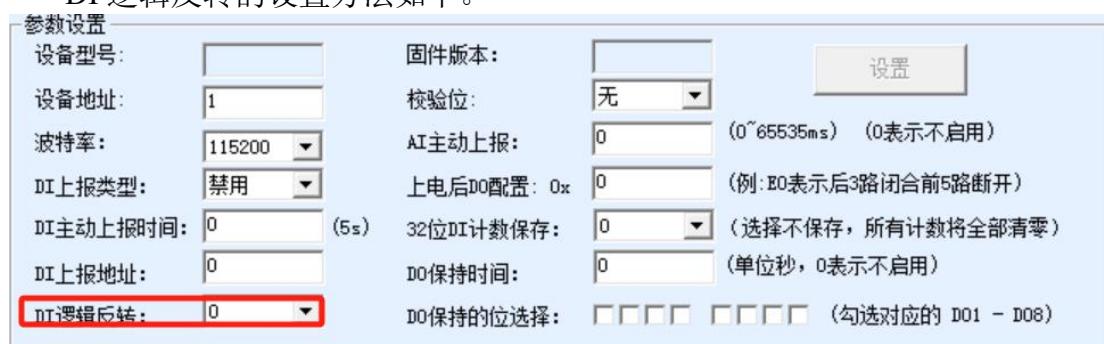


图 8 IO 控制器对话框中的 DI 反转设置

5.6 AI 使用说明

使用 Modbus 的 04 指令读取 0~3 寄存器的数值，就可以获得 AI1~AI4 的数值。数据为大端格式存放。

字节数	1	1	1	1	1	1	1	1	1
名称	设备地址	04	起始地址高	起始地址低	长度高	长度低	CRC 高	CRC 低	

例如读取 AI1 的数值的 Modbus RTU 指令为：

发送-> 01 04 00 00 00 01 31 ca

返回-> 01 04 02 01 82 38 c1

Modbus TCP 指令为：

发送-> 00 00 00 00 00 06 01 04 00 00 00 01

返回-> 00 00 00 00 00 05 01 04 02 01 82

这里返回数据 01 82 的具体使用方法跟 AI 的类型有关, 将 01 82 转化为十进制, 则为 $V_{in}=386$ 。对于不同的 AI 类型计算公式如下:

- 5V: 真实电压值 $= (V_{in}/ 1024)*5=1.8848$;
- 10V: 真实电压值 $= (V_{in}/ 1024)*10=3.7695$;
- 4~20mA: 真实电流 $= (A_{in} / 1024)*5/200*1000=9.4238$;

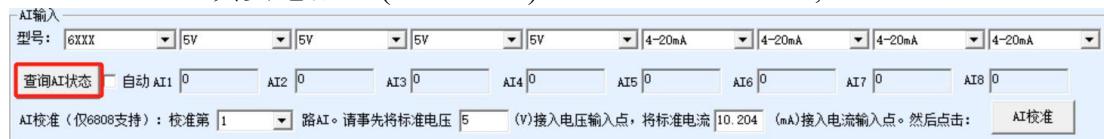


图 9 IO 控制器对话框中的 AI 读取

当 Vircom 连接设备成功之后, 点击“查询 AI 状态”开就可以查询 AI 的数值, 也可以点击“自动”来 1 秒钟查询一次。查询之前需要根据购买的型号先进行选择。选择型号之后, 会根据标准配置自动配置 AI1~AI4 的模拟量接口类型, 这样在数值对话框中就可以显示该接口的真实电流值。

5.7 AI 高精度使用

数据采集网关提供更高精度的 AI 数值计算方法。和普通的精度相比, 不对微小的波动自动过滤为 0 电压, 不对变化很小的数值自动设置为上一次采集的电压。所以能更加真实反应电压值, 但是可能存在更多噪声。

使用 04 功能码读取 32~35(0x20~0x23)寄存器的内容就可以获得 AI 高精度值。数据格式是大端格式。这是一个 12 位有效精度的值 Vh。

计算输入点电流为:

$$I_i = (((Vh)/1024)-1.0)*(Vri)*2.0)/200$$

其中 V_i ($i=1\sim 4$) 是每路的调整系数, 默认为 1.0。可以使用 03 功能码读取 0x4a~0x51 (十进制 74~81) 开始的寄存器, 获得分别对应 V1~V4 的浮点 (float) 型大端格式数据。例如 1.063 的 float 型数据读取的结果为 0x3F88 1062 形式的十六进制。

例如读取 A1 的调整系数:

即发送-> 01 03 00 4a 00 02 e5 dd

返回-> 01 03 04 **3f 80 00 00** f7 cf

其中的 **3f 80 00 00** 表示 1.0。

再读取第 1 路的 Vh:

即发送-> 01 04 00 20 00 01 30 00

返回-> 01 04 02 **07 c7** fa 92

其中 07 c7 代表 1991, 带入公式可得电压为:

$((((1991)/1024)-1.0)*(1.0)*2.0)= 1.8887$ 。

Vi 调整系数是出厂后校准过的, 可以保证产品的计算值的准确性。

5.8 DI 主动上报

ZLAN6408 是标准的 MODBUS 设备, 使用方式是问答形式, 但有的用户想要在 DI 输入一发生变化时就得到反馈, 即主动返回的功能。这里介绍 6408 的主动上报功能了。设置如图所示, 将“启用 DI 主动上报”设置为 1, 开启主动上报功能。DI 上报地址一般不要跟设备地址一样。

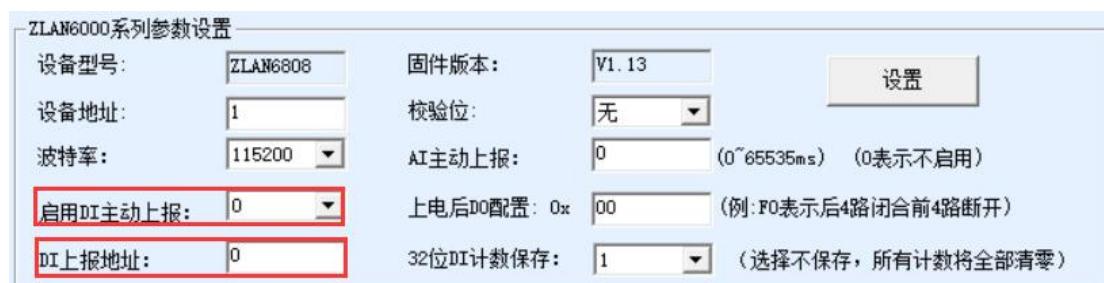


图 10 DI 主动上报设置

当 DI 的状态发生改变时, 开启 DI 主动上报后将发送 05 指令。05 指令可以实现 DI 的变化控制另外一台 Modbus 设备的 DO 的触发的功能。

字节数	1	1	1	1	1	1	1	1
名称	DI 上报地址	05	起始地址高	起始地址低	ff或者 00	00	CRC 高	CRC 低

例子如下:

DI1 变为高电平输入

00 05 00 10 00 00 CD 2E

DI1 变为低电平输入

00 05 00 10 ff 00 8C 2E

DI2 变为高电平输入

00 05 00 11 00 00 9C 1E

DI2 变为低电平输入

00 05 00 11 ff 00 DD EE

DI3 变为高电平输入

00 05 00 12 00 00 6C 1E

DI3 变为低电平输入

00 05 00 12 ff 00 2D EE

DI4 变为高电平输入

00 05 00 13 00 00 3D DE

DI4 变为低电平输入

00 05 00 13 ff 00 7C 2E

当使用 ZLVircom 测试的时候，DI 的主动上报会更新当前的 DI 的状态值。

DI 主动上报会同时发往 485-IO 和 4G。

当主动上报时间设置为 0 的时候禁用主动上报，设置为 1 的时候启动 DI 变化主动上报。设置为其它数值的时候会按照定时上报。其中设置为偶数的时候，会定时按照 15 指令上报 8 个 DI。设置为奇数的时候，会启用 DI 和 AI 同时上报，具体参考本章后续内容。当设置为 n，n 为非零偶数的时候，上报 DI 的时间为 $(n-1)*5$ 毫秒。例如设置前 4 路 DI 短接到 GND，后 4 路悬空，发送 DI 主动上报：

发送-> 01 0F 00 10 00 04 01 0F bf 51

通过新版本 Vircom 直接可以配置，不需要奇数/偶数设置。



*固件版本 V28 以下 DI 主动上报时间为 X 乘以 5ms, 例如 200, 则为 1000ms 上报, V28 及以上 DI 主动上报时间大于 100 时, 例如为 108, 则是(108-100)秒 上报。

5.9 AI 主动上报

AI 的主动上报功能是让采集的模拟量能够自动发送到上位机。这个方式无需上位机进行 Modbus 的指令查询, 对于基于 Internet 的网络模拟量监控非常有用。

可以设置模拟量的上报时间, 时间间隔从 0~65535 可以选择, 单位是 ms, 如果设置 0 表示不启用主动上报。在 IO 控制器对话框中可以直接设置。



图 11 在 IO 控制器对话框设置 AI 主动上报时间

AI 主动上传的指令为:

- 转化协议为 Modbus RTU 时: 01 04 08 H1 L1 H2 L2 H3 L3 H4 L4 C1 C2。
- 转化协议为 Modbus TCP 时: :00 00 00 00 00 0B 01 04 08 H1 L1 H2 L2 H3 L3 H4 L4

这里 H1 L1 表示 A1 的采集量, H2 L2 表示 A2 的采集量, 以此类推, 大端格式。C1、C2 是 CRC。

AI 主动上报之前如果有设备参数搜索, 则 AI 主动上报会暂停 5 秒钟, 这可

以防止 AI 主动上报和参数搜索冲突。

5.10 DI 和 AI 同时上传



图 12 DI 和 AI 同时主动上报设置

在软件中，当设置启用 DI 主动上报设置为大于 1 的数值的时候（2~255），此时设置值-1 再乘以 5 就是上报 AI 和 DI 的周期。例如设置为 201 的时候上报周期就是 $(201-1)*5=1000$ ms。

*V28 及以上主动上报时间大于 100 时，例如为 108，则是(108-100)秒上报。

该功能允许同时上报 AI 和 DI 的当前数值。上发的 Modbus RTU 格式如下：

00 04 0A 03 01 00 00 00 00 00 00 08 2f a5

其中第一个 00 为 DI 上报地址所设置，使用 04 功能码上报 4 个 AI 寄存器和 4 个 DI 数据。其中 03 01 表示 AI1 的数据，08 是 4 个 DI 的状态， 08 表示第 4 路为 1。

AI 和 DI 同时上报的时候，在 IO 控制器界面可以同时看到 AI 的数据和 DI 的数据，此时不需要点击“自动”查询也能看到数据。AI 和 DI 主动上报会同时发往-IO，+IO 和 4G。

DI 和 AI 主动上报之前如果有设备参数搜索，则 DI 和 AI 主动上报会暂停 5 秒钟，不至于 DI 和 AI 主动上报和参数搜索冲突。

5.11 DI 触发电平再保持 2 秒功能

ZLAN6408 的 V30 开始版本支持写 DI 触发电平消失后再保持 2 秒功能，即触发 DI 为 1 后即使 DI 状态已经变成 0 了，还是会将 1 再保持 2 秒。例如触发 DI 为 0.5 秒，实际读到的 DI 状态为 1 的时间为 2.5 秒（0.5 秒加 2 秒）。

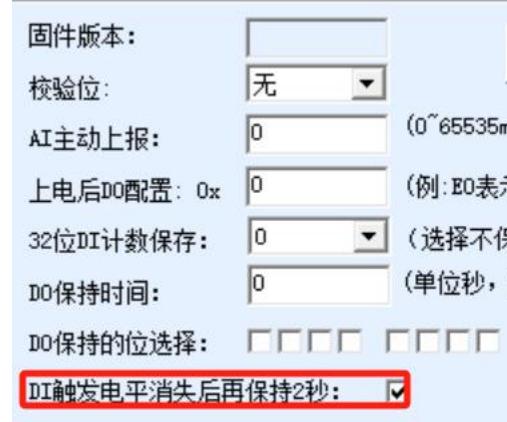


图 13 DI 保持 2 秒功能

6. 4G 配置方式 (6408-8)

网关可以通过串口配置参数。

6.1 串口 AT 指令

下载 ZLVircom 配置工具，这个软件可以通过串口配置采集网关。

将 USB 转 RS485 线连接到采集网关的 RS485 的-4G 和+4G 接口，同时给采集网关上电，打开 ZLVircom (下面简称配置工具)，进入配置工具主界面图 14。

点击设备管理，选择串口搜索，如图 15，弹出串口参数选择界面，如图 16，选择串口号，这里为 COM15，波特率为 115200，这里的 115200 是出厂的默认设置，如果用户之前将网关设置为其它波特率的（比如 9600），也能够搜索到。

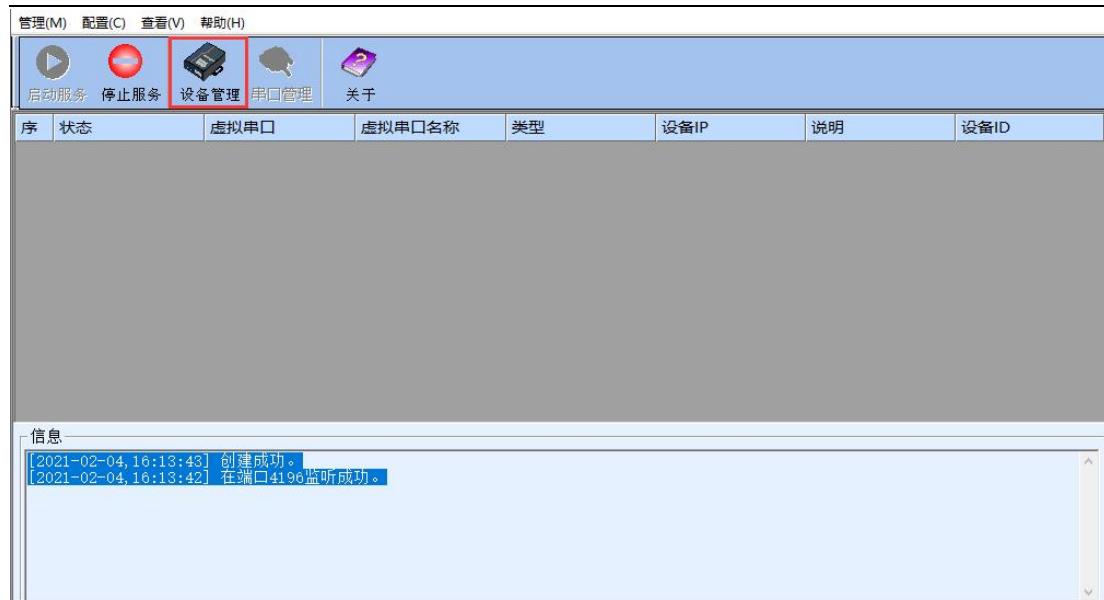


图 14 配置工具主界面

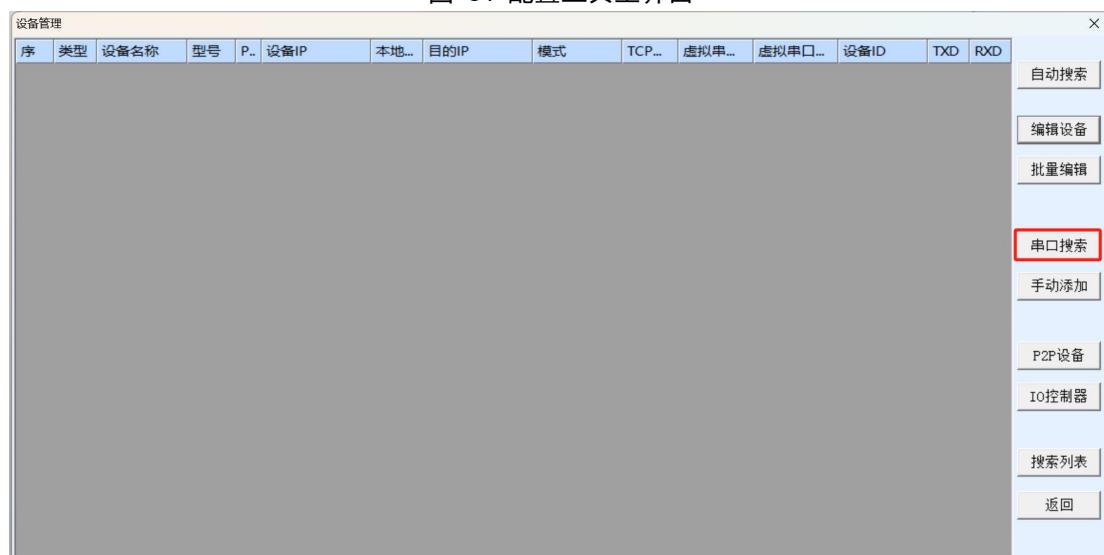


图 15 串口搜索界面



图 16 串口参数设置

上电后等待 15~20 秒，即拨号灯开始闪烁之后，选择点击“搜索”串口，此时配置工具会和设备尝试进行通讯，如果成功则会进入 ConfTool 界面。如下图 17 所示：

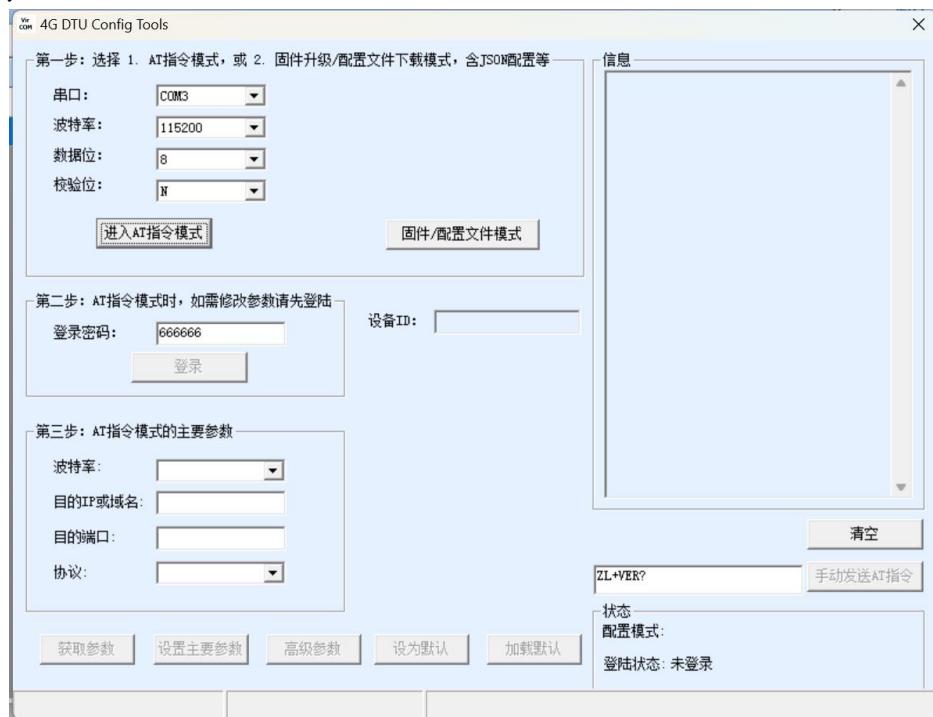


图 17 ConfTool 界面

点击进入 AT 指令模式，配置工具会和设备尝试进行通讯，通信成功，右侧会显示 AT 指令返回信息，配置模式显示为已进入配置模式，如下图 18：

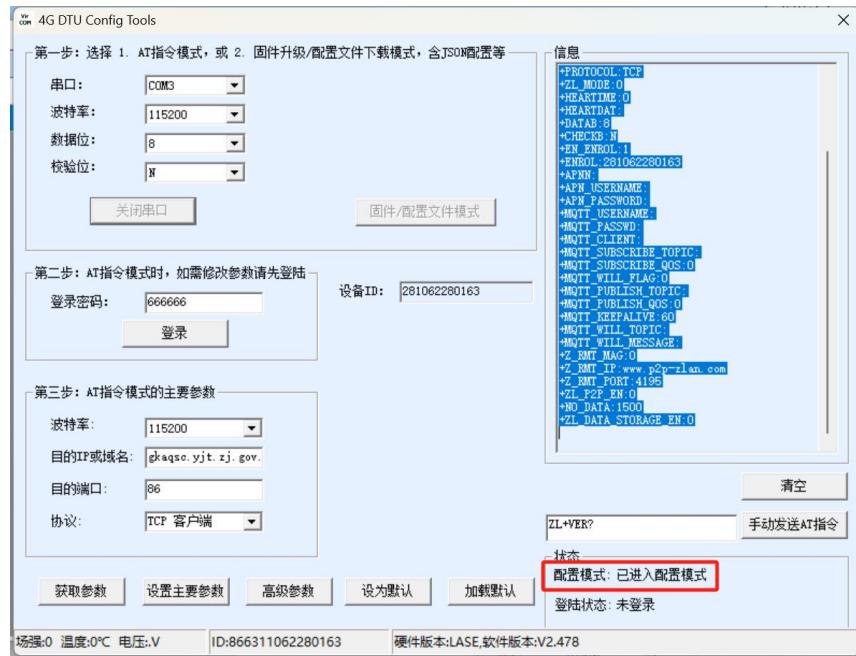


图 18 进入配置模式界面

登录密码默认为 666666，在点击“登录”之前，参数都是只读，无法设置或修改。点击“登录按钮”：

可以看到登录后，登录状态变为“已登录”，如图 19 所示。



图 19 登陆界面

AT 指令模式的主要的参数，包括波特率、目的 IP、目的端口和协议。协议支持 TCP 或者 UDP 协议。修改相应的参数后点击“设置参数”可以将新的参数设置到设备，同时设备会返回设置成功的参数，如图 20 所示。

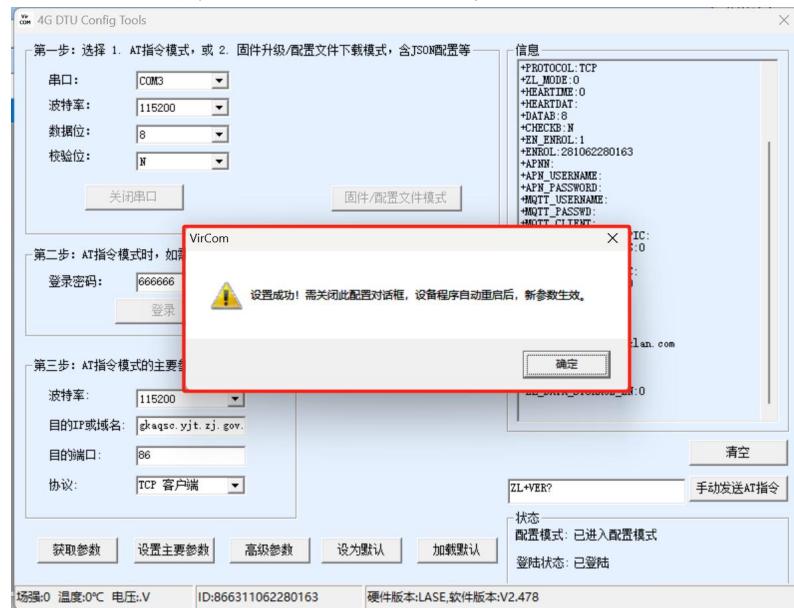


图 20 设置参数

“获取参数”按钮可以获得当前的设备的参数，获取参数是通过发送 AT 指令来获得参数的，右边列出的就是 AT 指令的返回数据。关于 AT 指令，可以参考本文的其它章节。由于“打开”成功之后会自动执行一次“获取参数”，所以一般不用点击“获取参数”按钮。

“获取参数”按钮可以获得当前的设备的参数，获取参数是通过发送 AT 指令来获得参数的，右边列出的就是 AT 指令的返回数据。关于 AT 指令，可以参考本文的其它章节。由于“打开”成功之后会自动执行一次“获取参数”，所以一般不用点击“获取参数”按钮。

点击“高级参数”，高级参数框如图 21 所示，常用的参数是：

1. 心跳间隔：可以设置间隔为 15s 的心跳包。
2. 心跳内容：设置心跳包内容。
3. 串口数据位
4. 串口校验位
5. 启用注册包：是否启用注册包。
6. 注册包内容：连接上服务器之后发送的注册包内容。

7. APN: APN 的接入点名称。
8. APN 用户名
9. APN 密码
10. MQTT 参数: 用于设置接入 MQTT 服务器的参数
11. 设备远程管理: 用于带有远程管理功能的设备接入远程服务器

选择好参数后点击“生效高级参数”按钮, 观察右侧信息栏查看设备返回的设置信息与填入的信息是否一致, 如图 22 所示。



图 21 高级参数

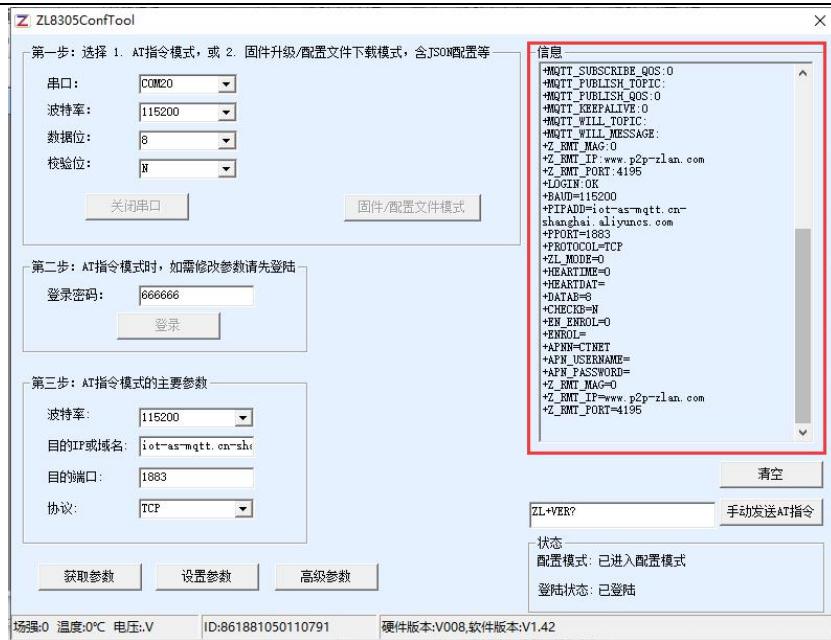


图 22 设置高级参数返回信息

6.2 固件/配置文件模式

进入 ConfTool 界面后，点击固件/配置文件模式按钮，如图 23 所示。

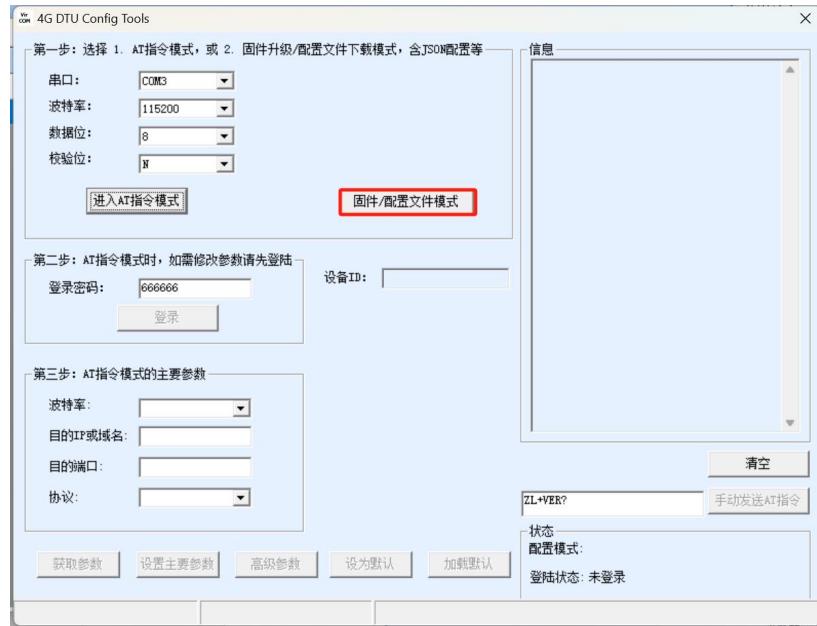


图 23 配置界面

跳转到固件/配置文件界面，先创建本地配置网页根目录存放配置文件，如果需要 MQTT 传输，那么就需要点击 MQTT 配置可以输入连接 MQTT 服务器

的信息，设置完成点击保存 MQTT 配置。



图 24 网页配置页面



图 25 MQTT 配置界面

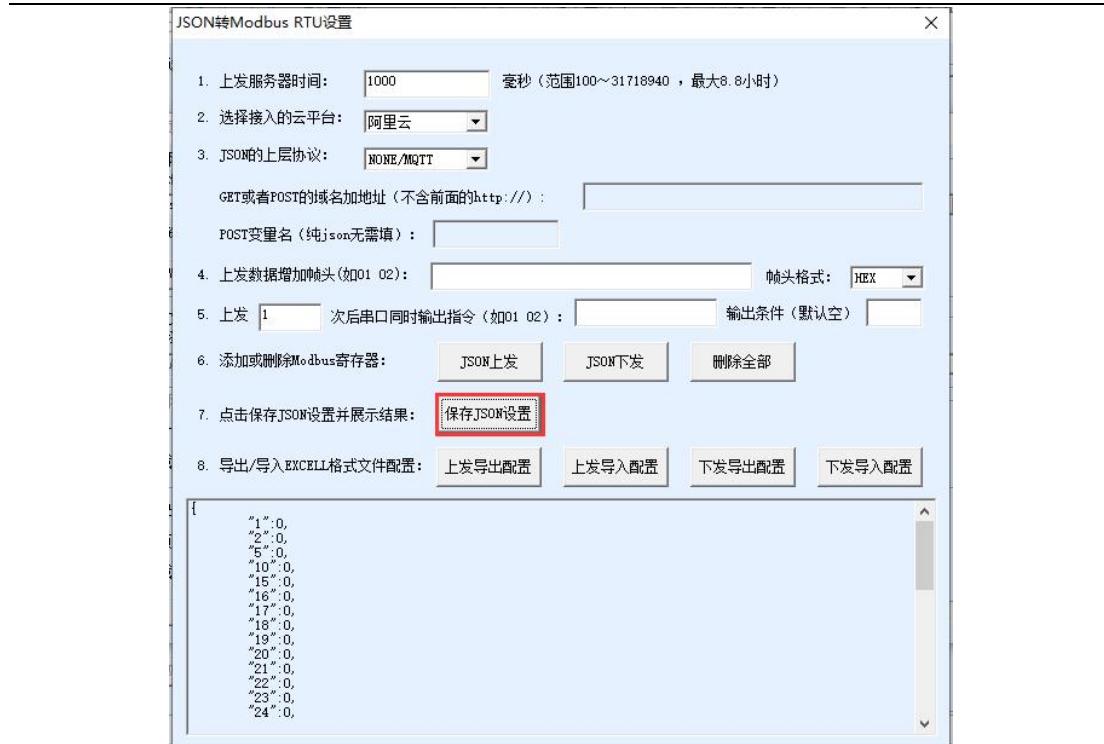


图 26 JSON 配置界面

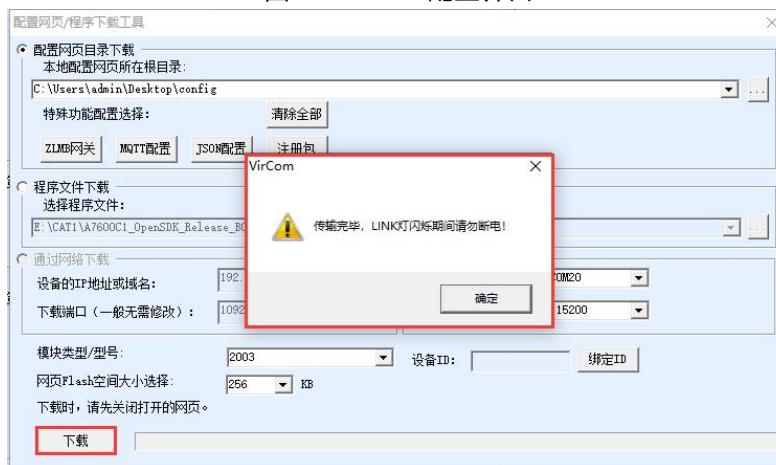


图 27 下载界面

7. 配置与使用方法 (6408-8)

7.1. 基础参数设置

参考 6.1 串口 AT 指令, 进入 AI 指令模式后可以配置网关的波特率, IP 地址, 端口等参数。配置完毕后点击设置主要参数。

7.2. MODBUS RTU 转 JSON 配置

7.2.1. 软件配置

参考 6.2 固件/配置文件模式，进入 ConfTool 界面后，点击固件/配置文件模式按钮，再点击 JSON 配置，进入配置页面。

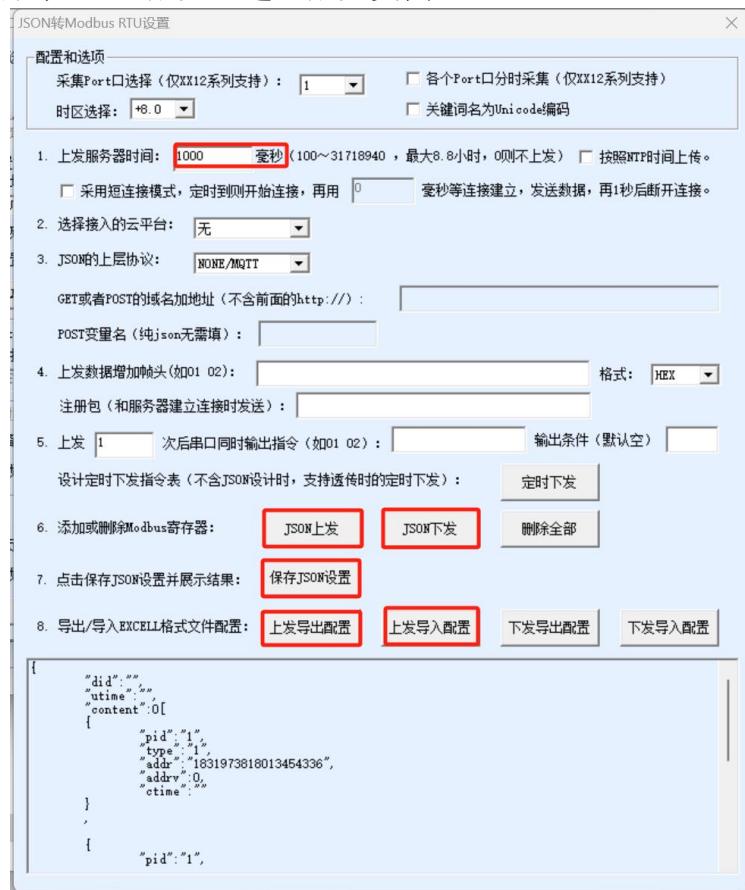


图 28 JSON 配置

如上图所示，几个较为重要的参数介绍如下：

上发服务器时间：默认 JSON 数据每隔多少时间上发到服务器，单位毫秒。

添加或删除 Modbus 寄存器：点击 JSON 上/下发可以进行 JSON 上/下发参数的配置，点击删除全部会删除全部已经配置的 JSON 参数。

保存 JSON 设置：点击后可以保存 JSON 设置并展示结果。

上发导入导出：将 JSON 上发配置导出 CSV 表格或者将 CSV 表格导入。

点击 JSON 上发：可以进入 JSON 配置页面。

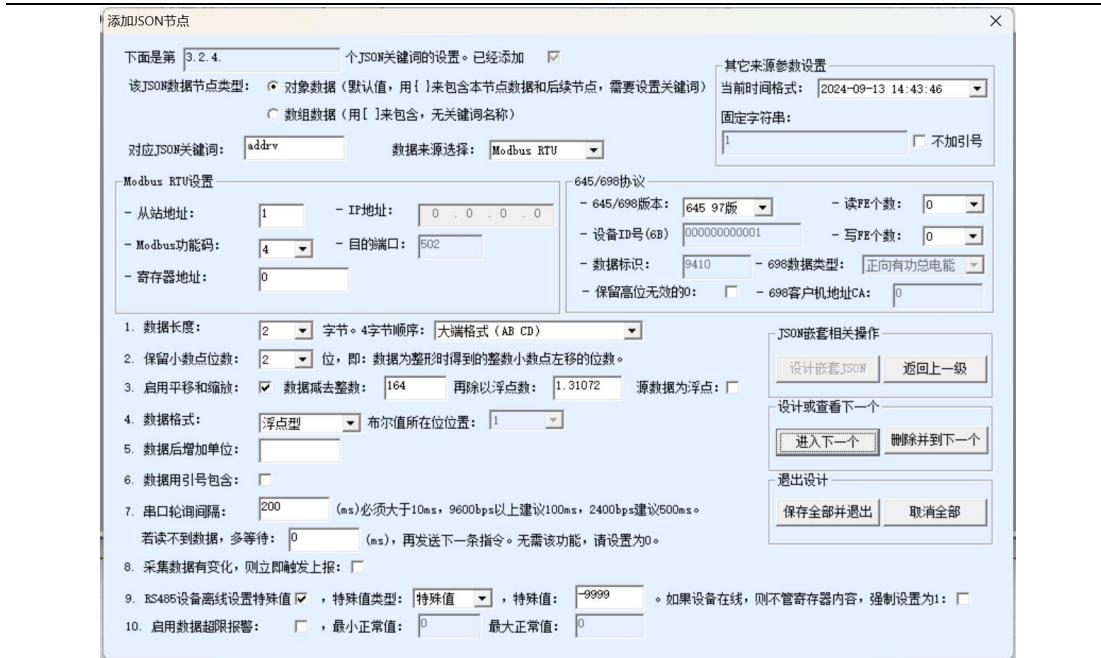


图 29 JSON 上发

如上图所示，几个较为重要的参数介绍如下：

1. 下图是第 3.2.4. 个 JSON 关键词：这里的“3.2.4”表示现在的设计界面第几个 JSON 关键词。
2. 已经添加过：如果打勾说明是已经添加的，查看已经配置的信息的时候会出现打勾的情况，表示处于编辑状态。如果出现未打勾的则是处于添加状态。
3. 对应 JSON 关键词：这个 JSON 节点的名字。
4. 数据来源：选择 JSON 数据的来源
 - a) Modbus RTU：例如 addrv:123.45 的形式，表示数据来源于某个 Modbus RTU 表并通过串口采集。图中左半部份都是设计 Modbus RTU 相关参数。
 - b) 固定字符串：例如 DevName:"MyDev" 的形式，在右侧的固定字符串中输入 MyDev，JSON 名字是 DevName，这样就可以产生固定字符串的 JSON 节点。
 - c) 设备 ID：如果 JSON 节点名字为 DevID，则上发节点字符串为 DevID:"285301020304" 其中的“285301020304”就是设备的 mAC 地址或者唯一编号。
 - d) 当前时间：如果 JSON 节点名字为 ColletTime，则上发字符串为 ColletTime:" 2019-05-13 22:23:31"。时间是系统通过 NTP 协议获得时间。

- e) 嵌套 JSON：如果节点名称为 Alarm，则它上发格式具有 Alarm:{temp1:"25.1",temp2:"26.2"} 的形式，即 Alarm 的内容还是一个 JSON 集合。

5. Modbus 相关设置

- a) 从站地址：Modbus 的表地址。
- b) Modbus 功能码：目前支持 01.02.03.04 功能码。
- c) 寄存器地址：这里对应的 0。
- d) 数据长度：这里对应是 2 字节。
- e) 启动平移和缩放：可以对获取到的值进行先减再除的运算。
- f) 数据格式：这里对应是浮点型（因为启用了平移缩放，所以 2 字节也得设置为浮点型）。
- g) 保留小数点：这里保留 2 位。
- h) 串口轮询时间：这里设为 200ms。是指这个寄存器和下个寄存器的轮询的间隔，而不是这条指令的轮询间隔。
- i) 离线设置特殊值：当串口不返回数据时将数据设置为特殊值例如-9999。

6. 固定字符串：来源选择为固定字符串的时候可以输入字符串内容。

7. 按钮

- a) 嵌套 JSON：当当前节点来源选择为“嵌套 JSON”类型的时候，必须点击这个按钮进入嵌套 JSON 的设计，如果当前为“2.”，则会进入“2.1”这个节点的设计。
- b) 返回上一级：如果当前节点的处于第 N 级嵌套，则点击这个按钮会返回到 N-1 级节点的设计，且停留在 N-1 级的一个新增节点上。
- c) 设计下一个：点击后进入本地的下一个 JSON 节点，如果之前设计不存在下一个节点则“已经添加过”的勾会取消，表示处于新增节点。
- d) 保存设计：完成设计，在最后一个设计节点界面时，点击“保存设计”。之后返回主界面，然后点击“保存 JSON 配置”即可。
- e) 取消设计：取消当前的所有设计，如果是查看设计内容，可以点击这个按钮退出。

7.2.2. excel 方式编辑

JSON 参数也可以在 CSV 表格中配置，CSV 表格如下，当已经有 CSV 模版的时候只需要编辑部分需要变动的参数即可，例如 modbus 寄存器一栏。

序号	JSON关键词	数据来源	固定字符串	Modbus从站	Modbus寄存器ID	数据类型	数据长度	大小端格式	数据格式	布尔值所在位	保留小数点	数据后增加数据	用引号串	轮询间隔	645版本	功能项	(1为JSON数)	平移	缩放	平移缩放	缩放	读FE个数	
1	1 did	设备ID	2.81E+11	1	3	0	1D:00000000	9410	2	0 无符号整型	0	0	0	100	97	0	0	0	0	0	1	C	
2	2 utime	当前时间	2.83E+11	1	3	1	1D:00000000	9410	2	0 无符号整型	0	0	0	100	97	0	0	0	0	0	1	C	
3	3 content	解析JSON	2.3E+11	1	3	2	1D:00000000	9410	2	0 无符号整型	0	0	0	100	97	0	0	0	0	0	1	C	
4	5 3.1.	解析JSON	2.3E+11	1	3	3	3D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	1	0	0	0	1	C
6	6 3.1.1.	pid	固定字符串	1	1	3	4 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	1	0	0	0	1	C
7	7 3.1.2.	type	固定字符串	1	1	3	5 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
8	8 3.1.3.	addr	固定字符串	1.83E+18	1	3	6 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
9	9 3.1.4.	addrv	Modbus RTU	0	1	1	7 D:00000000	9410	2	0 布尔型	0	0	0	0	200	97	1	0	0	0	0	1	C
10	10 3.1.5.	ctime	当前时间	0	1	3	8 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
11	11 3.2.	pid	固定字符串	1	1	3	9 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
12	12 3.2.1.	pid	固定字符串	1	1	3	10 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
13	13 3.2.2.	type	固定字符串	1	1	3	12 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
14	14 3.2.3.	addr	固定字符串	1.83E+18	1	2	13 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
15	15 3.2.4.	addrv	Modbus RTU	1	1	3	0	1D:00000000	9410	2	0 浮点型	0	2	0	200	97	1	0	0	164	1.31072	C	
16	16 3.2.5.	time	当前时间	0	1	3	8 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
17	17 3.3.	嵌套JSON	1	1	3	17 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C	
18	18 3.3.1.	pid	固定字符串	1	1	3	18 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	1	0	0	0	1	C
19	19 3.3.2.	type	固定字符串	0	1	3	19 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
20	20 3.3.3.	addr	固定字符串	1.33E+18	1	2	20 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
21	21 3.3.4.	addrv	Modbus RTU	203.18	1	4	1 D:00000000	9410	2	0 浮点型	0	2	0	200	97	1	0	0	164	32.76799	C		
22	22 3.3.5.	ctime	当前时间	0	1	3	8 D:00000000	9410	2	0 无符号整型	0	0	0	0	100	97	0	0	0	0	0	1	C
23																							

图 30 CSV 表格

编辑完毕后点击上发导入配置，选择编辑完毕的 CSV 文件，再点击导入，

导入完毕后点击保存 JSON 设置。

一般建议在已有的模版基础上进行修改。或者先在 vircom 配置软件上配置一组参数后再进行导出。在导出的 CSV 表格的基础上进行复制修改。

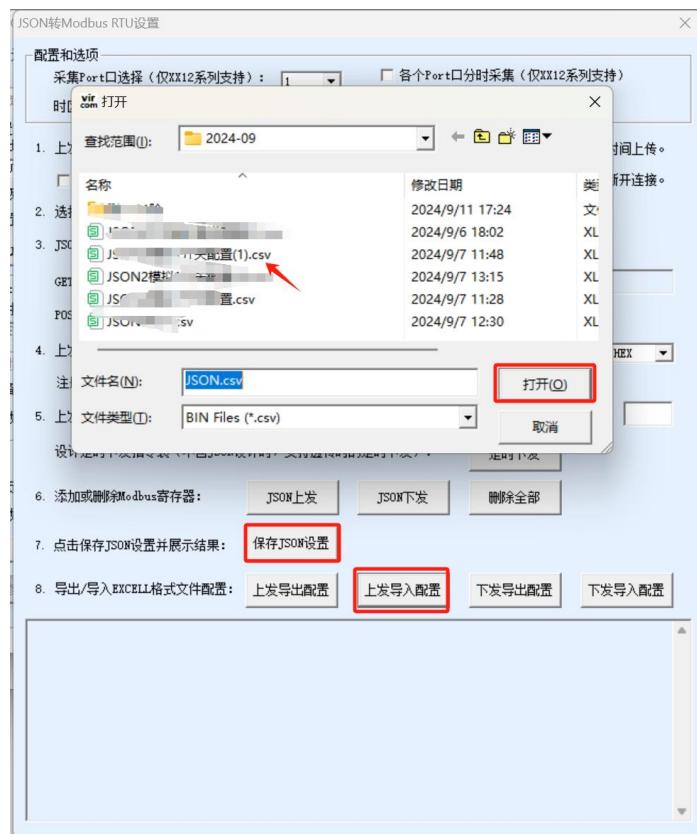


图 31 CSV 导入

7.3. 服务器透明传输测试

假设有如下的联网结构如下图，6408 配置为连接到服务器***.***.***.***的***端口。请通过“配置方法”一节的方法进行配置。配置完成后，重新上电，需要 20~40 秒才能连接上服务器。

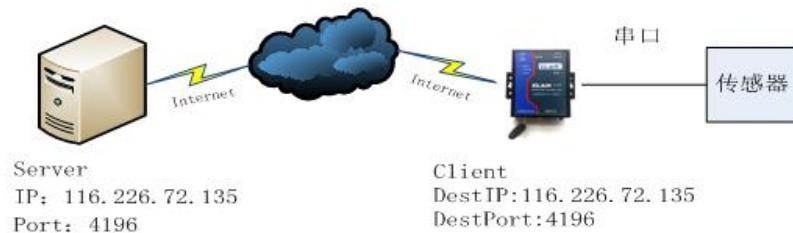


图 32 联网结构图

我们在服务器上运行 SocketDlgTest 这个 TCP 工具 (http://www.zlmcu.com/document/tcp_debug_tools.html)。



图 33 服务器端工具

如图选择本地端口为 4196 (注意如果运行 ZLVircom 工具，则需要换一个端口)，然后点击“打开”按钮。当 6408 设备连接上服务器之后，会显示“The NO... is accepted!” 的信息。

现在将 6408 设备的-4G,+4G 串口连接 USB 转 485 串口线，并且打开串口调试工具 (http://www.zlmcu.com/document/com_debug_tools.html)，并打开正确的

COM 口。

现在串口发送数据，则在服务器端会回复相应的数据，同样在设备接收服务器回复的消息通过串口输出，串口工具这里收到同样的数据。这样就演示了串口到 4G 的网络双向通讯，如下图 34 所示：

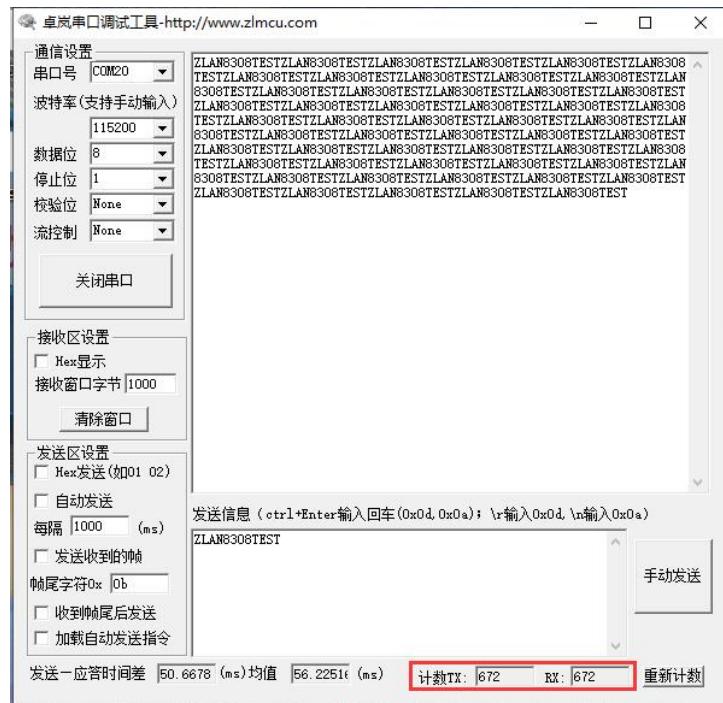


图 34 设备端串口调试工具

7.4. Modbus 协议转化测试

配置参数与无协议透传测试基本一样，只需要将转换协议更改为 MODBUS 协议即可。

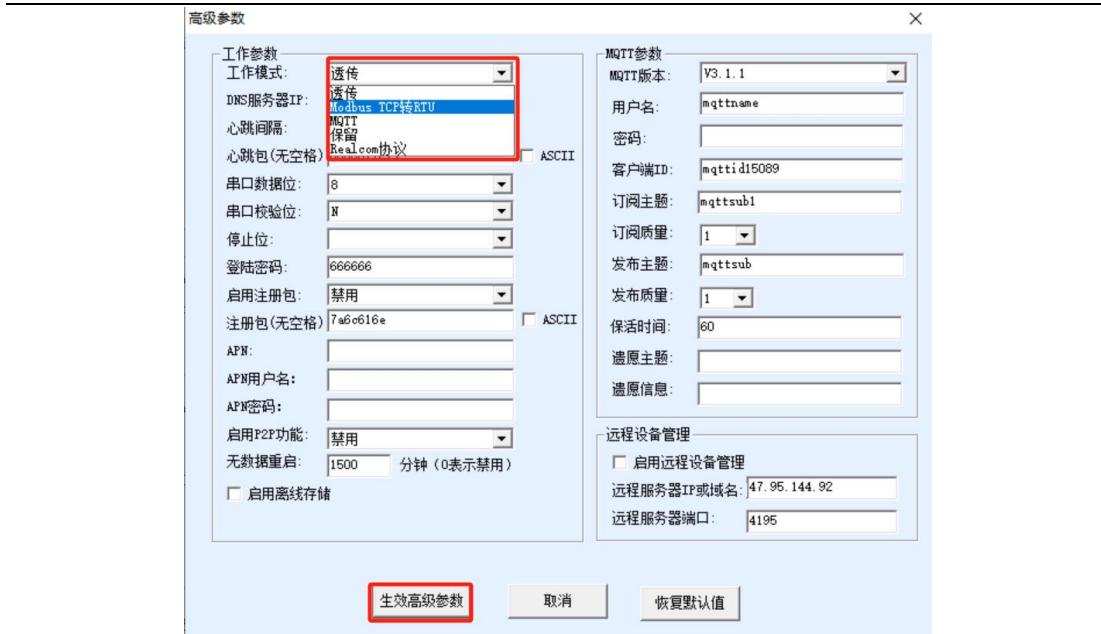


图 35 Modbus 协议转化设置

即可实现串口 MODBUS RTU 协议转换为网络的 MODBUS TCP 协议，将网络的 MODBUS TCP 协议转换为串口的 MODBUS RTU 协议。

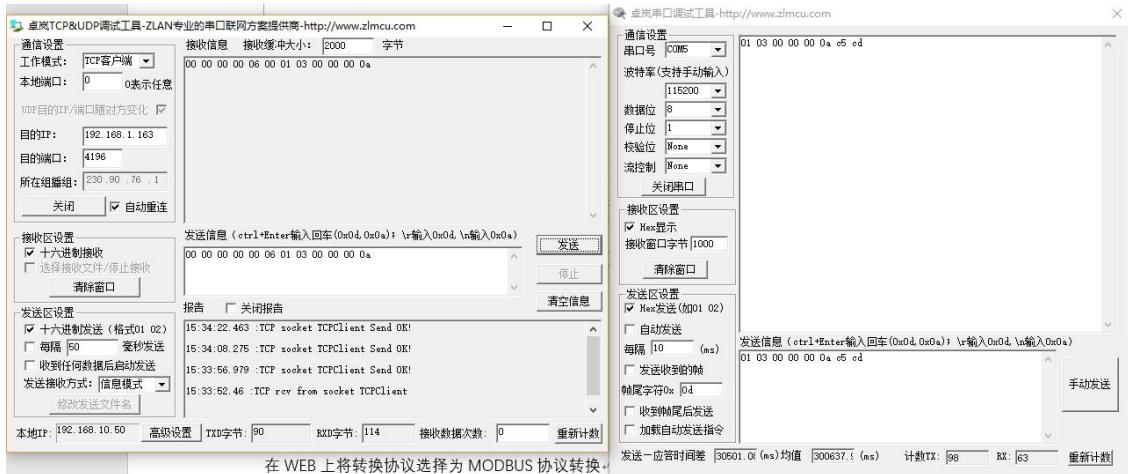


图 36 Modbus 协议转化测试

7.5. MQTT 协议测试

本次测试为连接阿里云做测试。在阿里云上新建一个名为 zlan_test 的订阅主题，一个名为 zlan_1 的发布主题，如图 37 所示。按照说明，先将 MQTT 服务器的 IP 和端口配置填入，保存参数，参数填写如图 38 所示。再通告高级参数页面将 MQTT 的 ID、用户名、密码包括订阅发布主题，保活时间填入，参数填

写如图 39 所示。注意选择工作模式为 MQTT 模式。

自定义 Topic	操作权限	描述
/a1WSVHIXkDh/\${deviceName}/user/zlan_test	订阅	-
/a1WSVHIXkDh/\${deviceName}/user/zlan_1	发布	-

图 37 阿里云添加主题

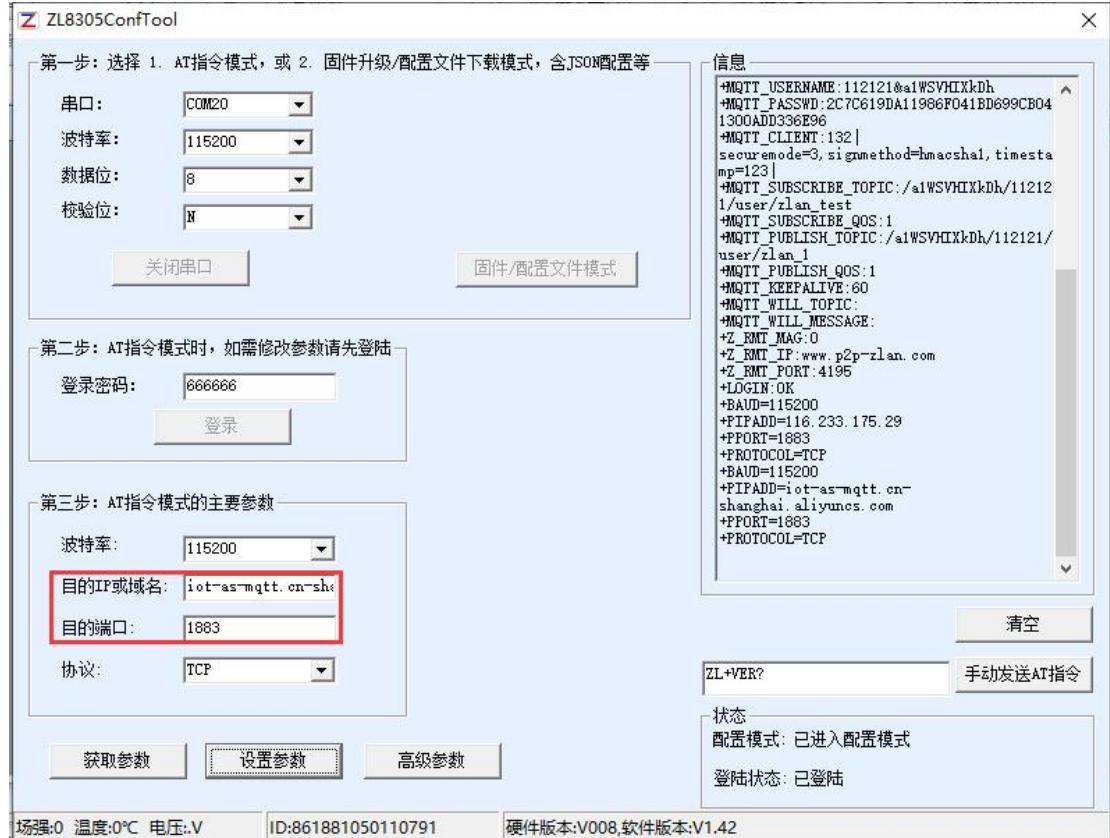


图 38 阿里云 IP 和端口

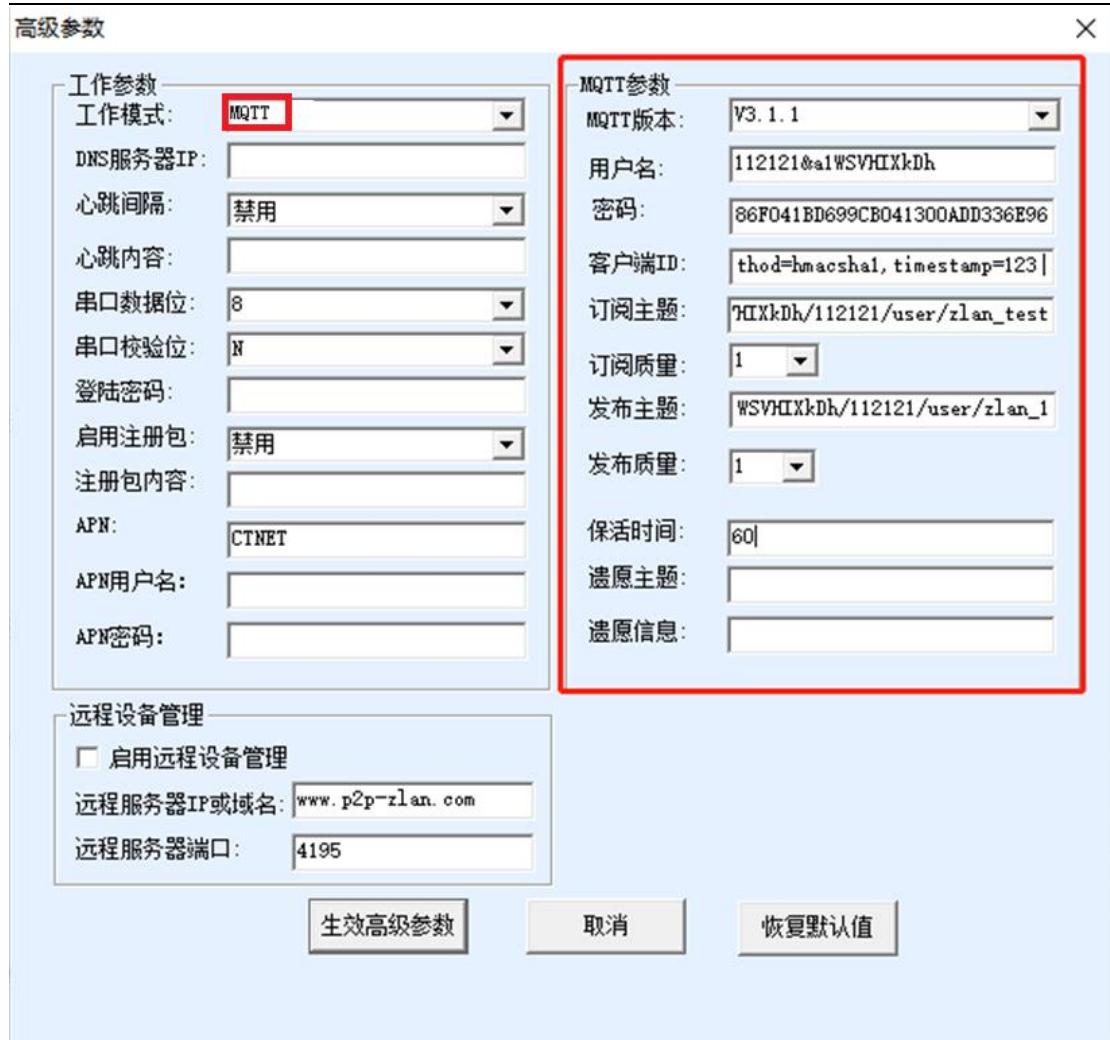


图 39 阿里云 MQTT 配置

设置完成后打开阿里云设备管理界面，进入日志服务页面查看设备上发的信息，如图 40 所示。通过设备的串口发送数据，通过 zlan_1 的主题发送消息 (“ZLAN8308TEST”) 到阿里云的 MQTT 服务器，阿里云收到数据如图 41 所示，阿里云服务器通过 zlan_test 主题发送消息 (“ALI_send”) 到设备串口，如图 42 所示，这就完成了 MQTT 收发测试。

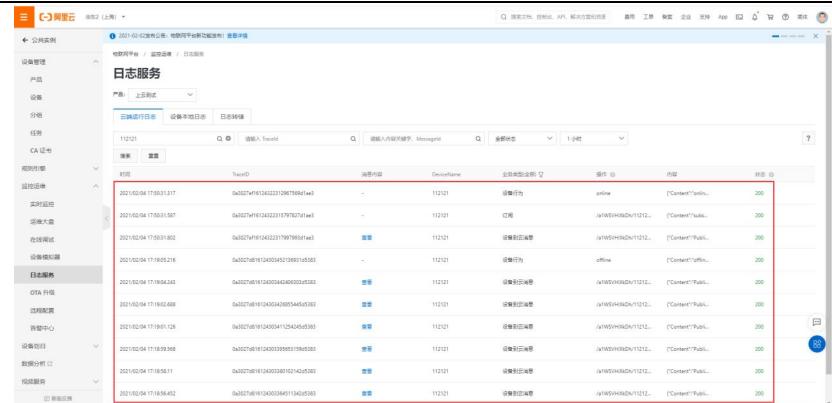


图 40 阿里云日志服务

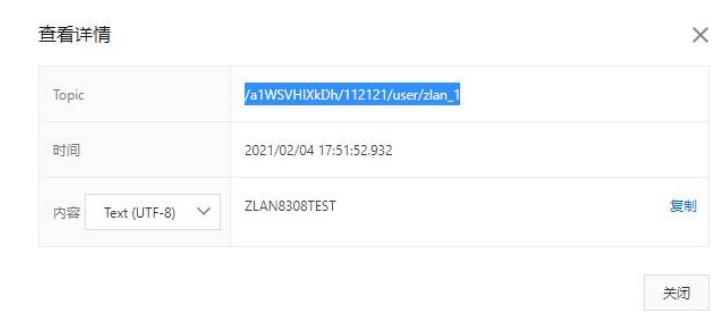


图 41 阿里云接收到串口数据

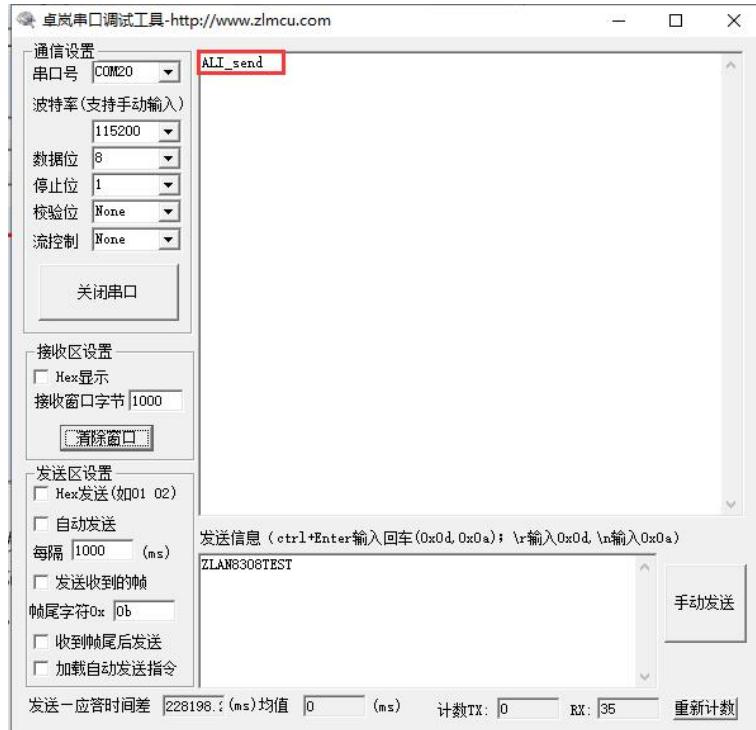


图 42 串口接收到阿里云数据

附录 1：参数总表

本章内容主要涉及参数设置和读取的技术细节。并有助于用户用自己的软件配置和修改参数。对于普通应用，可以略过此节内容。

从寄存器总表中将参数读取和设置相关的单独列出来如下。

表 2. 参数相关读取操作

功能码	功能	地址范围	地址范围
03	读基础参数	63~67	63~67
03	读扩展参数	68~162	68~162
06	设置参数	63~67	63~67
06	设置扩展参数	68~162	69~162
16	设置基础参数	63~67	63~67
16	设置扩展参数	68~162	68~162

从表中可知，参数使用 03 功能码读取，使用 06 和 16 指令设置。参数分为基础参数和扩展参数 2 个部分，分别对应寄存器 63~67 和 68~162。

表 3. 基础参数寄存器

寄存器地址	参数名	长度(字节)	说明
63(0x3F)	addr/设备地址	1	寄存器值的高字节
63(0x3F)	upLoad/启用 DI 主动上报	1	寄存器值的低字节，1 表示启用，2~255 表示按周期上报。
64(0x40)	dst_addr/DI 上报地址。	1	寄存器值的高字节
64(0x40)	baud/设备波特率	1	寄存器值的低字节，只设置 485-IO 的 RS485 接口的波特率。 1200 0; 2400 1; 4800 2 9600 3; 19200 4;

			38400 5; 57600 6; 115200 7
65(0x41)	ver/固件版本	1	寄存器值的高字节, 只读
65(0x41)	复合参数设置	1	寄存器值的低字节。 Bit1: 32 位 DI 计数保存, 1 表示保存 Bit2: DI 逻辑反转, 1 表示反转 Bit3: DI 延迟功能, DI 变为 1 之后, 从 DI 输入端变为 0 之后还持续保持 2 秒为 1, 即 2 秒内还能读到 DI 为 1。选择该功能后, DI 增加 50ms 的防抖功能。
66(0x42)	A1UpLoadH/AI 上报周期高字节	1	寄存器值的高字节
66(0x42)	A1UpLoadL/ AI 上报周期低字节	1	寄存器值的低字节
67(0x43)	A2UpLoadH/AI 上报周期高字节	1	寄存器值的高字节, 请和 A1UpLoadH 设置一样值
67(0x43)	A2UpLoadL/ AI 上报周期低字节	1	寄存器值的低字节, 请和 A1UpLoadL 设置一样值

表 10. 扩展参数寄存器

寄存器地址	参数名	长 度 (字节)	说明 (DO 无效)
68(0x44)	dostate/ 上电后 DO 配置	1	寄存器值的高字节, 0xF0 表示后 4 路吸合
68(0x44)	checkb/校验位	1	寄存器值的低字节。 0: 无校验 1: 奇校验 2: 偶校验

			3: 标记 4: 空格
69(0x45)	baud_UART_0_2/网络通讯和 485-4G 的波特率	1	寄存器值的高字节, 目前只读, 是通过网络模块自适应, 无需设置。
69(0x45)	datab/数据位	1	寄存器值的低字节。留待后续扩展。
70(0x46)	stopb/停止位	1	寄存器值的高字节, 留待后续扩展
70(0x46)	TCP_LINK_FLAG/预留	1	寄存器值的低字节。留待后续扩展。
71(0x47)	FirmwareType/设备类型	1	寄存器值的高字节。 0: 6002/6042 1: 6808-1(6408-1) 3: 6808-2, 6808-3, 6808-8, 6808-7, 6808(6408-8) 4: 6802/6842 9: 6808-9
71(0x47)	DO 保持时间	1	寄存器值的低字节。DO 状态保持的时间长度。
72(0x48)	DI 控制自身 DO	1	寄存器值的高字节的第一位(Bit0)。 1: 开启 0: 关闭 (无效)
72(0x48)	上电后保持上次断电前的 DO 状态	1	寄存器值的高字节的第二位(Bit1). 1: 开启;0: 关闭
72(0x48)	AI 主动上报改为上报高精度数值	1	寄存器值的高字节的第三位(Bit2). 1: 开启;0: 关闭

72(0x48)	写 DO 无返回指令	1	寄存器值的高字节的第四位(Bit3). 1: 开启;0: 关闭
72(0x48)	reserver/预留	1	寄存器值的低字节。留待后续扩展。
73(0x49)	reserver/预留	2	留待后续扩展。
74~89 (0x4a~0x59)	V1~V8是AI每路的 调整系数	32	大端格式数据, 具体参考“AI 高精度使用章节。”
90 (0x5a)	AI 校准状态	2	1 表示处于 AI 校准状态
91~98 (0x5b~6a)	32 位计数	32	共 8 个寄存器, 4 个 DI, 每个 2 个寄存器。
107 (0x6b)	单/多路 DO 保持	2	设置 DO1-DO8 单路/多路是否 保持
108~130 (0x6c~82)	预留	46	共 23 个寄存器
131~162 (0x83~a2)	DI 组合对 DO 逻辑 控制	32	共 16 个寄存器

附录 2: AI 校准

步骤: 以 RS485-IO 串口通讯方式校准为例

1. 发送 01 06 00 5a 00 01 68 19, 设置“AI 校准状态”为 1, 进入校准模式。
2. 发送 01 04 00 00 00 04 f1 c9 查询 4 路 AI 的数据。对收到的数据 01 04 08 **02 81** 00 00 00 00 00 34 dc, 计算每路 V1~V4 的数值。例如对于第一路。
 - a) 数值为 **02 81**, 转换为十进制为 Vin=641, 按照“AI 使用说明”中的公式计算输入点电压的为: $Vi = (Vin / 1024) * 5$, 其中 Vin 就是 641, Vi 为已知电压, 例如 3.3V。这样调整系数 $V1 = Vi / ((Vin / 1024) * 5) = 3.3 / ((641 / 1024) * 5) = 1.0543525$ 。
 - b) 将 V1 用 float 型数据表示, 并且转化为 HEX 大端格式为 0x3F86 F506。

- c) 将 0x3F86 写入 V1 对应的第一个寄存器 0x4a，将 0xF506 写入 V1 对应的第二个寄存器 0x4b。即发送 01 06 00 4a **3f 86** 38 4e 和 01 06 00 4b **f5 06** 3e 8e。
3. 发送 01 06 00 5a 00 00 a9 d9 退出校准模式。

使用 ZLVircom 的“IO 控制器”对话框的“AI 校准功能”用户可以自行校准。但是每台 ZLAN6408 设备都是出厂后已经用专业一起校准，如无必要，用户无需进行校准。校准步骤如下：

1. 在型号中请选择正确的产品子型号：只有选择了正确的型号，才能确定每路的 AI 类型是 5V、10V、4~20mA。才能进行校准。



图 43 AI 校准

2. 在校准第几路里选择将要校准的路。因为用户可能无法同时连接 4 个测试点，所以一路一路调准较为方便。
3. 点击“AI 校准”按钮，系统开始校准。校准完毕后 AI 的数值更加准确。校准完毕无需重启，系统自动保存校准参数。

售后服务和技术支持

上海卓岚信息技术有限公司

地址：上海市闵行区园文路 28 号金源中心 2001 室

电话：021-64325189

传真：021-64325200

网址：<http://www.zlmcu.com>

邮箱：support@zlmcu.com