

# 卓岚 MQTT 和 JSON 转 Modbus 网关用法

嵌入式设备联网解决方案

版权©2008 上海卓岚信息科技有限公司保留所有权力

ZL DUI 20190425.1.0



版权©2008 上海卓岚信息科技有限公司保留所有权力

## 版本信息

对该文档有如下的修改：

			修改记录
日期	版本	文档编号	修改内容
2019-4-25	Rev.1	ZL DUI 20190425.1.0	发布版本

## 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 .....	4
2. JSON.....	4
2.1. Modbus RTU 转 JSON.....	4
2.2. Modbus 表 .....	4
2.3. 设备配置 .....	5
2.4. 新建 Modbus 模拟表 .....	9
3. MQTT.....	10
3.1. 设备配置 .....	10
3.2. 数据测试 .....	14
4. MQTT+JSON 转 MODBUS RTU .....	16
5. 售后服务和技术支持 .....	16

## 1. 概述

本文介绍卓岚支持 MQTT 和 JSON 的设备的用法，支持的设备包括 ZLAN5143、ZLAN8303-7、ZLAN1043、ZLAN7144 等。其中 ZLAN5143、ZLAN8303-7、ZLAN1043 的固件版本要在 1.577 及以上，ZLAN7144 需要在 1.489 及以上。需要和 zlvircom 的 5.09 版本配合使用。

MQTT 和 JSON 可以单独使用也可以合起来使用。其中的 JSON 支持 Modbus RTU 格式转化为 JSON 格式。

主要特点是：

1. 采用 MQTT 为基础的协议和服务器建立连接，采用订阅发布的形式进行数据通讯。
2. 支持 Modbus RTU 寄存器的自主设计和自动采集。
3. 支持将特定的 Modbus 寄存器内容转化为 JSON 格式定时主动上发。
4. 支持在 JSON 格式中加入设备 ID，方便云端识别设备。
5. 支持无符号数据和有符号数据，支持小数点表示，支持 4 字节长度数据。
6. 所有配置都可以界面化配置完成，用户自主配置无需定制。

## 2. JSON

### 2.1. Modbus RTU 转 JSON

卓岚 Modbus RTU 转 JSON 可以实现 Modbus RTU 表的自动采集，并按照 JSON 格式自动上发给云端服务器。

这里我们通过一个具体的案例来讲解这个用法。

### 2.2. Modbus 表

假设现在有一个功能码为 3，地址为 1 的 Modbus 表，其寄存器地址和参数名如下。其中字节长度为 4 的表示需要连续读 2 个寄存器。

寄存器地址	参数名	字节长度	备注
0	当前总有功电能	4	无符号，保留 2 位小数。
97	A 相电压	2	无符号，保留 1 位小数
98	B 相电压	2	

99	C 相电压	2	无符号，保留 2 位小数
100	A 相电流	2	
101	B 相电流	2	
102	C 相电流	2	
119	频率	2	
356	A 相有功功率	4	有符号，保留 3 位小数
358	B 相有功功率	4	
360	C 相有功功率	4	
362	总有功功率	4	

所谓有符号是指，2 字节或者 4 字节最高 bit 为符号位，例如 0xFFFF 将被认为是-1。保留 2 位小数是指将数据当作整型转化之后，小数点从最右边向左移动 2 位。

### 2.3. 设备配置

我们按照设备作为客户端的方式进行配置。

使用 SocketDlgTest ([http://www.zlmcu.com/document/tcp\\_debug\\_tools.html](http://www.zlmcu.com/document/tcp_debug_tools.html))，在本地计算机 1883 端口监听一个 TCP 服务器。

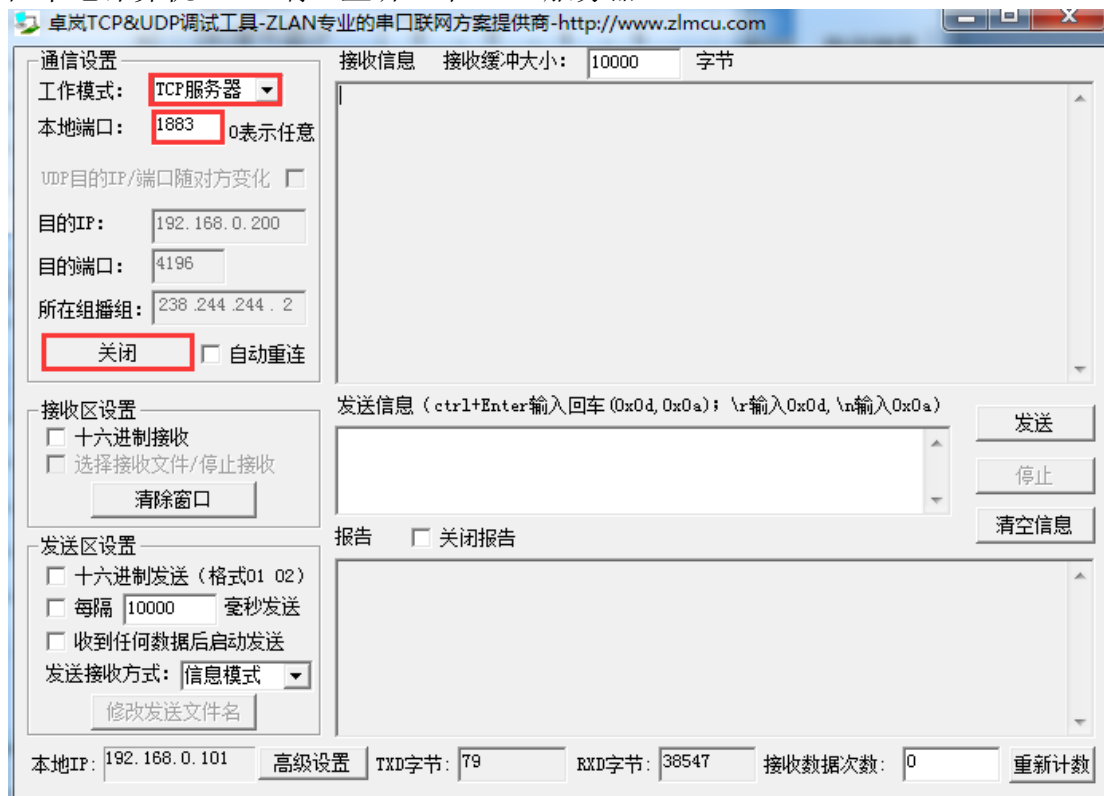


图 1Socket 模拟服务器接收数据

使用 ZLVircom5.09 对设备进行配置。

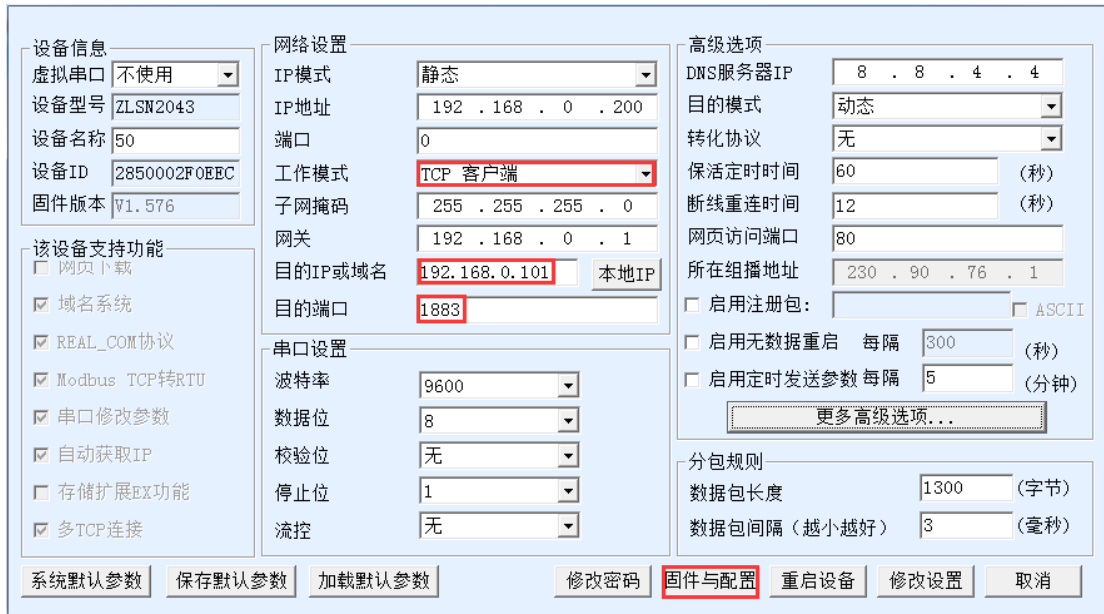


图 2 设备配置

点击修改配置，可以让设备连接上 SocketDlgTest 工具。再次进入设备编辑对话框。点击“固件和配置”按钮。



图 3 下载界面

首先点击“网页目录下载”进入配置下载模式。然后选择一个新的空目录，比如 MQTTHTTPD 目录。防止之前的设计有残留，请先点击“清除全部”按钮，这样可以清空之前的设计内容。设计文件将会保存在这个目录，等一下可以通过点击“下载”按钮下载到设备内部。

点击“JSON 配置”按钮。

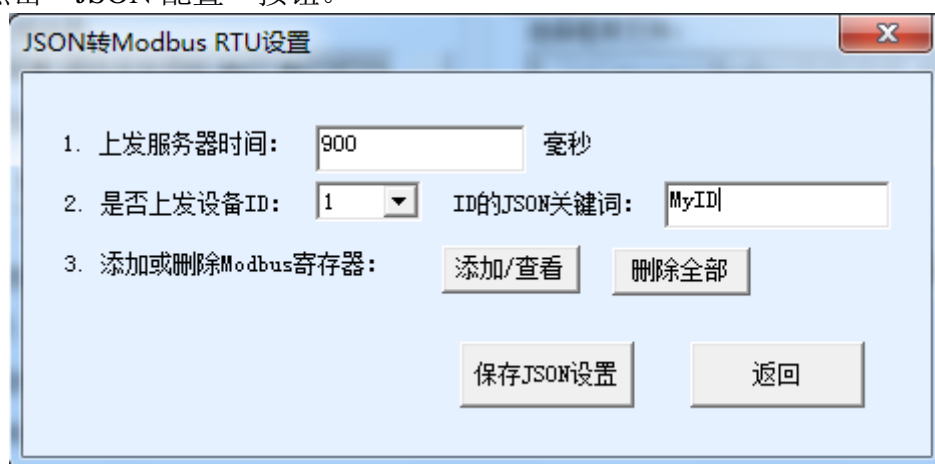


图 4JSON 配置主界面

这里的参数说明如下：

1. 上发服务器时间：默认 JSON 数据每隔多少时间上发到服务器，服务器就是刚刚在设备配置界面设置的 IP，单位是毫秒。
2. 是否上发设备 ID：如果选择 1 则会在每次上发的 JSON 数据中包含有设备的 ID，设备 ID 就是在设备编辑界面左上角看到的设备的 MAC 地址。
3. ID 的 JSON 关键词是自己命名的 ID 对应的的 JSON 关键词，这里写 MyID，最长为 32 字节。这样以后 JSON 格式中会增加“MyID”:”2850002F0EEC”的内容。这类似于注册包，有利于服务器识别是哪一台设备发来的数据。
4. 添加/查看：点击后可以进行 Modbus 寄存器的设计，也可以查看目前已经内容。
5. 删除全部：删除由“添加/查看”按钮设计的所有的 Modbus 寄存器，方便重新开始设计。
6. 保存 JSON 设置：设计完成后只有点击这个按钮才可以保存数据到刚才的下载目录，之后可以下载到设备内部。

现在点击“添加/查看”按钮。对于之前的 Modbus 表的第一行：

寄存器地址	参数名	字节长度	备注
0	当前总有功电能	4	无符号，保留 2 位小数。

对应的配置如下：

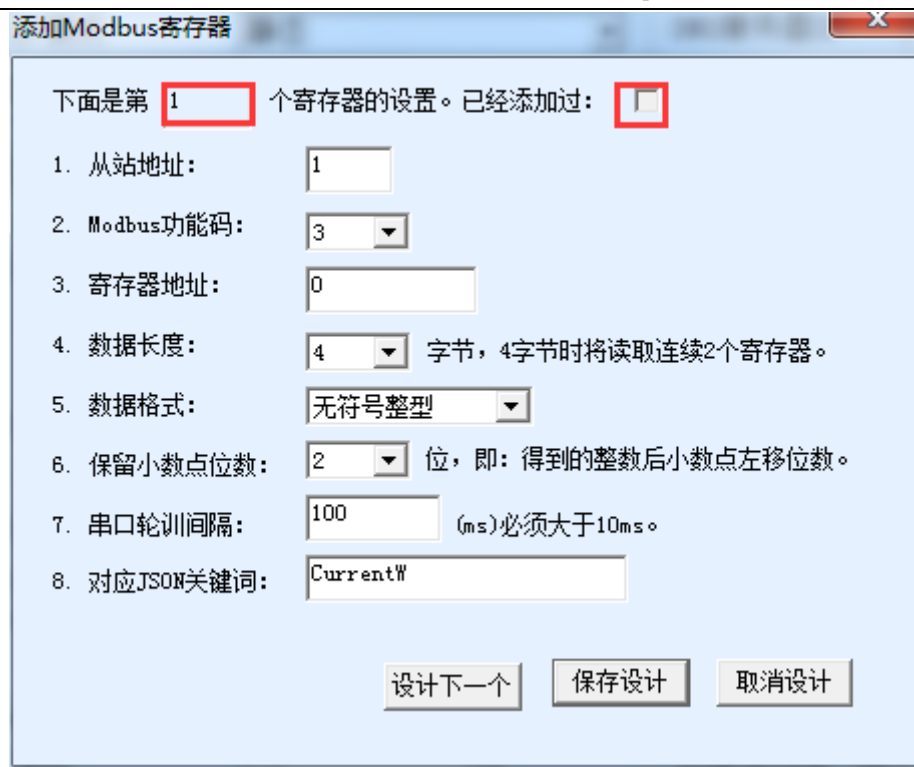


图 5 寄存器设置

这里的参数说明如下：

1. 第 1 个寄存器：这里的 1，表示现在的设计界面是配置第一个寄存器。
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。
3. 从站地址：Modbus 的表地址。
4. Modbus 功能码：目前支持 03 和 04 功能码。
5. 寄存器地址：这里对应的 0。
6. 数据长度：这里对应是 4 字节。
7. 数据格式：这里对应是无符号整型。
8. 保留小数点：这里保留 2 位。
9. 串口轮询时间：这里设为 100ms。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。
10. 对应的 JSON 关键词：对应上发的 JSON 中的 keyword。例如"CurrentW":232.12。
11. 设计下一个：点击后进入下一个寄存器的设置。
12. 保存设计：完成设计，之后点击上一个界面的“保存 JSON 配置”即可。
13. 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。

这里点击“设计下一个”按钮继续设计 Modbus 表中的其它寄存器。全部设计完表格中的寄存器后，点击“完成设计”，然后点击“保存 JSON 配置”后退出。然后点击“下载 web”页面的“下载按钮”



图 6 下载

然后点击“确定”，设备会自动重启。如果没有重启请手动重启。

## 2.4. 新建 Modbus 模拟表

这里使用 Modbus Slave 来模拟一个表

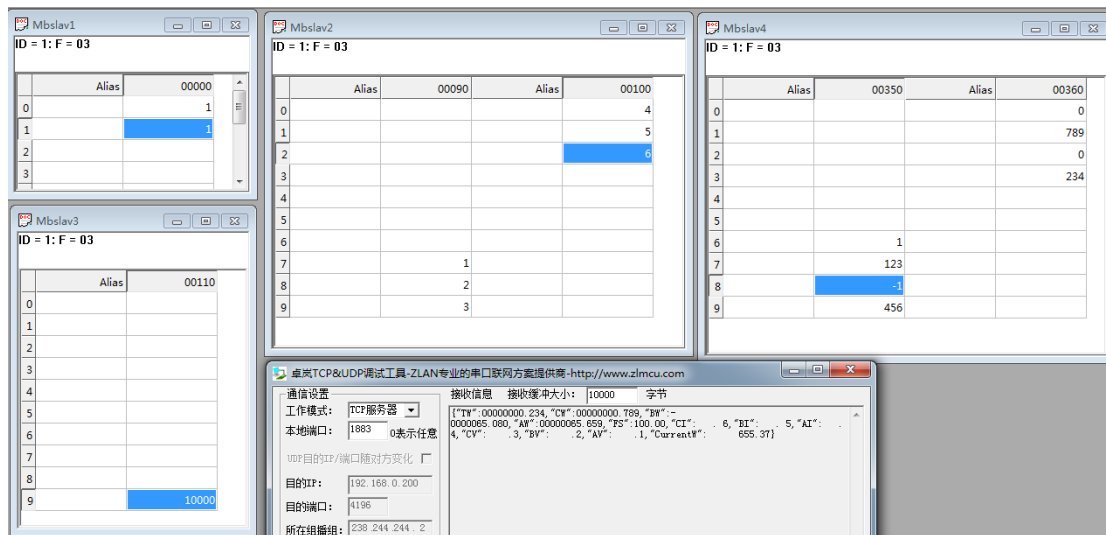


图 7 测试结果

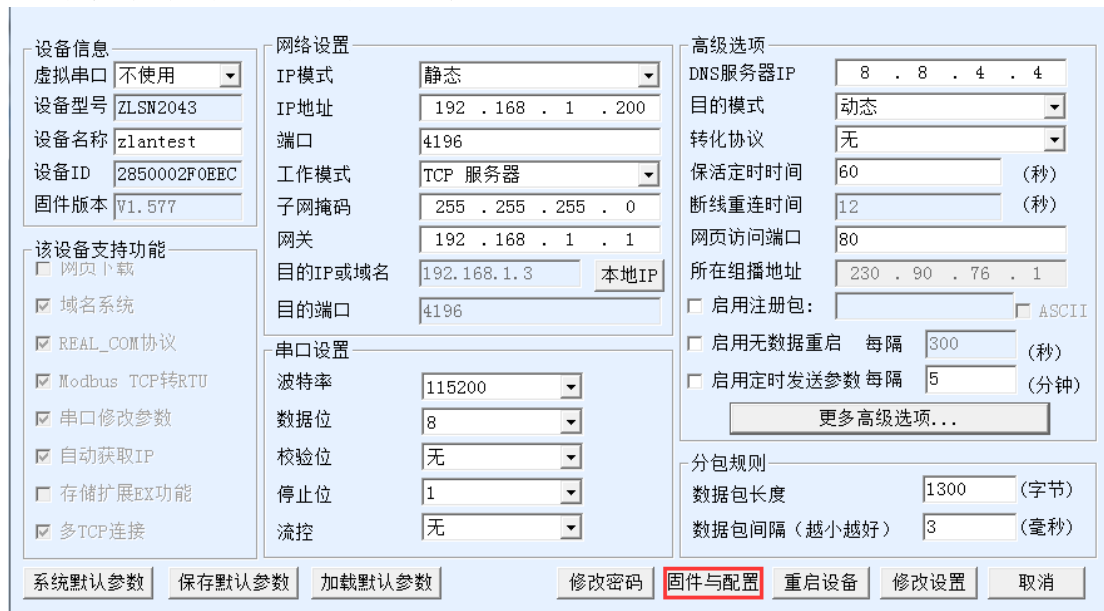
测试结果显示，使用 Modbus slave 工具模拟的仪表能够被网关所采集。同时可以定时，按照 json 格式上发到 SocketDlgTest 模拟的服务器软件端。

### 3. MQTT

MQTT 可以单独使用也可以和 JSON 功能配合使用。当单独使用的时候 MQTT 功能将串口数据透传上发 MQTT 服务器。也就是将串口接收的数据作为 MQTT 的载荷。同时 MQTT 的载荷和会透传从串口输出。实现串口转 MQTT。

#### 3.1. 设备配置

首先搜索到设备，然后点击编辑设备：



The image shows a configuration window for MQTT settings, divided into several sections:

- 设备信息 (Device Information):** Includes fields for 虚拟串口 (Virtual Serial Port) set to '不使用' (Not used), 设备型号 (Device Model) 'ZLSN2043', 设备名称 (Device Name) 'zlantest', 设备ID (Device ID) '2850002F0EEC', and 固件版本 (Firmware Version) 'V1.577'.
- 该设备支持功能 (Supported Features):** A list of checkboxes including '网页下载' (Web Download), '域名系统' (DNS System), 'REAL\_COM协议' (REAL\_COM Protocol), 'Modbus TCP转RTU' (Modbus TCP to RTU), '串口修改参数' (Serial Port Parameter Modification), '自动获取IP' (Automatic IP Acquisition), '存储扩展EX功能' (Storage Extension EX Function), and '多TCP连接' (Multiple TCP Connections).
- 网络设置 (Network Settings):** Includes IP模式 (Static), IP地址 (192.168.1.200), 端口 (4196), 工作模式 (TCP 服务器), 子网掩码 (255.255.255.0), 网关 (192.168.1.1), 目的IP或域名 (192.168.1.3), and 目的端口 (4196).
- 串口设置 (Serial Port Settings):** Includes 波特率 (115200), 数据位 (8), 校验位 (无), 停止位 (1), and 流控 (无).
- 高级选项 (Advanced Options):** Includes DNS服务器IP (8.8.4.4), 目的模式 (动态), 转化协议 (无), 保活定时时间 (60秒), 断线重连时间 (12秒), 网页访问端口 (80), 所在组播地址 (230.90.76.1), and checkboxes for 启用注册包 (ASCII), 启用无数据重启 (300秒), and 启用定时发送参数 (5分钟).
- 分包规则 (Packet Rules):** Includes 数据包长度 (1300字节) and 数据包间隔 (3毫秒).

At the bottom, there are buttons for '系统默认参数', '保存默认参数', '加载默认参数', '修改密码', '固件与配置' (highlighted in red), '重启设备', '修改设置', and '取消'.

图 8MQTT 配置 1

点击“固件与配置”，弹出配置下载和设计对话框：

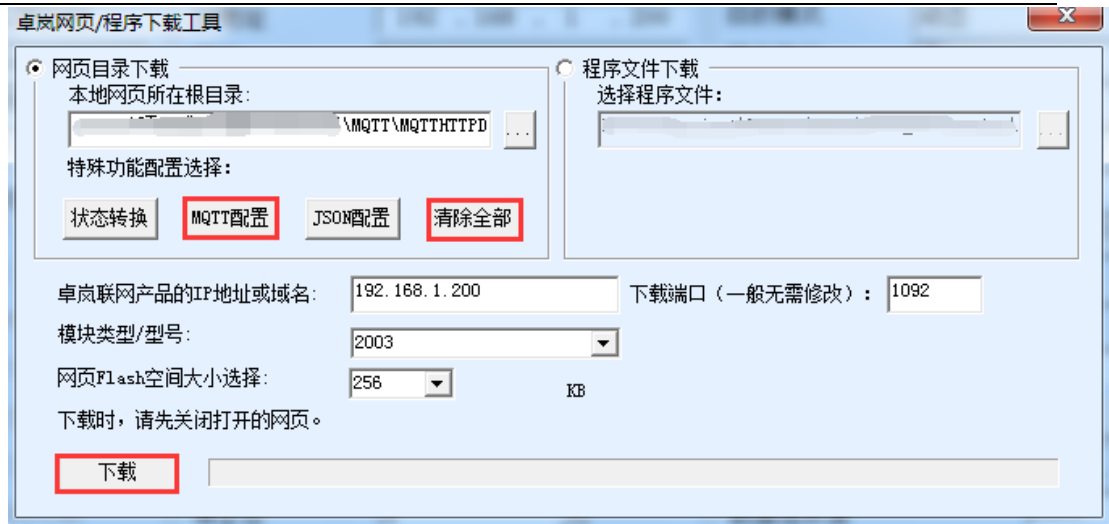


图 9MQTT 配置 2

这里选择“网页目录下载”，然后选择一个空的目录，比如 MQTTHTTTPD 目录，然后先点击“清除全部”清除之前的设计（注意如果之前按有 JSON 设计的，则不要清除全部，否则会清除之前的 JSON 设计）。然后点击 MQTT 配置。

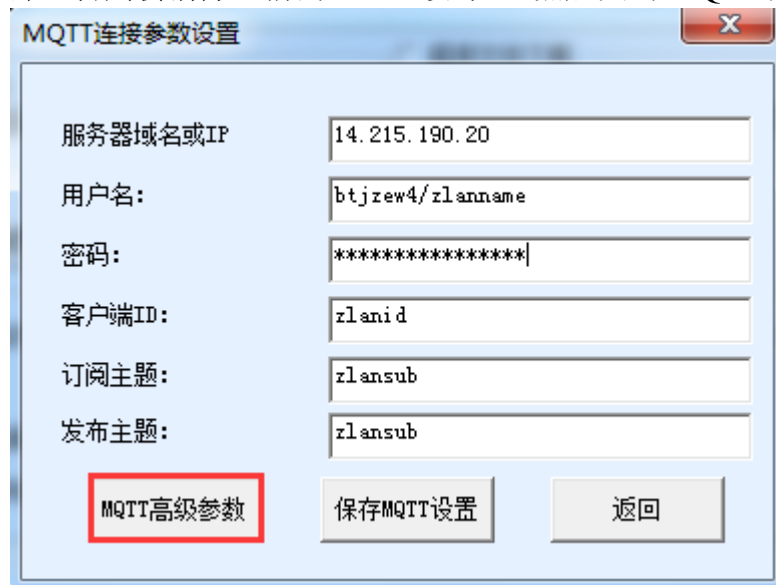


图 10MQTT 配置 3

这里配置说明如下：

- 1 服务器域名或 IP：这里填写的是 MQTT 服务器的 IP，最长为 30 个字符。
- 2 用户名：是 MQTT 服务器的用户名。
- 3 密码：是这个用户的登录密码。
- 4 客户端 ID：是 MQTT 的 client ID。

- 5 订阅主题：是这个设备订阅的主题，有其它设备发布这个主题的时候，服务器会下发给这个设备。如果只是发布，一般无需填写这个字段。
- 6 发布主题：这个设备串口转 MQTT 的时候向服务器发送的数据的主题。
- 7 MQTT 高级参数：用于配置高级参数。
- 8 保存 MQTT 设置：设计完毕点击这个按钮进行保存，然后点击网页下载目录的“下载按钮”下载。

现在先点击“MQTT 高级参数”（一般无需配置高级参数）：

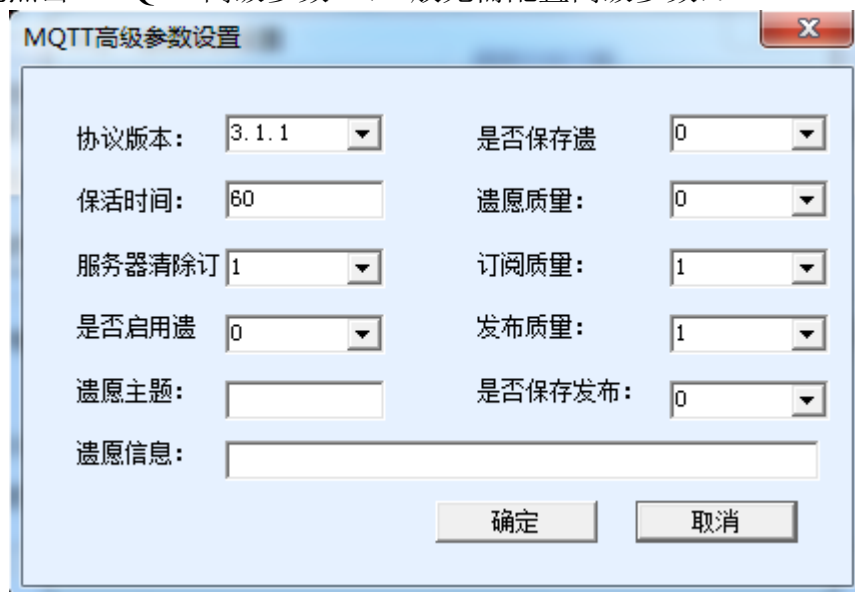


图 11MQTT 高级参数配置

说明如下：

- 1 协议版本：目前主流的是 3.1.1 版本，如果需要选择 3.1 版本，请在这里选择。
- 2 保活时间：MQTT 的心跳时间，最少为 10 秒，默认为 60 秒。
- 3 服务器清除订阅：客户端断线后服务器是否清除订阅信息。
- 4 是否启用遗嘱：是否有遗嘱。
- 5 遗嘱主题：遗嘱主题。
- 6 遗嘱信息：遗嘱的信息。
- 7 是否保存遗嘱：服务器是否需要保持客户端异常离线时发送给客户端的遗嘱消息。
- 8 遗嘱质量：服务器发送的遗嘱消息的交付质量级别。

- 9 订阅质量：订阅的交付质量级别。
- 10 发布质量：客户端发布消息的交付质量级别。
- 11 是否保存发布：服务器是否保持最后一条消息（如果有新的客户端订阅则发送给客户端）。

我们这里不修改高级参数。直接点击“保存 MQTT 设置”。然后点击“下载”

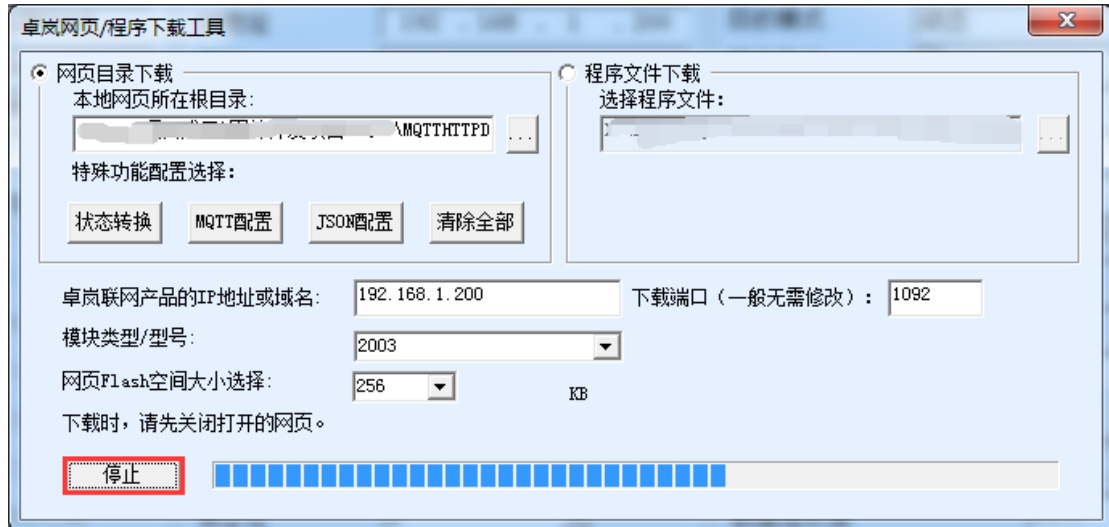


图 12 下载

下载完毕点击确定，此时会回到设备管理对话框，看到设备的目的 IP、工作模式、目的端口已经自动修改为 MQTT 的设置：

序	类型	设备名称	P.	设备IP	本地..	目的IP	模式	TCP连..	虚拟串口..	虚拟串口状..	设备ID	TXD	RXD
1	内网	ZLDEV0001		192.168.1.241	4196	115.239.210.27	TCP Server	未建立	未设置	未联通	D4EF6936	0	0
2	内网	zlanetest		192.168.1.200	0	14.215.190.20	TCP Client	未建立	未设置	未联通	002F0EEC	0	0

图 13 自动修改

如果没有自动修改则需要设备编辑对话框下，设置目的 IP、工作模式、目的端口。然后点击“修改设置”。

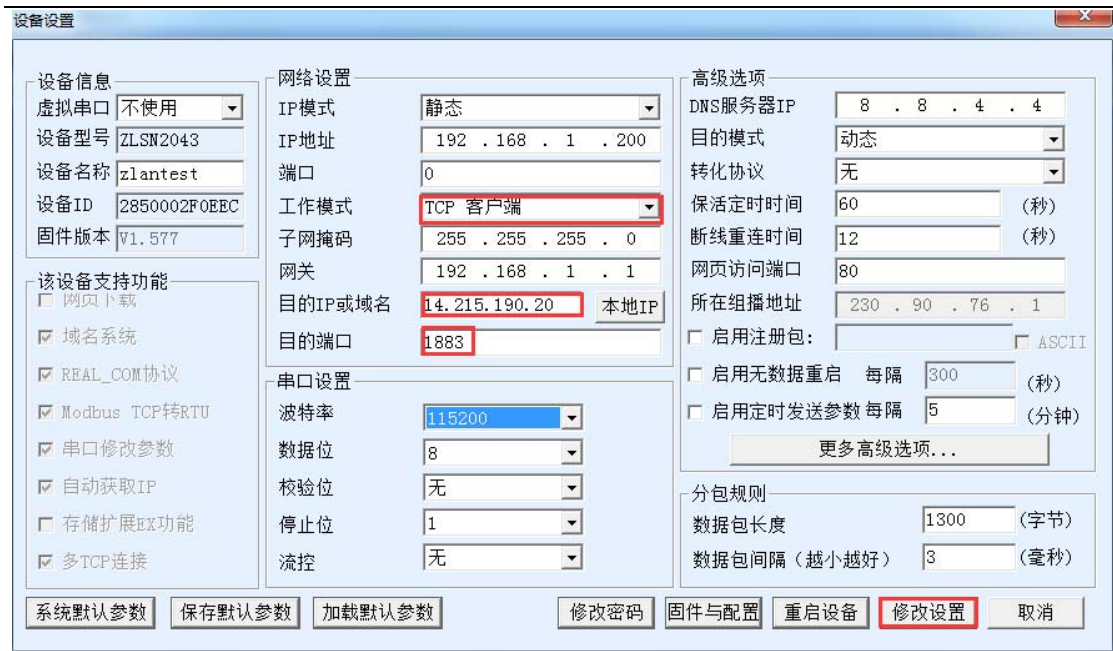


图 14IP 配置

这样配置就完成了。

### 3.2. 数据测试

连接完成后设备的 LINK 灯（一般为中间的蓝色灯）变亮。说明设备正常连接上 MQTT 服务器。

现在打开串口工具：

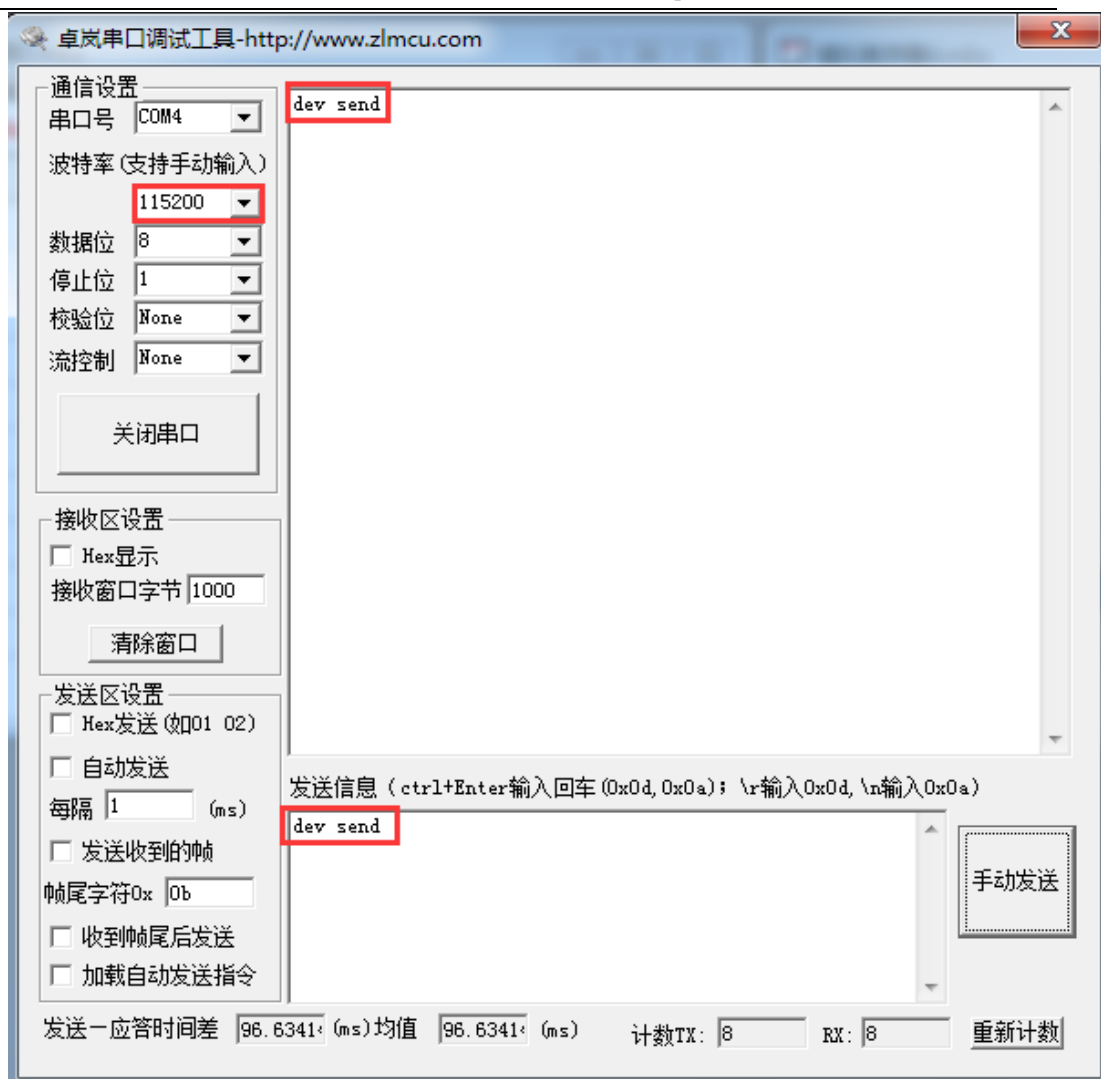


图 15 串口发送和接收

使用和设备一样的波特率打开串口，并发送数据“dev send”，此后在接收窗口看到返回的数据“dev send”。这是由于我们将 dev send 这个信息以 zlan-sub 的主题发布给 MQTT 服务器。但是同时我们的设备也订阅了 zlan-sub 这个主题，所以服务器立刻会给我们发送一个订阅信息，订阅信息的内容就是 dev send。这个信息作为 MQTT 的载荷发送下载，通过透传的方式从串口输出。

如果其它设备发布信息过来，这个设备也能够收到数据。

一般来说，用户可以直接将串口指令（比如 Modbus RTU）指令直接透传给 MQTT 服务器。另外也可以使用 JSON 功能，采用自动 Modbus RTU 格式采集，定时 JSON 格式上发的形式。另外也可以找上海卓岚定制一些非标的仪表和上位机协议格式。

## 4. MQTT+JSON 转 Modbus RTU

将上述的 JSON 和 MQTT 相组合可以实现如下的功能：

1. 采用 MQTT 为基础的协议和服务器建立连接，采用订阅发布的形式进行数据通讯。
2. 支持 Modbus RTU 寄存器的自主设计和自动采集。
3. 支持将特定的 Modbus 寄存器内容转化为 JSON 格式定时主动上发。
4. 支持在 JSON 格式中加入设备 ID，方便云端识别设备。

如果需要 MQTT+JSON 转 Modbus RTU 功能的，可以分别设计 MQTT 和 JSON，不分先后次序。设计完一种不要点击“清除设计”按钮，两个设计完，一起点击“下载”按钮下载到设备内容。

一般可以下载后手动重启设备，加载设置。

## 5. 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

电话：021-64325189

传真：021-64325200

网址：<http://www.zlmcu.com>

邮箱：[support@zlmcu.com](mailto:support@zlmcu.com)