LG01 LoRa 物联网网关用户手册

文件版本: 1.6

固件版本: IoT Mesh v4.3.5

Version	Description	Date
0.1	初版	2016-Oct-29
1.0	正式发布，添加 ThingSpeak 服务器例。	2016-Dec-9
1.1	添加如何连接到 TTN LoRaWAN 服务器例子	2017-May-17
1.2	添加 RN2483 链接。	2017-Jul-14
1.3	添加 OLG 天线说明，修改备用 IP 描述,更新软件源代码链接。 更新例子说明文字。	2017-11-15

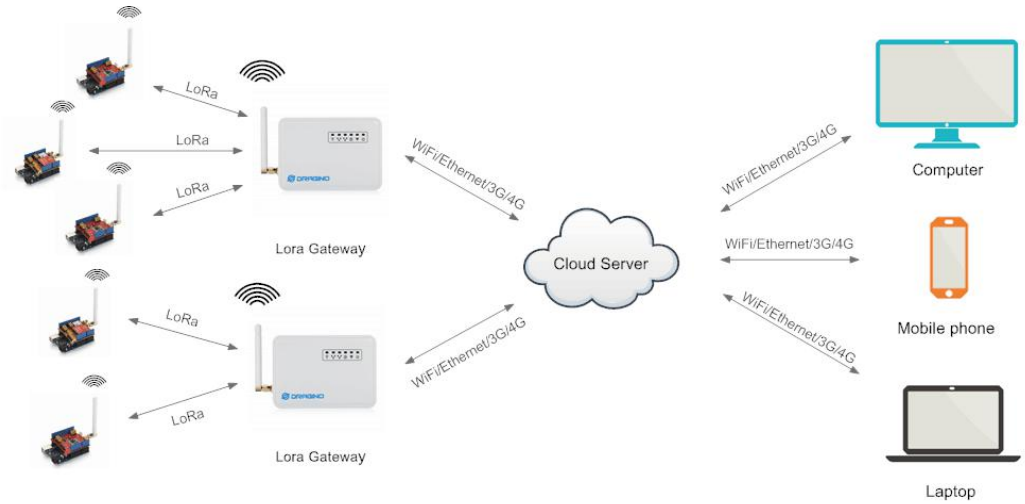
1.4	添加乐联网服务器示例，增加 MQTT 功能示例说明。	2018-04-03
1.5	添加 TCP 示例，修复 lg01_pkt_fwd 进程 bug。	2018-05-08
1.6	修复安全漏洞，修复互联网/ DNS 检查问题。	2018-06-20

目录

1 介绍

1.1 概述

LG01 是一个开源的单通道 LoRa 网关，它可以将 LoRa 网络通过 WiFi，以太网口，3G 或者 4G 来连接到 Internet IP 网络。LG01 在开源嵌入式 Linux 系统上运行；它有一个 USB 主机端口，2 个以太网口和 802.11 b / g / n WiFi 功能. USB 主机端口可用于连接蜂窝模块，因此 LG01 非常灵活，可以将 LoRa 网络连接到不同类型的网络，以满足用户的需求。



1.2 产品说明

硬件系统：

Linux 部分:

- 400Mhz AR9331 处理器
- 64MB RAM
- 16MB Flash

微处理器（MCU）部分:

- 单片机: ATMega328P
- Flash: 32KB
- SRAM: 2KB
- EEPROM: 1KB

接口:

- 输入电压: 9 ~ 24v DC
- 两个 RJ45 接口
- 一个 USB 2.0 host 外部接口
- 一个 内部 USB 2.0 host 接口

WiFi 规格:

- IEEE 802.11 b/g/n
- 频率范围: 2.4 ~ 2.462GHz
- 发射功率:
  - ✓ 11n tx power : mcs7/15: 11db mcs0 : 17db
  - ✓ 11b tx power: 18db
  - ✓ 11g 54M tx power: 12db
  - ✓ 11g 6M tx power: 18db
- Wifi 灵敏度
  - ✓ 11g 54M : -71dbm
  - ✓ 11n 20M : -67dbm

LoRa 规格:

- 频率范围:
  - ✓ Band 1 (HF): 862 ~ 1020 Mhz
  - ✓ Band 2 (LF): 410 ~ 528 Mhz
- 最大链路预算可达 168db
- +20 dBm - 100 mW 电压变化时恒定的射频输出与.
- +14 dBm 高效功率放大器
- 可编程比特率最高可达 300 kbps.
- 高灵敏度: 低至 -148 dBm.
- 高可靠性前端: 输入三阶截点 = -12.5 dBm.
- 卓越的阻断免疫.
- 10.3mA 低接收电流，200nA 寄存器保持电流
- 分辨率为 61Hz、完全集成的频率合成器
- FSK, GFSK, MSK, GMSK, LoRaTM 和 OOK 调制.
- 时钟恢复的内置位同步器.
- 前导码检测.
- 127 dB 的 RSS 动态范围.

- 自动射频信号检测，CAD 模式和超高速 AFC
- 带有 CRC、高达 256 字节的数据包
- 内置式温度传感器和低电量指示器.

蜂窝网络-4G LTE (可选):

- 上海移远 EC20 LTE module
- Micro SIM 卡槽
- 内部 4G 天线+外部 4G 船桨天线.
- 高达 100Mbps 的下行链路和 50Mbps 的上行链路数据率。
- 世界范围内的 LTE、um/hspa+和 gsm/gpr/edge 的覆盖
- MIMO 技术满足了现代无线通信系统中数据速率和链路可靠性的要求

蜂窝网络 - 3G UMTS/HSPA+ (可选):

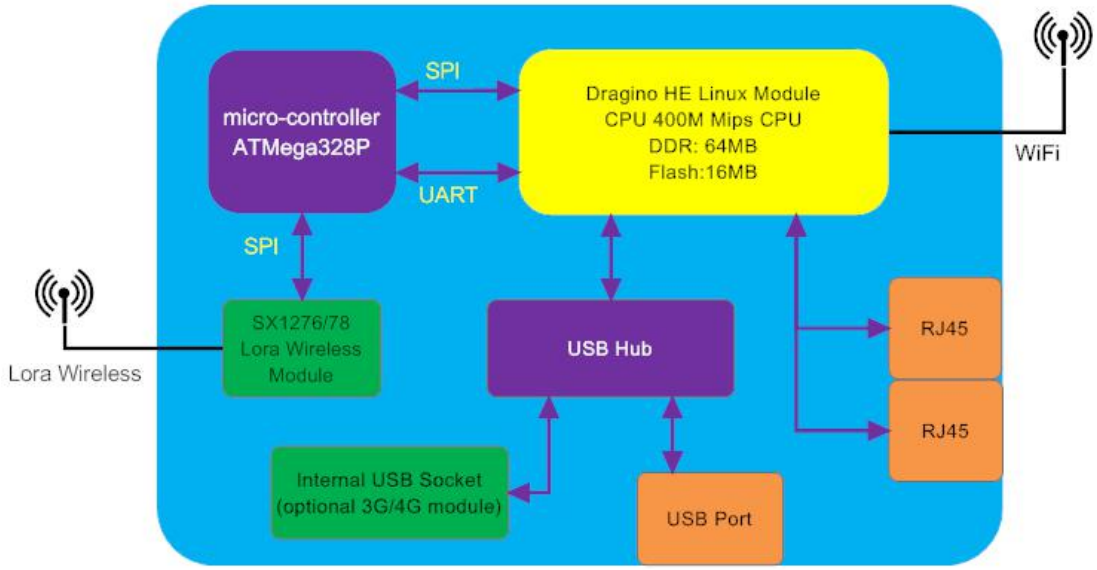
- 上海移远 UC20 LTE module
- Micro SIM 卡槽
- 内部 3G/4G 天线+外部 3G/4G 标签天线.
- 高达 14.4 Mbps 的下行链路和 5.76 Mbps 的上行数据速率
- 世界范围内的/hspa+和 gsm/gpr/edge 覆盖率
- 高质量的数据和图像传输即使在恶劣的环境中
- 主要和多样性接收路径是为等效的噪声图形性能设计的

1.3 产品特征

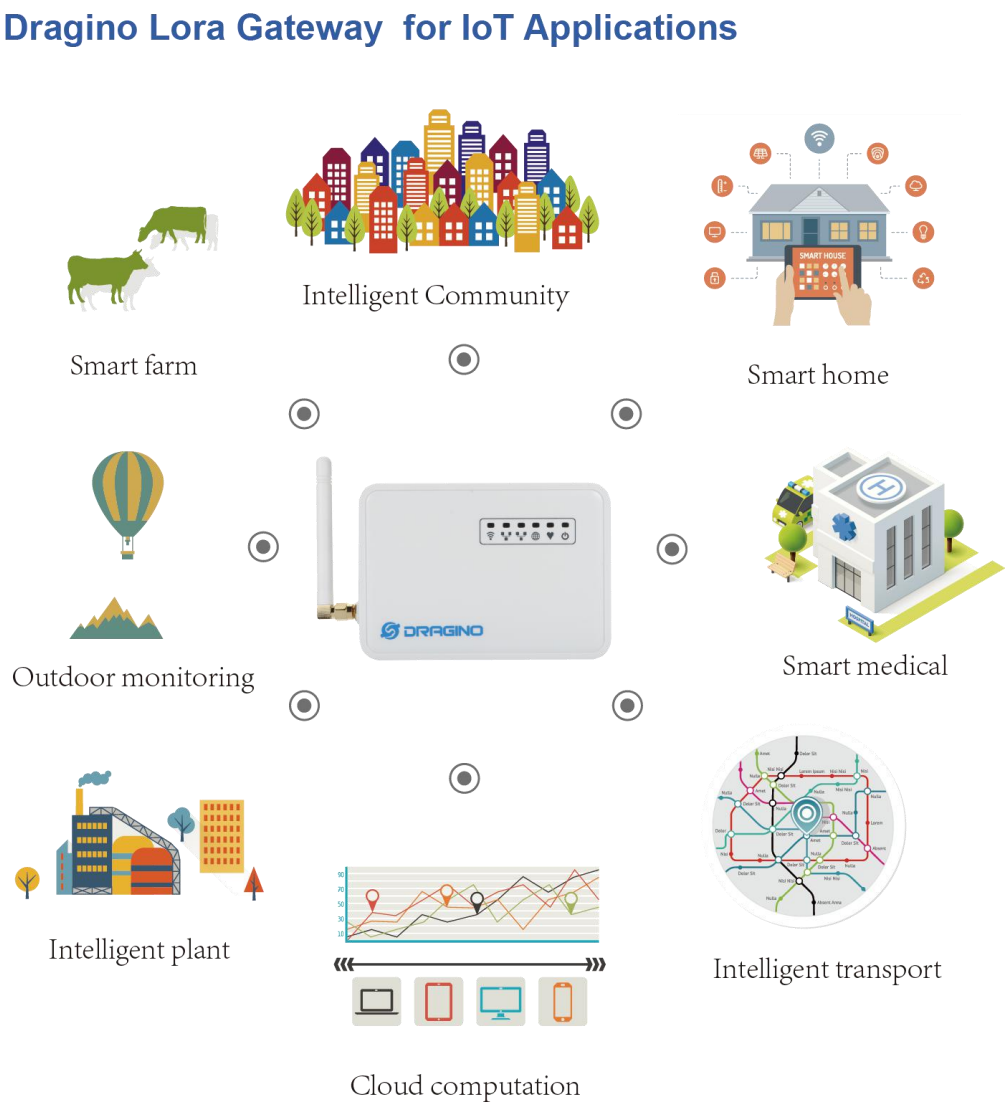
- 内置开源 Linux(OpenWrt)系统，用户可根据自身需求来修改或编译固件。
- 低功耗.
- 兼容 Arduino IDE 1.5.4 或更高版本, 用户可以通过 Arduino IDE 编程、调试或上传固件到 MCU.
- 由 Web GUI、SSH 通过 LAN 或 WiFi 管理.
- 软件通过网络升级.
- Auto-Provisioning.
- 内置 Web 服务器.
- 通过 RJ45 端口、WiFi 或 3G /4G 网络来连接 Internet 网络。
- 提供可靠的恢复系统.

1.4 系统构造

LG01 System Overview:



1.5 应用领域


Dragino Lora Gateway for IoT Applications
Intelligent Community
Smart farm
Smart home
Outdoor monitoring
Smart medical
Intelligent plant
Intelligent transport
Cloud computation

1.6 硬件版本

对于不同的使用环境，有不同的 LG01 版本。下表显示了这些硬件版本之间的差异：

型号	产品图片	描述
LG01-P		最通用的版本可以用作 LoRa 基础网关
LG01-S		在基础网关上增加了 连接外部传感器的接线端子
OLG01		户外版本，这个版本不包括 LoRa 天线而是提供一个 SMA 连接器，用户可以将它连接到一个高增益的 LoRa 天线。OLG01 可以由一个被动的 PoE 适配器来供电。

1.7 将 SIM 卡安装在 3G/4G 模块

在含有 3G/4G 上网模块的设备中，用户请按照下面的图示安装 Micro SIM 卡

2 快速启动向导

2.1 LG01 的使用和配置

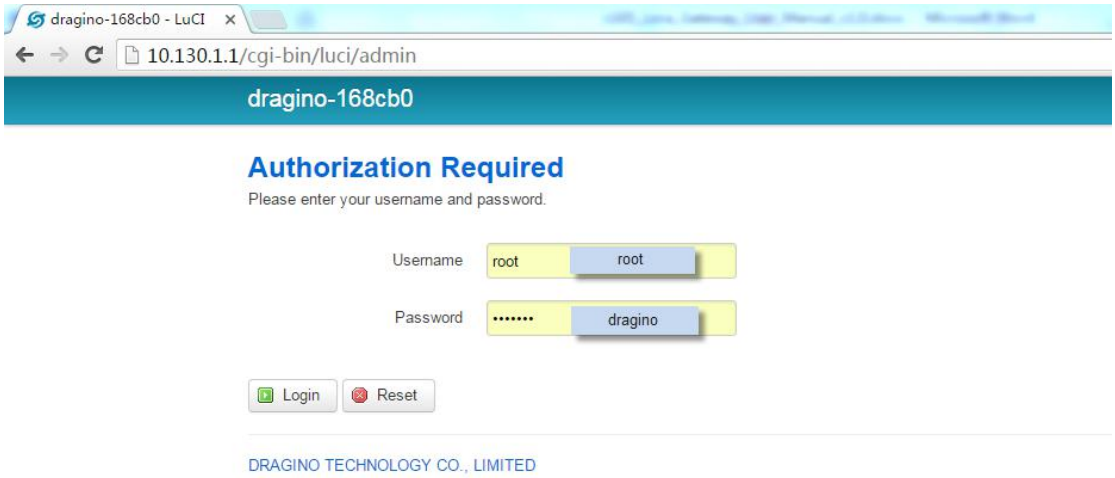
LG01 默认配置为一个 WiFi AP. 用户可以在连接到它的 WiFi 网络后访问和配置 LG01。

在 LG01 的第一次启动中，它将自动生成一个不安全的 WiFi 网络，叫做 ***dragino2-xxxxxx***

用户可以使用笔记本连接到这个 WiFi 网络. 这台笔记本电脑将获得一个 IP 地址 10.130.1.xxx， LG01 的默认 IP 是 10.130.1.1

打开笔记本电脑浏览器，键入 10.130.1.1
用户将看到 LG01 的登录界面.
Web 登录的账户是:

User Name: root
Password: dragino



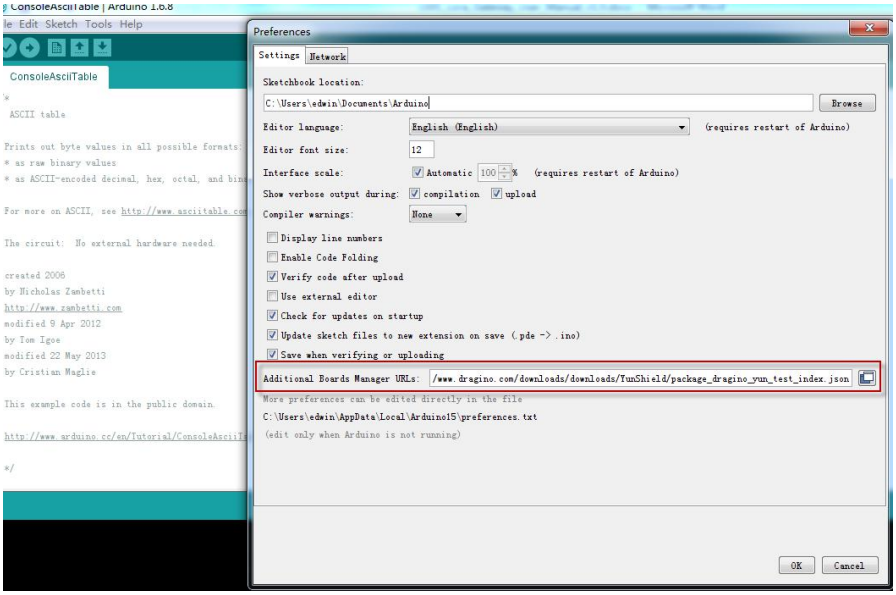
2.2 微控制器程序.

MCU(微控制器) ATMega328P 用于与 LoRa 模块和 Dragino Linux 模块通信。MCU 的程序语言是基于 C 语言的，编程工具是 Arduino IDE。下面我们展示了如何进行编程。

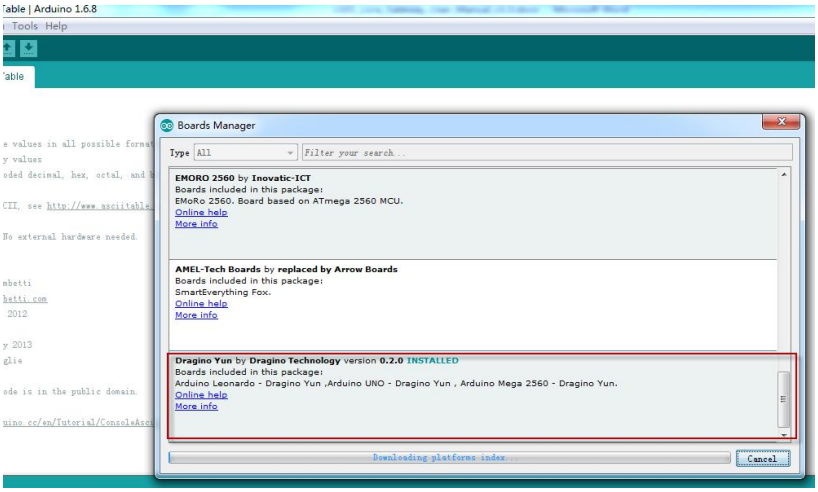
2.2.1 下载和安装 Arduino IDE

- 从 Arduino 官方网站下载最新的 Arduino 软件(IDE)：
  https://www.arduino.cc/en/Main/Software

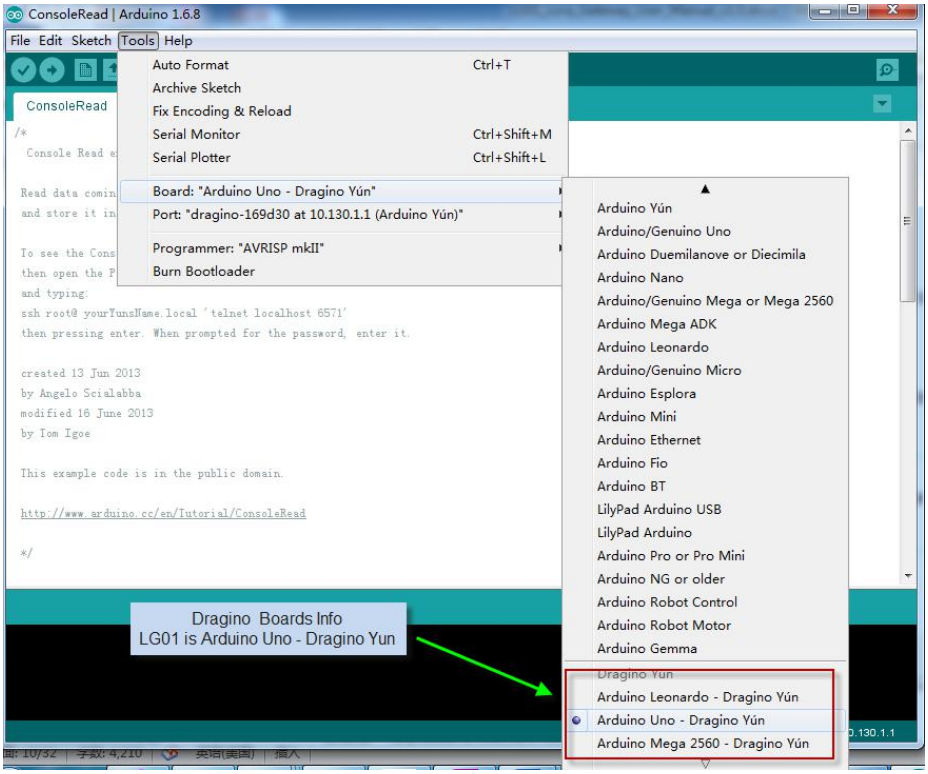
在 PC 上安装 IDE，打开并点击 **File --> Preference**，在 **Additional Boards Manager URLs** 里添加以下 URL http://www.dragino.com/downloads/downloads/YunShield/package_dragino_yun_test_index.json



- 转到 **tools --> Boards --> Boards Manager**，找到 Dragino boards 信息并安装它.



- 在 IDE 中安装 Dragino board 信息之后， 我们可以看到来自 IDE Board 信息，就像下面的截图一样。对于 LG01，我们应该选择: **Arduino Uno – Dragino Yun**.

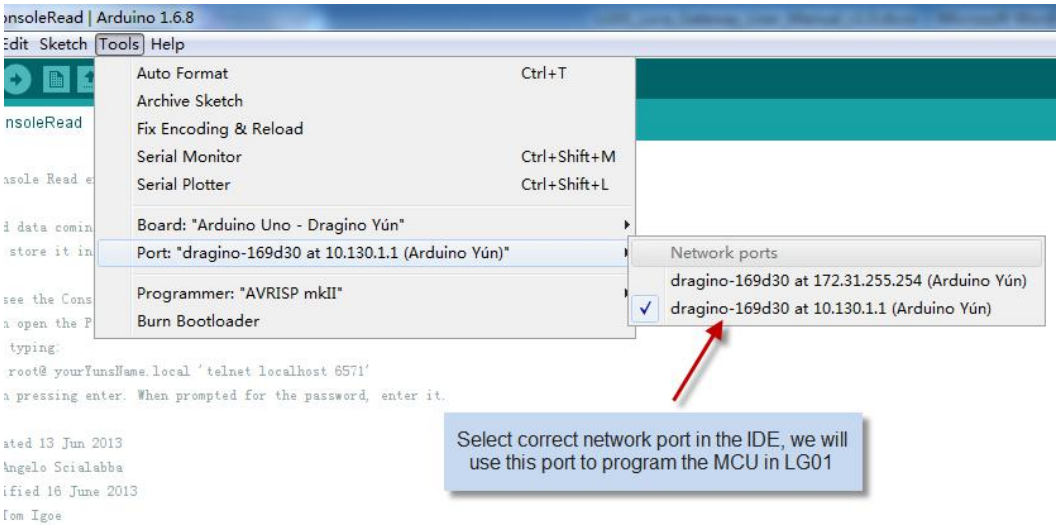


注意: 如果用户在自动安装 Dragino Boards 时候出现问题，用户可以**手动添加配置文件**。

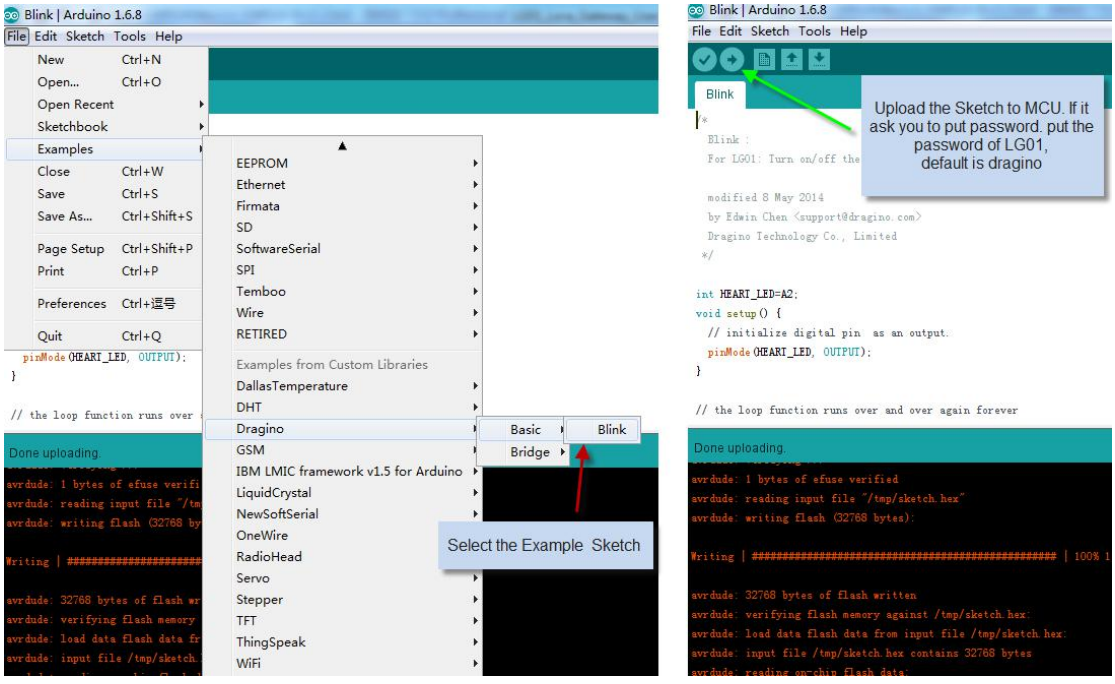
2.2.2 给 MCU 上传一个固件

首先，我们可以向 MCU 上传一个简单的固件，看看它是如何工作的

- 如果您已经连接到 LG01 WiFi SSID,请确保您的计算机和 LG01 是在相同的网络中， 然后这两个设备都在同一个 WiFi 网络中。在 IDE 中，选择正确的端口，如下图所显示：



- 从 **IDE --> File --> Examples --> Dragino --> Basic --> Blink** 中选择例子上传，点击上传，将固件上传到 LG01,若 LG01 提示输入密码，请输入 LG01 密码.



- 检查结果

Blink 例子将把 MCU 的 A2 引脚设置为周期性的高电平和低电平. 这个引脚连接到 LG01 的 **HEART 灯**. 如果成功上传了这个固件,用户可以看见 **HEART 灯**被周期性地点亮和熄

灭.

2.3 简单的 LoRa 无线范例

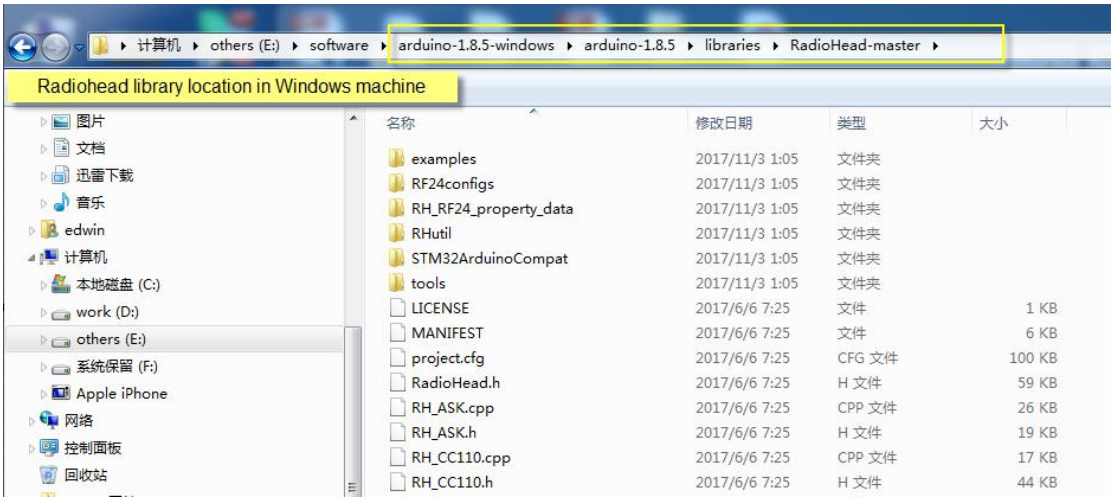
为了测试 LoRa 无线收发功能，我们至少需要两个支持 LoRa 的设备. 在本例中，我们将使用以下设备:

- LoRa 服务器: LG01 ;
- LoRa 客户端：LoRa Shield + Arduino Uno

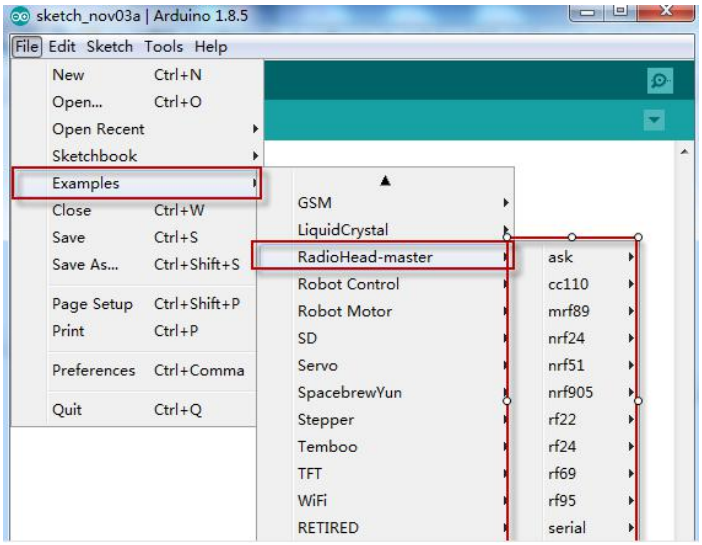
在这个例子，我们会演示基本的 LoRa 通信。LoRa 客户端通过 LoRa 无线广播数据包。 LG01 网关收到这些数据包之后会把他们在电脑上的调试窗口显示出来。

2.3.1 安装 LoRa 库

这里的库是 Radiohead 库，能够从 https://github.com/dragino/RadioHead/archive/master.zip 下载将它解压并放到 Arduino 库文件夹中，最后的路径应该如下图:

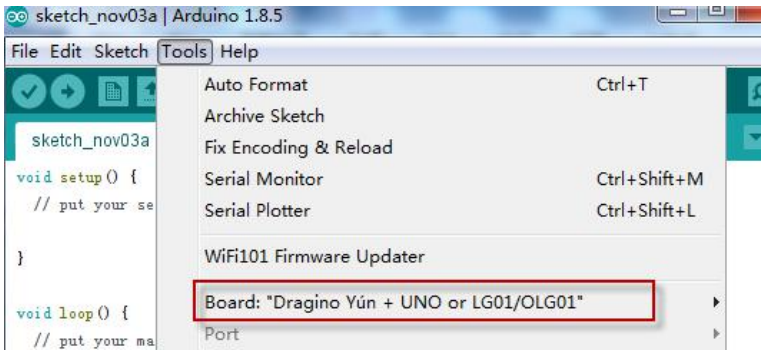


为了确保 Radiohead 的库是正确安装的，我们重启 Arduino IDE，之后我们会看到 Radiohead 出现在 Examples 的目录里面，如下:

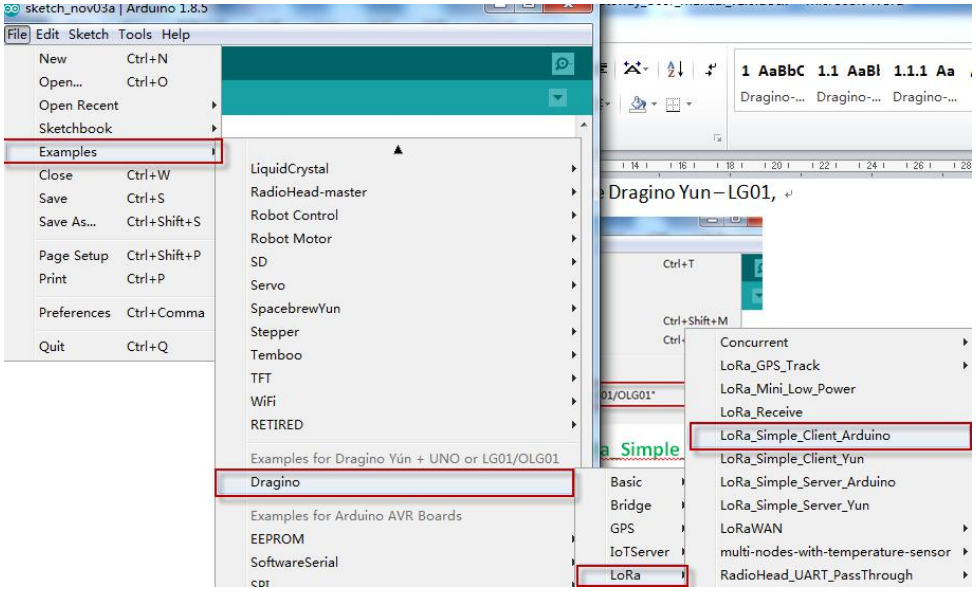

sketch_nov03a | Arduino 1.8.5
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
GSM
LiquidCrystal
RadioHead-master
Robot Control
Robot Motor
SD
Servo
SpacebrewYun
Stepper
Temboo
TFT
WiFi
RETIRED
ask
cc110
mrf89
nrf24
nrf51
nrf905
rf22
rf24
rf69
rf95
serial

2.3.2 上传 LoRa 客户端的固件

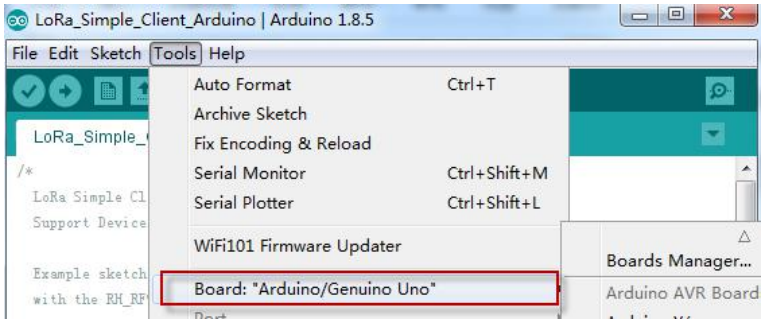
a) 首先打开 Arduino IDE，选择 Dragino Yun – LG01，



b) 然后选择例子: **LoRa_Simple_Client_Arduino**



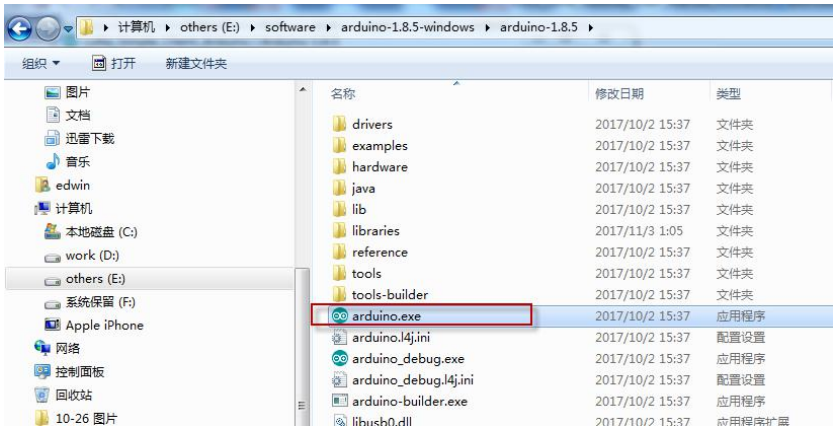
c) 在 **LoRa_Simple_Client_Arduino** 的编程窗口中，把板子的信息选回 Arduino UNO，这个代表的板子是 LoRa Shield + UNO:



d) 通过 USB com 端口上次例子 **LoRa_Simple_Client_Arduino** 到 LoRa Shield + UNO. 同时打开串口监视器查看输出。

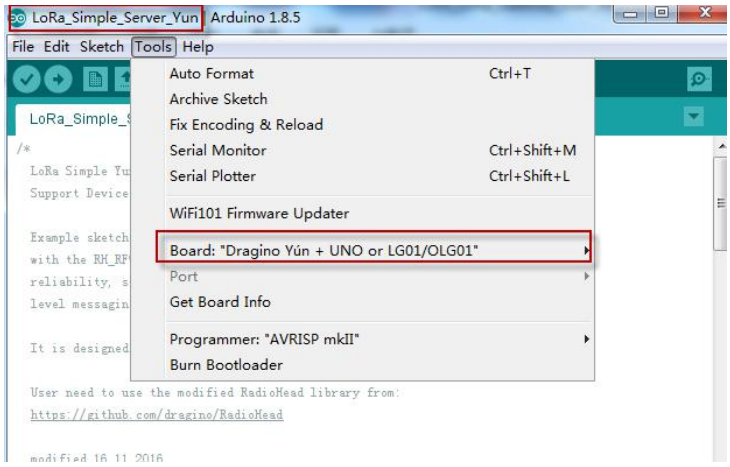
2.3.3 上传 LG01 LoRa 网关端固件

a) 再次点击 Arduino.exe 来打开另外一个 Arduino IDE 窗口。这很重要，因为我们需要两个独立的窗口监视器，一个是监视客户端，另外一个监视 LG01 网关

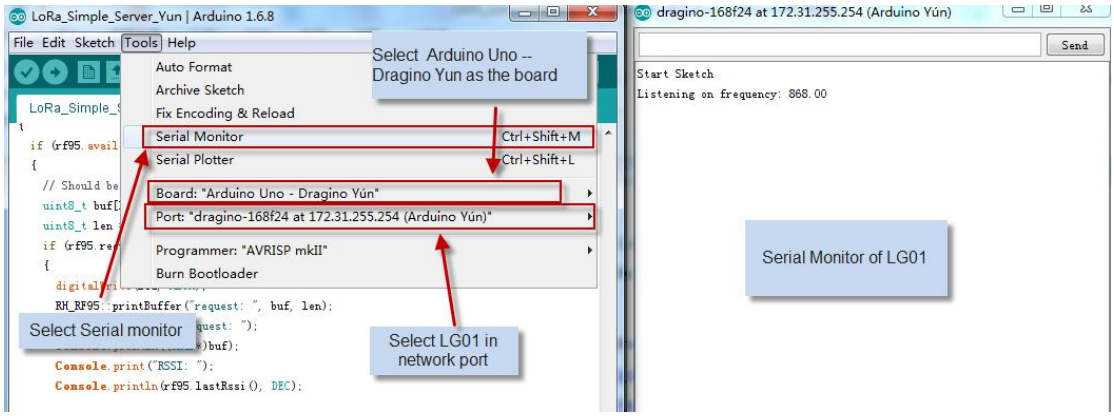


b) 在新的窗口中，选择 LG01 作为需要使用的板子 ，然后选择并上传例子:

LoRa_Simple_Server_Yun



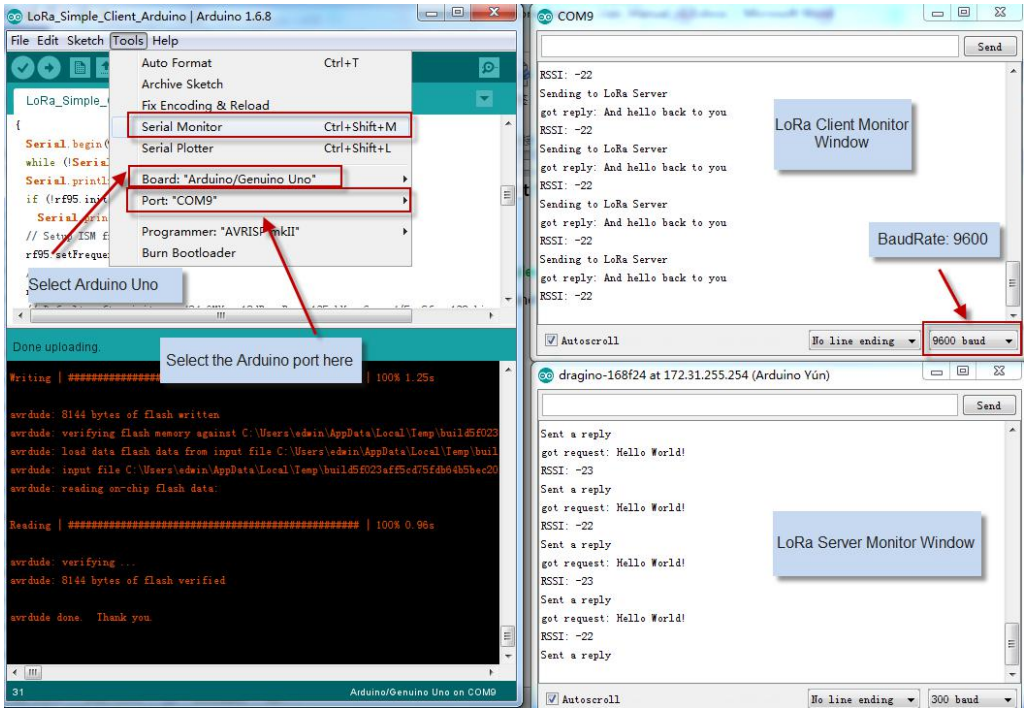
c) 例子上传结束后，打开串口监视器查看程序输出。



2.3.4 分析测试结果

底下的窗口显示了输出的结果.

- ✓ 上面的窗口显示了 LoRa 客户端在持续向外广播 LoRa 数据包，并等待回复。
- ✓ 下面的窗口是 LG01 网关的窗口， 他显示了 LG01 等到了一个”Hello, World”的数据包并回复“And hello back to you”，当 LoRa 客户端收到这个回复包之后会打印到自己的窗口上。



注意: 这个例子中，LoRa 客户端上电之后就会广播 LoRa 数据包. 而 LG01 会在用户打开串口监视器之后才会接收数据包并回复。原因是我们在网关中有这段代码:

```
while (!Console) ; // Wait for console port to be available
```

表示一直循环直到用户通过串口连接来。

如果希望 LG01 不等待串口监视器连接就发送，那么可以直接把这段代码屏蔽掉。

当使用 另一个 LG01 作为 LoRa 节点时候

方法与上面的方法相同，但是使用以下例子:

IDE --> File --> Examples --> Dragino --> LoRa --> LoRa_Simple_Client_Yun

3 典型的网络设置

3.1 概述

LG01 支持用于不同环境的灵活网络连接. 本节描述可以在 LG01 页面中设置的典型网络拓扑结构. 这些网络设备模式包括:

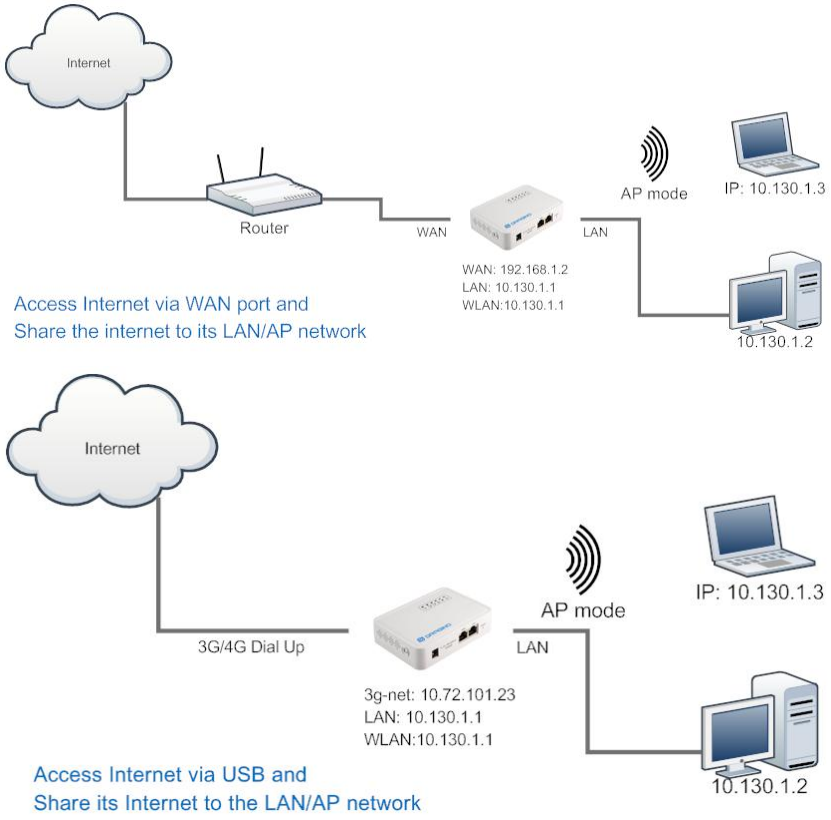
- ✓ **WAN 网络连接模式**
- ✓ **WiFi 客户端模式**
- ✓ **WiFi AP 模式**
- ✓ **无线网状网络(mesh)**
- ✓ **USB 拨号模式**
- ✓ **USB 接入方式**

在设置网络参数前, 用户可以在 **Web --> Network --> Internet Access** 访问中设置通过 LED 显示网络连接，方便地检查网络状态 ，LG01 将检查网络连接到主机，并在 **GLOBAL LED** 中显示状态， 如果 LG01 与该主机有网络连接，**GLOBAL LED** 将会闪烁

网络设置在 Network 下拉菜单下，在本章的后续部分中，我们将展示如何配置 LG01 以进行典型的网络配置。

3.2 一般的无线 AP 网络

在一般的 AP 模式下, LG01 通过它的 WAN 端口或者 USB 3G/4G/GPRS 获得互联网接入。LG01 本身作为一个 WiFi 接入点并提供一个私有 AP 网络。LG01 将互联网提供到它的 AP 网络和 LAN 接口，图示所下:



如何配置 WiFi AP 模式:

Network --> Internet Access:

✓ 通过 WAN 端口或 USB 调制解调器访问互联网

Network --> LAN and DHCP

✓ 在其局域网端口启用 DHCP 服务器

Network --> Access Point

✓ 启用 无线 AP

✓ 输入 SSID/ Encryption/ Passphrase

Network --> Mesh Network:

✓ 禁用 Mesh 网络

3.3 WAN 端口网络模式

LG01 将使用 WAN 端口进行互联网连接，当将 LG01 的 WAN 端口连接到路由器时, LG01 将从路由器获得 IP 和互联网接入。LG01 还将互联网提供到其 LAN 端口和 WiFi AP 网络，用于其他设备接入 Internet 使用。

WAN 端口网络模式是 LG01 出厂时的默认配置。

3.4 WiFi 客户模式

在 WiFi 客户端模式下，Dragino 充当 WiFi 客户端，并通过 WiFi 连接从上级路由器获取 DHCP 和网络连接。它还将互联网提供到其 LAN 端口.

在 Web UI 中设置

Network --> Internet Access:

- ✓ 通过 WiFi Client 访问互联网
- ✓ 获得 IP 的方法: DHCP
- ✓ 输入正确的 WiFi SSID,密码和加密方式

Network --> LAN and DHCP

- ✓ 在其局域网端口启用 DHCP 服务器

Network --> Access Point

- ✓ 禁用 WiFi AP

Network -->Mesh Network

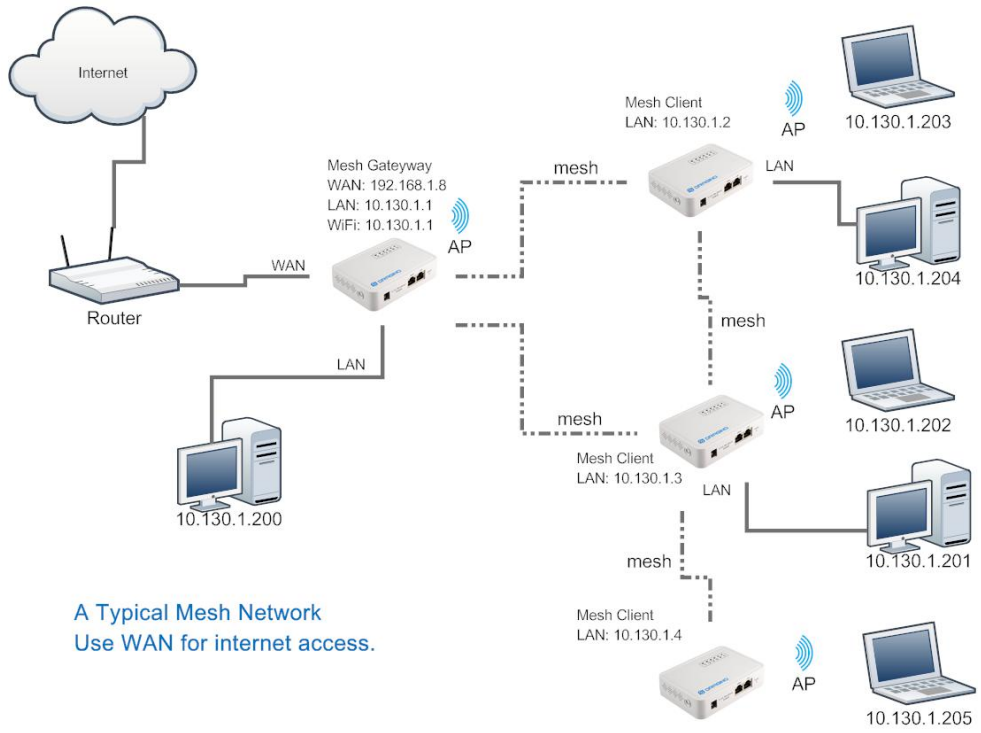
- ✓ 禁用无线 mesh 网络

3.5 无线网状网(mesh)

在网状网拓扑中，用户可以选择设备为网状网网关或网状网客户端节点

mesh 网关:使用 WAN 端口或 USB 3G/4G 调制解调器从上级路由器获取互联网接入。它还将互联网提供给该 mesh 网络的 mesh 节点. mesh 网关也充当它的 mesh 网络的 DHCP 路由器

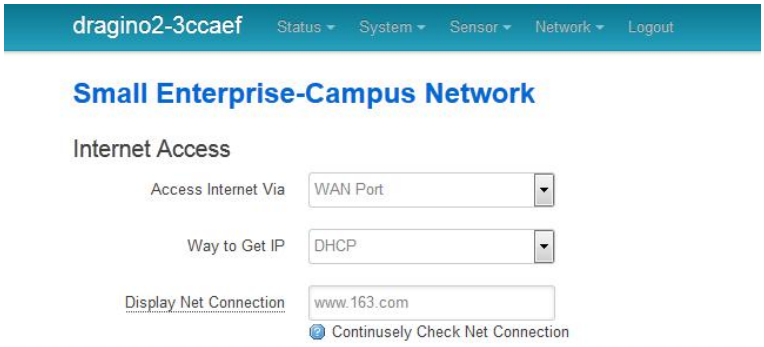
mesh 节点: 通过无线 mesh 协议连接到 mesh 网关，并通过 mesh 网关连接互联网



3.5.1 mesh 网关设置

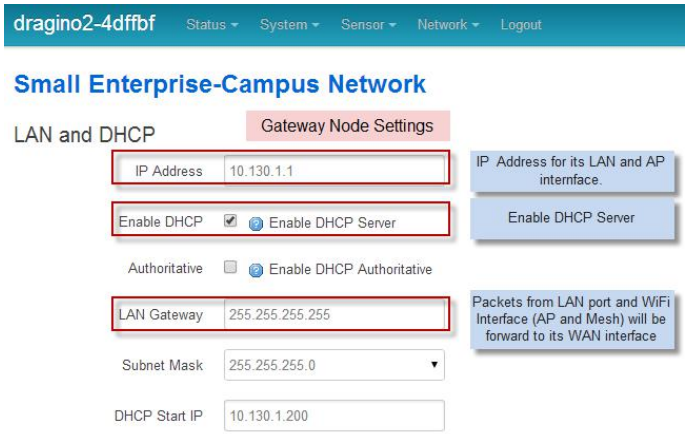
Network --> Internet Access

通过 WAN 端口或 USB 调制解调器访问互联网



✓ 为 mesh 网关选择互联网连接方法

Network --> LAN and DHCP

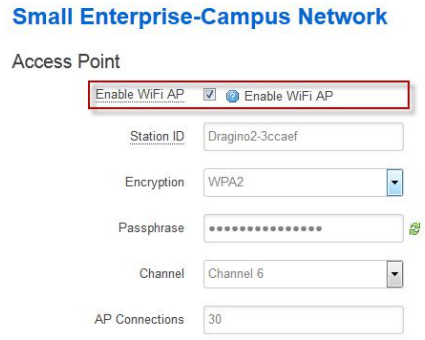


✓ 在其局域网端口勾选 Enable DHCP

✓ 设置 LAN Gateway: 255.255.255.255

Network --> Access Point

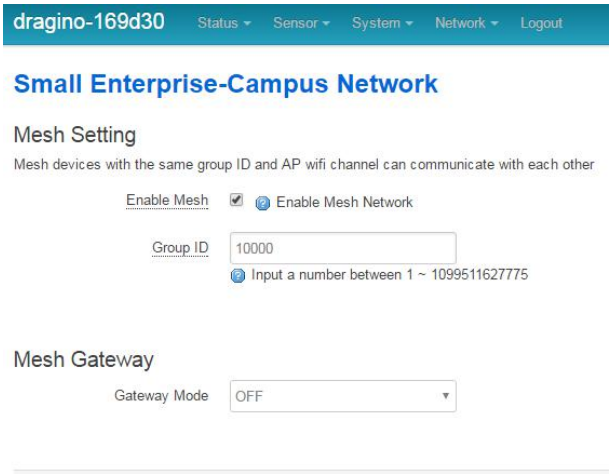
✓ Enable WiFi AP(可选)，mesh 网络中的网关和节点可以设置相同的 AP SSID。



Network --> Mesh Network

✓ Enable mesh

✓ 输入 Group ID//**注:只有同一组内的 Mesh 设备可以相互通信.**



3.5.2 网客户端设置

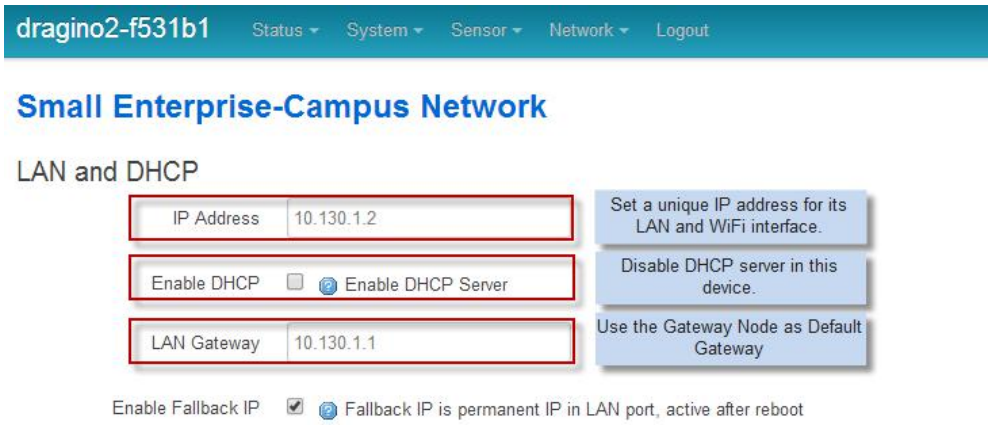
Network --> Internet Access

✓ 设置 Access Internet Via 为 Disable

Network --> LAN and DHCP

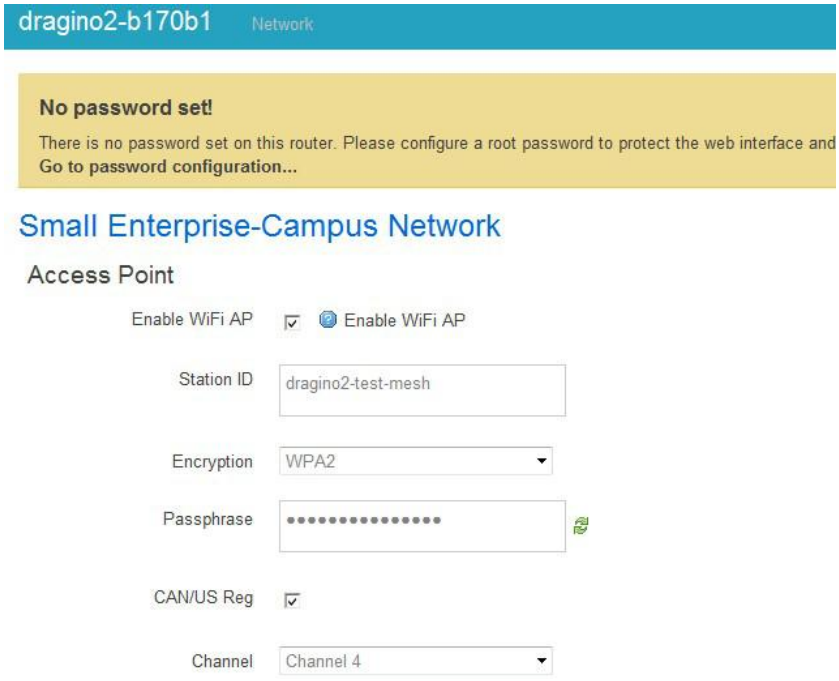
✓ 禁用 Enable DHCP

✓ LAN Gateway 设置为 Mesh 网关的地址



Network --> Access Point

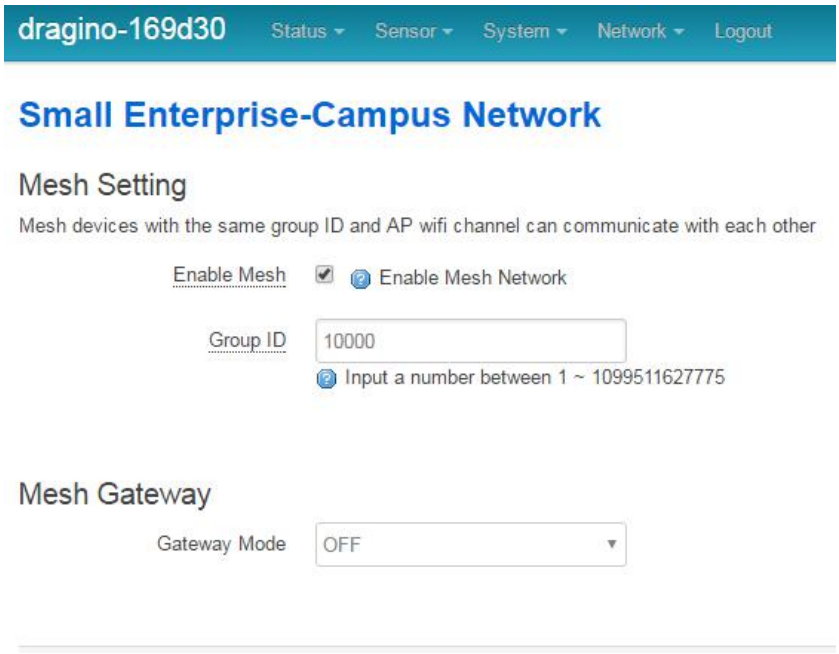
✓ Enable WiFi AP(可选)，mesh 网络中的网关和节点可以设置相同的 AP SSID。



Network --> Mesh Network

✓ Enable Mesh

✓ 输入 Group ID//**注:只有同一组内的 Mesh 设备可以相互通信.**

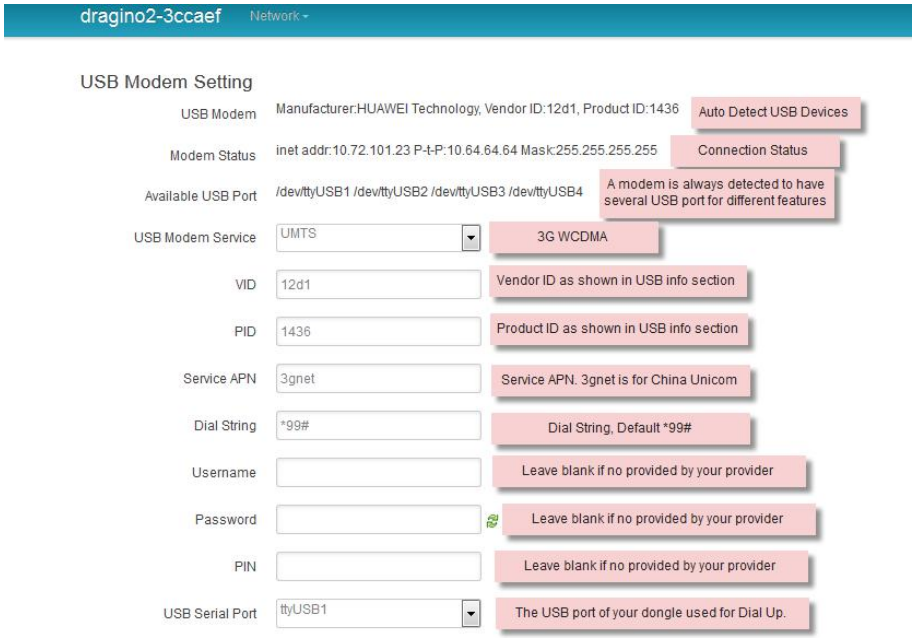


3.6 USB Modem 拨号上网

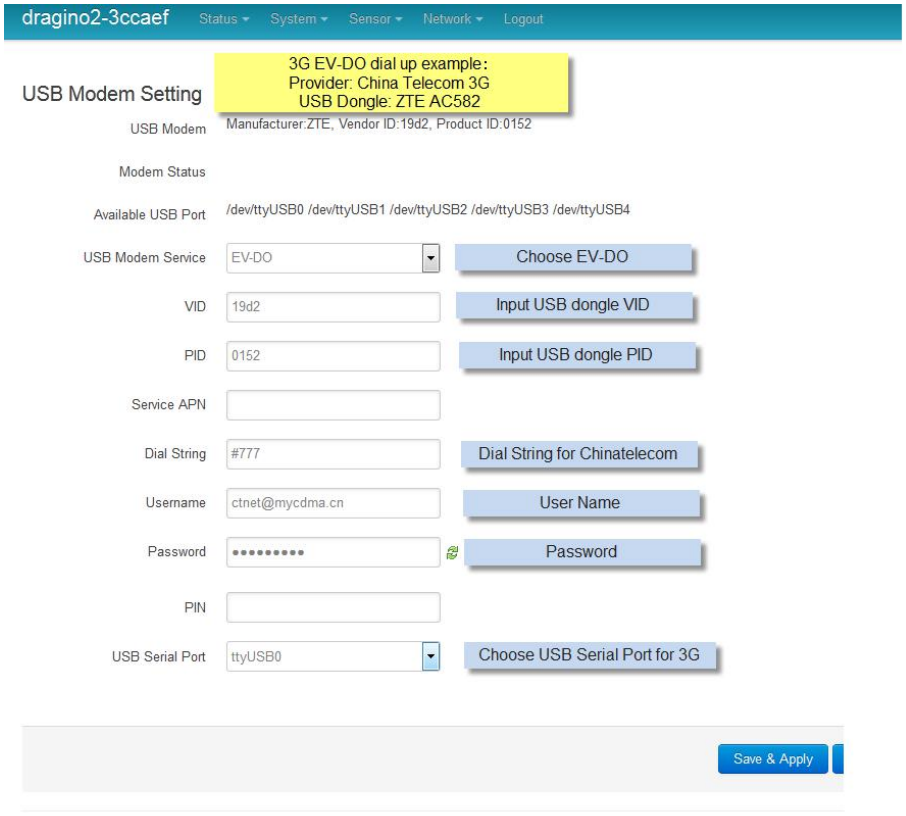
Dragino 的 USB 接口可用于连接 GPRS/3G/4G 等 USB 上网卡来上网.以下列举了一些例子.

注意:如果用户使用 ec20/uc20 模块，只需要在互联网设置中选择 USB Modem，然后重新启动设备，该设备将自动配置以支持 ec20/uc20 模块。

联通 WCDMA 设置示例:



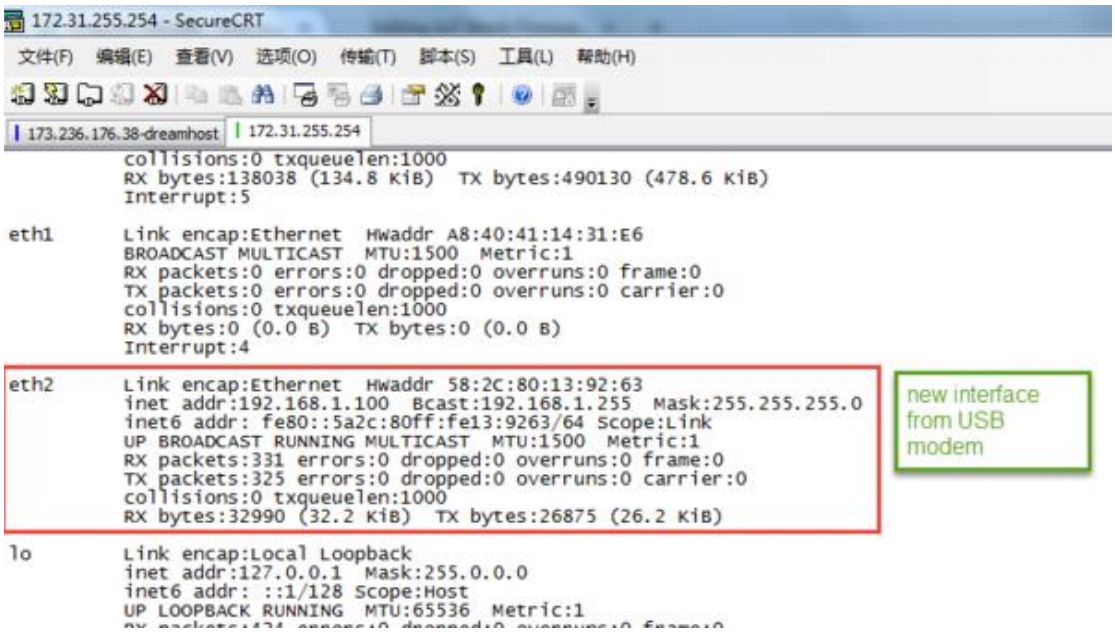
中国电信 3G EV-DO/CDMA2000 配置例子:



3.7 USB 3G/4G 以太网上网卡

一些 USB 上网卡没有通过拨号来连接互联网。相反，它们是作为一个网络接口出现，并具有内置的路由器特性。华为的 Hilink dongles 是典型的例子。当用户把这样的 USB 上网卡插入电脑时,它会自动连接到互联网和重定向到一个 web 界面,电脑会从这种 USB 上网卡获取一个局域网的 IP 地址来上网。

这样的 USB 上网卡插入 LG01 时,LG01 会出现一个新的接口(通常 eth2 或 usb0), 通过运行命令“ifconfig -a“可以看到这个新接口。用户可以使用 web UI 为网络连接配置使用这些 USB 上网卡。



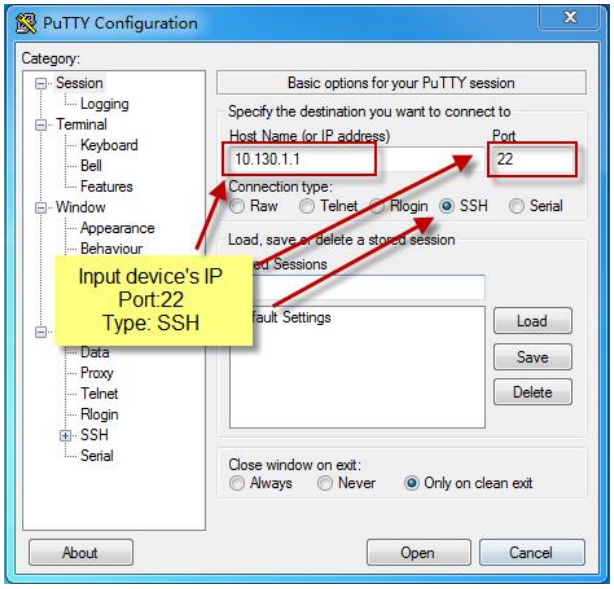
一个由 USB 上网卡生成的新接口

4 Linux 系统

LG01 内置开源 OpenWrt Linux 系统，用户可以自由地配置和修改内部 Linux 设置。

4.1 Linux 控制台的 SSH 访问

用户可以通过 SSH 协议访问 Linux 控制台。确保您的 PC 和 LG01 在同一个网络中，然后使用 SSH 工具(例如 putty)访问它，下面是截图:

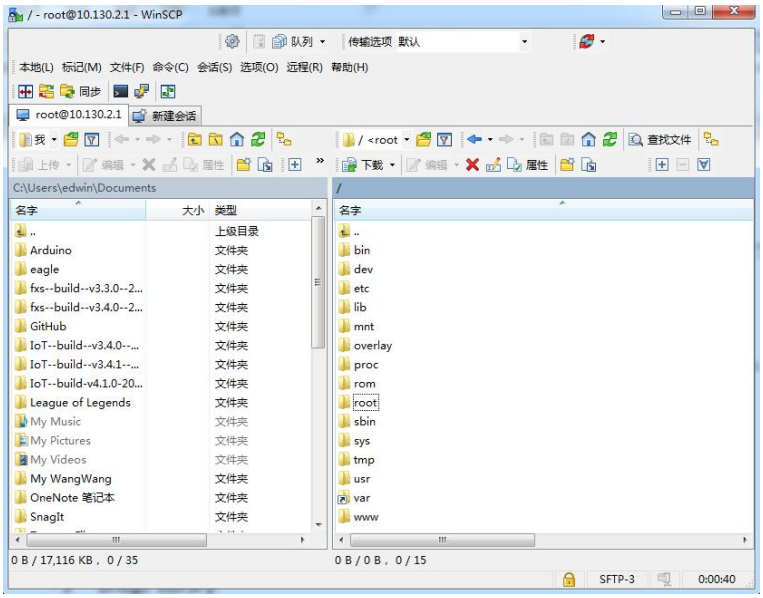


IP 地址: LG01 的 IP 地址
端 口: 22
用户名: **root**
密 码: **dragino** (默认设置)

登录之后，您将看到 Linux 控制台

4.2 文件的编辑和传输

LG01 支持 **SCP 协议**，并有一个构建的 **SFTP 服务器**。有很多方法可以使用这两种协议来编辑和传输文件。最简单的一种是通过 WinSCP 实用程序。通过 WinSCP 访问设备后，可以使用一个 FTP 相似的窗口将文件拖拽到 LG01，或者直接在 window 中编辑文件，截图如下：



4.3 文件系统

LG01 有 16MB 的 flash 和 64MB 的 RAM。/var 和/tmp 目录位于 RAM 中，这意味着在重新启动设备后，/tmp 和/var 中的内容将被删除。除了/var 和 /tmp 目录，其他文件存储在 flash 中，在重新启动后将继续保存。

Linux 系统使用大约 8MB~ 10MB 的 flash 大小，这意味着用户在 LG01 闪存中存储数据的空间并不大。用户可以使用外部 USB 闪存来扩展存储空间。

4.4 软件包维护系统

LG01 使用 **OPKG 软件包维护系统**。在我们的软件包服务器中有超过 3000 多个软件包供用户为他们的应用程序安装。例如，如果用户想要添加 MQTT 支持，他们可以安装相关的包并配置 LG01 来支持 MQTT。

下面是一些 OPKG 命令的例子，更多的请参考 **OPKG 软件包维护系统**。

在 Linux 控制台运行:

root@dragino-169d30:~# opkg update // 获取最新的包列表

root@dragino-169d30:~# opkg list //显示可用的包

root@dragino-169d30:~# opkg install mosquitto-client //安装 MQTT 客户端，它自动安装所需的软件包.

Installing mosquitto-client (1.3.5-1) to root...

Downloading http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/mosquitto-client_1.3.5-1_ar71xx.ipk.

Installing libcares (1.10.0-1) to root...

Downloading http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/libcares_1.10.0-1_ar71xx.ipk.

Installing libmosquitto (1.3.5-1) to root...

Downloading http://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/packages/packages/libmosquitto_1.3.5-1_ar71xx.ipk.

Configuring libcares.

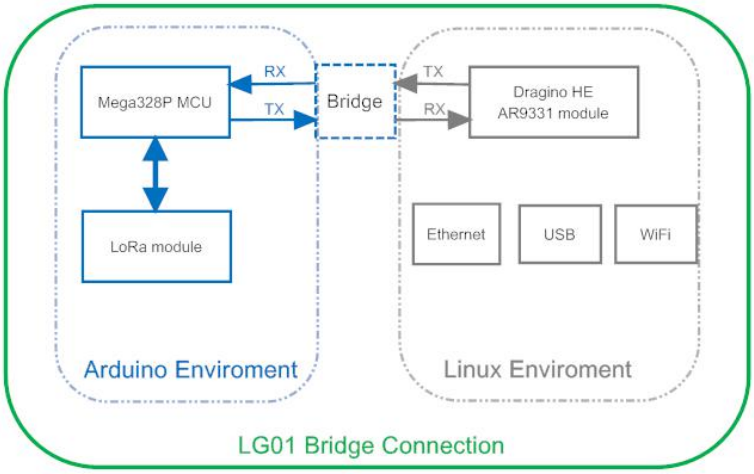
Configuring libmosquitto.

Configuring mosquitto-client.

5 Bridge 库

Bridge 库是 LG01 最重要的部分。Bridge 库定义了 MCU 如何与 CPU 通信(ar9331)的机制。在 Bridge 库中，MCU 可以将数据发送到 Linux CPU，从 Linux CPU 中获取结果，或者在 Linux CPU 中调用命令。

Bridge 库使用 UART 接口在 MCU 和 ar9331 之间进行通信。下图显示了 ATMega328p MCU 和 Linux 之间的 Bridge 连接。

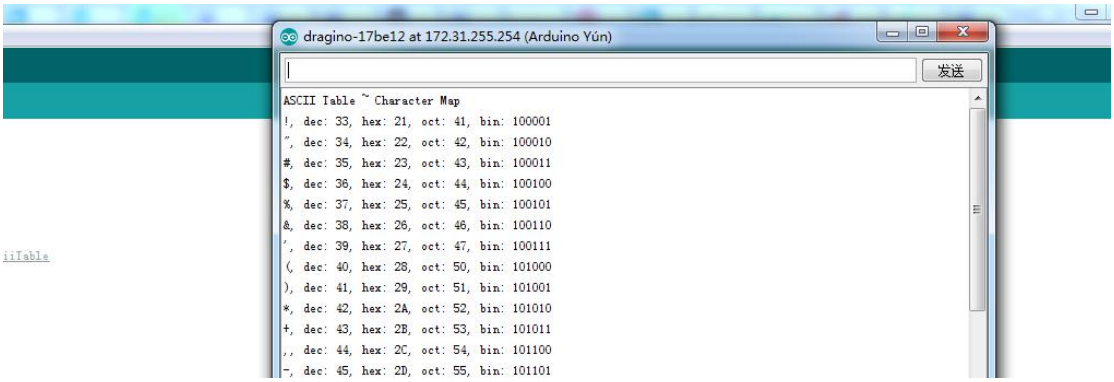


在 Arduino Bridge 库中可以找到如何使用 Bridge 库的详细说明。由于硬件的不同，我们在阅读和使用 Arduino 网站上的例子时，有一些要点:

- 我们建议用户先尝试使用 **Arduino IDE --> Files --> Examples --> Dragino** 下的例子
- 当使用 Bridge 类时，用户需要调用 Bridge.Console。在 LG01 的 单片机固件中调用 Bridge.Console (115200)
- 在 Arudino IDE 的默认 Bridge 示例中，它使用 Serial 类打印调试信息。这在 LG01 中不起作用。因为 Serial 类将调用 ATMega328p 的硬件 Serial 端口，它将与 Bridge 库发生冲突。如果用户需要打印调试信息，请使用 Console 类。

5.1 使用 Console 库来打印调试信息

我们可以用 Console 库来打印程序的调试信息。 例子 **Arduino IDE --> Files --> Examples --> Dragino-->Bridge-->ConsoleRead** 展示了如何使用 Console 库来打印信息到 Arduino IDE. 如下



除了使用 Arduino IDE 的串口监视输出外，我们可以使用 SSH 登录到 LG01，然后运行 **telnet localhost 6571** 来获取 Console 的结果，如下:


```
BusyBox v1.23.2 (2017-06-24 23:34:27 CST) built-in shell (ash

 _______                     ________        __
|       |.-----.-----.-----.|  |  |  |.----.|  |_
|   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
|_______||   __|_____|__|__||________||__|  |____|
         |__| W i F i, L i n u x, M C U, E m b e d d e d

OpenWRT Chaos Calmer 15.05
Version: Dragino-v2 IoT-4.2.2
Build Wed Jul 19 15:06:00 CST 2017

www.dragino.com
----------------------------------------------------

root@dragino-17be12:~# telnet localhost 6571
ASCII Table ~ Character Map
!, dec: 33, hex: 21, oct: 41, bin: 100001
", dec: 34, hex: 22, oct: 42, bin: 100010
#, dec: 35, hex: 23, oct: 43, bin: 100011
$, dec: 36, hex: 24, oct: 44, bin: 100100
%, dec: 37, hex: 25, oct: 45, bin: 100101
&, dec: 38, hex: 26, oct: 46, bin: 100110
', dec: 39, hex: 27, oct: 47, bin: 100111
(, dec: 40, hex: 28, oct: 50, bin: 101000
), dec: 41, hex: 29, oct: 51, bin: 101001
*, dec: 42, hex: 2A, oct: 52, bin: 101010
+, dec: 43, hex: 2B, oct: 53, bin: 101011
,, dec: 44, hex: 2C, oct: 54, bin: 101100
-, dec: 45, hex: 2D, oct: 55, bin: 101101
., dec: 46, hex: 2E, oct: 56, bin: 101110
```

6 进阶管理

6.1 重置网络和重置出厂设置

LG01 提供了用户重置设备的方法。当 Linux 系统运行时，用户可以按下 toggle 按钮来重置设备。按压的时间将决定要重新设置哪个部分.

- 按下 toggle 按钮，地球灯将闪烁，5 秒后松开按钮，设备将重置网络设置并重新启动(地球/LAN/WAN/WiFi 灯闪烁)，其他设置将被保留。
- 按下 toggle 按钮，地球灯将闪烁，30 秒后松开按钮，设备将重置所有设置为出厂默认设置，并重启(地球/LAN/WAN/WiFi 灯闪烁)。

7 升级 Linux 固件

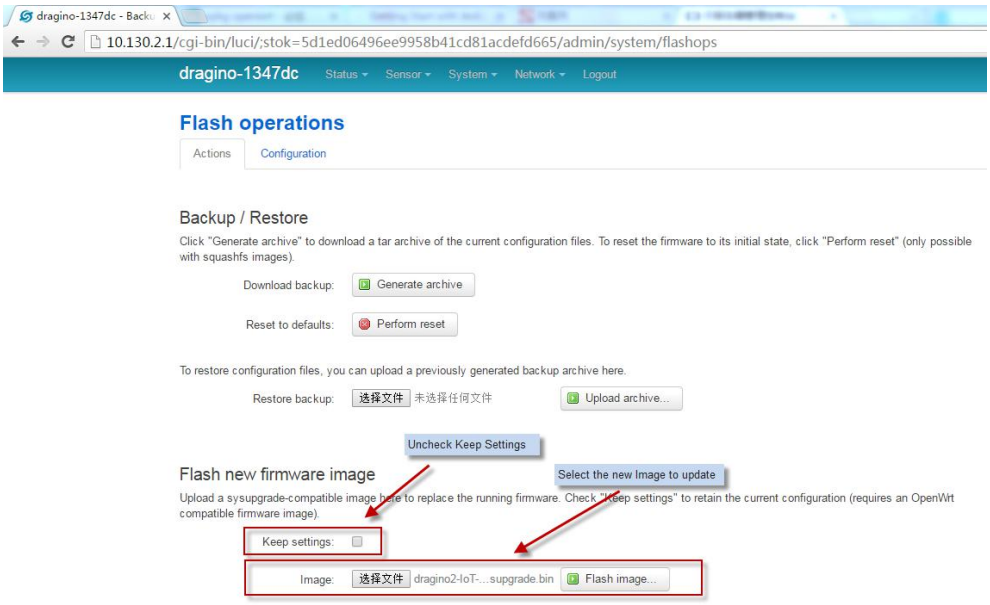
我们不断改进 LG01 Linux 侧固件，以添加新的特性和 bug 修复。最新的固件可以在 IoT Mesh Firmware 找到，并且可以在这里找到更改记录: Firmware Change Log.

这个文件命名为 **dragino2-IoT--xxxxx-squashfs-sysupgrade.bin** 是升级用的 Image。有不同的升级方法，如下：

7.1 通过 Web UI 升级

转到页面: **Web --> System --> Back Up and flash firmware**，选择 Image 并点击 flash Image，Image 将被上传到设备上，然后点击 Process Update 来升级。

系统将在升级后自动启动新固件.



7.2 经由 Linux Shell 升级

将固件发送到系统**/var** 目录，然后运行

root@OpenWrt:~# /sbin/sysupgrade -n /var/Your_Image

注意:将 Image 传输到在/var 目录很重要，因为这个固件超过 flash 可用空间的大小，如果传到其他目录中，会导致系统崩溃。

8 上传 MCU 固件

我们提供了三种方法供用户升级 LG01 的 MCU 侧的固件。

8.1 通过 Arduino IDE 上传

我们在上面已经用过这个方法，点击查看。

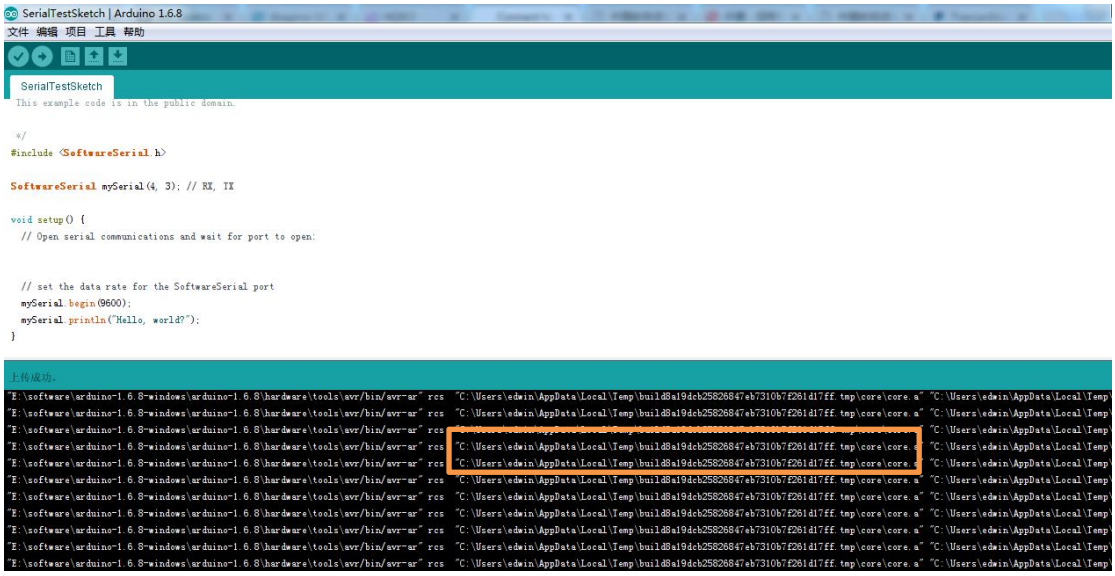
8.2 通过 Web UI 升级 MCU 固件

用户可以通过 WEB 页面来升级量产的 MCU 固件。 这个时候的固件是.hex 文件格式。用户可以在编译的临时文件夹中找到这个文件。下面是升级说明。

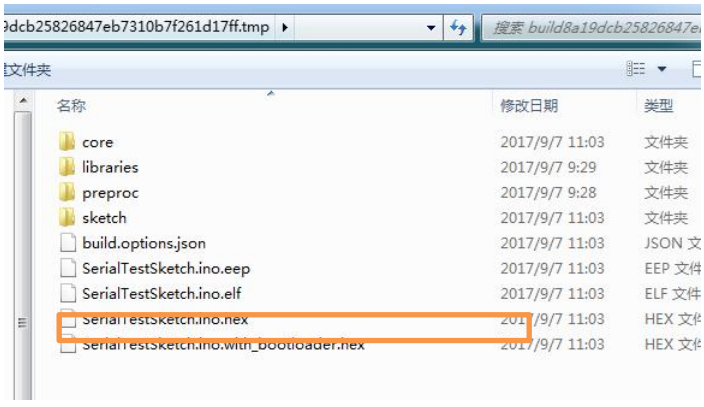
- 打开 Web 页面 **Sensor --> Flash MCU**，选择正确的 hex 文件上传.
- 重启 LG01，重启之后，检查配置页面 **Sensor --> MicroController**，如果固件有对版本定义，这边会显示 MCU 的版本号.

如何获取 hex 文件?

在 Arduino 编译 Sketch 的时候，会生成项目对于的临时文件夹， 用户可以在这个文件夹中找到对于的 hex 文件。



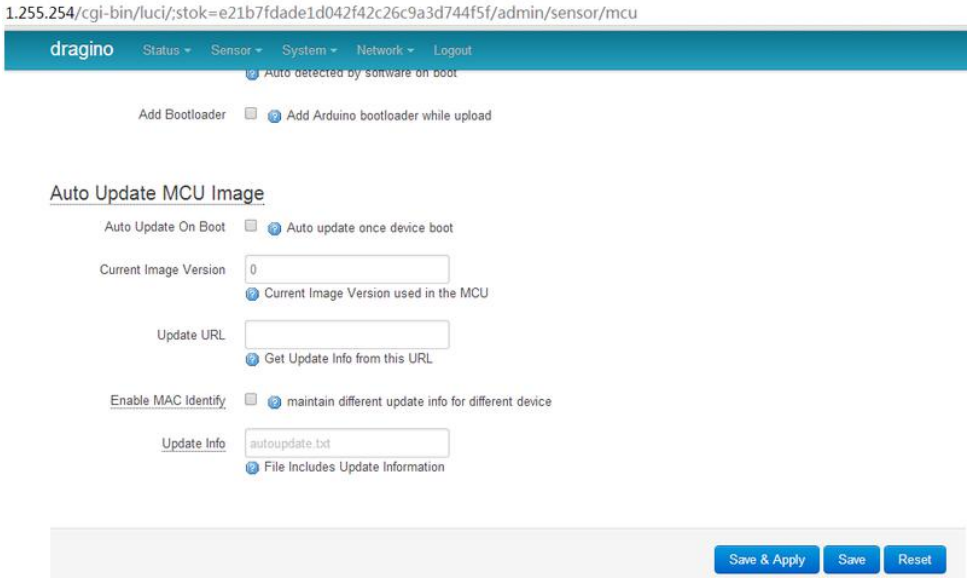
在临时文件夹中找到 hex 文件 (不要选择带 bootloader 后缀那个固件文件)



8.3 MCU 自动更新

从固件更新了 4.1.1 之后，linux 系统支持自动更新单片机侧固件。有了这个特性，Dragino 将连接到 http/https 服务器，获得最新的单片机固件，并将这个固件上传到单片机中。通过自动更新功能，我们可以降低远程安装的技术支持成本和时间。

该特性可以在页面 **sensors -> microcontroller** 中配置。



- **Auto Update On Boot**: 假如启用该选项。LG01 在每次启动时将连接到自动更新服务器，并检查是否有新版本的固件需要更新。如果设备在自动更新服务器上找到更新版本，设备将从服务器下载，并使用这个新版本更新 mcu。
- **Current Image Version**: 显示当前的固件版本。默认情况下是 0。当自动更新成功之后，LG01 会将这个版本更新到最新版本号。
- **Update URL**: 此 URL 包含更新信息和 sketch.hex 文件。LG01 通过连接到该 URL 来检查是否有更新的版本
- **Update Info**: 此文本文件包含更新信息。文件的一个示例格式可以在这里找到：example for update information file.它应该包括：
  - **image**: 用于自动更新的单片机固件
  - **md5sum**: md5sum 的固件
  - **version**: 最新的版本号
- **Enable MAC Identify**: 与在更新信息中指定的更新信息不同，该设备将从 Update URL 中寻找更新信息文件: **wifi_mac.txt**。这意味着，如果设备的 wifi mac 地址为 A840417867AF，设备将下载该文件: $Update_URL/A840417867AF.txt.用于自动更新信息。

自动更新程序步骤:

假设我们有以下配置：

Auto Update On Boot：checked
Update URL: http://www.dragino.com/downloads/downloads/tmp/autoupdate/
Update Info: update_info
Enable MAC Identify: unchecked

重新启动后，该设备将自动更新如下:

1. 从 http://www.dragino.com/downloads/downloads/tmp/autoupdate/update_info 下载更新信息
2. 比较最新版本和设备上的版本
3. 如果服务器有更高版本，设备将下载 http://www.dragino.com/downloads/downloads/tmp/autoupdate/sketch.hex
4. 执行 md5sum 检查来验证下载的固件是否正确
5. 使用新版本的固件更新 MCU
6. 将版本号更新为最新版本号

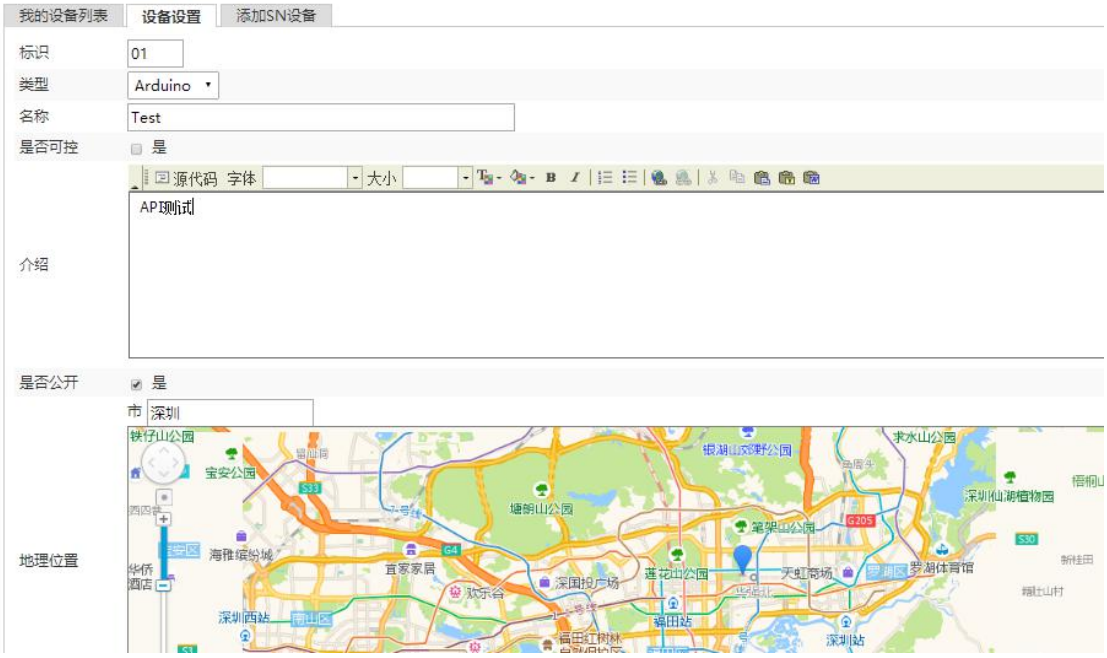
9 示例：将 LoRa 与 RESTFul API 结合

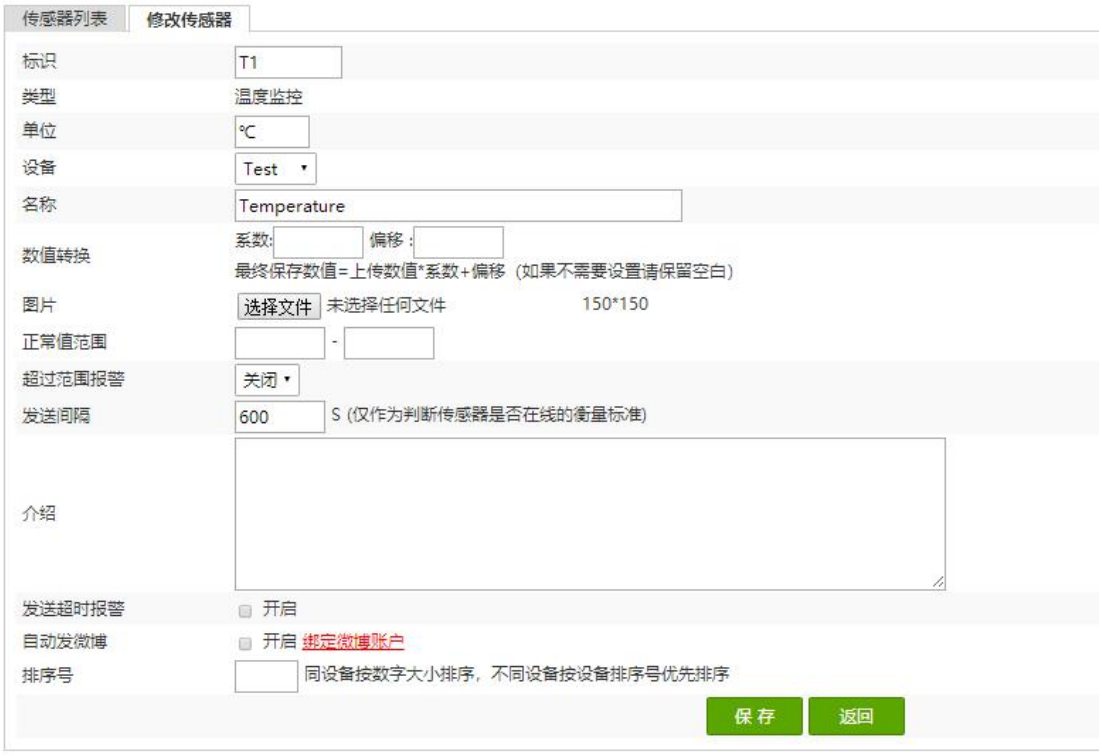
9.1 RESTFul API 是什么？

RESTful API 是一个应用程序接口(API)，它使用 HTTP 请求来获取、放置、发布和删除数据。许多物联网服务提供 RESTful API 作为传感器数据通信的上行/下行链路方法之一。这个示例将展示如何使用 LG01 通过 RESTful API 与 IoT 服务器进行通信，从而实现将传感器数据上传到 IoT 服务器或从 IoT 服务器下行命令。

9.2 配置 IoT 服务器

许多服务器支持 RESTful API，我们这里使用一个有直观图表来显示我们测试结果的服务器乐联网。方法是通用的，也可以在其他的 IoT 服务器一起使用。要使用服务器，我们需要在乐联网上注册一个账户，然后创建一个设备和增加一个传感器并键入信息。如下所示：

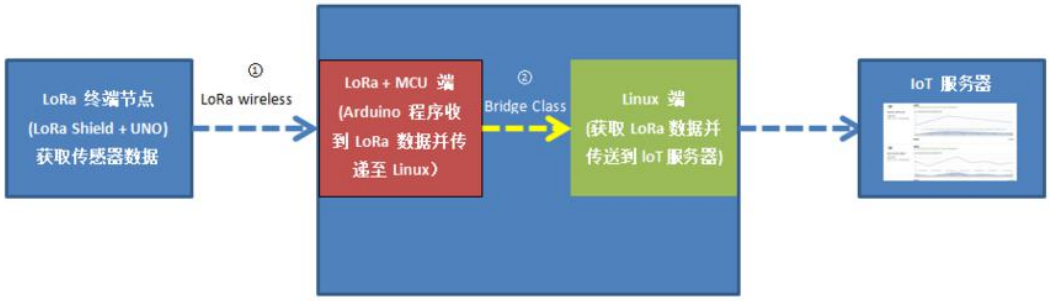




为了实现 LG01 与服务器通信，我们还需要 API Keys。在乐联网中，我们可以在用户组里找到 API Keys 界面：

9.3 逐步上传测试

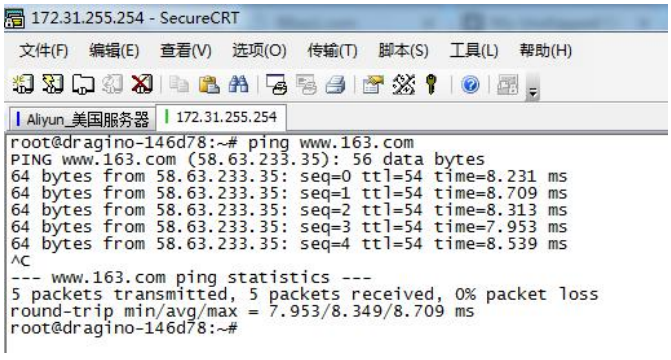
在本节中，我们将尝试对 LG01 进行编程以便数据上传传输到乐联网中。本例的结构图如下：



数据流：
①: LoRa 终端节点从传感器获取数据并通过 LoRa 无线协议发送出去。
②: LG01 中的 LoRa/MCU 部分从 LoRa 无线获取传感器数据，并把数据传给 Linux 端。
③: LG01 中的 Linux 部分将传感器数据以 RESTful API 格式发送给 IoT 服务器。

9.3.1 LG01 通过 Linux 命令尝试 RESFul API 调用

首先，我们需要确保 LG01 有互联网接入。我们可以登陆到 SSH 和 ping 一个因特网地址，看它是否通过。如下：



LG01 有内置的 Linux 工具 curl。它是一个非常强大的 http 通信工具，我们可以使用此工具在 LG01 中处理 RESTful API 调用。

上传传感器数据的命令如下：

curl --request POST http://www.lewei50.com/api/V1/Gateway/UpdateSensors/01 --data "[{'Name':'T1','Value':'23'}]" --header "userkey:c94a2ba527"

下面是输出窗口：

在服务器上查看结果：

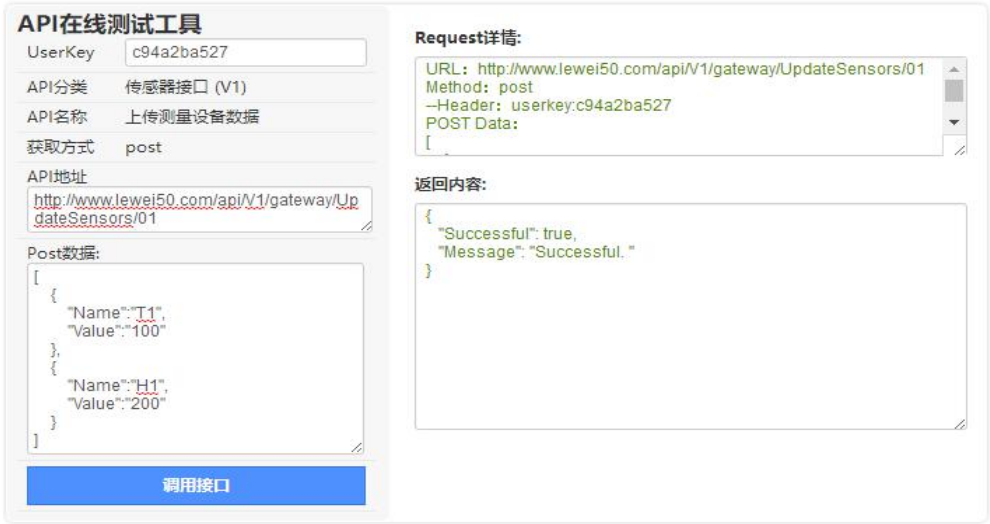
在这里，我们成功地使用 LG01 上传数据到乐联网, curl 命令在 Linux 端执行。最后，我们必须调用带有传感器数据变量的 curl 命令。

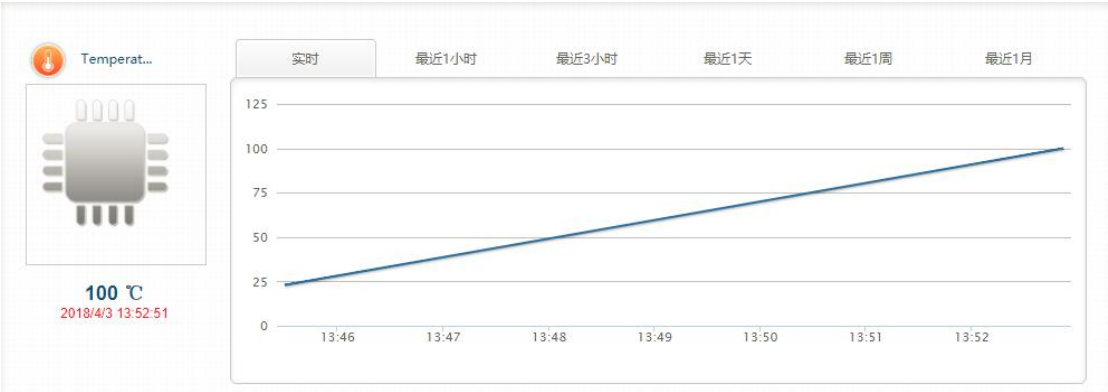
9.3.2 通过网页发送请求尝试 RESFul API 调用

我们可以直接使用乐联网网页上 API 在线测试工具。方法如下：

API Key: 你的 User Key。
请求方式：选择 POST 模式。
传感器名字：T1，01H1
网关号：你的设备号

结果如下：

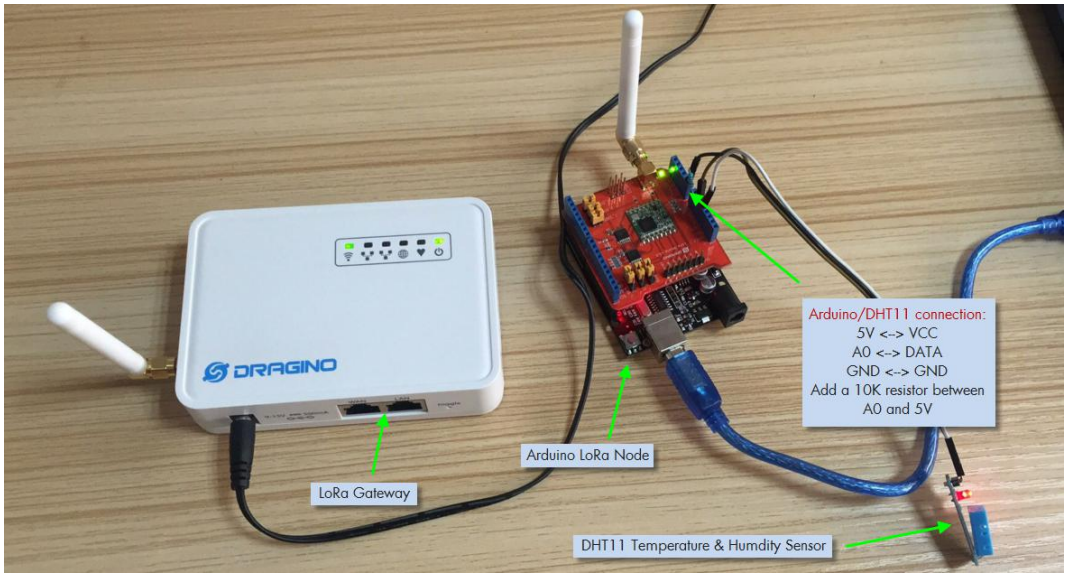




9.4 上传:从 LoRa 节点获取数据并发送到物联网（IoT）服务器

9.4.1 准备硬件

硬件设置如下，LoRa 节点将从 DHT11 获取传感器数据，并将这些数据发送到 LG01 网关。当网关获取数据时，它将把数据传递给物联网服务器。



9.4.2 建立物联网服务器账户

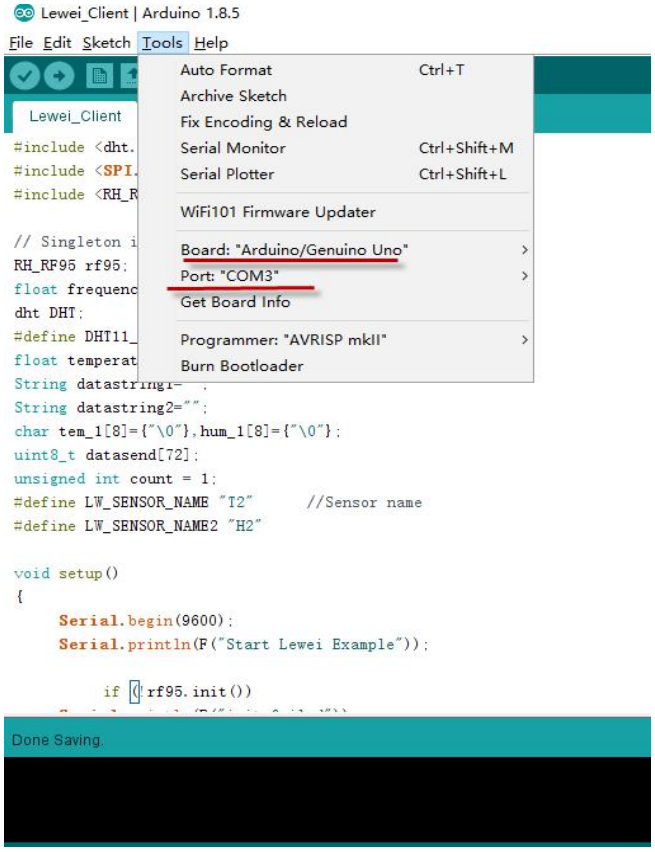
我们选择乐联网作为物联网服务器，上文配置 IoT 服务器里已经提到，如果您已有账户请直接登陆即可。（详情引导请参考：乐联网 开发指南）

9.4.3 上传单片机固件

步骤 1：打开 Arduino IDE，分别打开 server 和 client 两个文件，并添加一个 DHT 的库进去。（Lewei_Server），（Lewei_Client）。

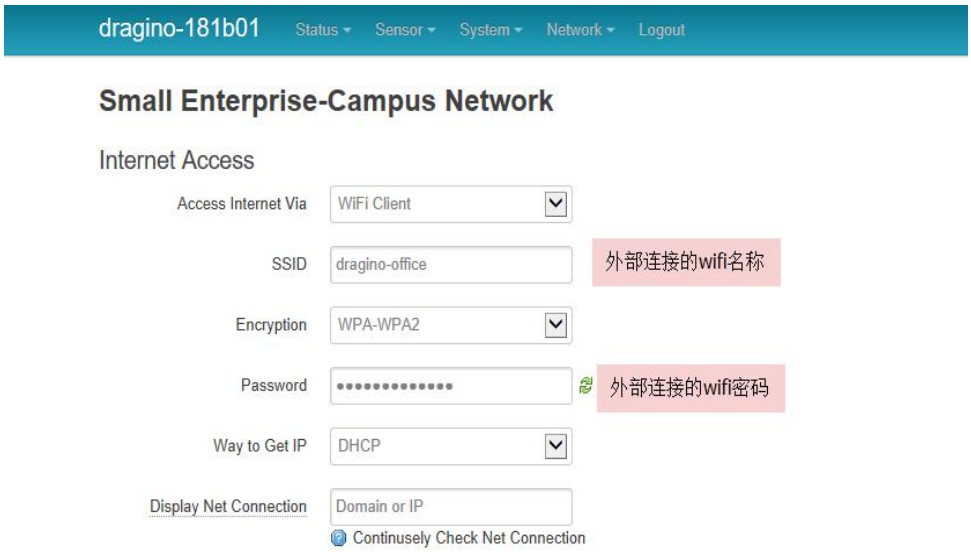
步骤 2：选择（Lewei_Client）文件，然后将这个固件上传到 LoRa 节点上。需要修改你所持设备的频率（例如：433 的设备可以更改为：433.0），和传感器的名称（示例：T1,H1）。

如图：

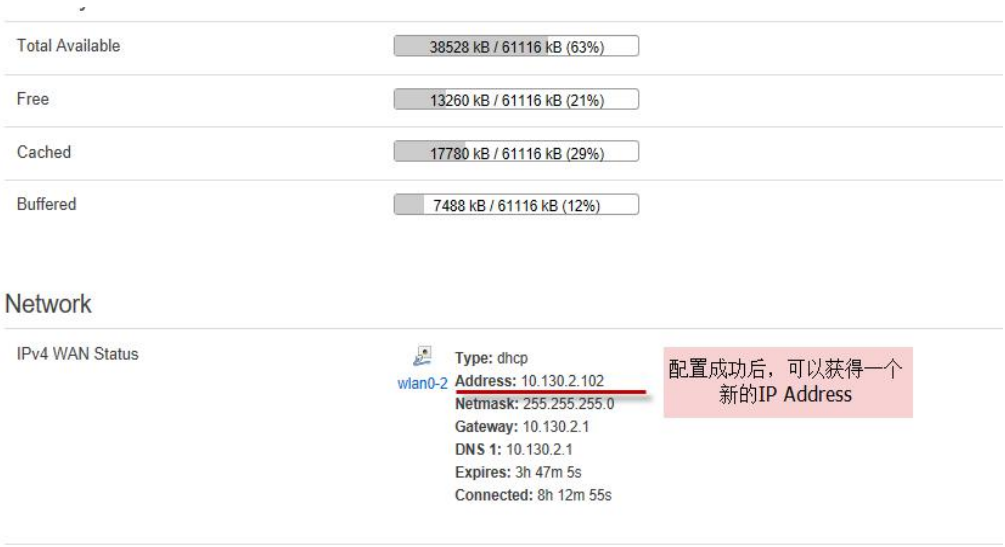


步骤 3：给 LG01 配置入网以及设定选项。操作如下：

打开 LG01 Web 控制台，依次进入 Network->Internet Access



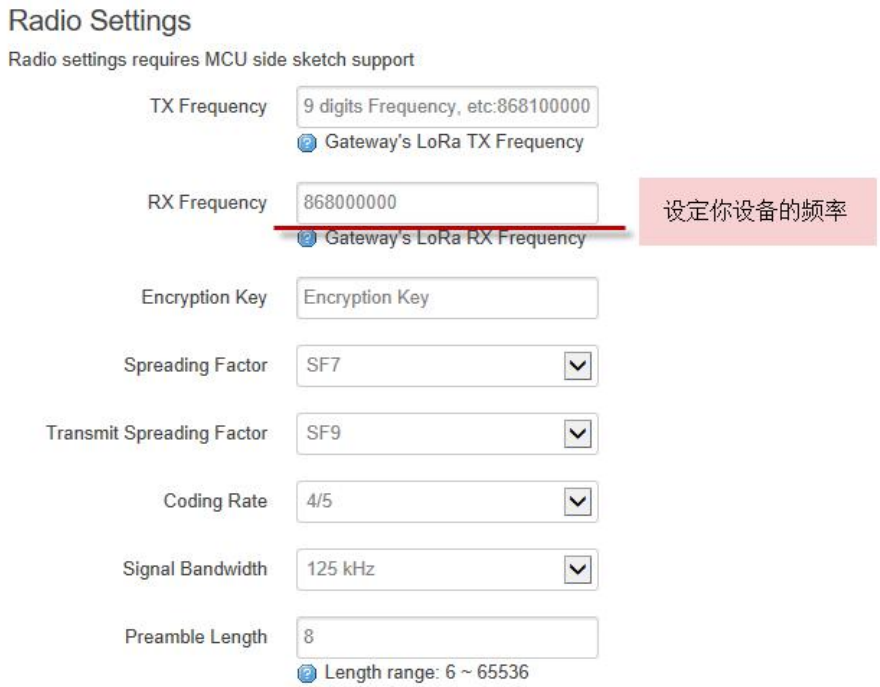
保存并且执行，你将在 Overview 上可以查看：



检查 ping 通结果，打开 SSH 控制台，访问新生成的 IP，输入指令。

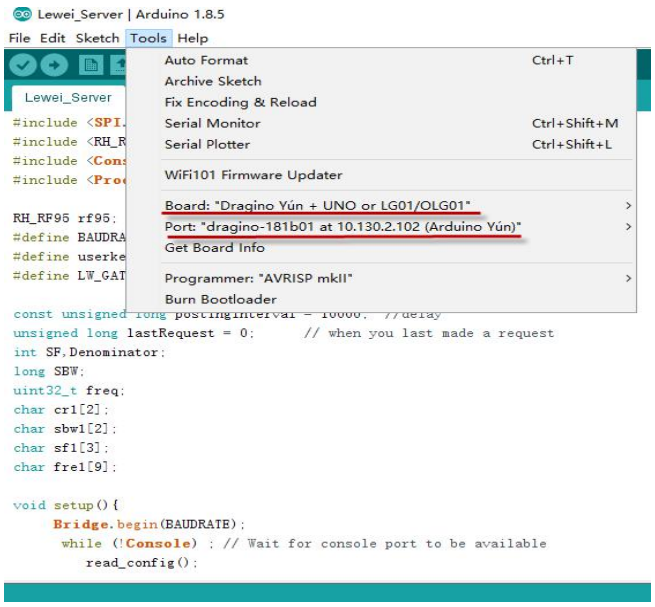
例：ping www.baidu.com

再回到 LG01 Web 控制台，进入 Sensor->LoRa/LoRaWAN,设定 RX 的频率。（示例是以 868 频率进行的。）



步骤 4：将 server 代码里 userkey 和 LW_GATEWAY 的参数替换为您的信息，并将这个固件上传到 LG01 网关。如图：

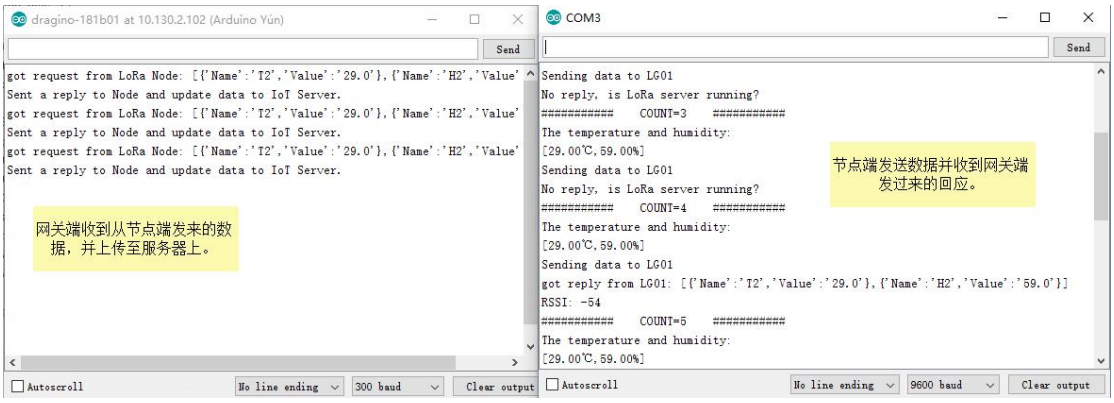
```
#define userkey "userkey:c94a2ba527"     //Userkey
#define LW_GATEWAY "01   "               //Gateway ID
```



9.5 检验结果

9.5.1 串口监视器查看结果

如下：



9.5.2 云服务器上查看结果

结果如下：

10 将 LoRa 与 MQTT 结合

10.1 什么是 MQTT?

MQTT 是机器对机器（M2M）/“物联网”连接协议。 它被设计成一个非常轻量级的发布/订阅消息传输。 对于需要较小代码空间和/或网络带宽较高的远程位置进行连接非常有用。例如，它已被用于通过卫星链路与经纪人进行通信的传感器，偶尔与医疗保健提供商进行拨号连接，以及家庭自动化和小型设备场景。大多数 IoT 服务器都支持 MQTT 连接，对于这些服务器，我们可以使用 MQTT 连接来发布数据或订阅频道。本示例将显示如何使用 LG01 通过 MQTT 连接到 IoT 服务器。

10.2 调用 MQTT API

10.2.1 工作原理

我们已经在上面尝试做过了①和②，接下来我们将进行③，在按照预期工作后，我们将把这三个步骤合为一个完整的上传示例。

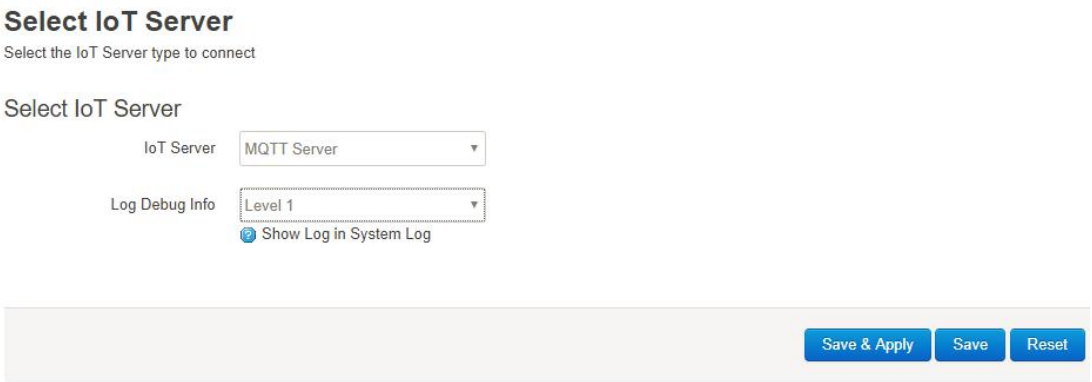
10.2.2 配置工作

要使用到 MQTT 功能，请确保 LG01 升级至版本 4.3.3。如升级成功，Sensor 选项里将会添加 MQTT 的选项。

步骤 1：

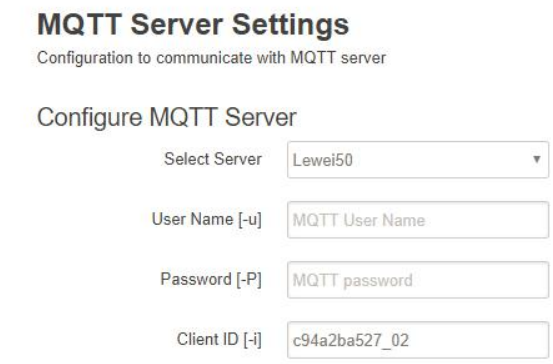
打开 LG01 web 控制台，进入 Sensor -> IOT Server

选择 MQTT 服务器。



步骤 2：

进入 MQTT 选项里，选择 Lewei50 服务器（乐联）

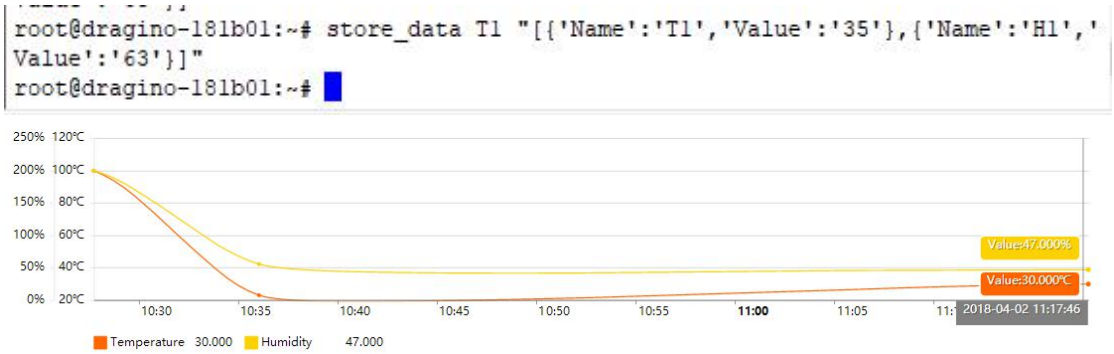


乐联网只需要配置 Client ID 即可，它的格式为：Userkey_xx(设备标识，需要创建)
图例：

步骤 3：
其余请参考入网和选项设定。

10.2.2 调用 MQTT API

打开 Linux SSH 控制台，输入指令



也可以使用 MQTT.fx 工具进行测试。具体步骤参考 wiki：调用 MQTT API。

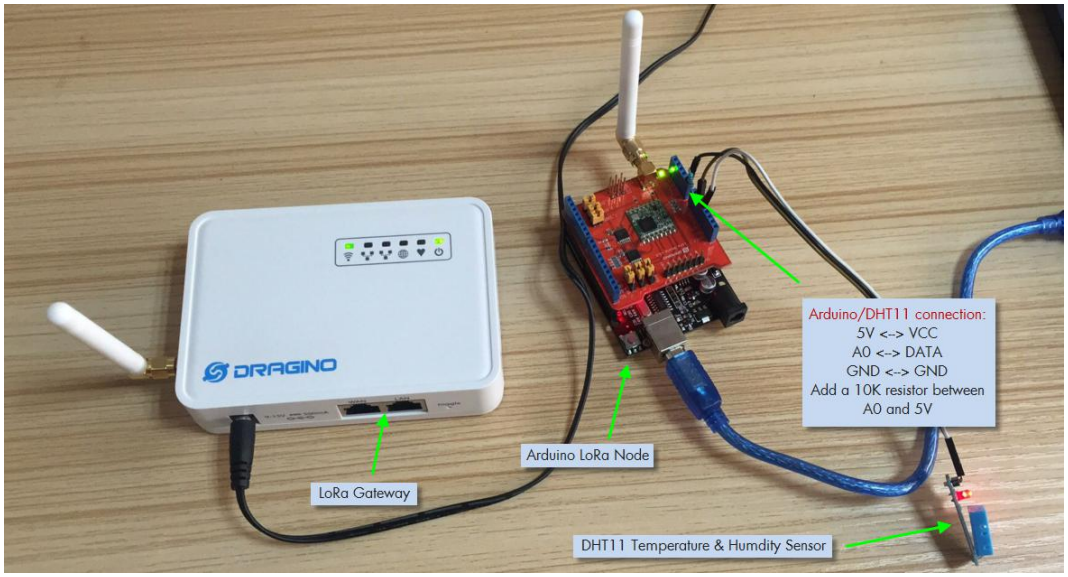
10.3 上传数据

这里将提供一个完整的示例。

硬件设置如下：

√ LoRa 终端节点： LoRa ShieLd + UNO + DHT11 温湿度传感器。LoRa 终端节点不断从传感器中获取温度和湿度的数据，并通过 LoRa 定期发送出去。

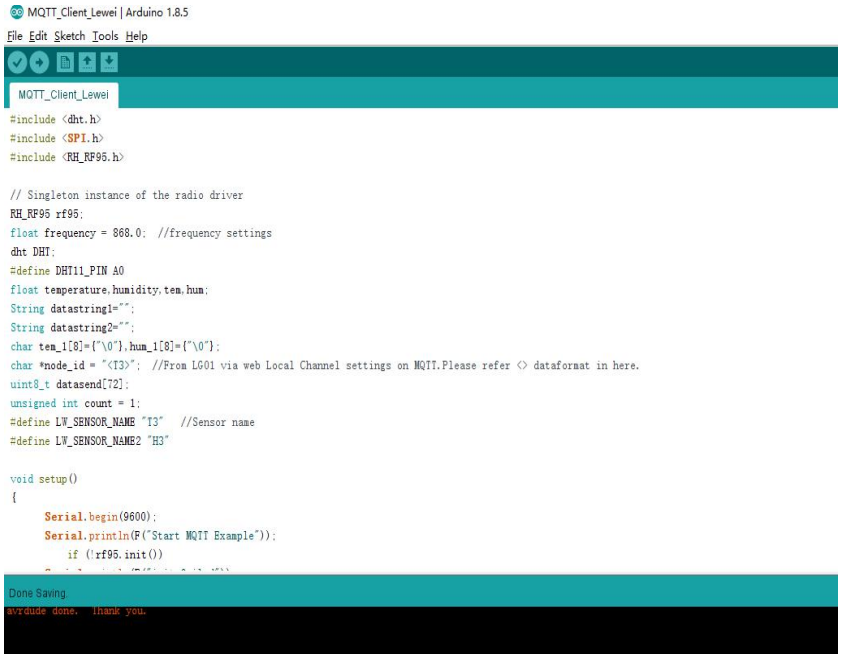
√ LoRa 网关 LG01：收听 LoRa 无线信道时，若有新的 LoRa 数据包到达，解析并发送至 IoT 服务器。



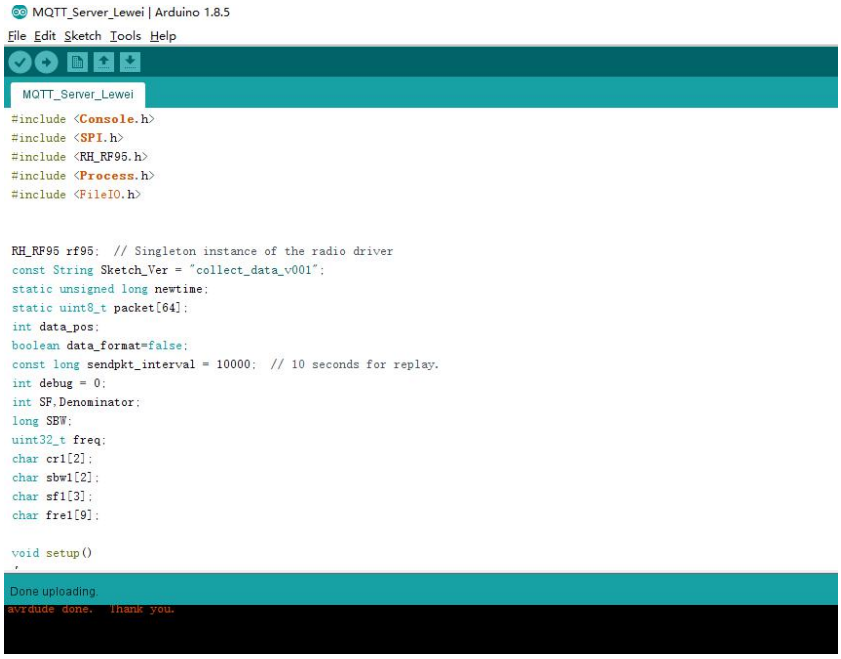
代码链接：

- LoRa Shield + UNO : MQTT_Client_Lewei
- LG01 LoRa Gateway: MQTT_Server_Lewei

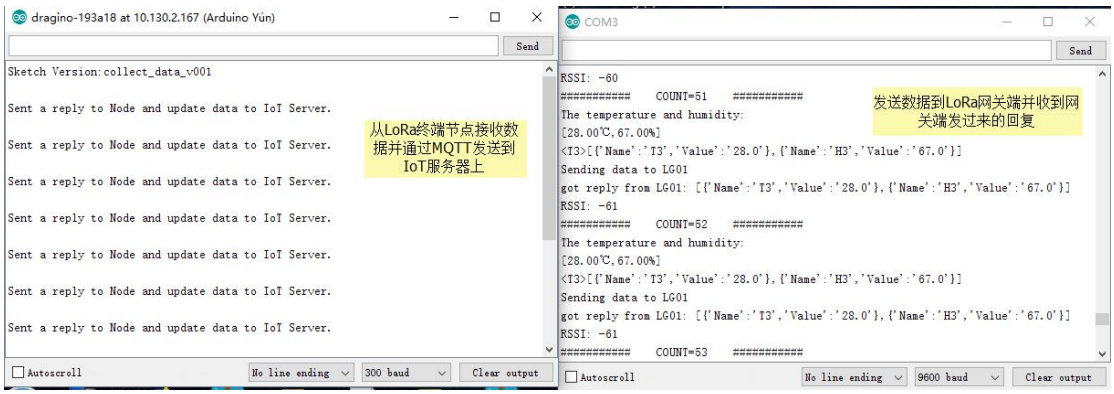
打开 MQTT_Client_Lewei，然后选择开发板 Arduino/Genuino UNO，端口选择 Serial port。



打开 MQTT_Server_Lewei，然后选择开发板 Dragino Yun + UNO Or LG01，端口选择 Network port。



打开串口监视器，查看结果：



前往服务器端，查看结果：

更多关于 MQTT，请参考：Lewei_Example

11 将 LoRa 与 TCP 结合

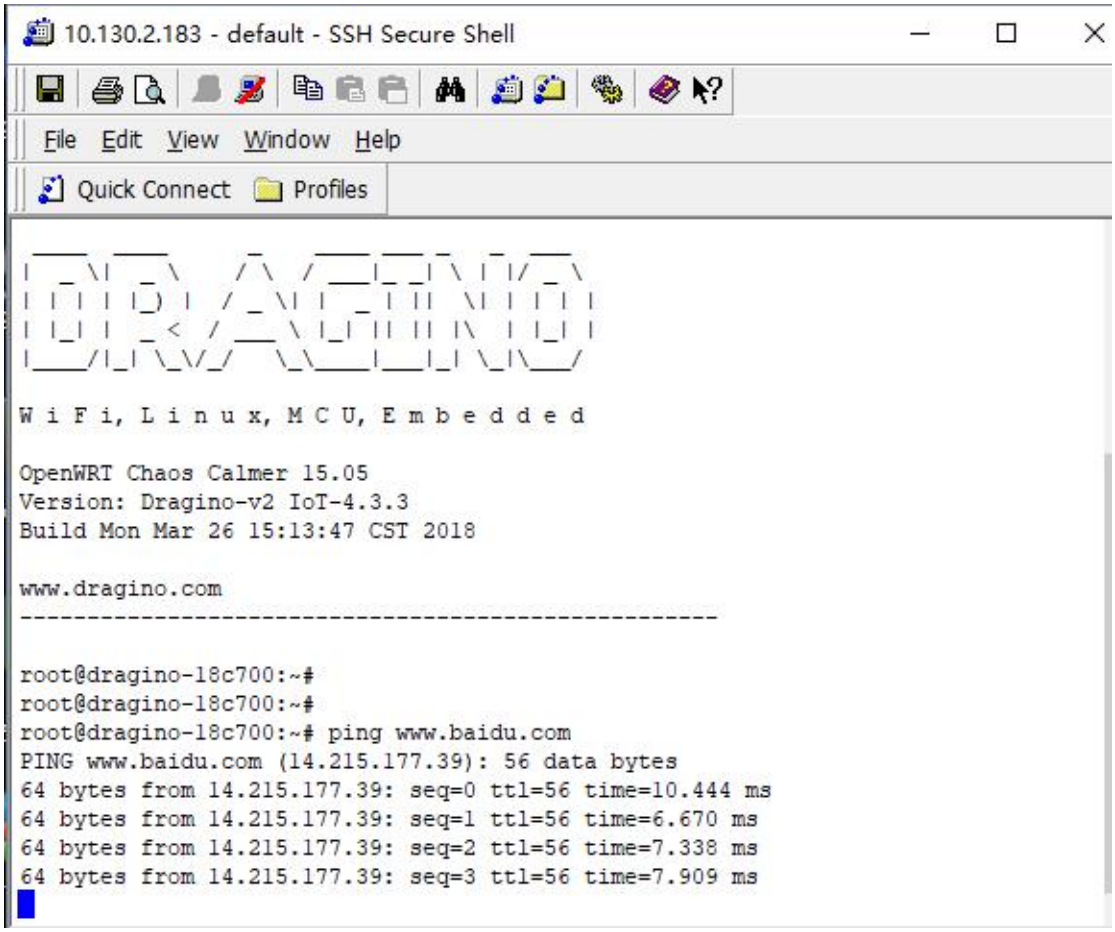
11.1 什么是 TCP

TCP 是一种面向连接的传输控制协议。它是计算机网络体系 OSI 模型中完成第四层传输层所指定的功能。TCP 层是位于 IP 层之上，应用层之下的中间层。不同主机的应用层之间经常需要可靠的、像管道一样的连接，但是 IP 层不提供这样的流机制，而是提供不可靠的包交换。 TCP 用于应用程序之间的通信，需要发送通信请求，在经过三次握手之后，客户端和服务端形成一个全双工可相互发送和接收消息。

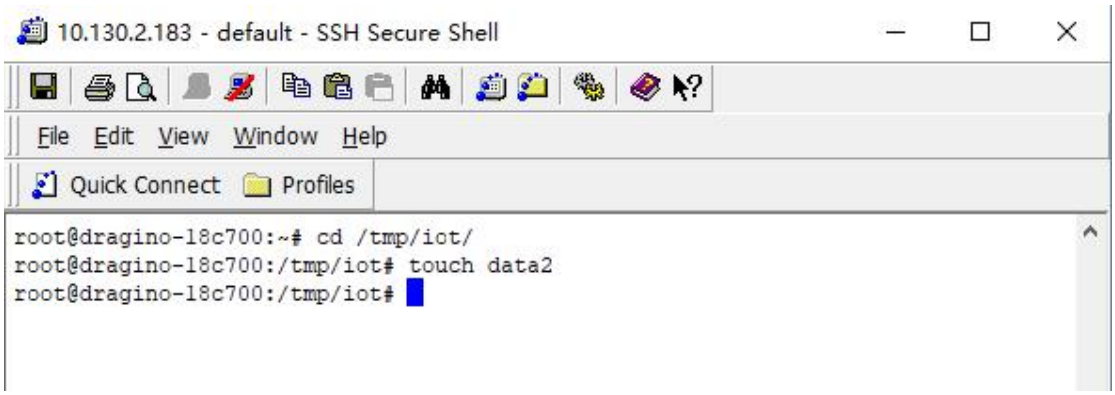
11.2 模拟数据发送

请升级至最新版本 4.3.4 。

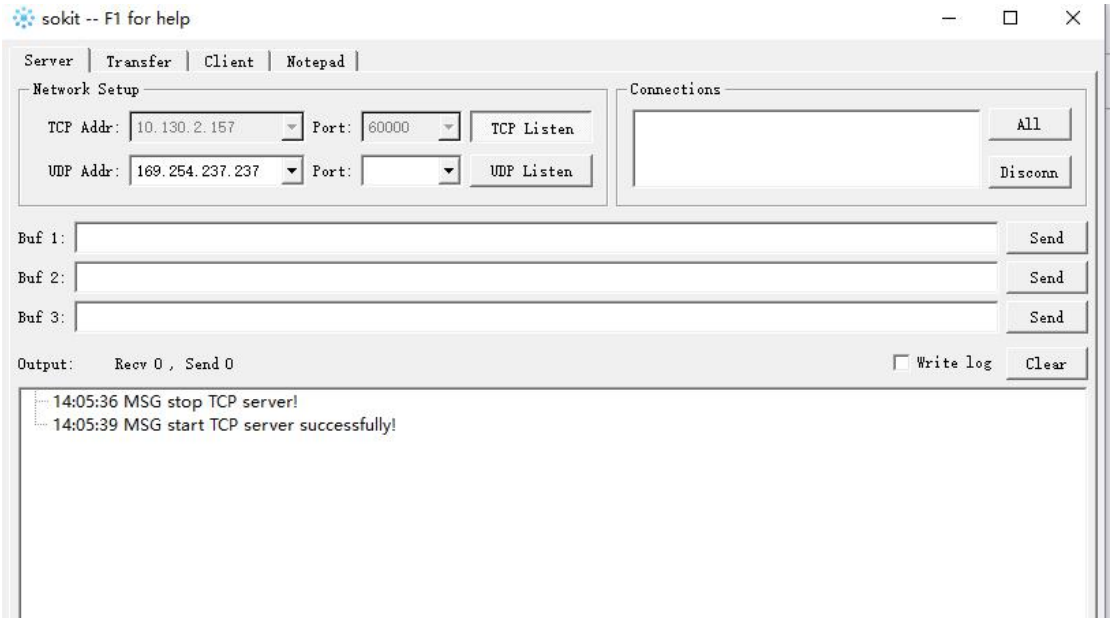
1. 打开 SSH，登陆后首先 ping 通一下入网。



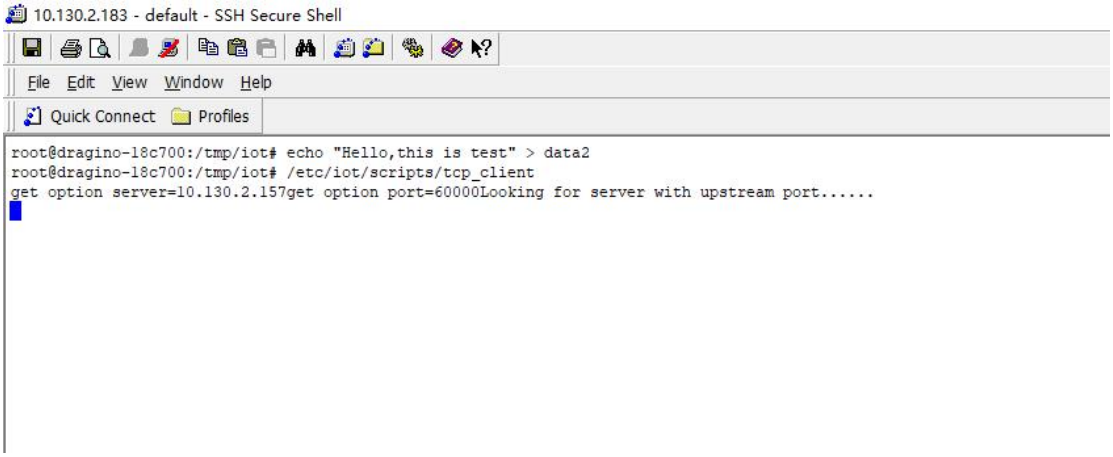
2. 输入指令创建一个文件命名为 data2。



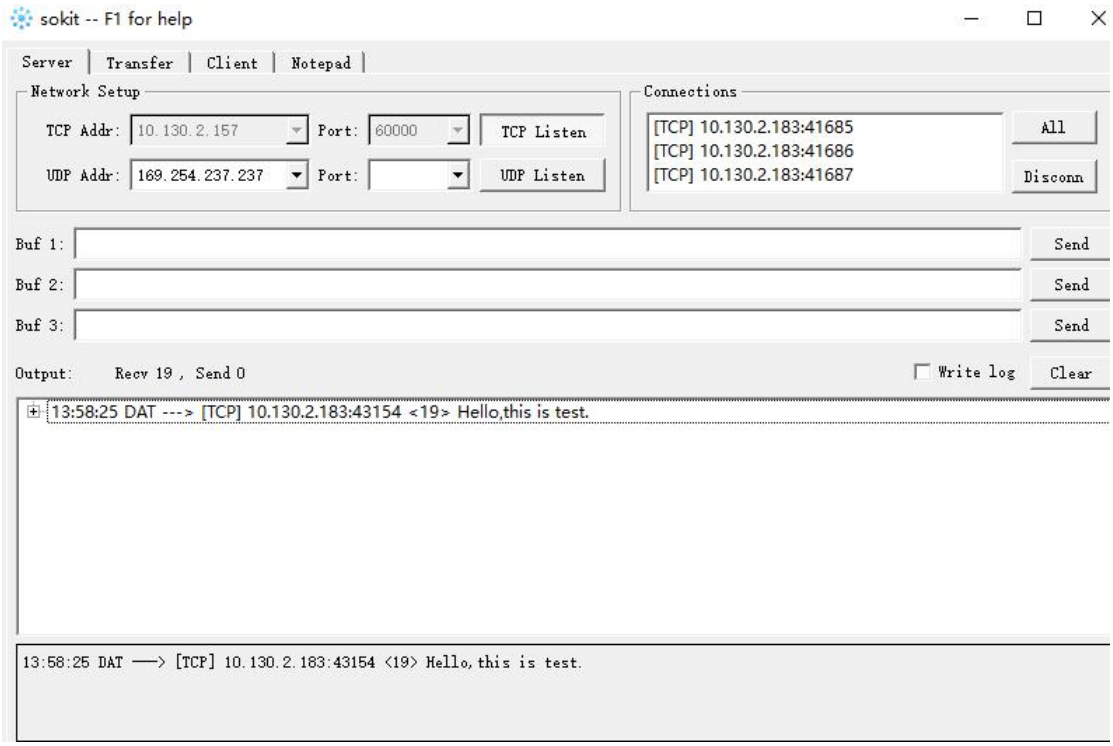
3. 打开 Sokit 创建一个 Server 端。



4. 输入测试指令，发送一个数据”Hello,this is test”。

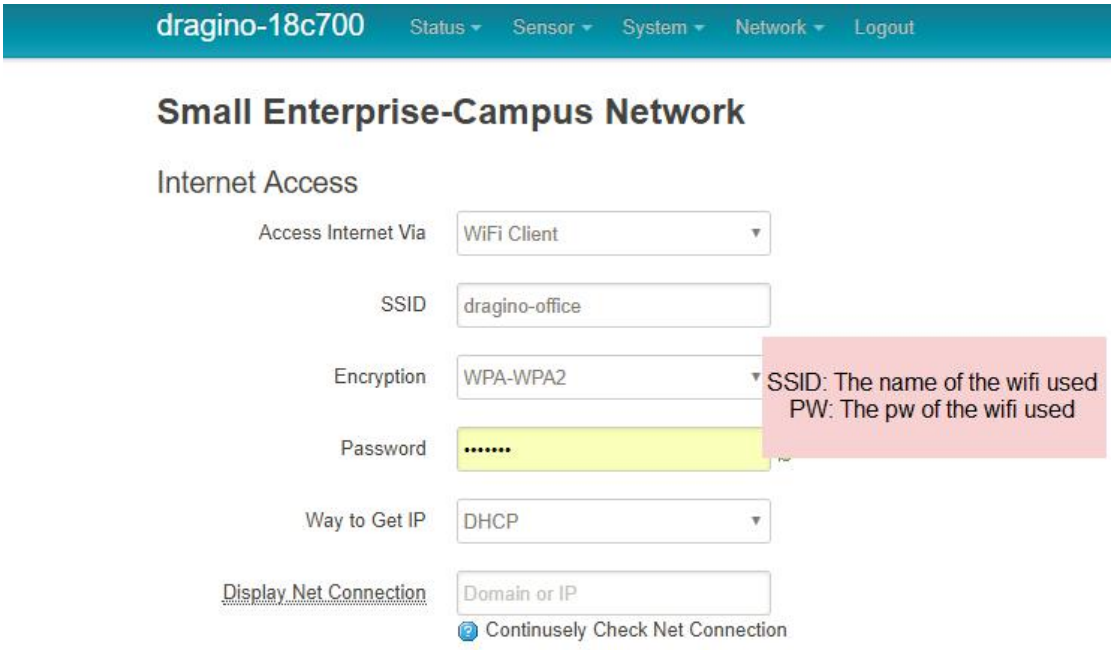


5. 查看结果

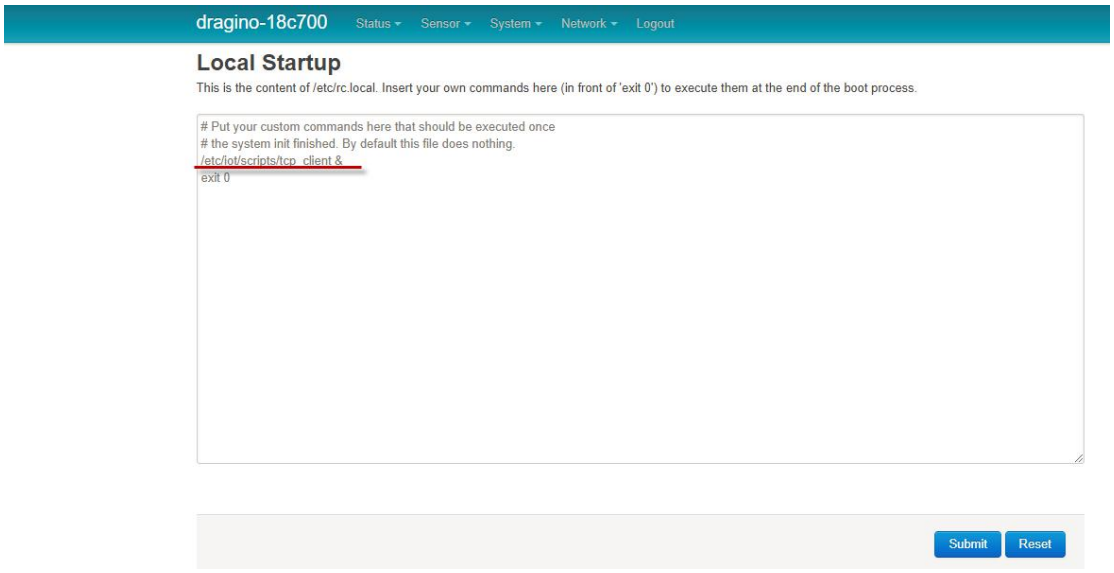


11.3 配置工作

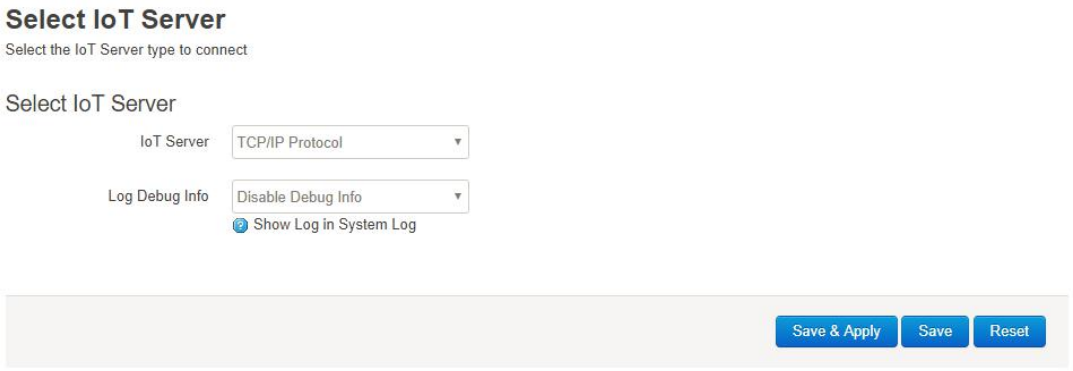
1. 首先配置入网。



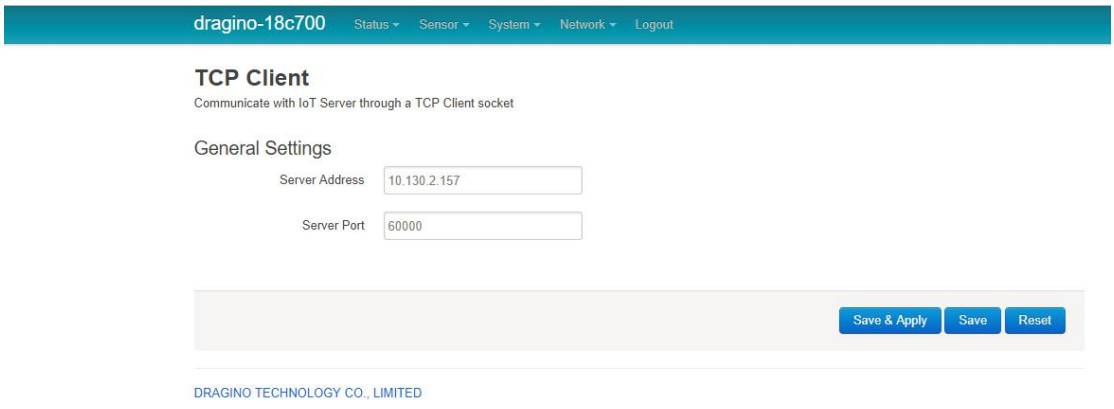
2. 点击 SyStem -> Startup，添加：/etc/iot/scripts/tcp_client &



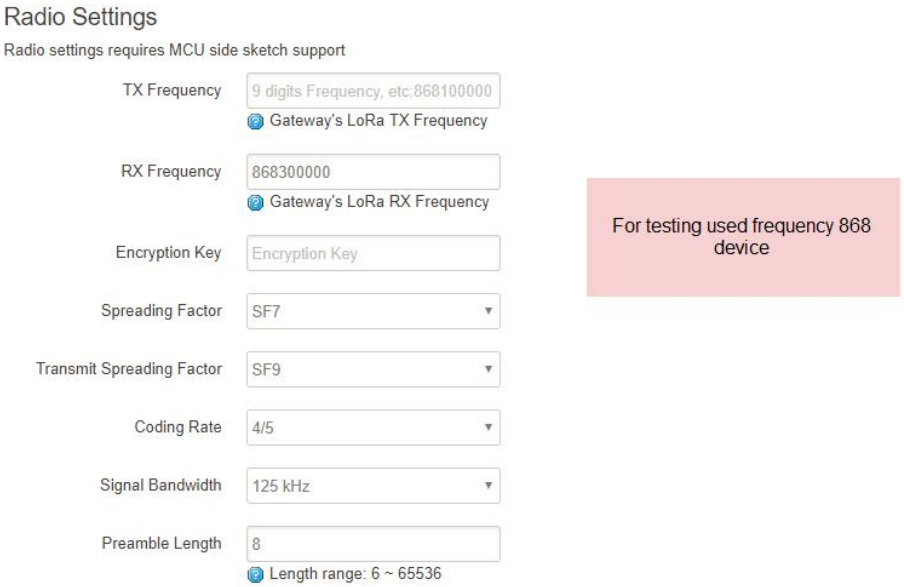
3. 启用 TCP 服务



4. 配置 TCP 的服务器端口号和 IP



5. 配置发射频率



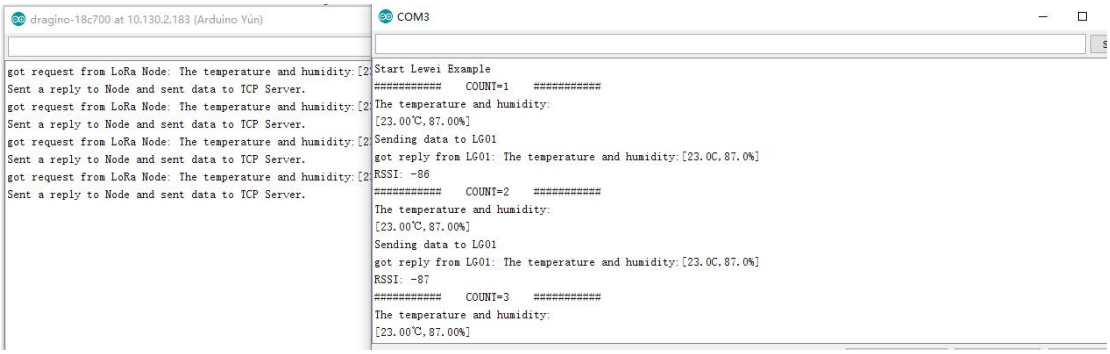
11.4 发送数据

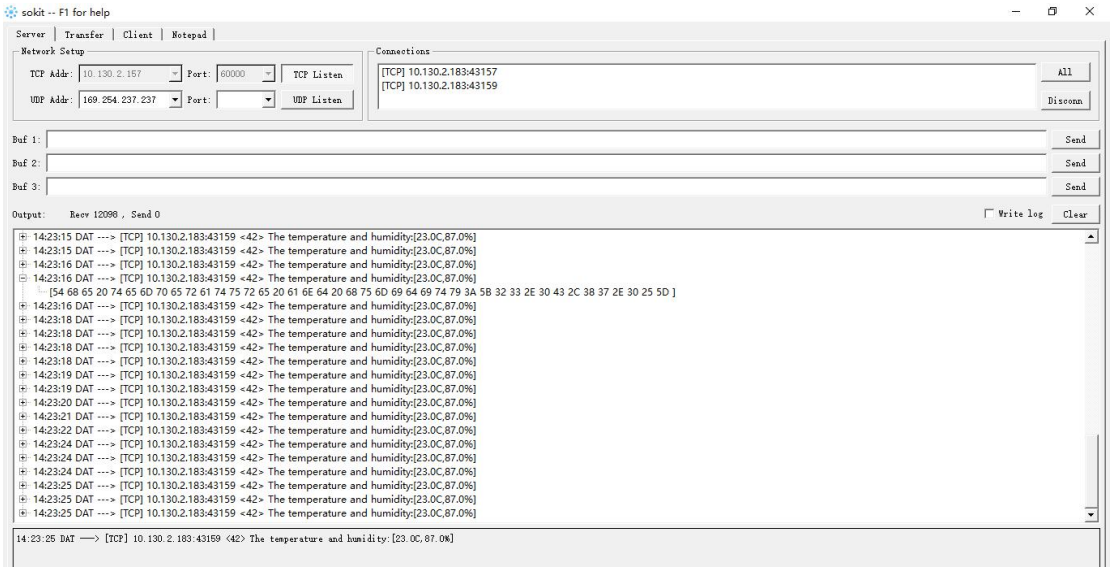
硬件设置如下：

√ LoRa 终端节点： LoRa ShieLd + UNO + DHT11 温湿度传感器。LoRa 终端节点不断从传感器中获取温度和湿度的数据，并通过 LoRa 定期发送出去。

√ LoRa 网关 LG01：收听 LoRa 无线信道时，若有新的 LoRa 数据包到达，解析并发送至 TCP 服务器。

Code: TCP Client, TCP Server。





更多示例及内容请参考：TCP/IP_Example.

12 进阶例子

12.1 连至 TTN LoRaWAN 服务器的例子

请仔细检查这个链接: **Connect to TTN**

12.2 多个节点的例子

这个示例展示了网关如何处理多个节点，可以多达数百个节点。这个例子可以从 **IDE --> File --> Examples --> Dragino --> LoRa --> Concurrent** 中找到。

它是如何运作的:

这个并发的客户端固件与并发的网关固件一起工作。在使用这个固件之前，请使用 write_client_id 固件在 EEPROM 中编写一个客户机 ID。客户机 ID 是 LoRa 网络中的每个客户机的唯一 ID。对网关的 write_gateway_id 不是必需的，如果不写，网关 id 将是 0XFF。

当客户端启动时，它将持续侦听来自 LoRa 网关的广播消息。

当网关固件启动时，它将广播一条消息来建立一个 LoRa 网络。如果客户端获得广播消息，客户端将向网关发送一个连接请求消息，当连接请求消息到达网关时，网关将发送带有客户端 id 的连接-ack 消息，并将该客户端添加到 LoRa 网络

如果客户端收到网关对客户端连接请求的 ACK 消息，客户端将进入接收模式，同时侦听来自网关的数据请求消息。在这种模式下，网关将会定期发送数据请求信息，如果客户端接收到为含有该客户端 ID 的数据请求消息，客户端将向网关发送一个数据消息。

在数据监听模式下的客户端，如果它在超时中没有从网关接收任何消息，客户端将返回到网络设置模式来监听网络组建消息

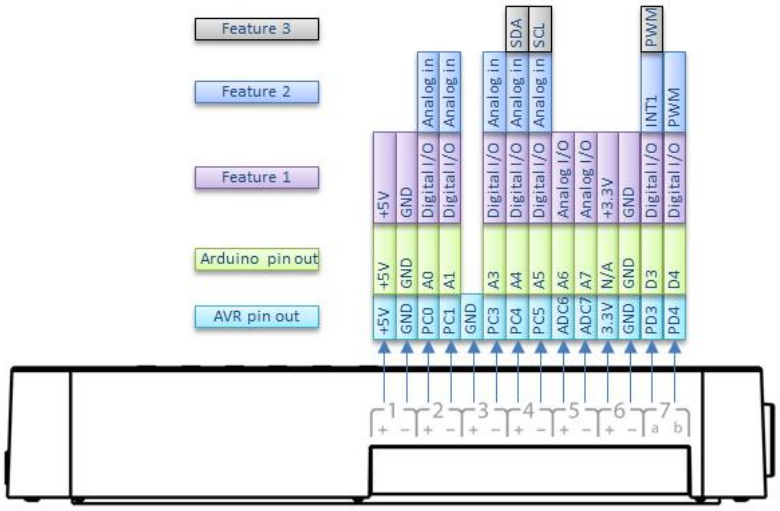
网关将定期刷新 LoRa 网络，以增加新的客户端或移除不可访问的客户端

这个示例使用了 LoRa 节点和网关之间的轮询方法，它将使 LoRa 数据包在空中的传输最小化，从而避免拥塞。它适用于一个非实时的 LoRa 工作。

效率: 在一个有 100 个节点和 1 个网关的房间里进行性能测试：
(a) 网关需要大约 1.5 分钟来建立这 100 个节点网络
(b) 网关需要大约 2 分钟的时间来对这 100 个节点进行轮询

12.3 如何使用 LG01-S 的传感器引脚?

LG01-S 有来自 ATMega328P MCU 的外部传感器引脚，它可以连接到外部传感器，下面是 LG01-S 的引脚定义:

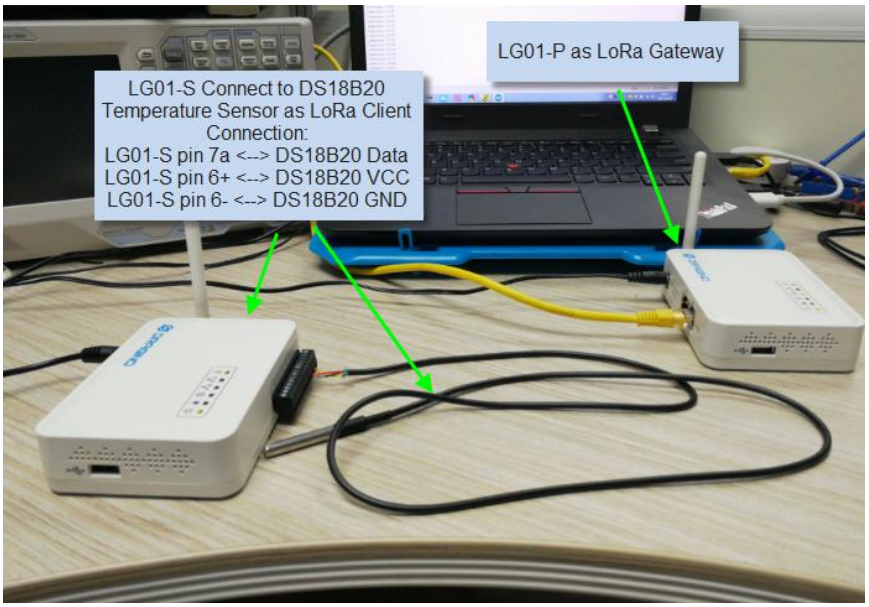


LG01-S Pinout

这些引脚的程序方法和 Arduino 一样，应该注意的是，引脚是 3.3v 的输入/输出。

下面是一个如何使用 DS18B20 温度传感器的例子：

硬件设置如下：

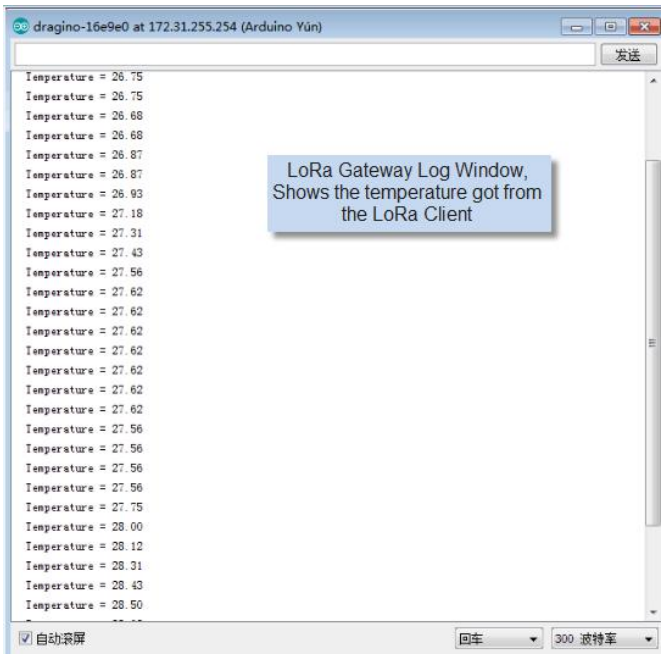


源代码在这个链接中:

✓ 传感器和 LoRa 客户端

✓ 网关侧代码

结果如屏幕截屏:


dragino-16e9e0 at 172.31.255.254 (Arduino Yún)
发送
Temperature = 26.75
Temperature = 26.75
Temperature = 26.68
Temperature = 26.68
Temperature = 26.87
Temperature = 26.87
Temperature = 26.93
Temperature = 27.18
Temperature = 27.31
Temperature = 27.43
Temperature = 27.56
Temperature = 27.62
Temperature = 27.62
Temperature = 27.62
Temperature = 27.62
Temperature = 27.62
Temperature = 27.62
Temperature = 27.62
Temperature = 27.56
Temperature = 27.56
Temperature = 27.56
Temperature = 27.56
Temperature = 27.75
Temperature = 28.00
Temperature = 28.12
Temperature = 28.31
Temperature = 28.43
Temperature = 28.50
LoRa Gateway Log Window, Shows the temperature got from the LoRa Client
自动滚屏
回车
300 波特率

12.4 更多例子

我们不断更新例子来支持更多的服务器和设备，更多的示例请参考链接: Dragino Examples Catalog

13 常见问题

13.1 为什么 LoRa 部分有 433/868/915 等不同频率版本?

不同的国家对使用 LoRa 的 ISM 频段有不同的规定。虽然 LoRa 芯片可以支持更广泛的频率，但我们在 LoRa 部分提供了不同频率版本的最佳调校。这就是我们提供不同版本的 LoRa 的原因。

13.2 LG01 LoRa 部分的频率范围是多少？

LoRa 部分使用的芯片是:

版本	LoRa IC	支持频率范围	最佳调校频率
433	Semtech SX1278	Band2(LF): 410 ~525Mhz Band3(LF): 137 ~175Mhz	433Mhz
868	Semtech SX1276	Band1(HF): 862 ~1020Mhz	868Mhz
915	Semtech SX1276	Band1(HF): 862 ~1020Mhz	915Mhz

用户可以在软件中设置 LoRa 的实际工作频率.

13.3 网关支持什么类型的 LoRa 设备?

LoRa 部分软件运行在 ATMega328p MCU 上。我们用 Radiohead Library 作为例子。如果其他的 LoRa 设备运行相同的 Radiohead 库，相同的频率和相同的加密，它们应该能够与这个网关通信。

用户还可以在 MCU 上运行其他 LoRa 协议，以支持他们想要的其他 LoRa 设备。
这边是一个支持 Microchip RN2483 的例子。

13.4 LG01 可以支持多少个节点？

最大的支持端节点取决于终端节点和网关之间的通信(频率)。在一个使用简单的 LoRa 示例的实验室测试中，如果终端节点试图每 5 分钟向网关发送数据，那么在网络有 20 个 30 个节点由于通道冲突，就会导致丢失数据
如果用户想要支持更多的节点，用户可以考虑使用轮询方法来确保每次只在频率上有一个 LoRa 信号传输。如果网关使用轮询方法从最终节点获取数据，它可以支持几百个节点或更多。示例可以看到：Polling example for LoRa

13.5 LG01 可以支持什么类型的服务器？

LG01 的 Linux 端是 OpenWrt，它是开放源码的，用户可以在它上面开发应用程序。基本上，

如果使用正确的 API，它可以支持大多数物联网服务器。我们有一些例子来说明如何通过典型的协议(MQTT，RESTful)来连接一些服务器，比如物联网、MQTT 或 RESTful. 参考此链接: IoT Server Examples.

13.6 我可以为 LG01 创建自己的固件吗?哪里可以找到 LG01 的源代码?

是的，用户可以为 LG01 创建自己的固件，或者添加自定义的应用程序。可以在此找到 LG01 源代码和编译指南: https://github.com/dragino/openwrt-cc-15.05

13.7 如何为这个设备获取更多的示例?

我们在 Arduino IDE 下的 Dragino 示例目录中不断发布 Arduino 示例。如果用户在早期安装了 dragino 版，我们就会发布新的示例。除了用户更新了 board 的配置文件外，新版本不会出现在 IDE 中。如果要更新，用户可以在 Arduino board manager 中移除 board 文件，然后重新安装。

13.8 OLG01 使用什么天线合适呢?

OLG01 在发货时候提供了一个小的弹簧天线用于测试，如下图。

用户可以用这个天线作为测试使用，对于商业使用，用户可以用高增益的，性能更好的天线替换，譬如说玻璃钢天线。 当使用外置天线时候请确保选择的天线的频率和购买的设备频率一致。

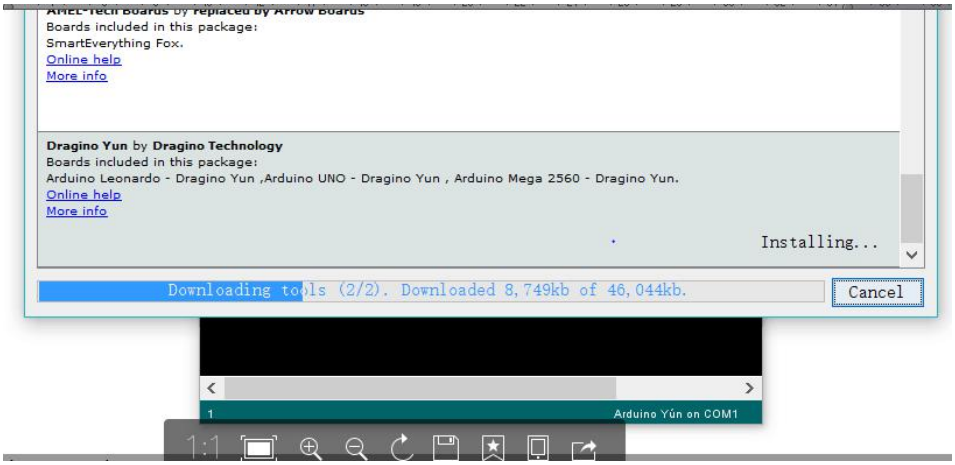
13.9 更加多的关于 LoRa 基本问题。

在我们的 WiKi，用户可以找到关于 LoRa 技术和我们产品配合 LoRa 使用时候的一些通用问题。 链接如下: http://wiki.dragino.com/index.php?title=LoRa_Questions

14 故障检修

14.1 我无法在 Arduino IDE 下载 Dragino 配置文件

如果 IDE 非常缓慢地下载了 board manager 里的 Dragino 配置文件，并在某个地方卡住了。正如下面所示，这是因为您的网络与 Arduino IDE 的一些软件包之间的连接速度很慢。



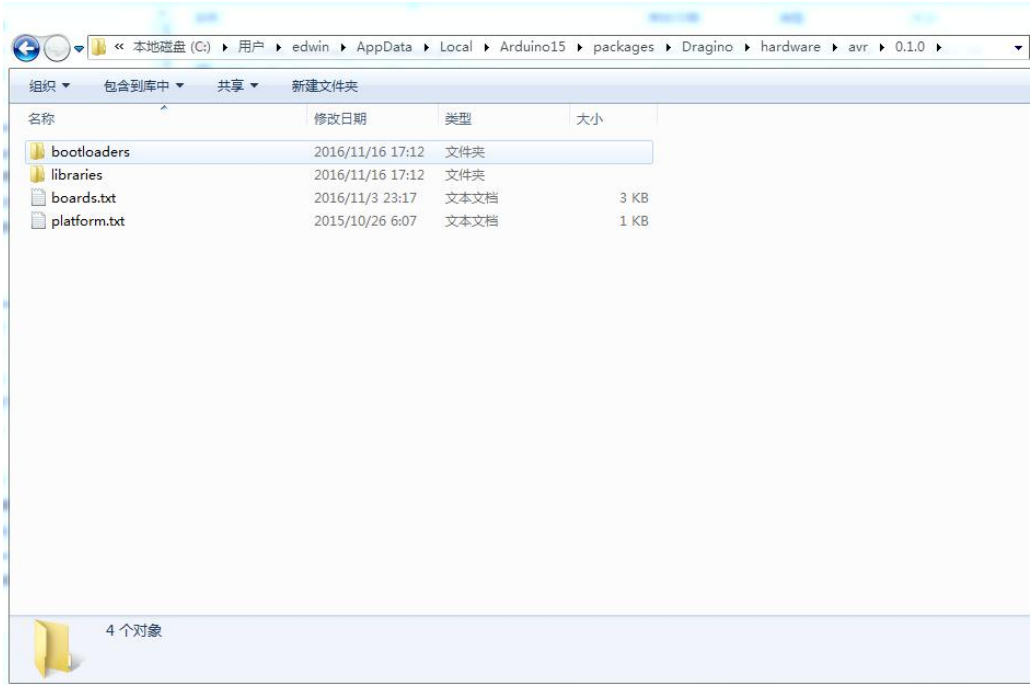
为了解决这个问题，用户可以手动添加 Dragino 板子文件。下面是步骤:

1/ 从 https://github.com/dragino/Arduino-Profile-Examples 中下载配置文件

2/ 解压并将内容放到该目录下：**C:\Users\xxx\AppData\Local\Arduino15\packages\Dragino\hardware\avr\0.1.0**

注意:不同的操作系统可能有不同的目录，如果您的系统没有 Dragino\hardware\avr\0.1.0，请在您的 Arduino 目录中创建它。

最终的目录内容应该如下所示。



14.2 MCU 和 Linux 模块之间的 Bridge 不工作

一些可能性:

1/ 您已经在 MCU 固件中使用了 **Serial 类**，比如 Serial.begin(9600)，在 ATMega328p 中的 Bridge 库使用相同的 Serial 接口。如果你在固件中有 Serial 代码。他们会发生冲突，而 Bridge 也不起作用

2/当您编译其他固件时，IDE 会在 Serial 设置中出现混乱。在这种情况下，您可以关闭 IDE 并再次打开它

14.3 Arduino IDE 没有检测到 LG01

如果这个问题发生，请检查以下几点:

- ✓ Arduino IDE 版本是 1.5.4 或更高版本
- ✓ 您的个人电脑和 LG01 都在同一个网络中
- ✓ 尝试通过 Web 或 SSH 访问 LG01，然后再次检查 IDE
- ✓ 如果上面仍然没有工作，那么 SSH 登录到 LG01 并运行: **/etc/init.d/avahi-daemon restart** 重启该服务，以便 IDE 能够检测到 LG01。

14.4 安装新包时，我得到了内核错误，如何修复?

在某些情况下，当安装包时，它会产生如下的内核错误:

```
root@dragino-16c538:~# opkg install kmod-dragino2-si3217x_3.10.49+0.2-1_ar71xx.ipk
Installing kmod-dragino2-si3217x (3.10.49+0.2-1) to root...
Collected errors:
* satisfy_dependencies_for: Cannot satisfy the following dependencies for kmod-dragino2-si3217x:
*  kernel (= 3.10.49-1-4917516478a753314254643facdf360a) *
 * opkg_install_cmd: Cannot install package kmod-dragino2-si3217x.
```

在这种情况下，用户可以使用–force-depends 选项来安装这样的包

```
opkg install kmod-dragino2-si3217x_3.10.49+0.2-1_ar71xx.ipk --force-depends
```

14.5 如果 Linux 固件崩溃，如何恢复 LG01

LG01 向用户开放了 Linux 系统的完全控制，在某些引导文件中出现了不适当的修改后，该设备可能会出现死机和无法引导的情况

在这种情况下，用户可以通过 Web 故障安全模式上传新固件来恢复整个 Linux 系统。

过程如下:

- 使用网线将 PC 直接连接到 LG01 的网口
- 将 PC 设置为 ip 192.168.255.x，子网掩码 255.255.255.0
- 在设备上按住复位按钮，然后接上电源
- 该设备的所有 LED 灯都会闪烁，在所有灯闪烁四次之后松开复位按钮。
- 所有的 LED 都会很快闪烁一次，这意味着设备检测到网络连接并进入到网络故障安全模式。在设备进入此模式后，您的 PC 应该能够 ping 192.168.255.1。
- 在网页浏览器上打开 192.168.255.1
- 选择一个 squashfs-sysupgrade 类型的固件并更新固件

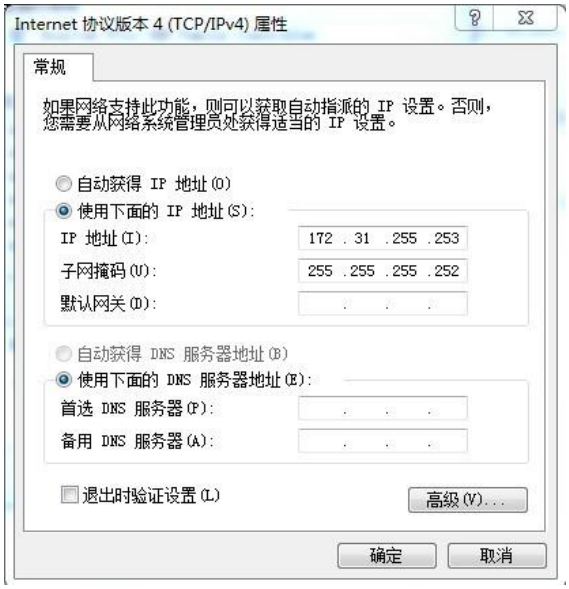
14.6 我为 WiFi 访问配置了 LG01 并失去了它的 IP，在该怎么做?

LG01 在它的 LAN 端口上有一个备用 IP。这个 IP 始终是启用的，因此用户可以使用备用 IP 访问 LG01，无论 WiFi IP 是什么。备用 IP 对于连接和调试设备非常有用。

(备注: 备用 IP 可以在 LAN and DHCP 配置页面中禁用)

通过备用 IP 连接的步骤:

1. 连接电脑的以太网端口至 LG01 的 LAN 端口
2. 配置 PC 的以太网端口有 IP: 172.32.255.253 和子网掩码:255.255.255.252
   如下图:



3. 在 PC 上，使用 172.31.255.254 通过 Web 或 SSH 控制台访问 LG01 .

15 订购须知

通用版:

- **LG01P-433**: LoRa Gateway best tune to 433 MHz.
- **LG01P-868**: LoRa Gateway best tuned to 868 MHz.
- **LG01P-915**: LoRa Gateway best tuned to 915 MHz

接线端子版:

- **LG01S-433**: LoRa Gateway best tune to 433 MHz.
- **LG01S-868**: LoRa Gateway best tuned to 868 MHz.
- **LG01S-915**: LoRa Gateway best tuned to 915 MHz.

户外版:

- **OLG01-433**: LoRa Gateway best tune to 433 MHz.
- **OLG01-868**: LoRa Gateway best tuned to 868 MHz.
- **OLG01-915**: LoRa Gateway best tuned to 915 MHz.

16 包装信息

LG01 套装包含:

- ✓ LG01P 或 LG01S LoRa 网关 x 1
- ✓ 用于 LoRa RF 部分的棒形天线。天线的频率是 433 或 868 或 915Mhz，取决于购买的设备频率版本。
- ✓ 电源适配器: EU/AU/US 类型的电源适配器取决于使用的国家
- ✓ 环保纸箱包装

外形尺寸及重量:

- ✓ 设备尺寸: 12 x 8.5 x 3 cm
- ✓ 设备重量: 150g
- ✓ 包装尺寸: 1.5 x 10 x 5 cm
- ✓ 重量 / pcs : 360g
- ✓ 包装规格: 45 x 31 x 34 cm. 36pcs 每箱
- ✓ 重量 / 箱 : 12.5 kg

17 技术支持

- 试着看看你的问题是否已经在 wiki 中得到了答案。
- 星期一至星期五，从 09:00 to 18:00 GMT+8.提供支持。我们将尽快回复你的问题。
- 提供尽可能多的信息关于你的询问 (产品模型，准确地描述您的问题和步骤用于重新问题等) 并发送邮件到

support@dragino.com

18 参考信息

- 更多的关于 LG01 使用例子

- LG01 LoRa 物联网网关 软件源代码
  https://github.com/dragino/openwrt-cc-15.05

- OpenWrt 官方网站
  http://www.openwrt.org/

- Arduino 官方网站
  https://www.arduino.cc

- Arduino bridge 库使用例子和详细说明:
  https://www.arduino.cc/en/Tutorial/Bridge

- 硬件源代码:
  LG01 硬件包含 2 部分:
  - ✓ MS14N Linux 主板:
    https://github.com/dragino/motherboard-hardware/tree/master/ms14n
  - ✓ LoRa 子板 LoRa G:
    https://github.com/dragino/Lora/tree/master/LoRa%20G/v1.3