# LG02/OLG02 LoRa Gateway User Manual

## Document Version: 1.5.2

## Firmware Version: LG02_LG08—v5.2

| Version | Description | Date |
|---|---|---|
| 1.0 | Release | 2018-Sep-11 |
| 1.1 | Enrich content for ABP | 2018-Oct-07 |
| 1.2 | Add TCP_IP Client Instruction, Add Radiohead support<br>(firmware ver > LG02_LG08--build-v5.1.1541419802-20181105-2011) | 2018-Nov-5 |
| 1.3 | Add notice for recover mode, add packet relay mode example.<br>Add Example for LoRa RAW.<br>Add Manual for MQTT mode<br>(firmware ver > LG02_LG08--build-v5.1.1545026772-20181217-1407) | 2018-Dec-27 |
| 1.4 | Add Customized Script Feature<br>(firmware ver >LG02_LG08--build-v5.1.1547896817-20190119-1921) | 2019-Jan-19 |
| 1.5 | -Add Downlink support and example.<br>(firmware ver >LG02_LG08--build-v5.1.1548820215-20190130-1151)<br>-Correct typo for the UNO code of example for lg02_single_rx_tx | 2019-Jan-30 |
| 1.5.1 | Add How to control LEDs, add hardware source, improve limitation description. | 2019-May-13 |
| 1.5.2 | Relay mode limitation, MQTT instruction | 2019-Jun-19 |
| 1.5.3 | Add photo for OLG02 SIM card installation | 2019-Nov-01 |

**1. Introduction ..... 6**

*1.1 What is LG02 & OLG02 ..... 6*

*1.2 Specifications ..... 7*

*1.3 Features ..... 9*

*1.4 System Structure ..... 9*

*1.5 Applications ..... 10*

*1.6 Hardware Variants ..... 11*

*1.7 Install SIM card in 4G module ..... 12*

*1.8 Firmware Change log ..... 12*

**2. Access LG02 ..... 13**

**3. Typical Network Setup ..... 14**

*3.1 Overview ..... 14*

*3.2 Use WAN port to access Internet ..... 14*

*3.3 Access Internet as a WiFi Client. ..... 15*

*3.4 Use built-in 4G modem for internet access ..... 17*

*3.5 Check Internet connection ..... 19*

**4. Example 1: Configure as a LoRaWAN gateway – LoRaWAN mode ..... 20**

*4.1 Create a gateway in TTN Server ..... 20*

*4.2 Configure LG02 Gateway ..... 22*

4.2.1 Configure to connect to LoRaWAN server ..... 22

4.2.2 Configure LG02's RX frequency ..... 23

*4.3 Create LoRa End Node ..... 24*

4.3.1 About Limited support for LoRaWAN ..... 24

4.3.2 Preparation ..... 25

4.3.3 Test with OTAA LoRa end node (LoRa Shield + UNO) ..... 26

4.3.4 Test with ABP LoRa end node (LoRa Shield + UNO) ..... 30

**5. Example 2: Control LoRa radio directly as general LoRa transceiver ..... 34**

*5.1 User LoRa Radio via pkt_fwd ..... 34*

5.1.1 Use pkt_fwd to receive ..... 34

5.1.2 Use pkt_fwd to transmit ..... 34

*5.2 Use LoRa radio device directly ..... 36*

**6. Example 3: MQTT Transfer Mode ..... 39**

**7. Example 4: TCP IP Client Mode ..... 40**

**8. Example 5: Write a customized script ..... 42**

**9. Example 6: LoRaWAN Packet Relay mode ..... 44**

*9.1 How it works ..... 44*

*9.2 Limitation ..... 46*

**10. Linux System ..... 47**

*10.1 SSH Access for Linux console ..... 47*

*10.2 Edit and Transfer files ..... 48*

*10.3 File System ..... 48*

*10.4 Package maintain system ..... 50*

**11. Upgrade Linux Firmware ..... 51**

*11.1 Upgrade via Web UI ..... 51*

*11.2 Upgrade via Linux console ..... 51*

**12. FAQ ..... 52**

*12.1 Why there is 433/868/915 version LoRa part? ..... 52*

*12.2 What is the frequency range of LG02 LoRa part? ..... 52*

*12.3 What does "Limited support on LoRaWAN"? ..... 52*

*12.4 Can I develop my own application for LG02? ..... 54*

*12.5 Can I make my own firmware for LG02? Where can I find the source code of LG02? ..... 54*

*12.6 Can I change the LoRa behavior of LG02? ..... 54*

*12.7 On OTAA mode, if I use the other frequency, how should I modify in the library? ..... 54*

*12.8 How can I reset the device to factory default? ..... 55*

*12.9 Can I control the LEDs? ..... 56*

*12.10 More FAQs about general LoRa questions ..... 56*

**13. Trouble Shooting ..... 57**

*13.1 I get kernel error when install new package, how to fix? ..... 57*

*13.2 How to recover the LG02 if firmware crash ..... 58*

*13.3 I configured LG02 for WiFi access and lost its IP. What to do now? ..... 59*

14. Order Info ..... 60

15. Packing Info ..... 60

16. Support ..... 60

17. Reference ..... 62

# 1. Introduction

## 1.1 What is LG02 & OLG02

LG02 & OLG02 are an open source dual channels LoRa Gateway. It lets you bridge LoRa wireless network to an IP network via WiFi, Ethernet, 3G or 4G cellular. The LoRa wireless allows users to send data and reach extremely long ranges at low data-rates. It provides ultra-long range spread spectrum communication and high interference immunity.

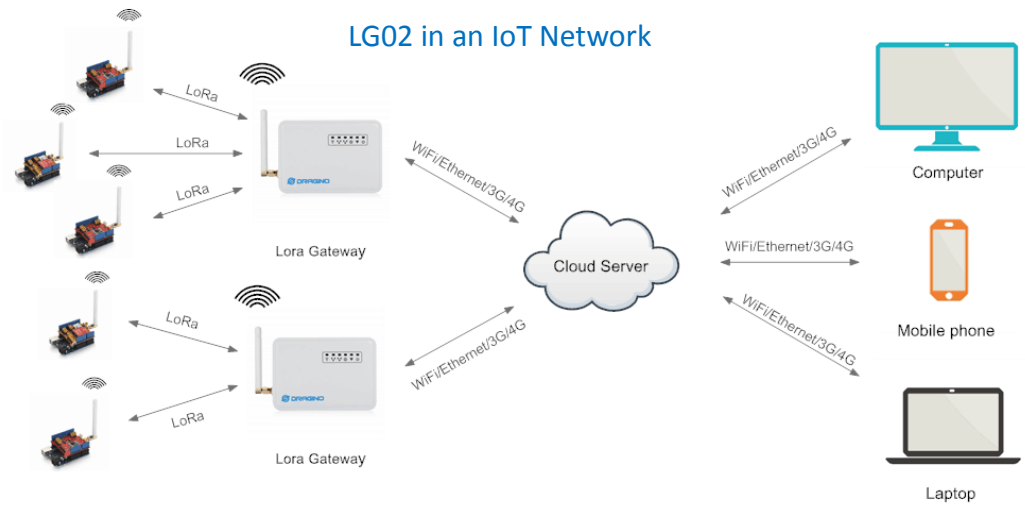
LG02 & OLG02 have rich internet connection method such as WiFi interface, Ethernet port and USB host port. These Interfaces provide flexible methods for users to connect their sensor networks to Internet.

LG02 & OLG02 can support the LoRaWAN protocol in single frequency and customized LoRa transition protocol. The design of LG02 is use the Linux to directly control two sx1276/sx1278 LoRa modules which lets the LoRa can works in full duplex LoRa mode and increase the communication efficiency.

LG02 can be used to provide a low cost IoT wireless solution to support 50~300 sensor nodes.

Except LoRaWAN mode, LG02 can support multiply working mode such as: LoRa repeater mode, MQTT mode, TCP/IP Client mode, TCP/IP Server mode to fit different requirement for IoT connection. Click this link for more info about the modes.

LG02 & OLG02 provide a low cost for your IoT network connection. Compare to the cost with normal SX1301 LoRaWAN solution. LG02 & OLG02 is only of its 1/4 or less cost. This makes the LG02 very suitable to set up small scale LoRa network or use it to extend the coverage of current LoRaWAN network.



## 1.2 Specifications

### Hardware System:

Linux Part:

- 400Mhz ar9331 processor
- 64MB RAM
- 16MB Flash

### Interface:

- 10M/100M RJ45 Ports x 2
- WiFi : 802.11 b/g/n
- LoRa Wireless
- Power Input: 12V DC
- USB 2.0 host connector x 1
- USB 2.0 host internal interface x 1
- 2 x LoRa Interfaces

### WiFi Spec:

- IEEE 802.11 b/g/n
- Frenquency Band: 2.4 ~ 2.462GHz
- Tx power:
  - ✓ 11n tx power : mcs7/15: 11db  mcs0 : 17db
  - ✓ 11b tx power: 18db
  - ✓ 11g 54M tx power: 12db
  - ✓ 11g 6M tx power: 18db
- Wifi Sensitivity
  - ✓ 11g 54M : -71dbm
  - ✓ 11n 20M : -67dbm

### LoRa Spec:

- Frequency Range:
  - ✓ Band 1 (HF): 862 ~ 1020 Mhz
  - ✓ Band 2 (LF): 410 ~ 528 Mhz
- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.

- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

**Cellular 4G LTE (optional):**

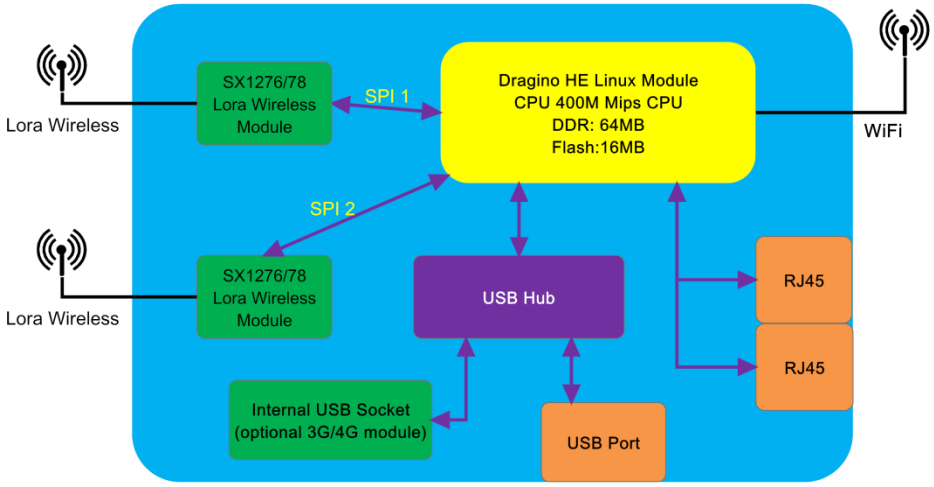
- Quectel **EC25 LTE module**
- Micro SIM Slot
- Internal 4G Antenna + External 4G Sticker Antenna.
- Up to 150Mbps downlink and 50Mbps uplink data rates
- Worldwide LTE,UMTS/HSPA+ and GSM/GPRS/EDGE coverage
- MIMO technology meets demands for data rate and link reliability in modem wireless communication systems

## 1.3 Features

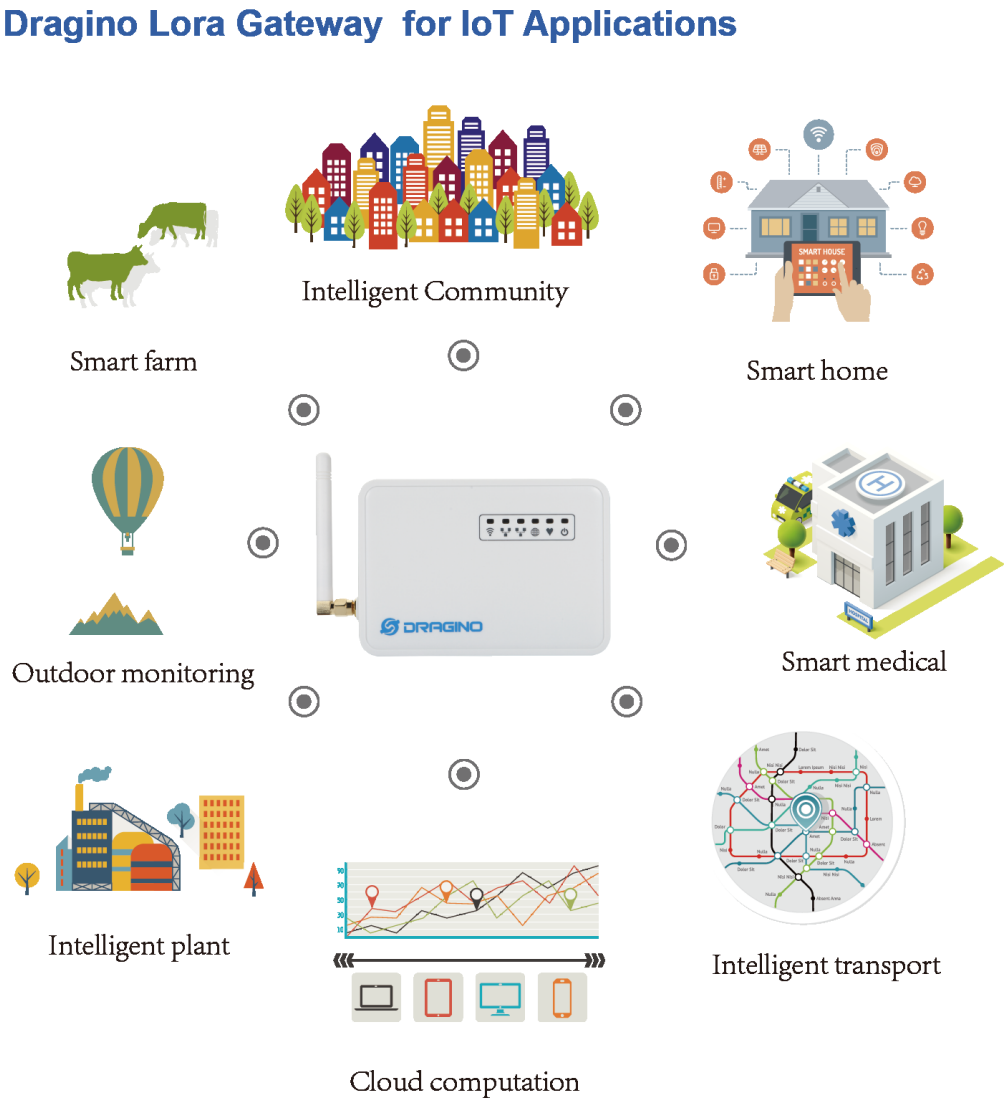
- ✓ Open Source OpenWrt LEDE system
- ✓ Low power consumption
- ✓ Firmware upgrade via Web
- ✓ Software upgradable via network
- ✓ Flexible protocol to connect to IoT servers
- ✓ Auto-Provisioning
- ✓ Built-in web server
- ✓ Managed by Web GUI, SSH via LAN or WiFi
- ✓ Internet connection via LAN, WiFi, 3G or 4G
- ✓ Failsafe design provides robustly system
- ✓ 2 x SX1276/SX1278 LoRa modules
- ✓ Full – duplex LoRa transceiver
- ✓ Two receive channels, and one transmit channel
- ✓ Limited support in LoRaWAN/ Support Private LoRa protocol
- ✓ Support upto 300 nodes
- ✓ LoRa band available at 433/868/915/920 Mhz
- ✓ Max range in LoRa: 5~10 km. Density Area:>500m

## 1.4 System Structure

LG02 System Overview:



### 1.5 Applications



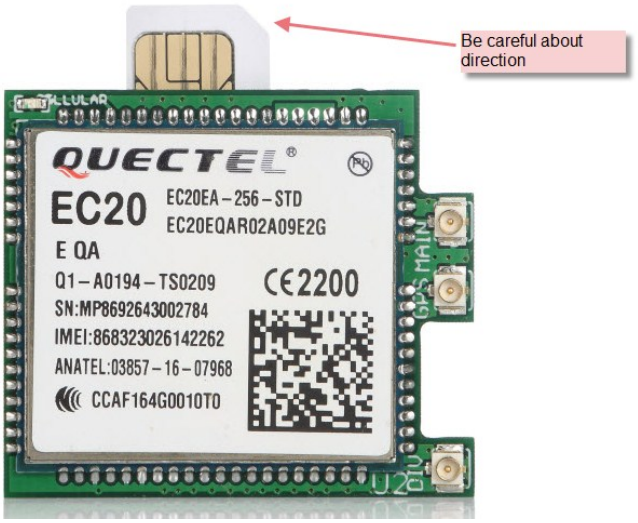
### 1.6 Hardware Variants

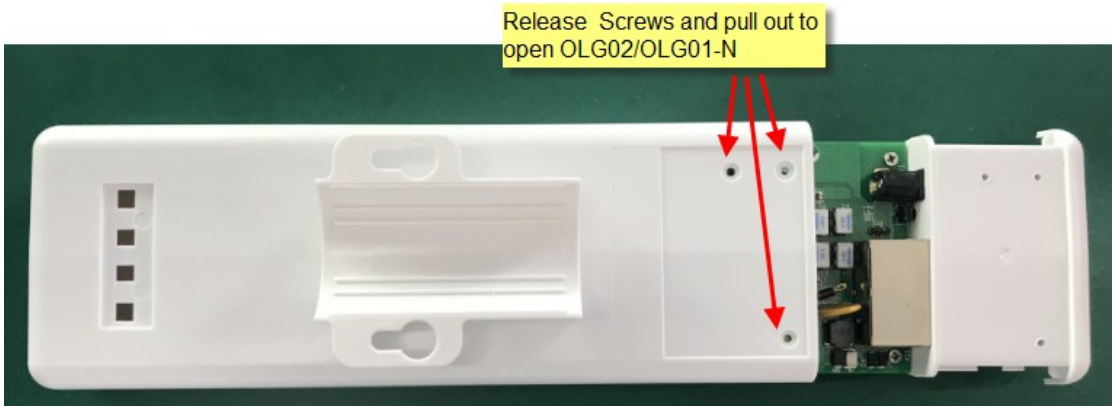
The LG02 and OLG02 use the same firmware and have the same feature in the software side. In this document, we will use LG02 as the model number to explain the feature.

| Model | Photo | Description |
|---|---|---|
| **LG02** | | Indoor version for dual channel LoRa Gateway, without 4G module |
| **LG02** | | Indoor version for dual channel LoRa Gateway, include 4G module. |
| **OLG02** | | Outdoor version for dual channel LoRa Gateway |

### 1.7 Install SIM card in 4G module

For LG02 & OLG02 4G version, the 4G module is inside the box, please open the box and use below direction to install the SIM card




### 1.8 Firmware Change log

Please see this link for firmware change log:

http://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LG02-OLG02/Firmware/&file=ChangeLog

## 2. Access LG02

Access and configure LG02

The LG02 is configured as a WiFi AP by factory default. User can access and configure the LG02 after connect to its WiFi network.

At the first boot of LG02, it will auto generate an unsecure WiFi network call dragino-xxxxxx

User can use the laptop to connect to this WiFi network. The laptop will get an IP address 10.130.1.xxx and the LG01 has the default IP 10.130.1.1

Open a browser in the laptop and type

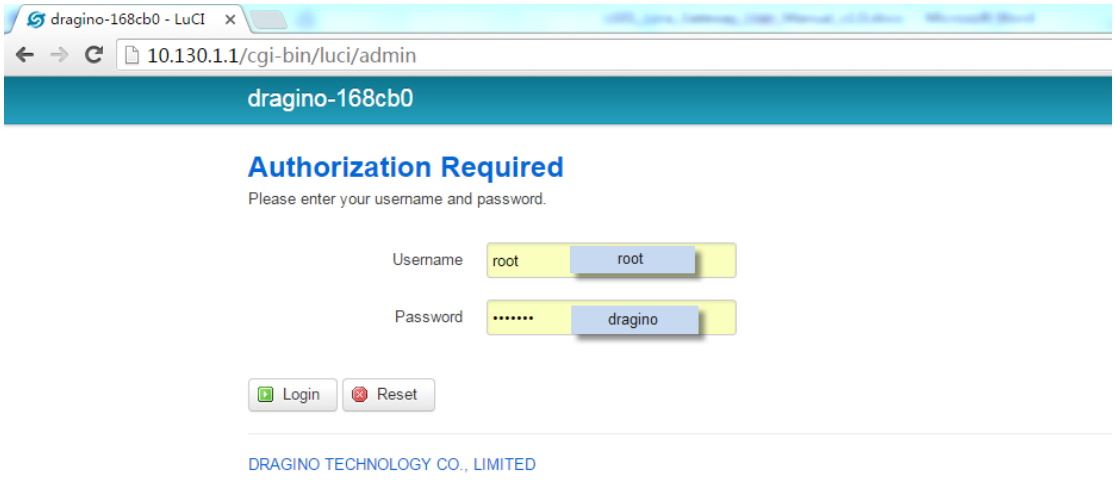
http://10.130.1.1/cgi-bin/luci/admin

User will see the login interface of LG02.

The account for Web Login is:

User Name: root

Password: dragino




Notice: In case the WiFi network is disabled, user can connect the PC to LG02's LAN port, the PC will get DHCP from LG02, and be able to access it.

# 3. Typical Network Setup

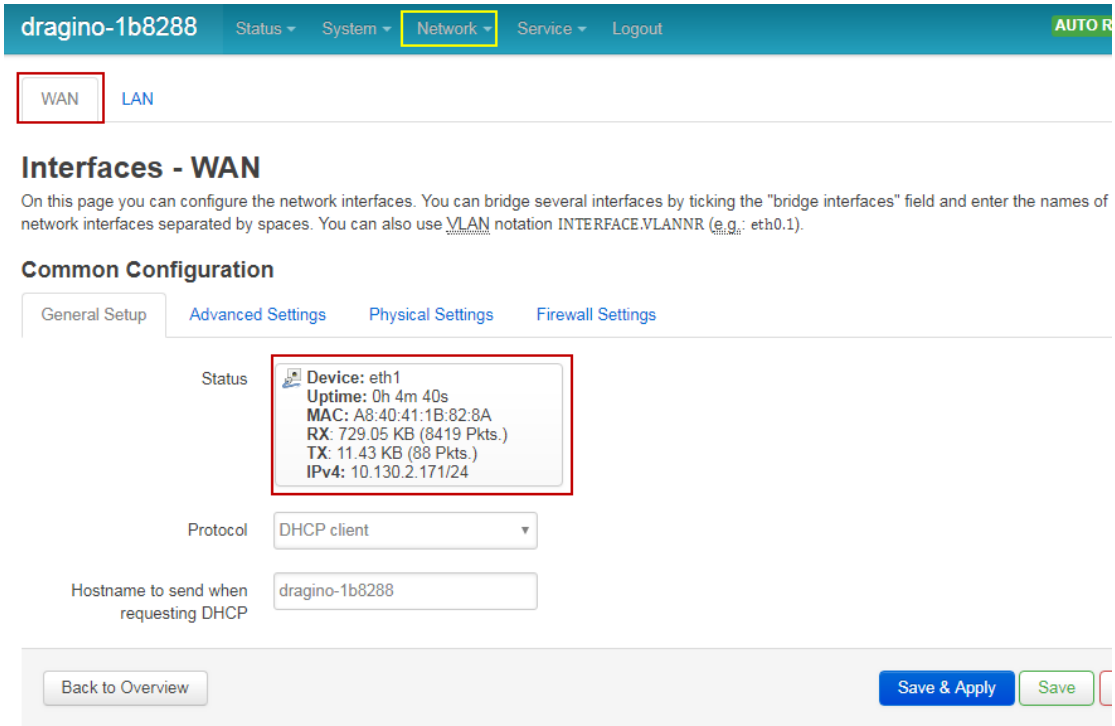
## 3.1 Overview

LG02 supports flexible network set up for different environment. This section describes the typical network topology can be set in LG02. The typical network set up includes:

- ✓ **WAN Port Internet Mode**
- ✓ **WiFi Client Mode**
- ✓ **WiFi AP Mode**
- ✓ **USB Dial Up Mode**

## 3.2 Use WAN port to access Internet

By default, the LG02 set to use WAN port as network connection. When connect LG02's WAN port to router, LG02 will get IP from router and have internet access. The network status can be checked as below:

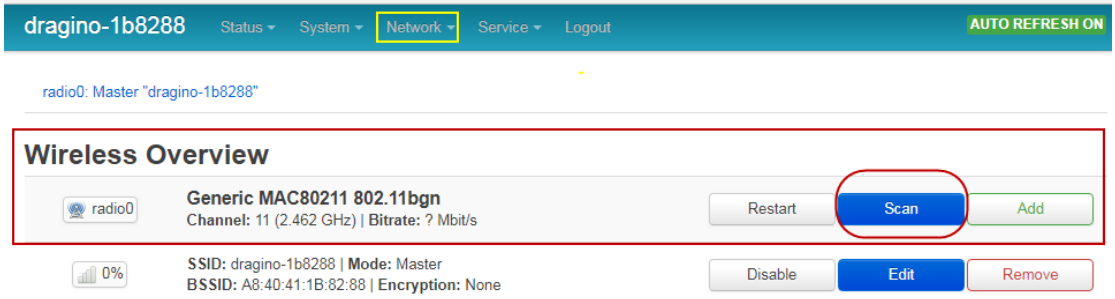


### 3.3 Access Internet as a WiFi Client.

In the WiFi Client Mode, Dragino acts as a WiFi client and gets DHCP from uplink router via WiFi. The step to set is as below:

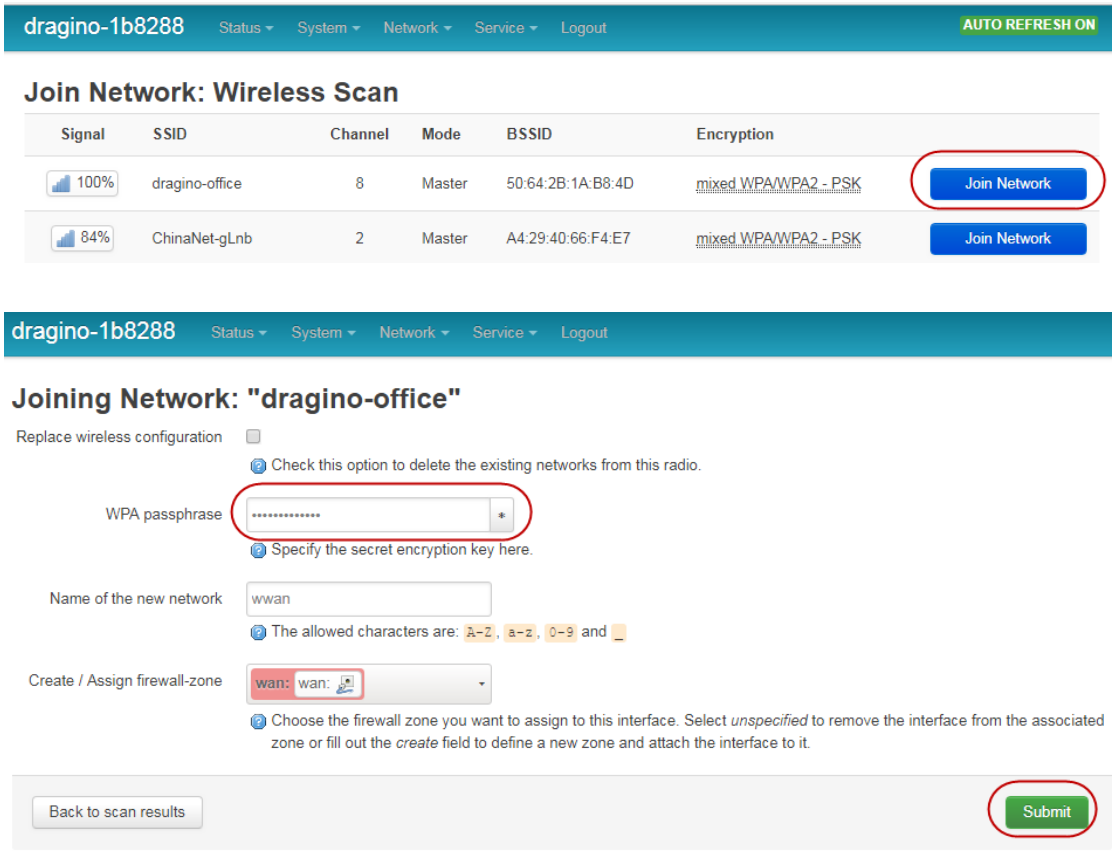
**Step1:**

In network -> Wireless, select Radio0 interface and scan.



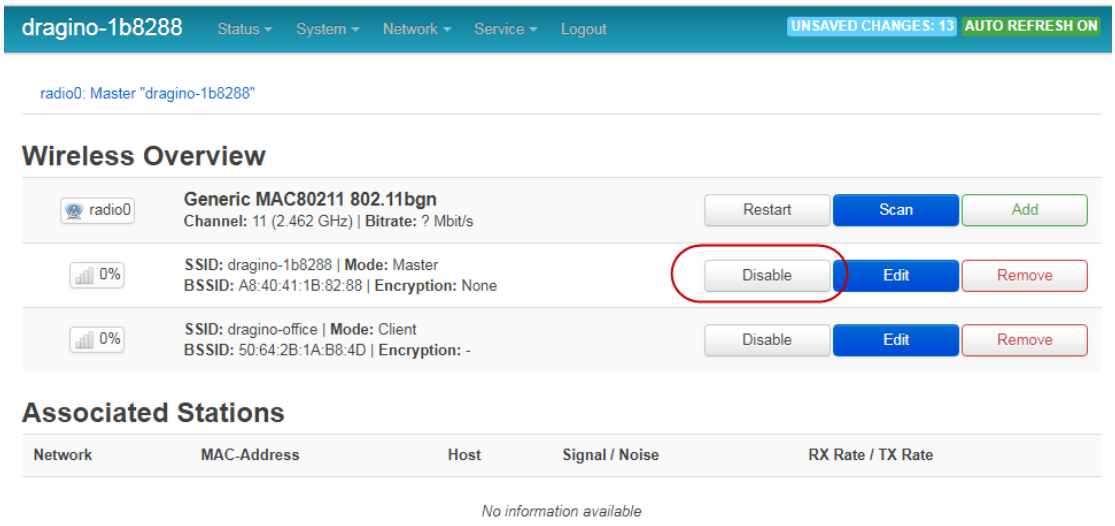

**Step2:**

Select the wireless AP and join:

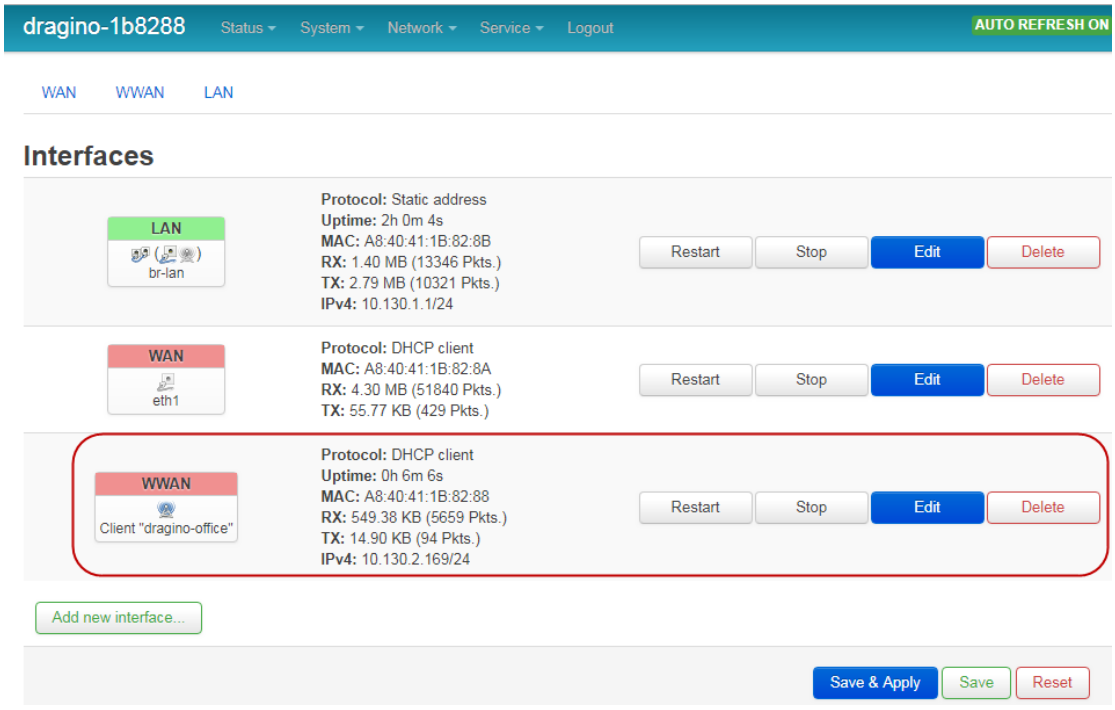



**Step3:**

In network->wireless page, disable WiFi AP network. Notice: After doing that, you will lose connection if your computer connect to the LG02 via LG02's wifi network.



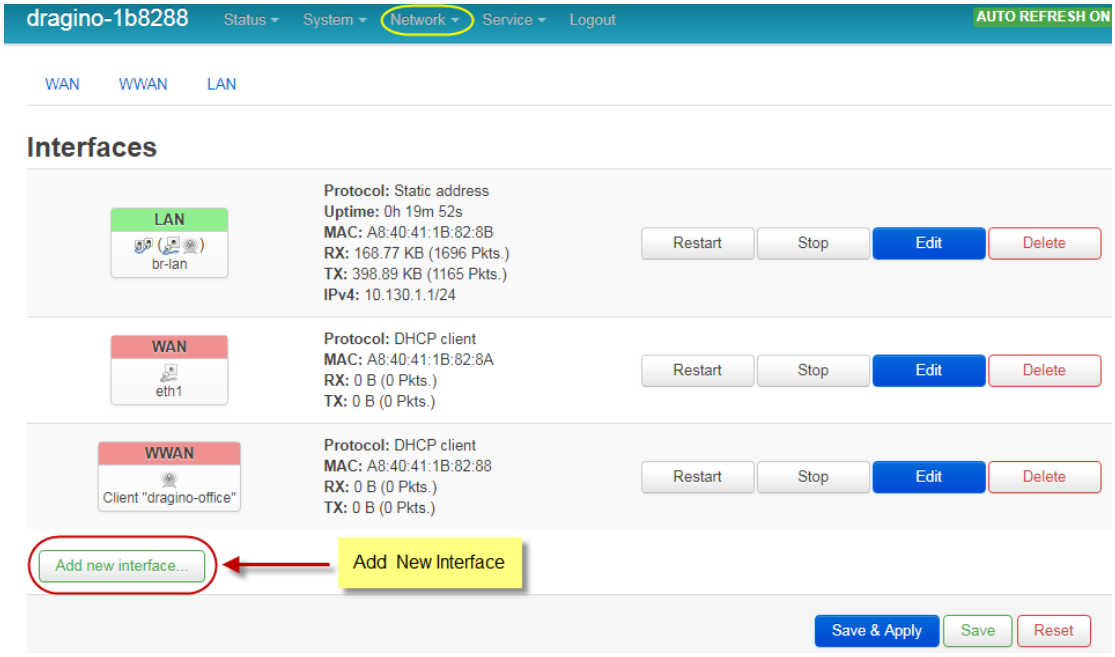

After successful associate, the WiFi network interface can be seen in the same page:

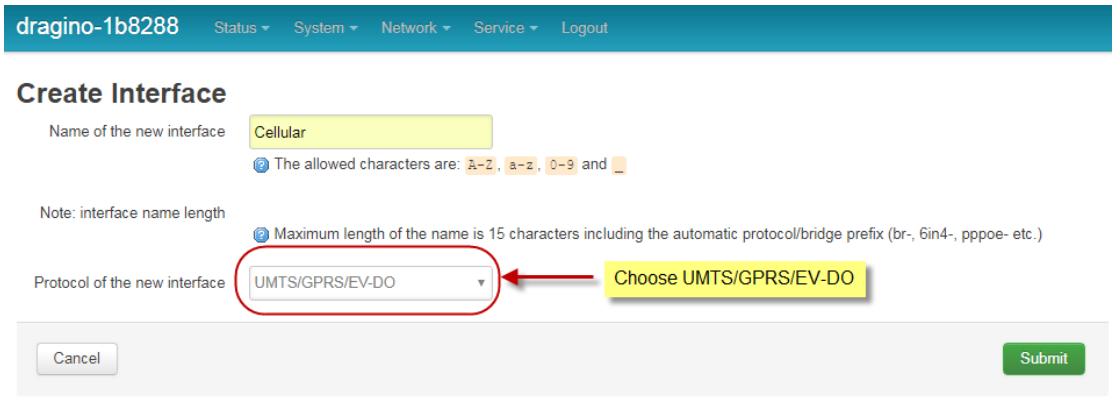


## 3.4 Use built-in 4G modem for internet access

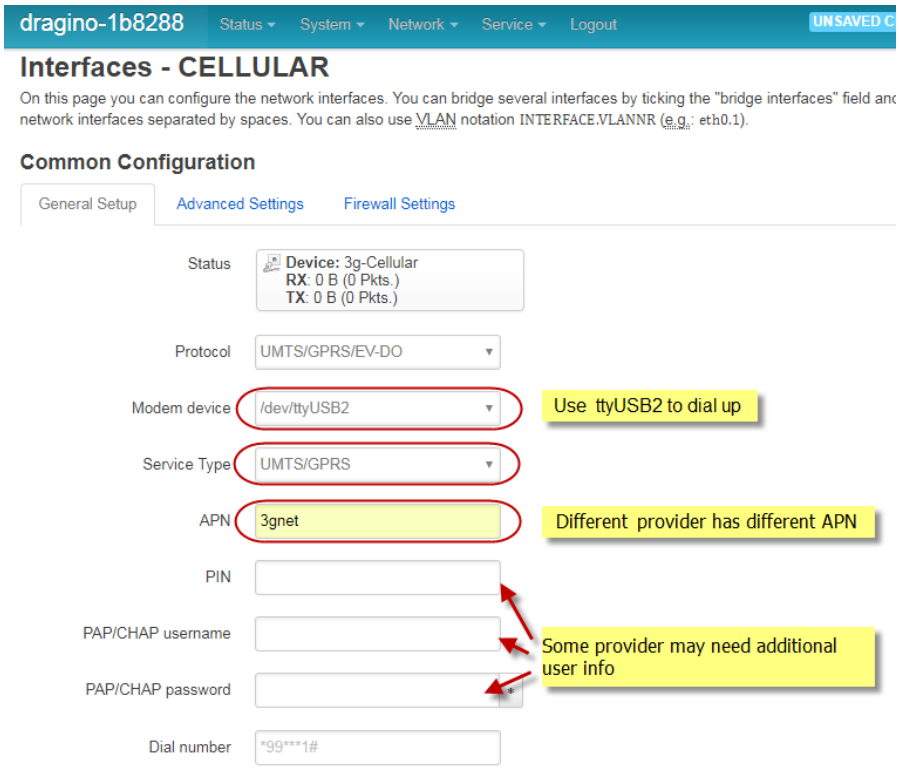
For the LG02 model with 4G version, user can configure the modem for internet access.

### Step 1: Add New Interface

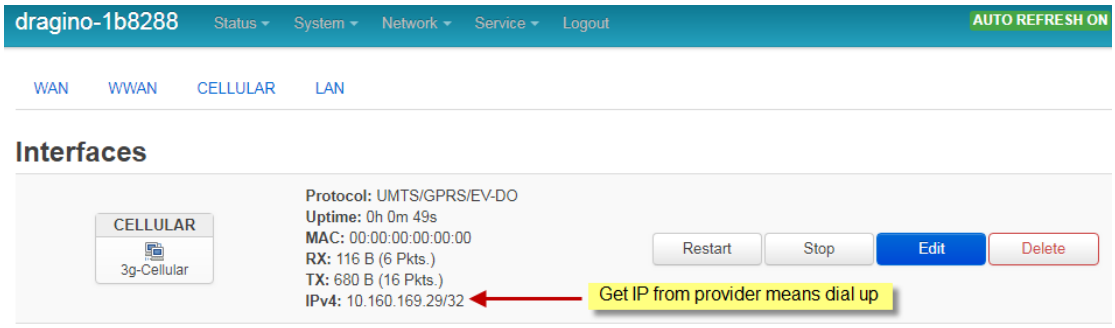


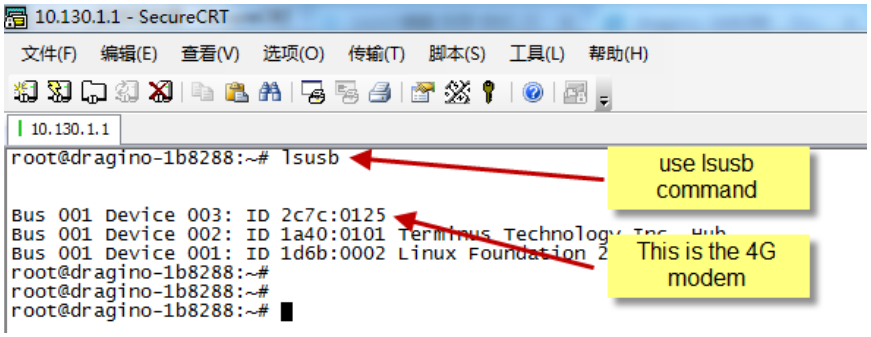
### Step 2: Configure cellular interface




### Step 3: Check Result

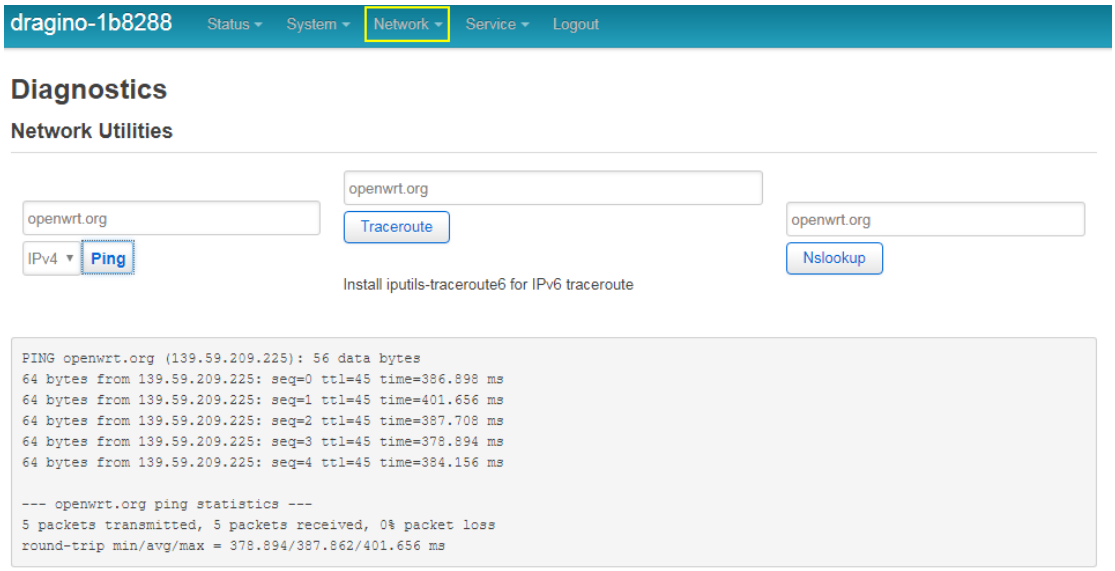



**Note: In case you don't know if your device has 4G modem, you can run lsusb command in SSH access to check, as below:**



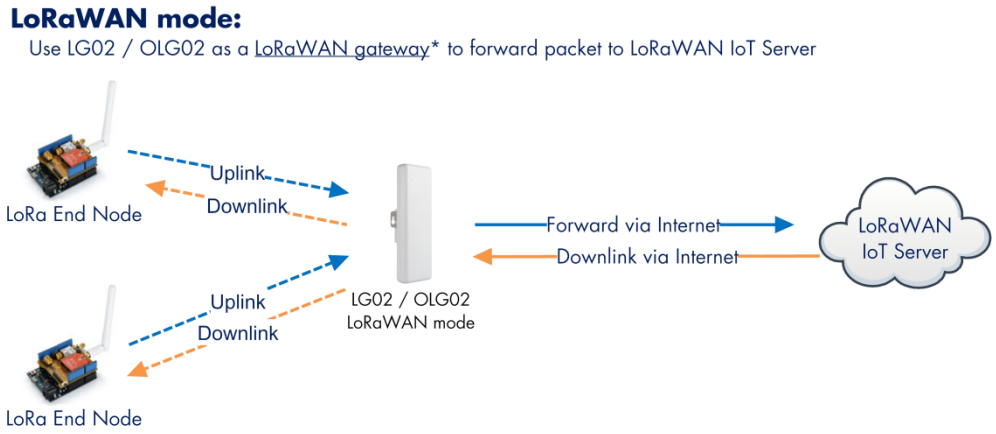
### 3.5 Check Internet connection

User can use the diagnostics page to check and analyze Internet connection.



## 4. Example 1: Configure as a LoRaWAN gateway – LoRaWAN mode

LG02 has two LoRa channels and can be configured as a Dual Channel LoRaWAN gateway. LG02 is pre-configured to support one uplink channel and one downlink channel in the LoRaWAN mode. This mode works as below diagram:

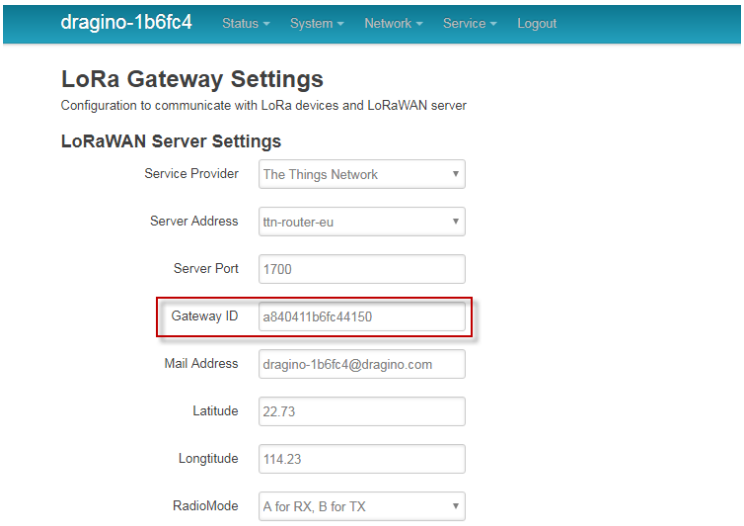



This chapter describes how to use LG02 to work with TTN LoRaWAN Server. The method to work with other LoRaWAN is similar.

### 4.1 Create a gateway in TTN Server

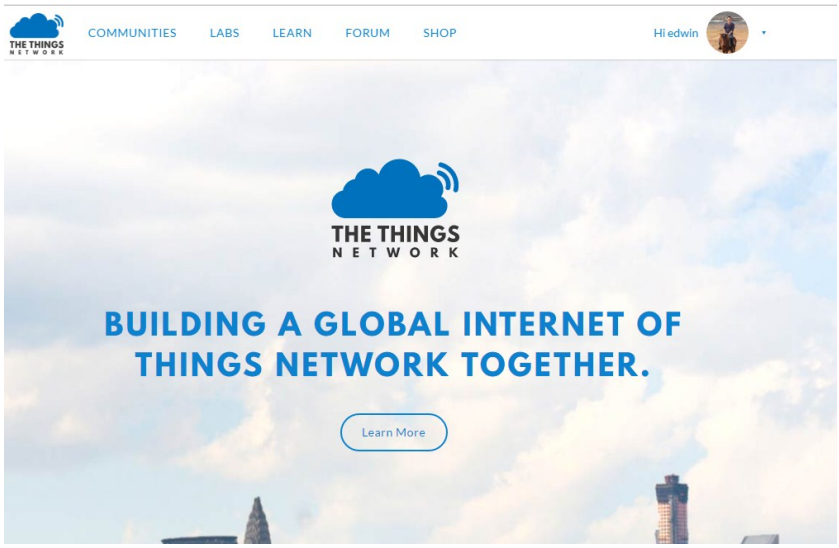
#### Step 1: Get a Unique gateway ID.

Every LG02 has a unique gateway id. The id can be found at LoRaWAN page:

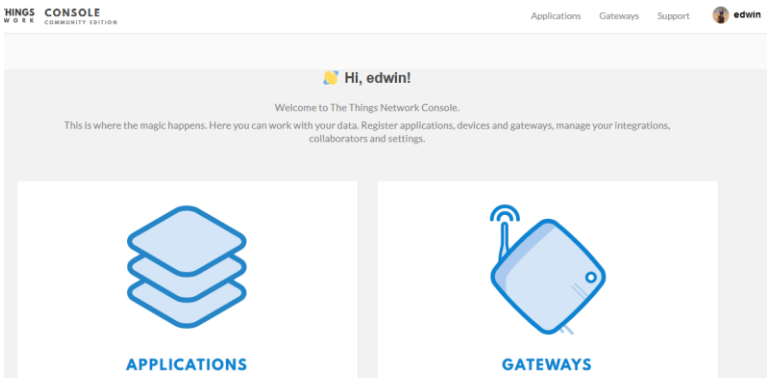


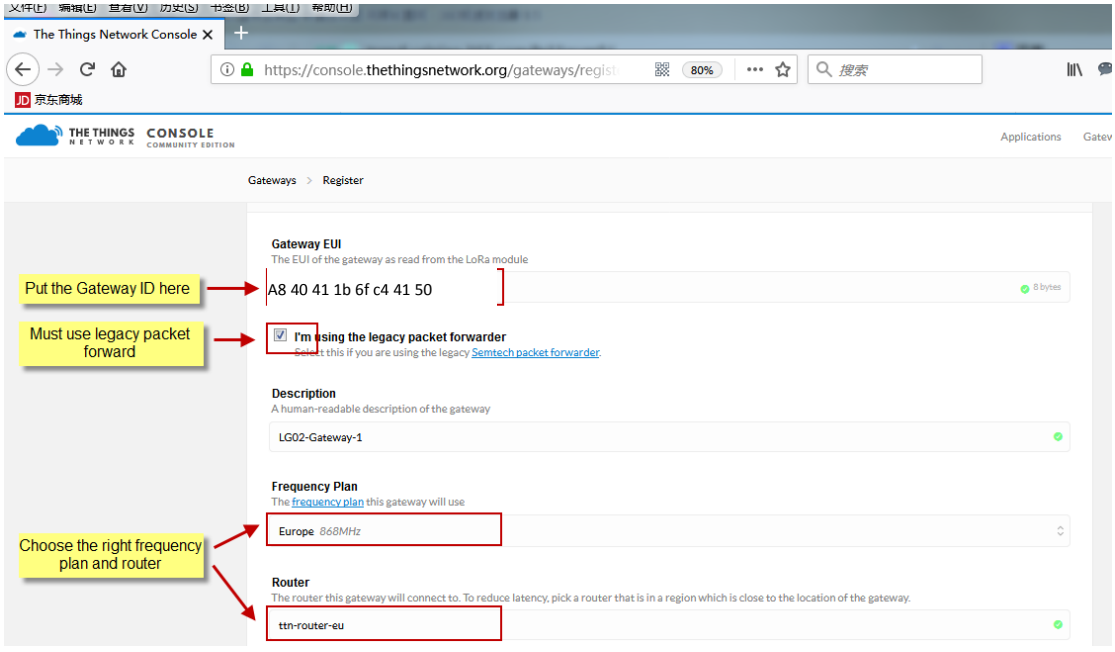

The gateway id is: **a840411b6fc44150**

#### Step 2: Sign up a user account in TTN server

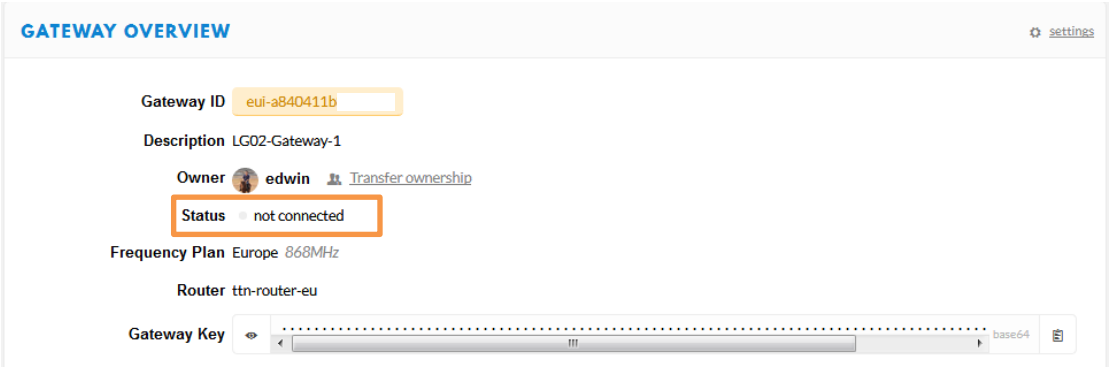



## Step 3: Create a Gateway in TTN



After create the gateway, we can see the gateway info, as below, the Status shows “not connected” because the LG02 doesn’t configure to send update status yet.

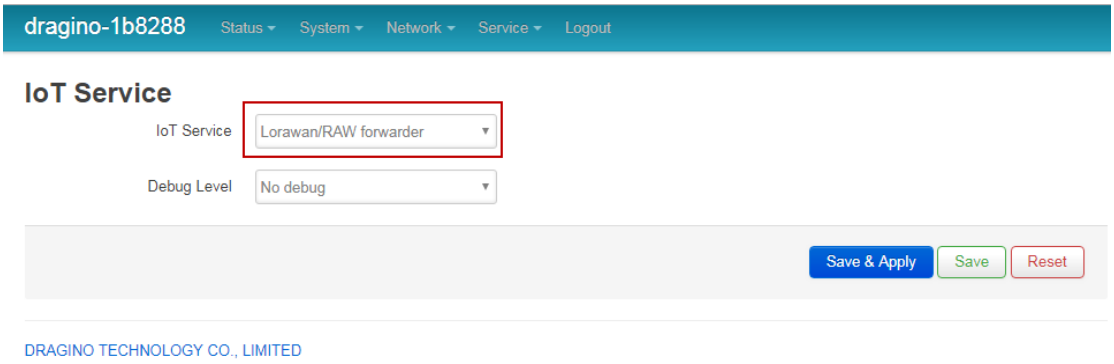



## 4.2 Configure LG02 Gateway

### 4.2.1 Configure to connect to LoRaWAN server

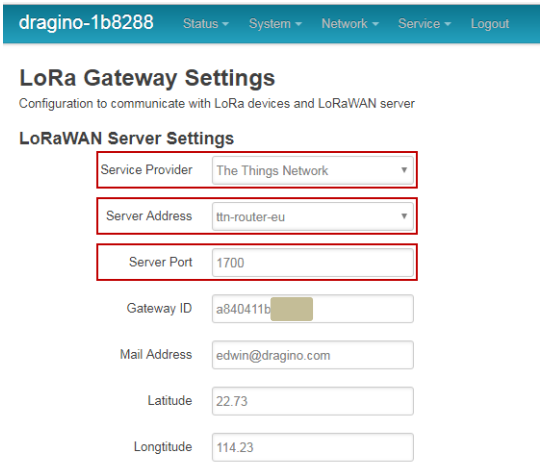
We should configure the LG02 now to let it connect to TTN network. Make sure your LG02 has Internet Connection first.

**Step1: Configure LG02 to act as raw forwarder**



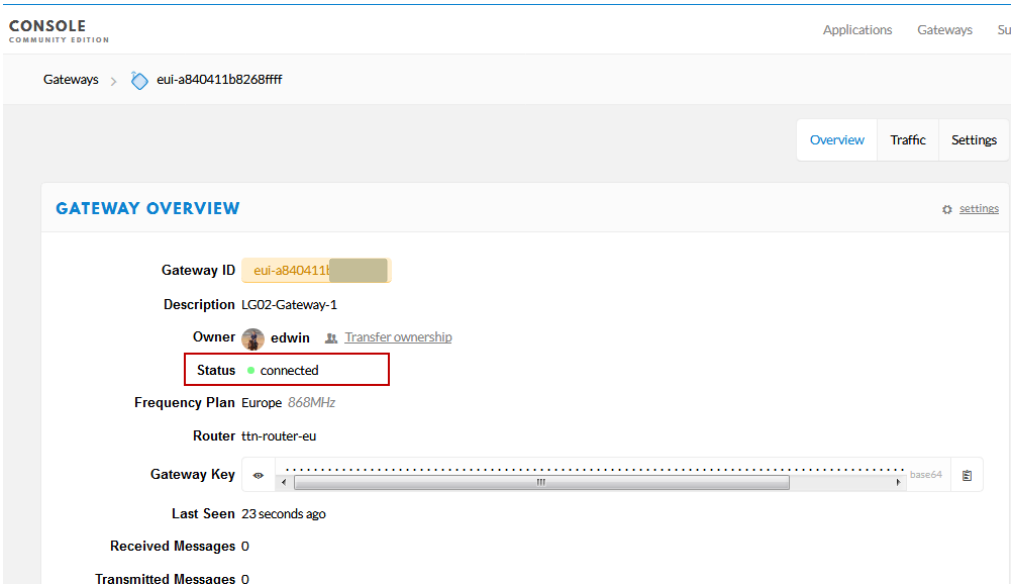

**Step2: Input server info and gateway id**

Choose the correct the server address and gateway ID.



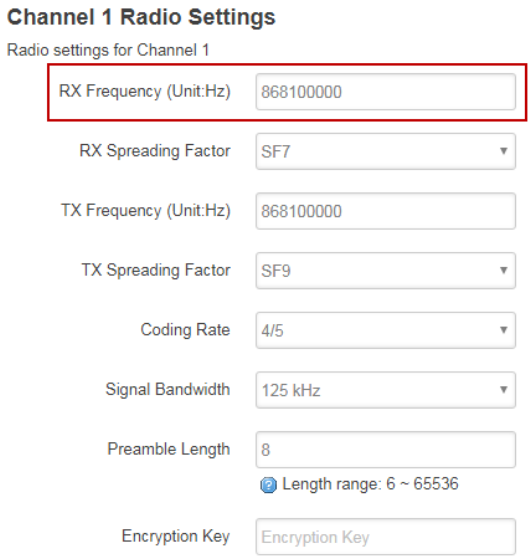
### Check Result

After above settings, the LG02 should be able to connect to TTN, below is the result seen from TTN:




## 4.2.2 Configure LG02's RX frequency

Now we should configure LG02 to act receive the LoRaWAN packets. This is in Channel 1, we configure is to use 868.1Mhz (868100000 Hz) as below.




Users only need to configure the RX Radio settings. In LoRaWAN protocol, the downlink packet from LoRaWAN server will specify the downlink frequency and SF (Spreading Factor), the gateway will use these parameters to send out Downlink LoRaWAN packets.

## 4.3 Create LoRa End Node

### 4.3.1 About Limited support for LoRaWAN

LG02 supports LoRaWAN End Node, in LoRaWAN protocol, it requires LoRaWAN node to send data in a hopping frequency. Since LG02 only support one single frequency, it will only able to receive the packet which is of the same Radio Frequency set up in LG02 Radio 1.

For example, in EU868, a standard LoRaWAN device may send the data in eight frequencies with different SF, such as:

```
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);        // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);        // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK,    DR_FSK),    BAND_MILLI);        // g2-band
```

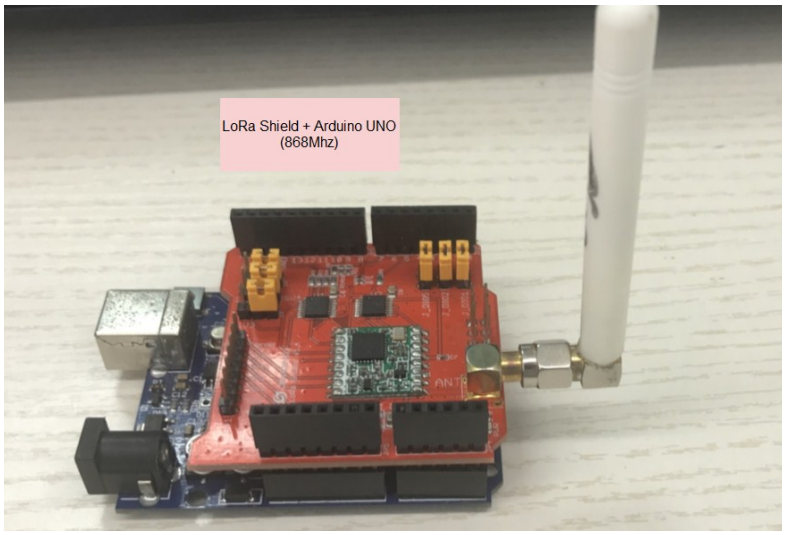
So the LG02 will only able to receive the 868100000, SF7 packet and will not receive others. Means only one packet will arrive the TTN server in every 8 packet sent from the LoRaWAN end node.

If user want all the packets from LoRaWAN end device can arrive LoRaWAN server, user need to set up the LoRaWAN node to send packet in a specify frequency.

In this section, we will use LoRa Shield and a modify LMIC Library to show how to configure LoRaWAN end node and let's work in single frequency.

### 4.3.2 Preparation

**LoRaWAN End device Hardware:**




**Software Library for LoRaWAN End device**:

Install this library https://github.com/dragino/arduino-lmic to the Arduino Library path. Before compiling the End Device software, User needs to change the Frequency Band to use with LG02. What user need to change is in the file **arduino\libraries\arduino-lmic \src\lmic\config.h.**

Changes are as below:

```
#define CFG_eu868 1
//#define CFG_us915 1
//#define CFG_au921 1
//#define CFG_as923 1
//#define CFG_in866 1

#define LG02_LG01 1

//US915: DR_SF10=0, DR_SF9=1, DR_SF8=2, DR_SF7=3, DR_SF8C=4
//       DR_SF12CR=8, DR_SF11CR=9, DR_SF10CR=10. DR_SF9CR=11. DR_SF8CR=12. DR_SF7CR
#if defined(CFG_us915) && defined(LG02_LG01)
// CFG_us915 || CFG_as923
#define LG02_UPFREQ   902320000
#define LG02_DNWFREQ  923300000
#define LG02_RXSF     3       // DR_SF7
#define LG02_TXSF     8       // DR_SF12CR
#elif defined(CFG_eu868) && defined(LG02_LG01)
// CFG_eu868
//EU868: DR_SF12=0, DR_SF11=1, DR_SF10=2, DR_SF9=3, DR_SF8=4, DR_SF7=5, DR_SF7B=1, DR_FSK, DR_NONE
#define LG02_UPFREQ   868100000
#define LG02_DNWFREQ  869525000
#define LG02_RXSF     5       // DR_SF7
#define LG02_TXSF     0       // DR_SF12
#endif
```

Choose the Frequency Band, same as in LoRaWAN server

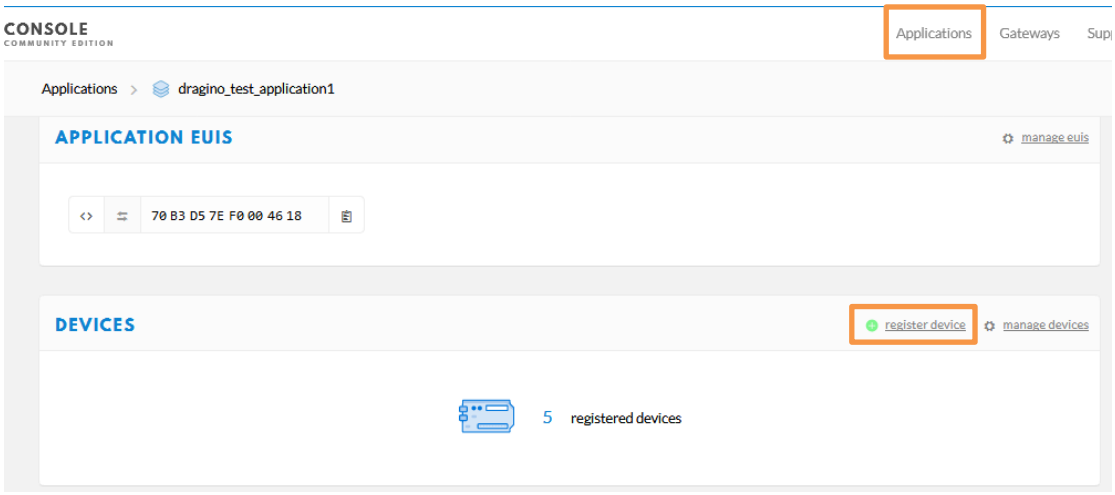
uncomment this for LG01 / LG02

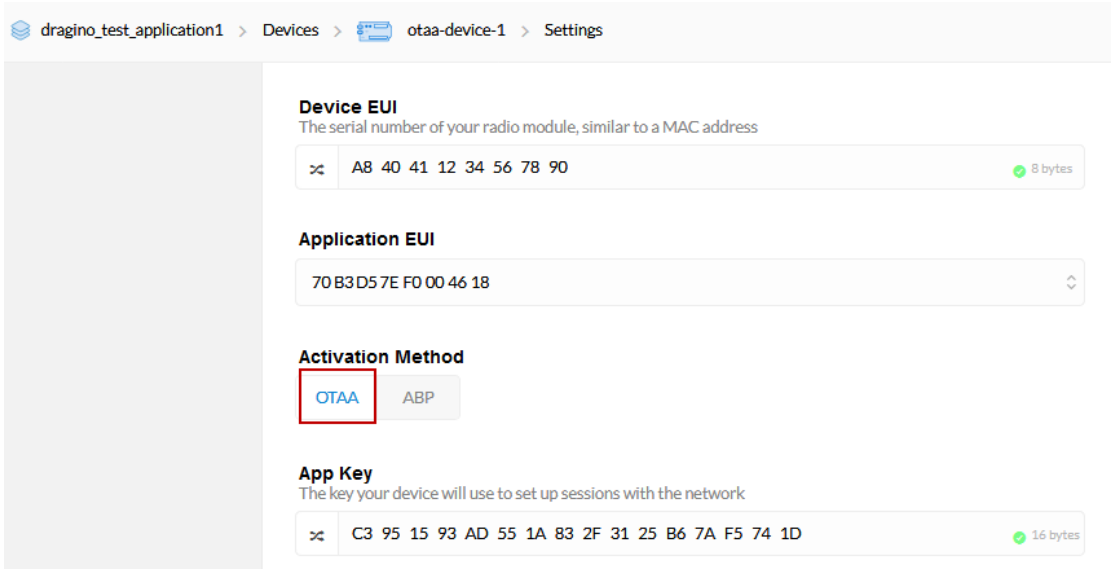
LG02_UPFREQ:End Device Uplink Frequency
LG02_DNWFREQ:End Device Uplink Frequency
LG02_RXSF: End Device Uplink (transmit) SF
LG02_TXSF: End Device Downlink (receive) SF

The TXSF is now set to default value:
US915/AS923 : 923300000 , SF12BW500
EU868: 869525000, SF12BW125

### 4.3.3 Test with OTAA LoRa end node (LoRa Shield + UNO)

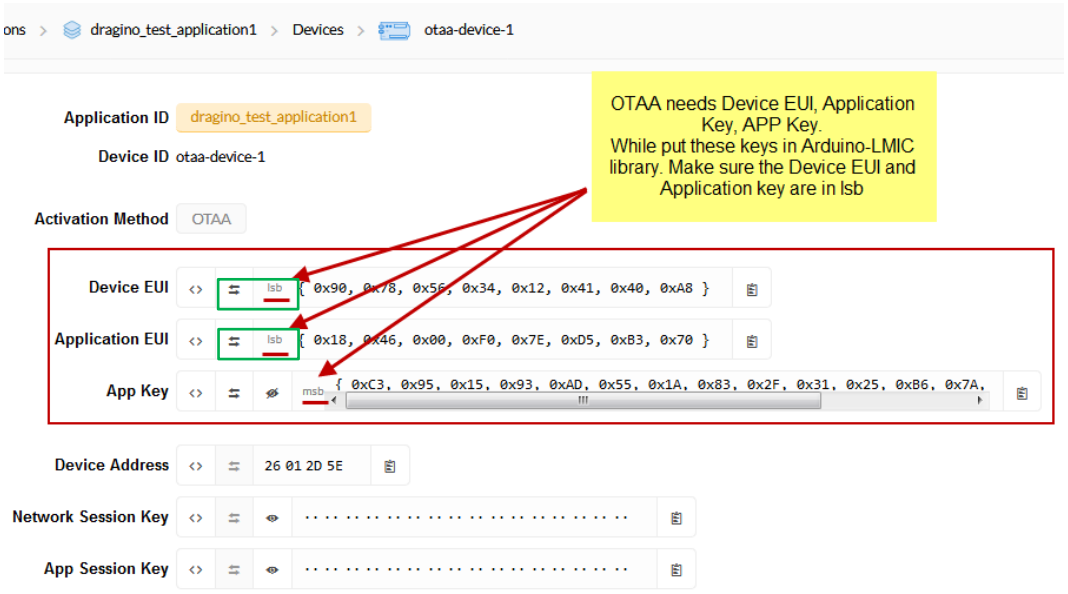
Step 1: Create an OTAA device in TTN server -- > Application page.

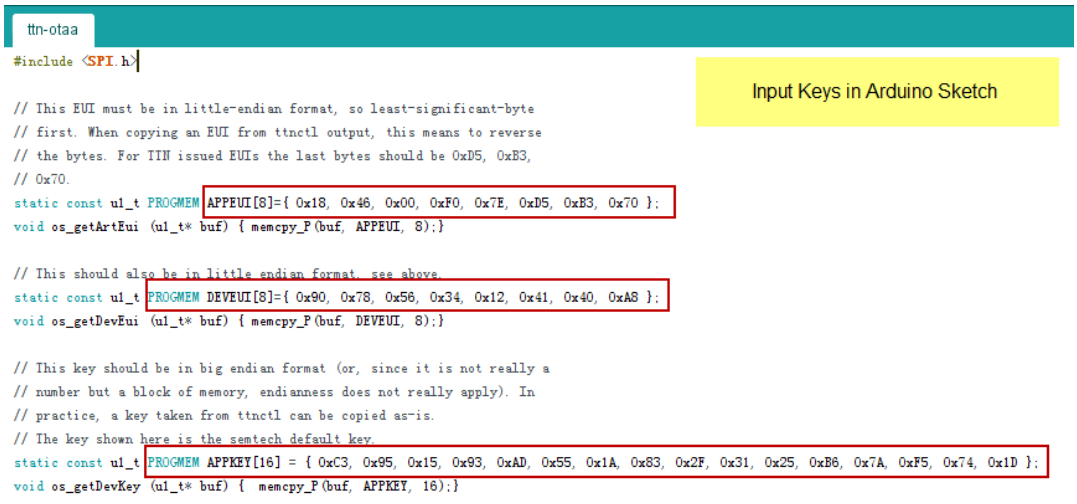


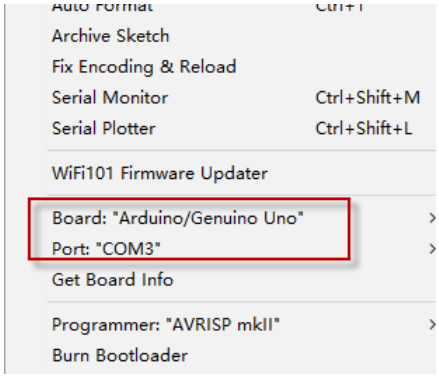
Step 2: Input keys into Arduino Sketch.

The sketch for the LoRa Shield is in Arduino –IDE --> Examples -->LMIC_Arduino→ ttn-otaa

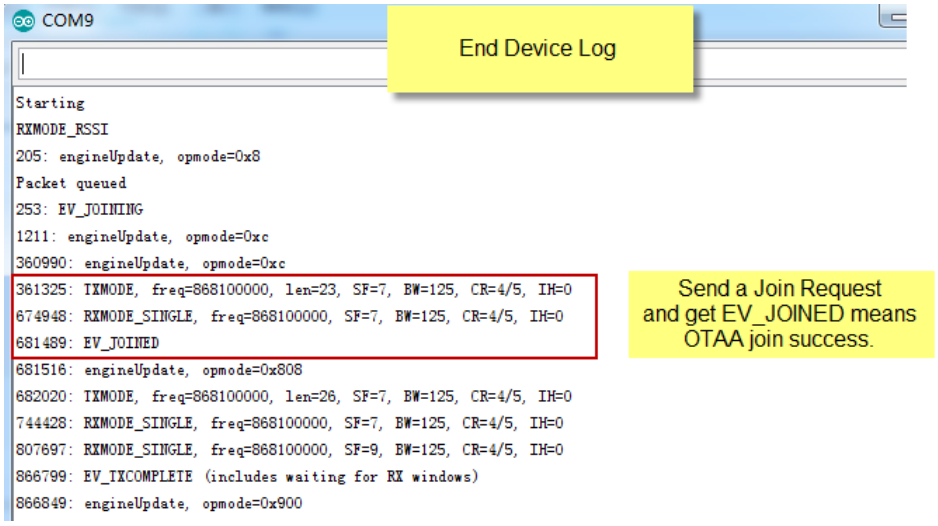


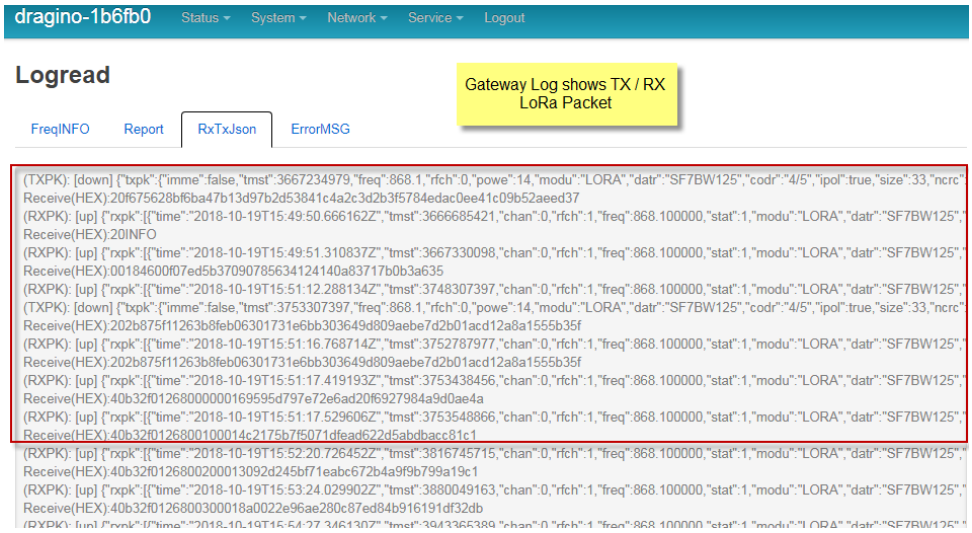



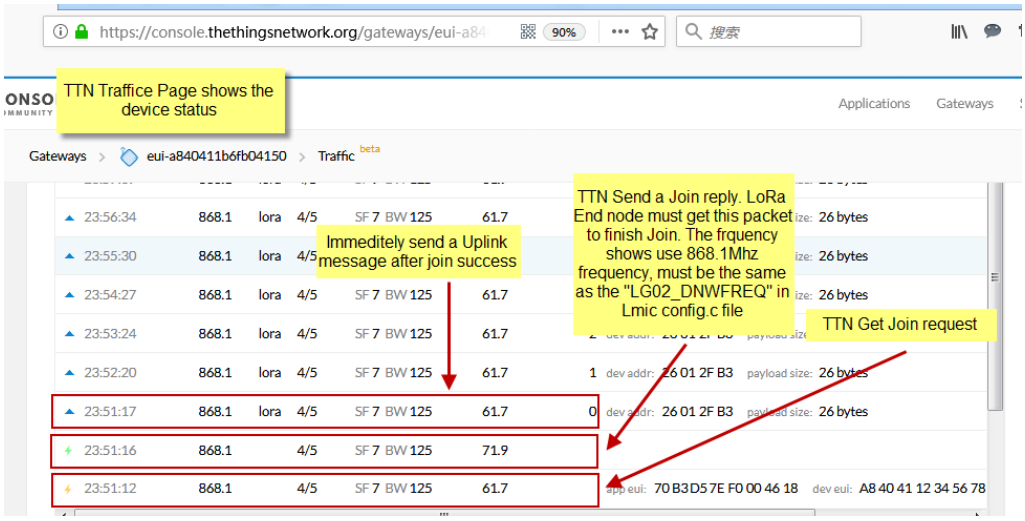

Choose Arduino UNO to upload the sketch to LoRa Shield and UNO



Step 3: Check Result for OTAA



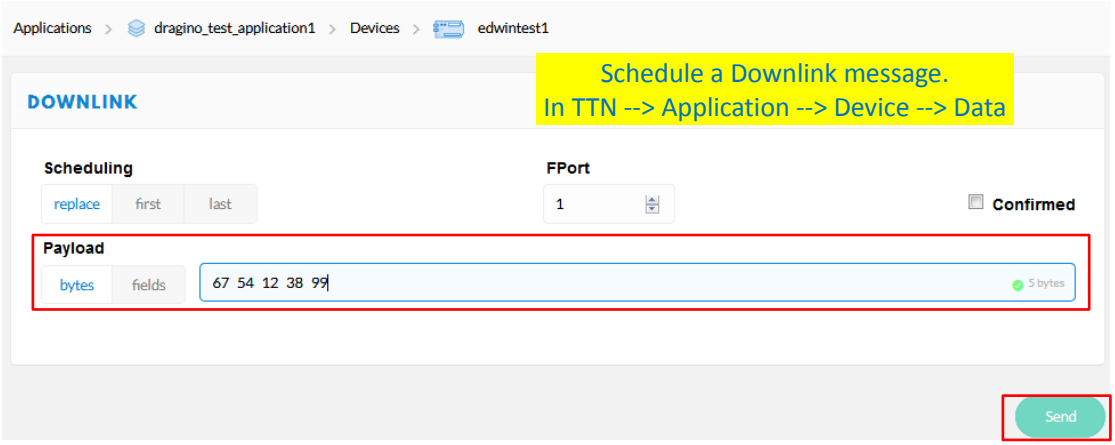



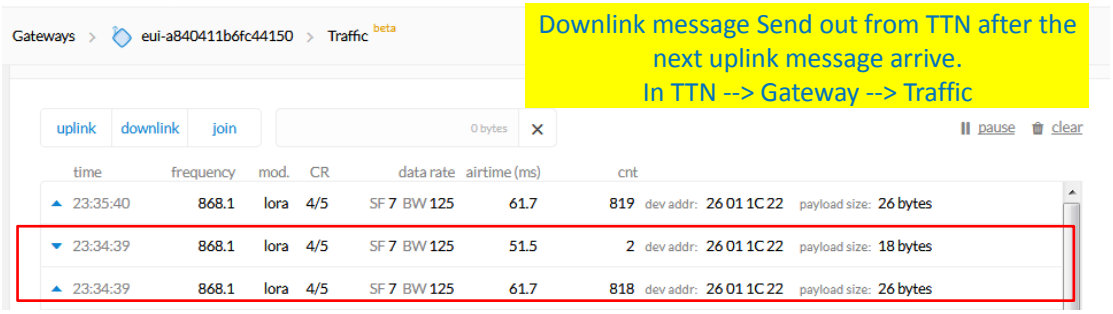



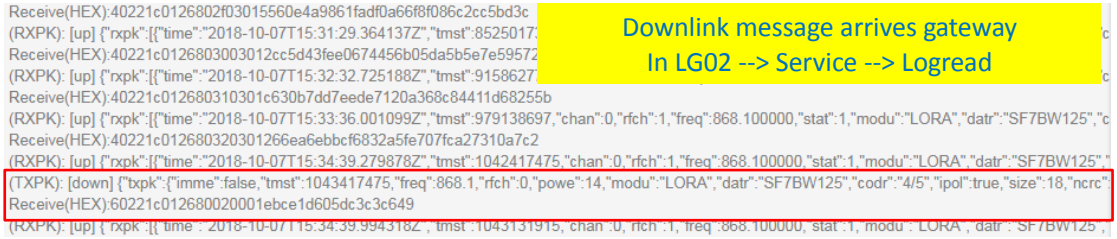

Note: The LG02_DNWFREQ value in Arduno_LMIC/src/lmic/config.h should match downlink frequency from TTN.

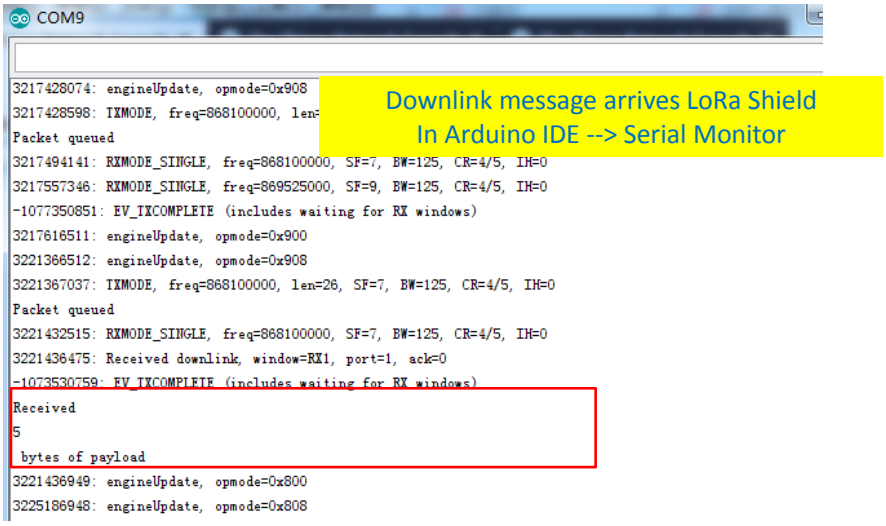
TTN shows 868.1 here, So LG02_DNWFREQ should be 868100000

Step 4: Test Downlink



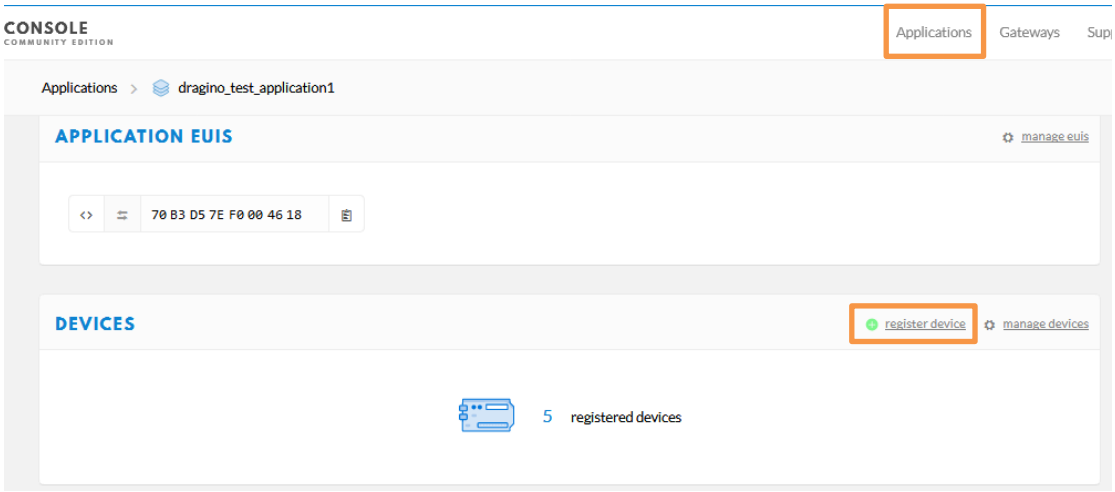



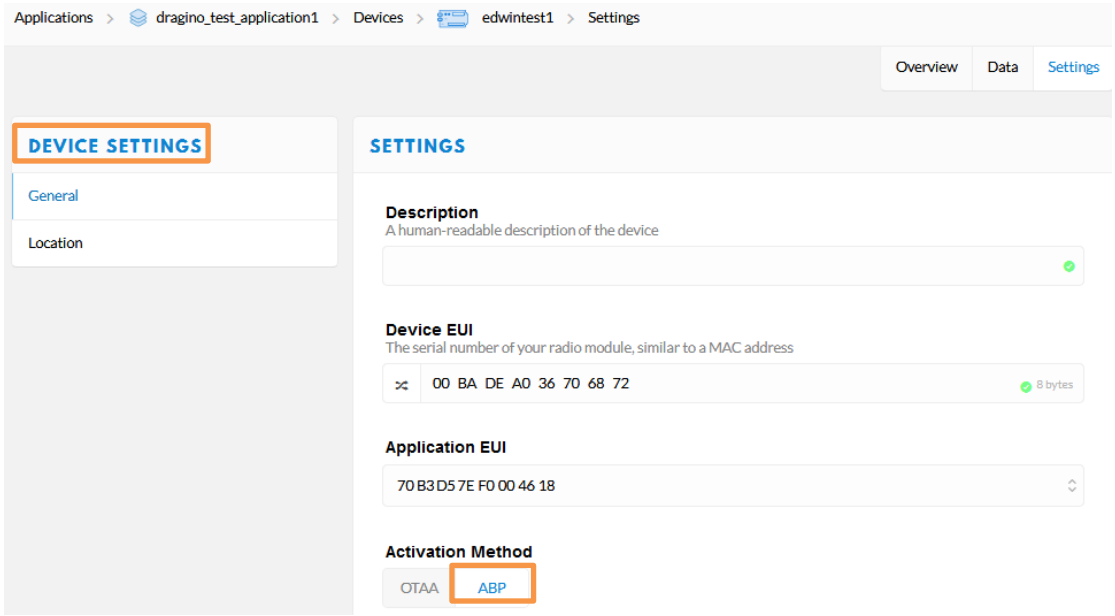



### 4.3.4 Test with ABP LoRa end node (LoRa Shield + UNO)

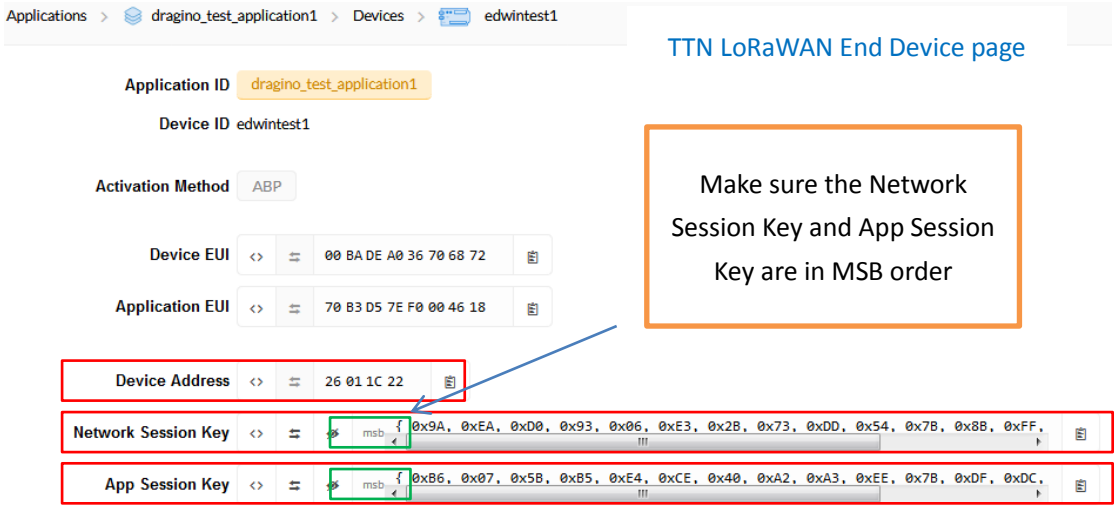
Step 1: Create an ABP device in TTN server -- > Application page. And change it to ABP mode.

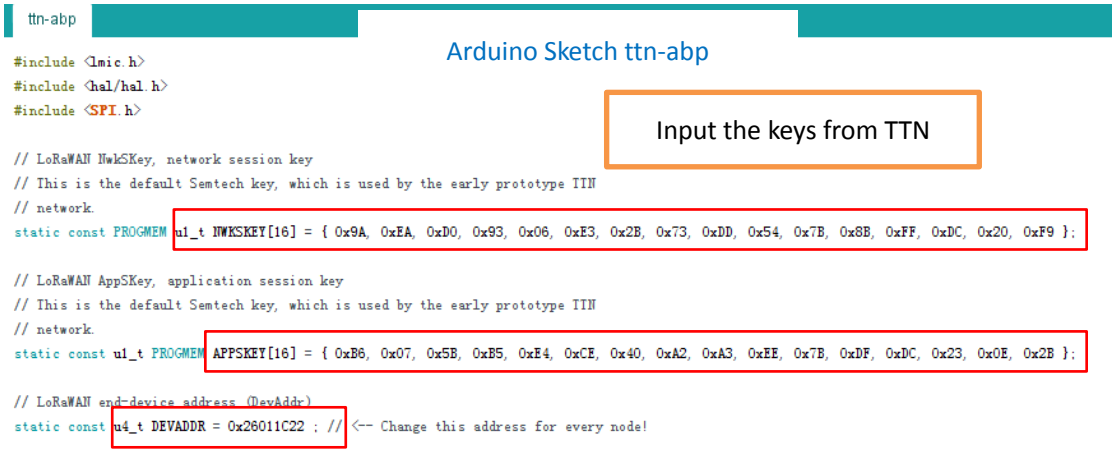


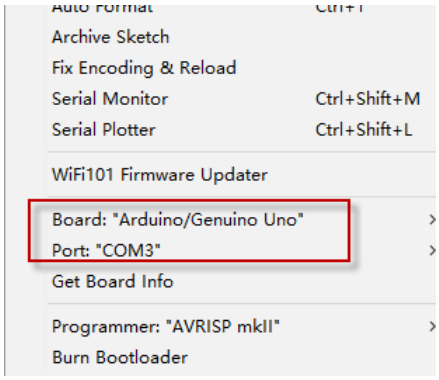
Step 2: Input keys into Arduino Sketch.

The sketch for the LoRa Shield is in Arduino –IDE --> Examples -->LMIC_Arduino→ ttn-abp

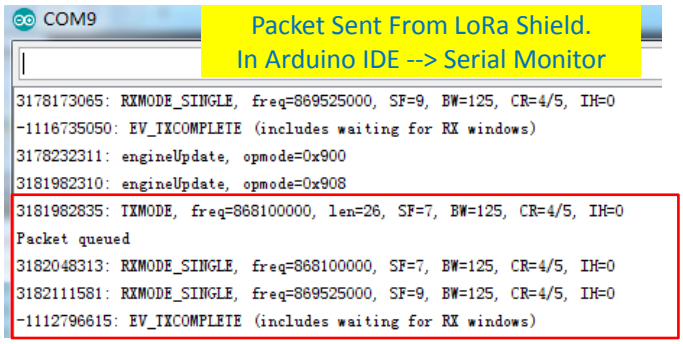


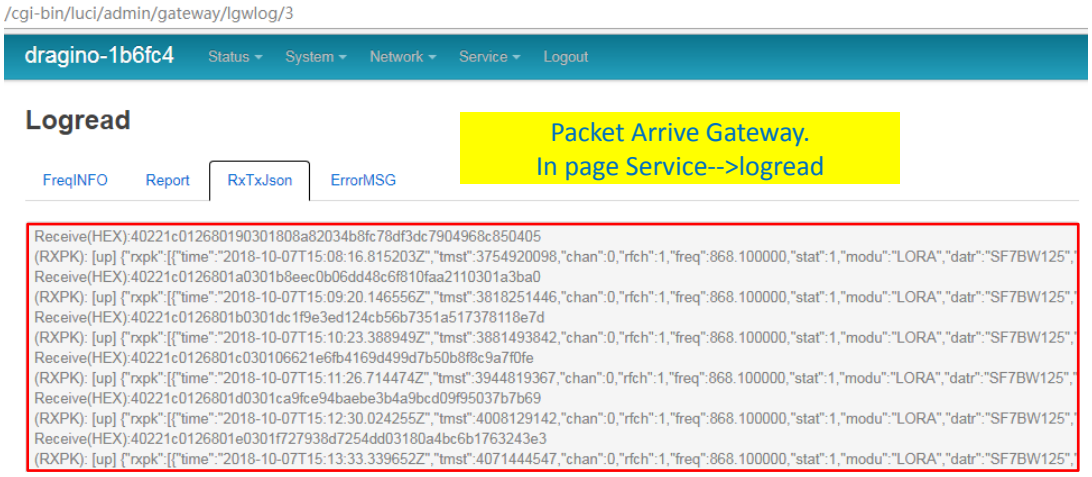



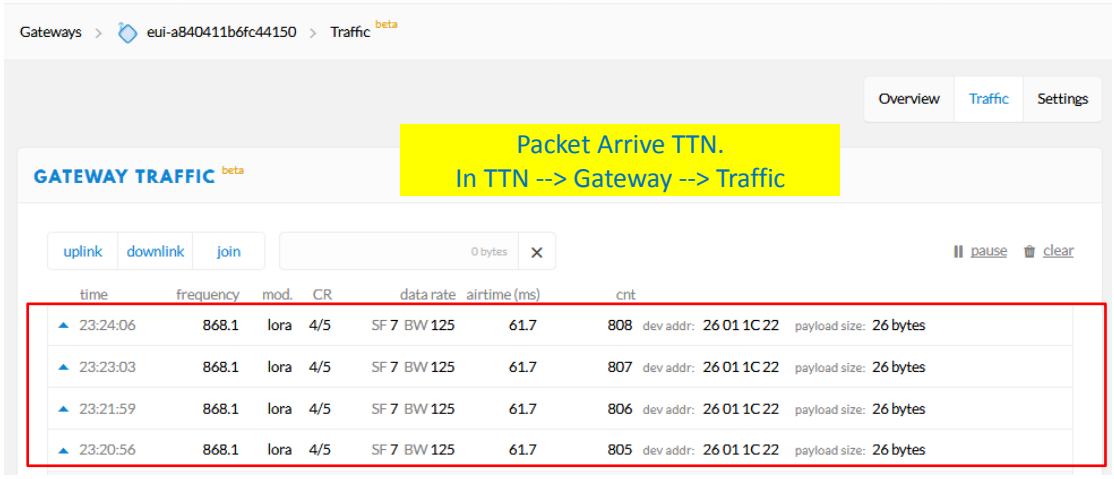

Choose Arduino UNO to upload the sketch to LoRa Shield and UNO

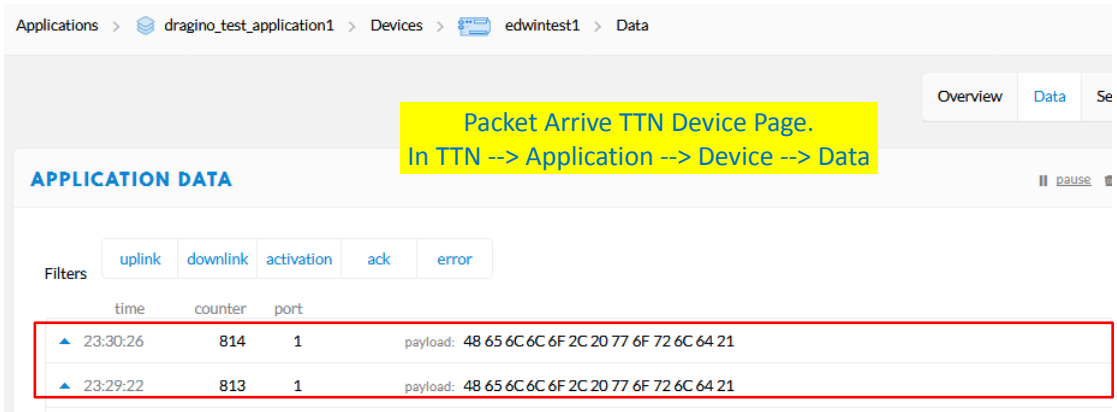


Step 3: Check Result for Uplink

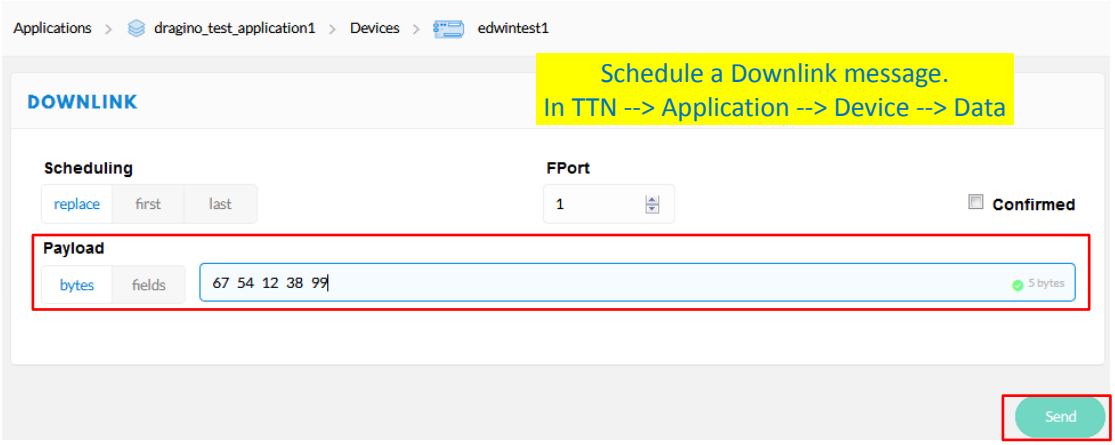


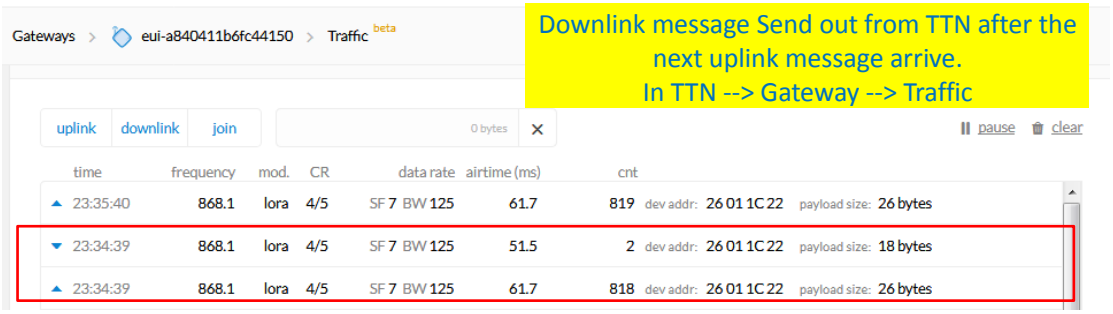



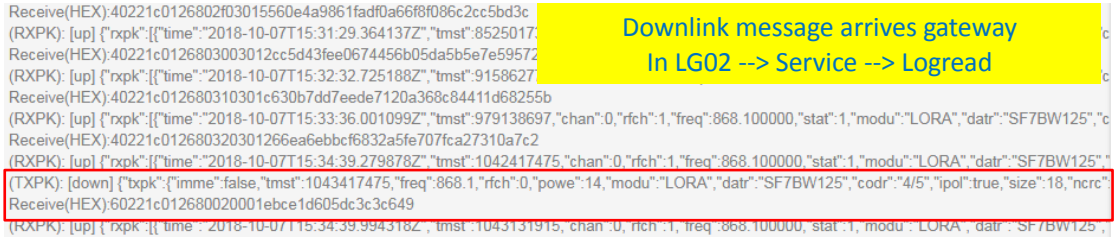



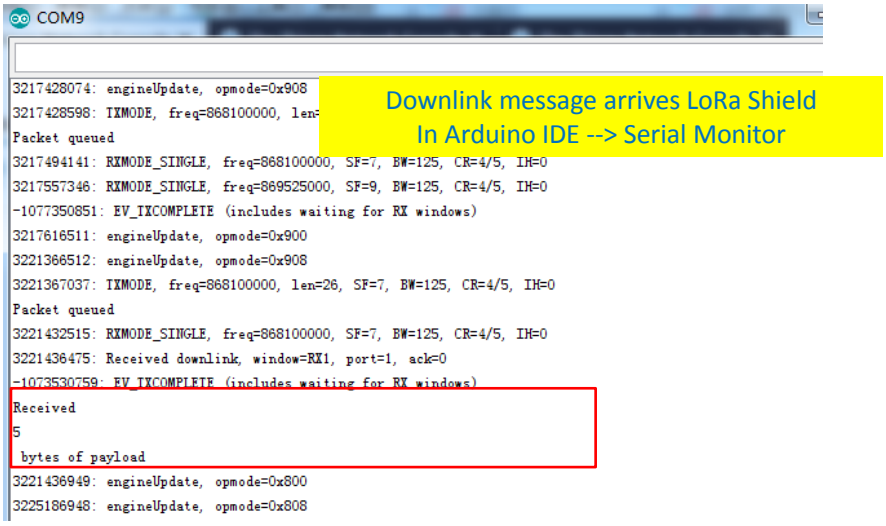



Step 4: Test Downlink



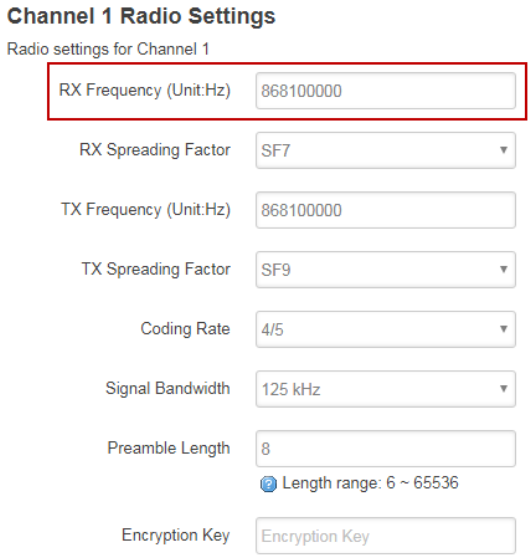
# 5. Example 2: Control LoRa radio directly as general LoRa transceiver

There are two ways to use the LoRa Radio of Gateway: a) Through pkt_fwd process , b) Use the Radio SPI device separately.

## 5.1 User LoRa Radio via pkt_fwd

### 5.1.1 Use pkt_fwd to receive

When user chooses the MQTT/TCP-IP/Customized mode, the lg02_pkt_fwd will auto start. It will listen the LoRa Radio Channel 1 base on the setting in the web setting.




If the LoRa end node send data in the match format, the pkt_fwd will store the data for further use, the logic of this receive part please see Customized_Script.

### 5.1.2 Use pkt_fwd to transmit

(This is a new feature since 2019-Jan-30)

The pkt_fwd also open a thread to listen to local files under directory **/var/iot/push/**. Once there is a file in this directory, the thread will check if it is an outgoing file and send out the LoRa message if format match. Below is the file example (json format):

{"txpk":{"imme":false,"tmst":861608339,"freq":925.1,"rfch":0,"powe":20,"modu":"LORA","datr":"SF7BW500","codr":"4/5","ipol":true,"size":22,"ncrc":true,"data":"YEklBCaqCgADQAIAcQM6AP8B9TYzUA=="}}

**Explain:**

| Name | Type | Function |
|---|---|---|
| imme | bool | Send packet immediately (will ignore tmst & time) |
| tmst | number | Send packet on a certain timestamp value (will ignore time) |
| tmms | number | Send packet at a certain GPS time (GPS synchronization required) |

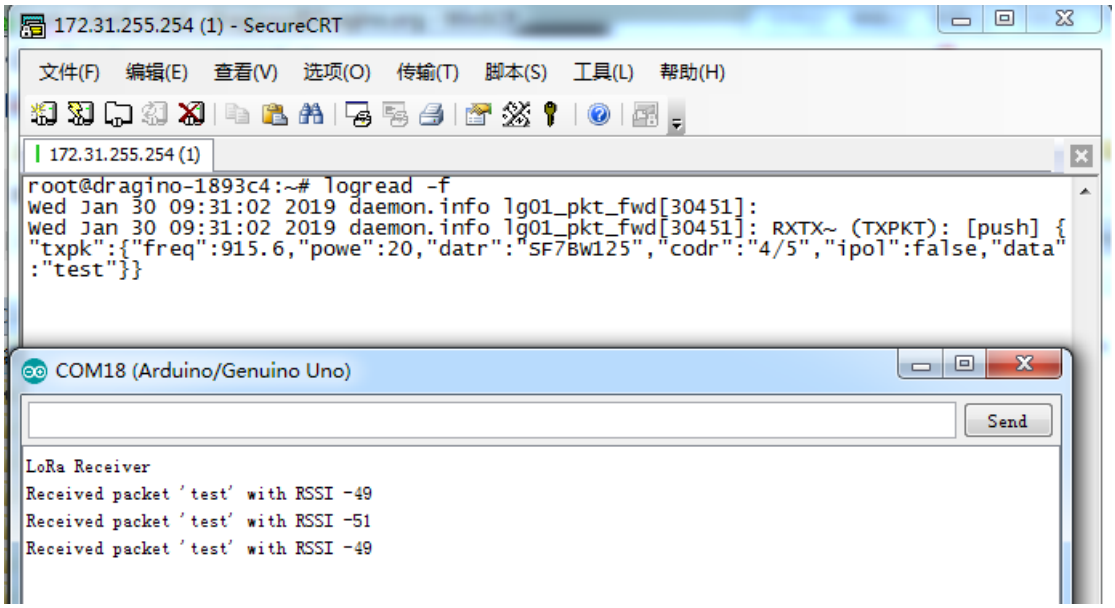
freq | number | TX central frequency in MHz (unsigned float, Hz precision)
rfch | number | Concentrator "RF chain" used for TX (unsigned integer)
powe | number | TX output power in dBm (unsigned integer, dBm precision)
modu | string | Modulation identifier "LORA" or "FSK"
datr | string | LoRa datarate identifier (eg. SF12BW500)
datr | number | FSK datarate (unsigned, in bits per second)
codr | string | LoRa ECC coding rate identifier
fdev | number | FSK frequency deviation (unsigned integer, in Hz)
ipol | bool | Lora modulation polarization inversion
prea | number | RF preamble size (unsigned integer)
size | number | RF packet payload size in bytes (unsigned integer)
data | string | Base64 encoded RF packet payload, padding optional
ncrc | bool | If true, disable the CRC of the physical layer (optional)

Not all field are necessary, below is an example:

1) First set up a LoRa Shield with this code: LoRaReceiver. So the LoRa Shield will receive the data at frequency 915.6Mhz, SF7BW125, CR: 4/5
2) Edit a file (any name) under **/var/iot/push/** with below content**.**

```
{"txpk":{"freq":915.6,"powe":20,"datr":"SF7BW125","codr":"4/5","ipol":false,"data":"test"}}
```

And then we can see below output

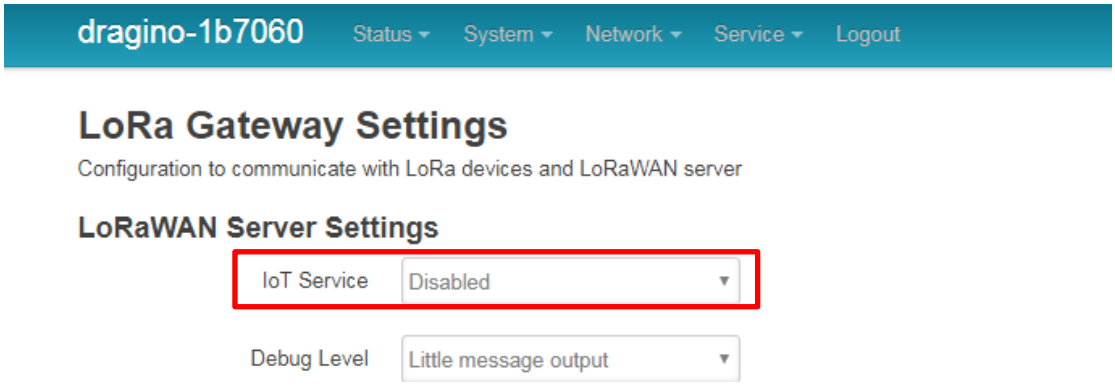


## 5.2 Use LoRa radio device directly

There are two SPI devices in LG02, user can use lg02_single_rx_tx to control this SPI device for transmit and receive. When use the lg02_single_rx_tx command to transmit, it will initiate the SPI device on each call and it will add delay to start transmit, this will be slower than the pkt_fwd mode

### Step 1: Disable packet forward

With firmware higher than version LG02_LG08--build-v5.1.1545908833-20181227-1908, select "Disabled" in IoT Service page.



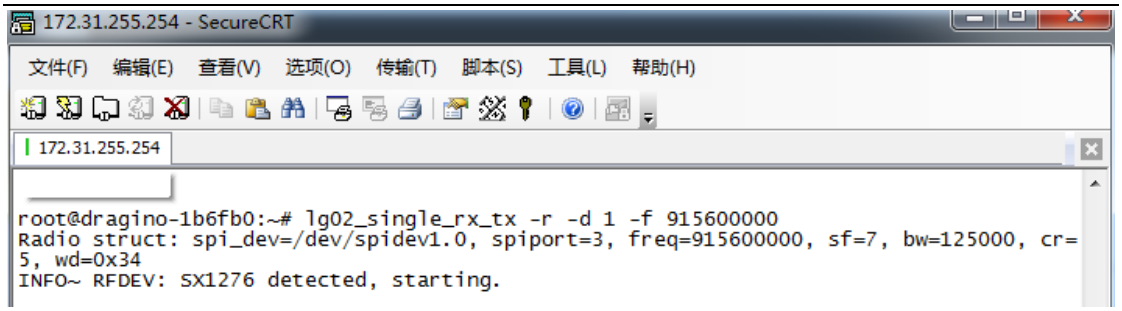

### Step 2: Use lg02_single_rx_tx to receive

Usage: lg02_single_rx_tx [-d radio_dev] select radio 1 or 2 (default:1)
[-t] set as tx
[-r] set as rx
[-f frequence] (default:868500000)
[-s spreadingFactor] (default: 7)
[-b bandwidth] default: 125k
[-w syncword] default: 52(0x34)reserver for lorawan
[-p message ] message to send
[-o filepath ] payload output to file
[-v] show version
[-h] show this help and exit Use Radio 1 to transmit:

Command:

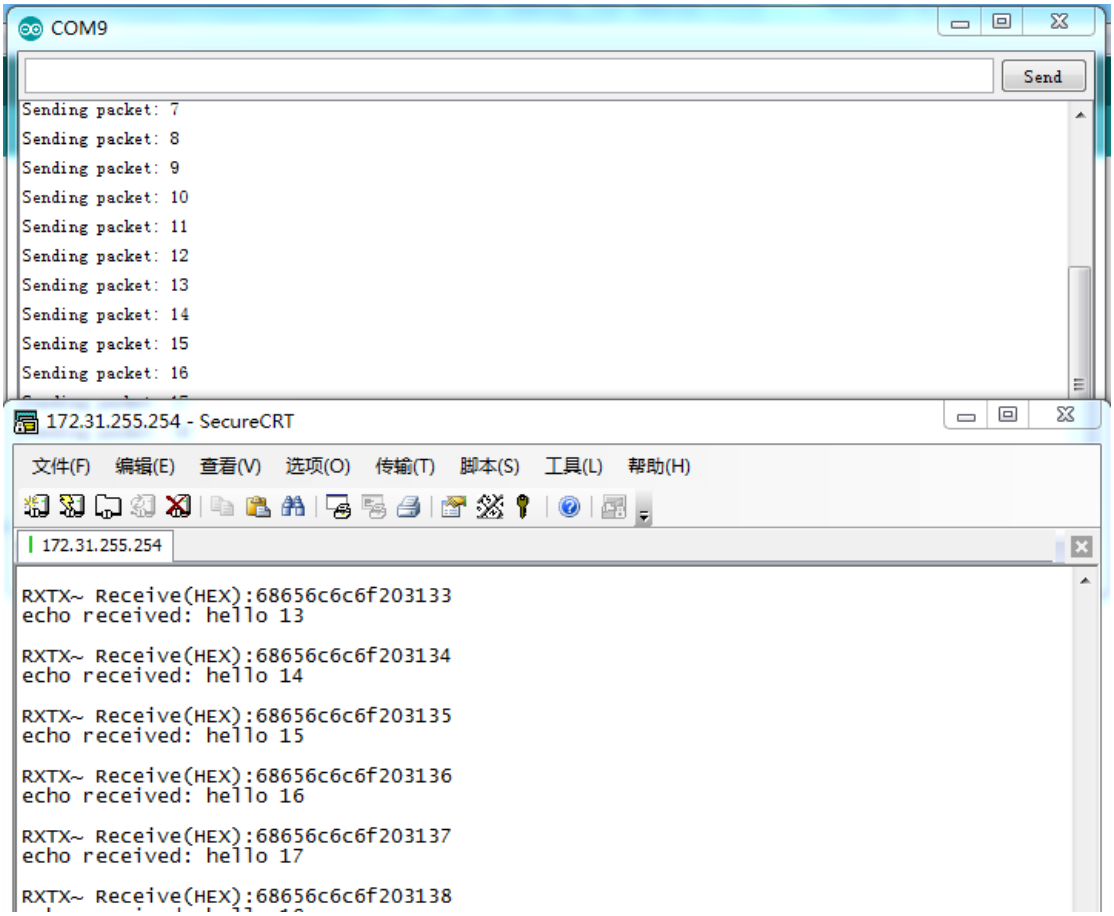
```
root@dragino-1b6fb0:~# lg02_single_rx_tx -r -d 1 -f 915600000
```

Use radio 1 to receive at frequency 9156000000




Then set up a LoRa node to send out LoRa packet, We use LoRa Shield + UNO in this example. The library use in Arduino UNO is LoRa-Master. And the source code is LoRaSender..

Result screen shot:



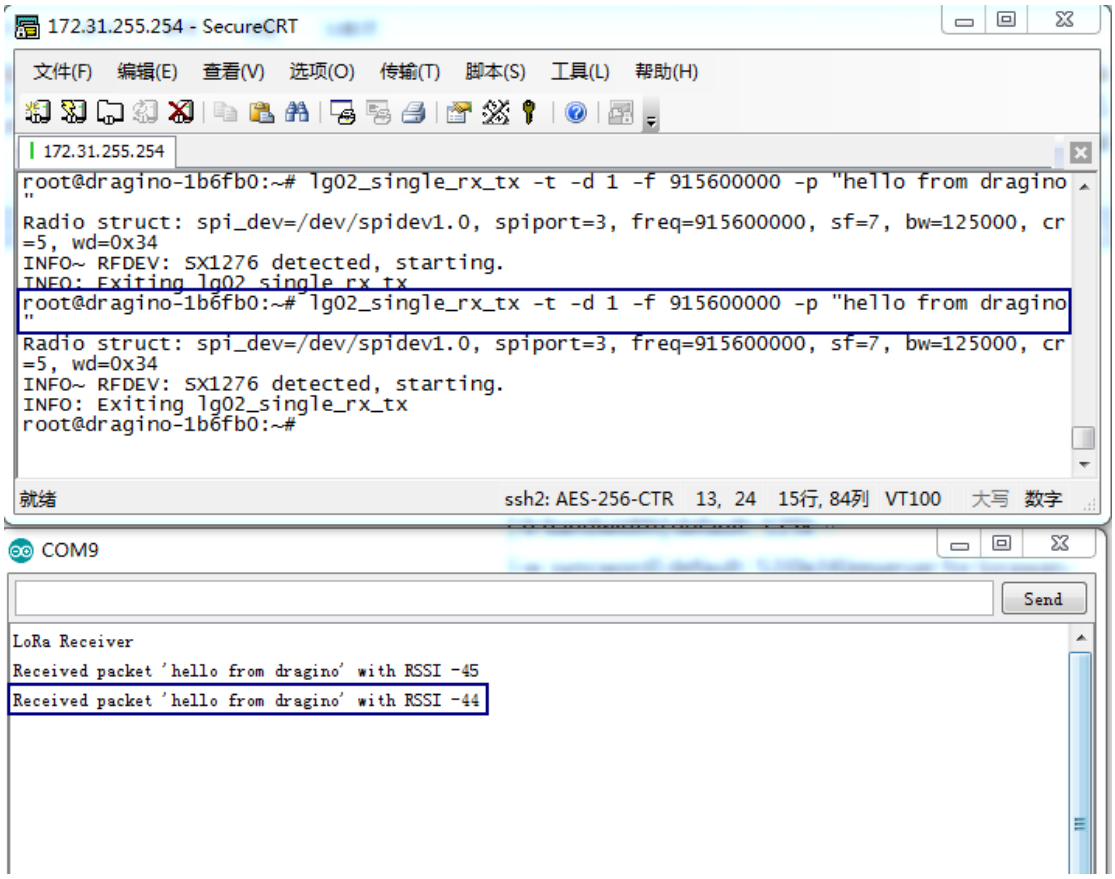
### Step 3: Use lg02_single_rx_tx to transmit

Command:

root@dragino-1b6fb0:~# lg02_single_rx_tx -t -d 2 -f 915600000 –m “hello from dragino”

Use radio 2 to transmit a message at frequency 9156000000

Set up a LoRa node to send out LoRa packet, We use LoRa Shield + UNO in this example. The library use in Arduino UNO is LoRa-Master. And the source code is LoRaReceiver.

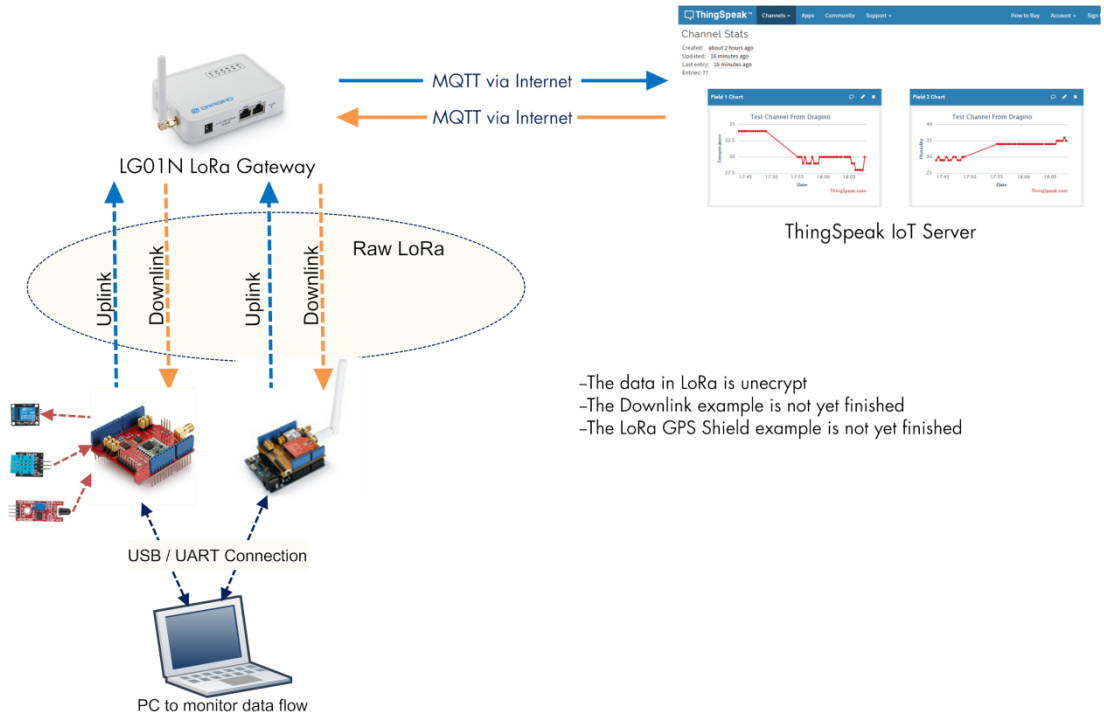


## 6. Example 3: MQTT Transfer Mode

MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios.

Most IoT server support MQTT connection, for those servers, we can use MQTT to connect it to publish data or subscribe to a channel.

**Topology for ThingSpeak Connection:**

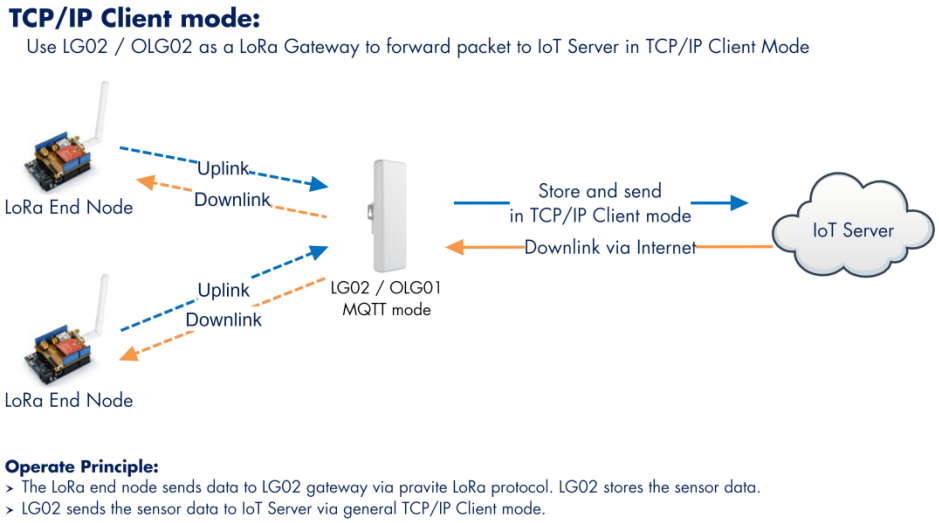



Most IoT server support MQTT connection, for those servers, we can use MQTT to connect it to publish data or subscribe to a channel.

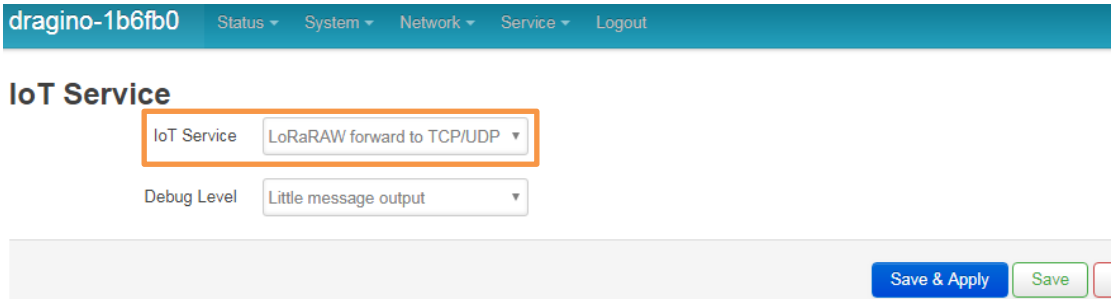
A detail of how to use MQTT plus Video instruction can be found at **Example 2: Test with a MQTT IoT Server** from user manual http://www.dragino.com/downloads/index.php?dir=LoRa_IoT_Kit/v2-Kit/ with version higher than v1.0.3

## 7. Example 4: TCP IP Client Mode

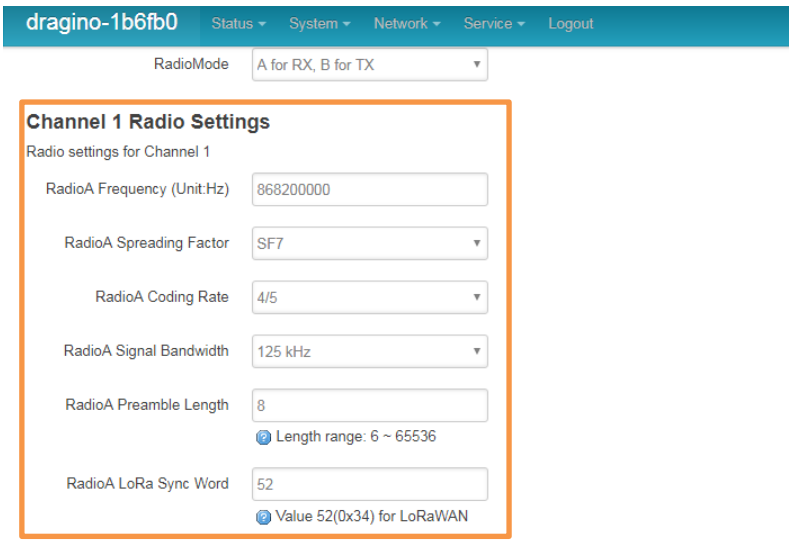
In the TCP IP Client mode, LG02 can accept LoRa packets and send it to the TCP-IP server. The working topology is as below. In this mode, The Uplink LoRa packets should use a customized format.




**Step1: Select TCP-IP Client mode**

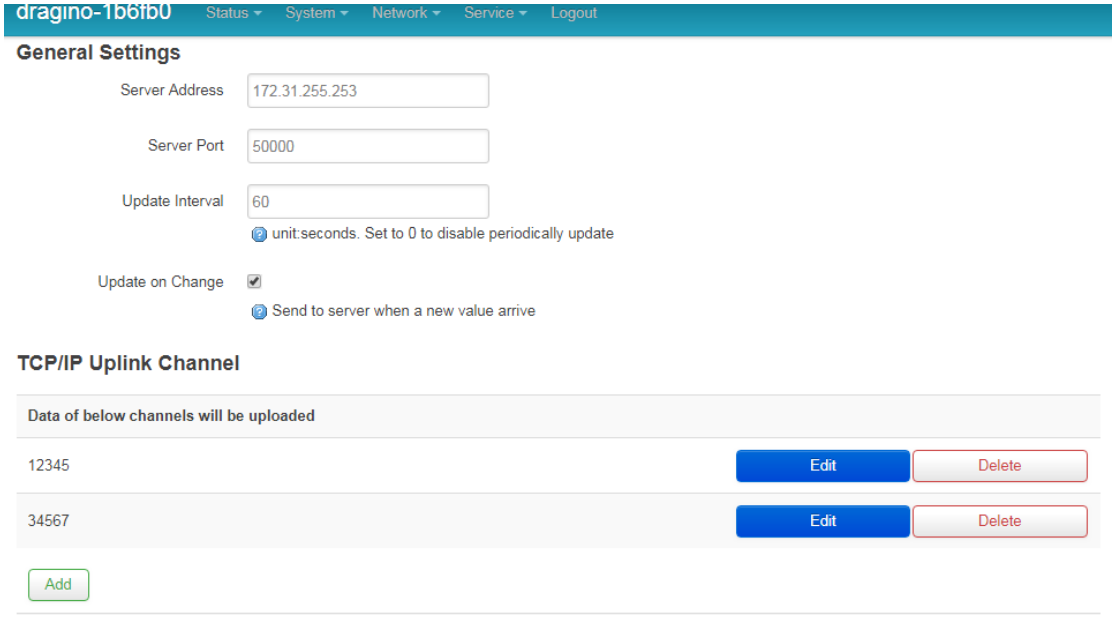



**Step2: Configure the Radio1 channel with the match radio settings frequency as the LoRa End Node**



**Step3: Configure TCP Server Info**

Note: Gateway may receive many LoRa packets, it will only transfer the packet with the same ID as specify in the channel.




**Step4: About uplink data format**

The LoRa end node should upload the data with below format:

Uplink Format: **<Channel_ID>data**

For example, if we have configured 2 channels 12345 and 34567.

And there is are three LoRa End nodes sending: 12345,34567,78

The LG02 will accept the data from 12345 and 34567, it will ignore the data from Node 78

**Case 1:**

Node 12345 send <12345>field1=0.0&field2=1102.0

Node 34567 doesn't send anyting

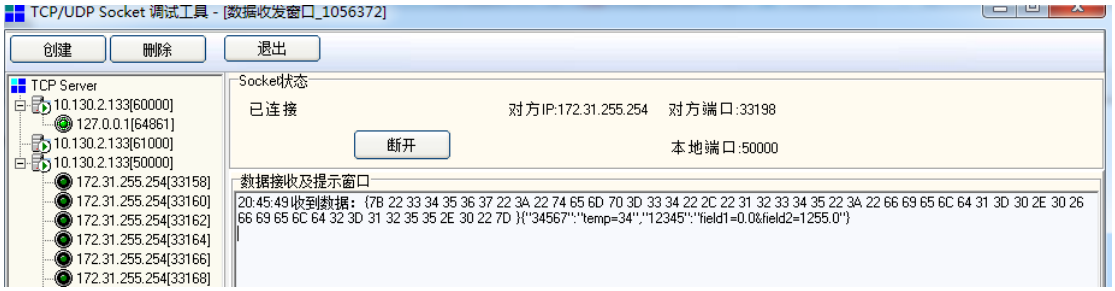
The TCP/IP server will get {"12345":"field1=0.0&field2=1102.0"}

**Case 2:**

Node 12345 send <12345>field1=0.0&field2=1102.0

Node 34567 send <34567>temp=34

The TCP/IP server will get {"34567":"temp=34","12345":"field1=0.0&field2=1102.0"}

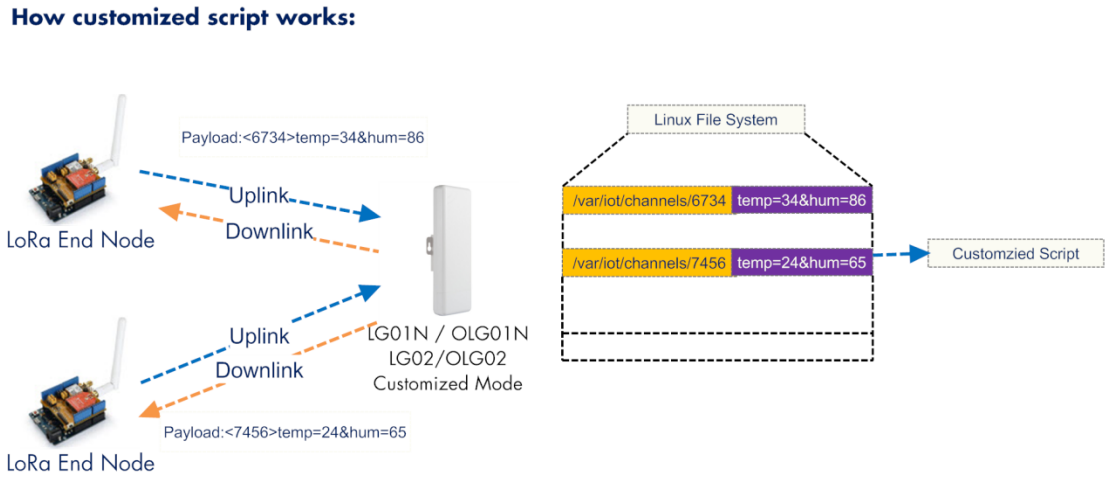



LoRa End Device reference source code: [check this link](#).

## 8. Example 5: Write a customized script

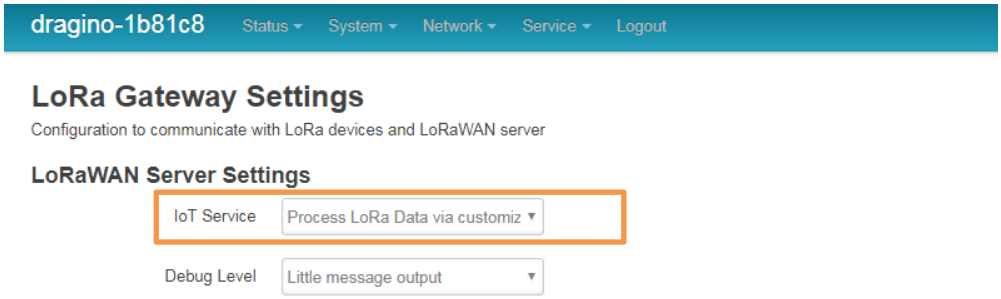
LG02 supports customized script to process LoRa data. This chapter describes about the data format from LoRa End node and How to write the script.

The data flow from LoRa End Node to LG02 is as below:

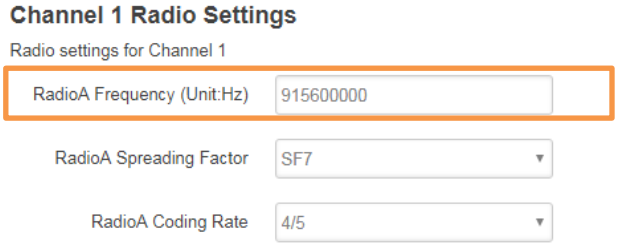



Example: Store Data in a file.

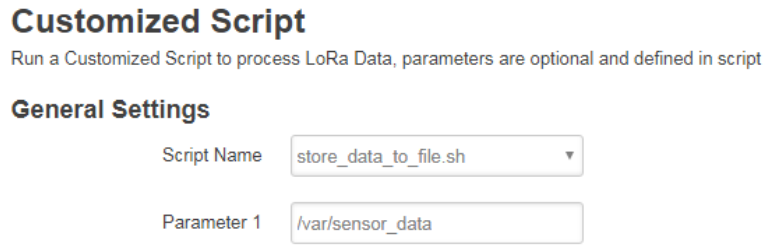
### Step 1: Choose LoRa customized script mode




### Step 2: Configure LoRa Frequency



### Step 3: Choose the customized script



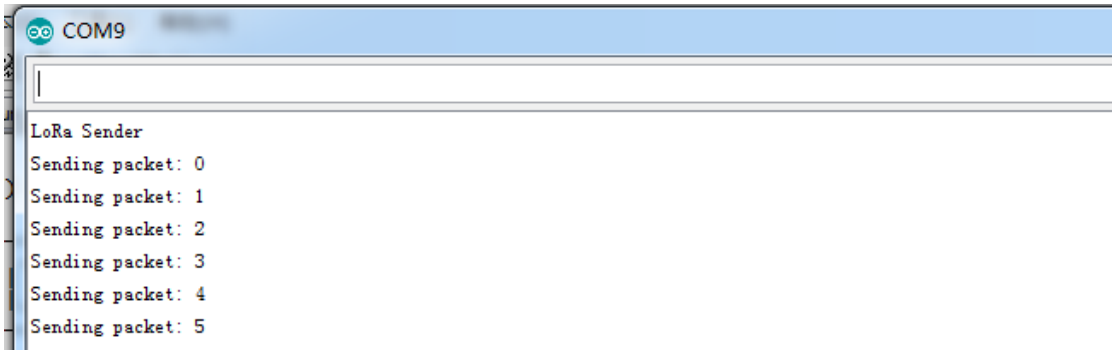

The directory to store customized script is in **/etc/lora/customized_scripts/**. User can write a new script and put it under this directory for their application. The web will auto detect it.

### Step 4: Configure the LoRa End Device to send sensor data.

Here is an example code for LoRa Shield: **End Device Code**

### Outputs:

End node send out packages:




Gateway receive packet & Script find packet

```
root@dragino-1b81c8:~# logread -f
Sun Jan  1 00:47:08 2012 user.notice root: [IoT]: Found field1=25&field2=87 at Local Channel: 10009
Sun Jan  1 00:47:08 2012 user.notice root: [IoT]: Append at /var/sensor_data
Sun Jan  1 00:47:13 2012 daemon.info lg02_pkt_fwd[31105]:
Sun Jan  1 00:47:13 2012 daemon.info lg02_pkt_fwd[31105]: RXTX~ Receive(HEX):3c31303030393e6669656c64313d3239266669656c64323d3933
Sun Jan  1 00:47:14 2012 user.notice root: [IoT]: Found field1=29&field2=93 at Local Channel: 10009
Sun Jan  1 00:47:14 2012 user.notice root: [IoT]: Append at /var/sensor_data
Sun Jan  1 00:47:23 2012 daemon.info lg02_pkt_fwd[31105]:
Sun Jan  1 00:47:23 2012 daemon.info lg02_pkt_fwd[31105]: RXTX~ Receive(HEX):3c31303030393e6669656c64313d3238266669656c64323d3934
Sun Jan  1 00:47:26 2012 user.notice root: [IoT]: Found field1=28&field2=94 at Local Channel: 10009
Sun Jan  1 00:47:26 2012 user.notice root: [IoT]: Append at /var/sensor_data
```

Script store data into file

```
root@dragino-1b81c8:~# cat /var/sensor_data
Sun Jan 1 00:15:26 UTC 2012 :<1234> 123443
Sun Jan 1 00:46:26 UTC 2012 :<10009> field1=32&field2=94
Sun Jan 1 00:46:44 UTC 2012 :<10009> field1=32&field2=94
Sun Jan 1 00:46:56 UTC 2012 :<10009> field1=28&field2=93
Sun Jan 1 00:47:08 UTC 2012 :<10009> field1=25&field2=87
Sun Jan 1 00:47:14 UTC 2012 :<10009> field1=29&field2=93
Sun Jan 1 00:47:26 UTC 2012 :<10009> field1=28&field2=94
Sun Jan 1 00:47:38 UTC 2012 :<10009> field1=25&field2=90
Sun Jan 1 00:47:44 UTC 2012 :<10009> field1=27&field2=87
Sun Jan 1 00:47:56 UTC 2012 :<10009> field1=32&field2=88
Sun Jan 1 00:48:08 UTC 2012 :<10009> field1=32&field2=94
Sun Jan 1 00:48:20 UTC 2012 :<10009> field1=25&field2=87
Sun Jan 1 00:48:26 UTC 2012 :<10009> field1=28&field2=94
Sun Jan 1 00:48:38 UTC 2012 :<10009> field1=34&field2=92
Sun Jan 1 00:48:50 UTC 2012 :<10009> field1=25&field2=88
Sun Jan 1 00:48:56 UTC 2012 :<10009> field1=34&field2=93
Sun Jan 1 00:49:08 UTC 2012 :<10009> field1=31&field2=90
Sun Jan 1 00:49:20 UTC 2012 :<10009> field1=32&field2=91
Sun Jan 1 00:49:26 UTC 2012 :<10009> field1=27&field2=92
Sun Jan 1 00:49:38 UTC 2012 :<10009> field1=25&field2=88
```

## 9. Example 6: LoRaWAN Packet Relay mode

The LG02 has two radio channels, we can use these two channels as LoRaWAN relay to improve the network coverage.

### 9.1 How it works

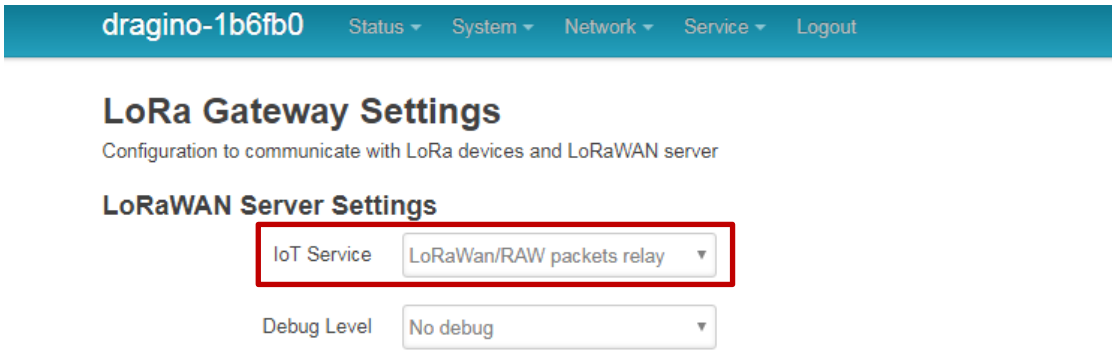
Basic concept: Get data in Radio 1 with match frequency, transmit the same data on Radio2 with pre-configure frequency.

This mode works for below LoRa protocol:

- ✓ LoRaWAN ABP mode and only be able to relay one frequency.
- ✓ Radiohead Library.
- ✓ Simple LoRa protocol.

Configuration is as below:

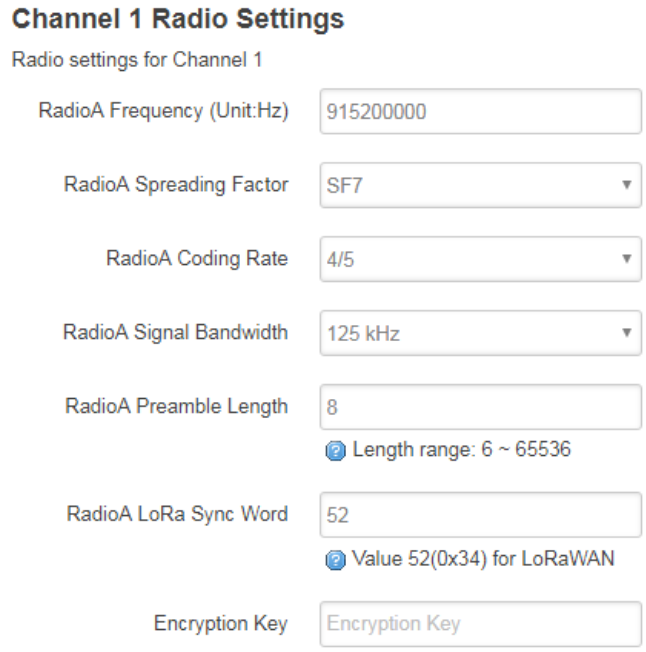
**Step 1: Choose LoRaWAN/Raw relay mode**




**Step 2: Configure receive channel info**

For example, if we want to receive all packets on:

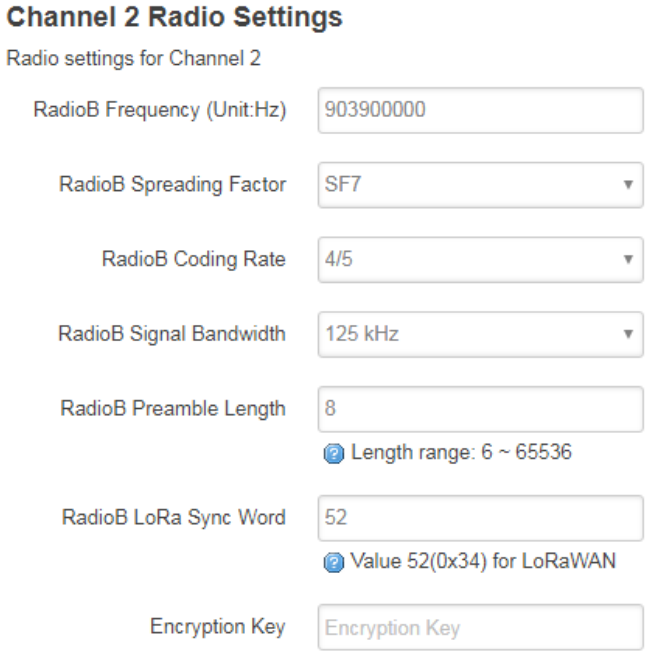
Frequency:915.2Mhz, sf=7, bw=125000, cr=5, wd=0x34,bandwidth= 125k. below is the configure.




### Step 3: Configure transmit channel info

For example, if we want to transmit all received packets on:

Frequency: 903.9Mhz, sf=7, bw=125000, cr=5, wd=0x34, bandwidth= 125k. Below is the configuration.



Now the LG02 will transfer all packets receive on Radio 1 (915.2Mhz, sf=7, bw=125000, cr=5, wd=0x34,bandwidth= 125k) to Radio 2 (903.9Mhz, sf=7, bw=125000, cr=5, wd=0x34, bandwidth= 125k)

Below is the debug info for reference:

```
root@dragino-1b6fb0:~# logread -f
Mon Dec 17 13:18:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:18:27 2018 daemon.info lg02_pkt_fwd[6370]: RXTX~ Receive(HEX):408b1c012600152402794 4e22ef8b78848a1005b60
Mon Dec 17 13:18:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:18:27 2018 daemon.info lg02_pkt_fwd[6370]: INFO~Transmit at SF7BW125 on 903.900000.
Mon Dec 17 13:19:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:19:27 2018 daemon.info lg02_pkt_fwd[6370]: RXTX~ Receive(HEX):408b1c012600162402f115475169ebb39c23b278d4
Mon Dec 17 13:19:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:19:27 2018 daemon.info lg02_pkt_fwd[6370]: INFO~Transmit at SF7BW125 on 903.900000.
Mon Dec 17 13:20:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:20:27 2018 daemon.info lg02_pkt_fwd[6370]: RXTX~ Receive(HEX):408b1c012600172402b10a8a261da8a7c70220c76f
Mon Dec 17 13:20:27 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:20:27 2018 daemon.info lg02_pkt_fwd[6370]: INFO~Transmit at SF7BW125 on 903.900000.
Mon Dec 17 13:20:30 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:20:30 2018 daemon.info lg02_pkt_fwd[6370]: RXTX~ Receive(HEX):f08c5484c0b4e9575b3e0926ee0da8a8082b0b6a31e51079d9813010688a
Mon Dec 17 13:20:30 2018 daemon.info lg02_pkt_fwd[6370]:
Mon Dec 17 13:20:30 2018 daemon.info lg02_pkt_fwd[6370]: INFO~Transmit at SF7BW125 on 903.900000.
Mon Dec 17 13:20:32 2018 daemon.info lg02_pkt_fwd[6370]: INFO~ Exiting Lora service program
```

## 9.2 Limitation

While use this mode to enlarge the LoRaWAN coverage, there is limitation because the LG02 only listen one Frequency and DR, while the LoRaWAN end node transmit the data on multiply frequency and different DR. For example, in EU868 LoRaWAN, the end node LoRaWAN settings are:

**EU863-870**

Uplink:

868.1 - SF7BW125 to SF12BW125

868.3 - SF7BW125 to SF12BW125 and SF7BW250

868.5 - SF7BW125 to SF12BW125

867.1 - SF7BW125 to SF12BW125

867.3 - SF7BW125 to SF12BW125

867.5 - SF7BW125 to SF12BW125

867.7 - SF7BW125 to SF12BW125

867.9 - SF7BW125 to SF12BW125

868.8 – FSK

Downlink:

Uplink channels 1-9 (RX1)

869.525 - SF9BW125 (RX2 downlink only)

**For uplink**: if we set LG02 to listen on 868.3 , SF7, BW125, and transmit at 868.3,SF7,BW125. The LG02 will receive all end node's packet at 868.3 SF7BW125 and can't receive other frequency from end node. Which means the LG02 can only extend the uplink for 868.3 SF7BW125.

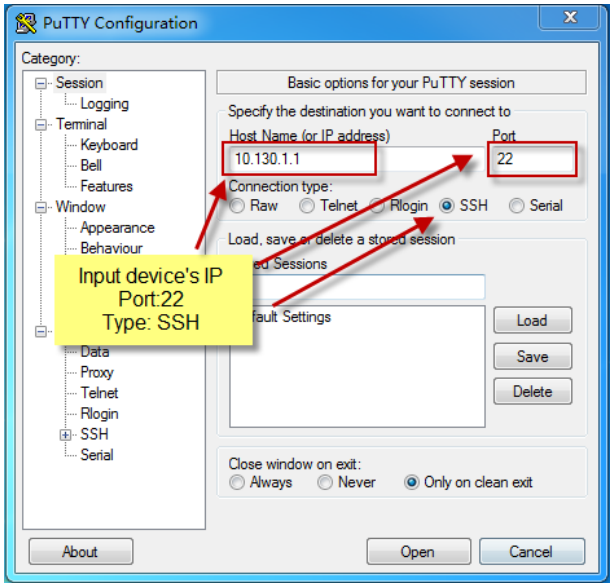
**For downlink**: Due to IQ inverter issue, the end node can't get relay packet from LG02. So OTAA mode can't work. User need to use ABP mode.

## 10. Linux System

The LG02 bases on OpenWrt Linux System. It is open source, and user are free to configure and modify the inside Linux settings.

### 10.1 SSH Access for Linux console

User can access to the Linux console via SSH protocol. Make sure your PC and the LG01 is in the same network, then use a SSH tool (such as putty) to access it. Below are screenshots:



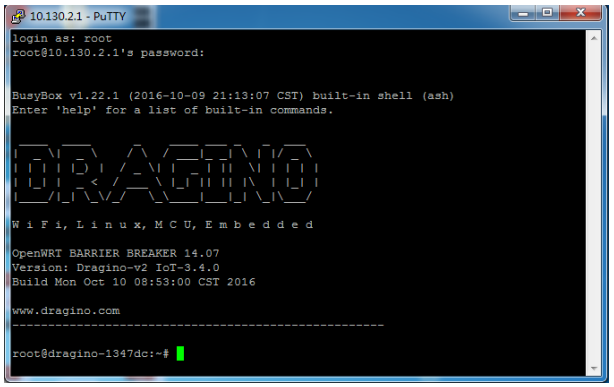

IP address: IP address of LG02

Port: 22

User Name: **root**

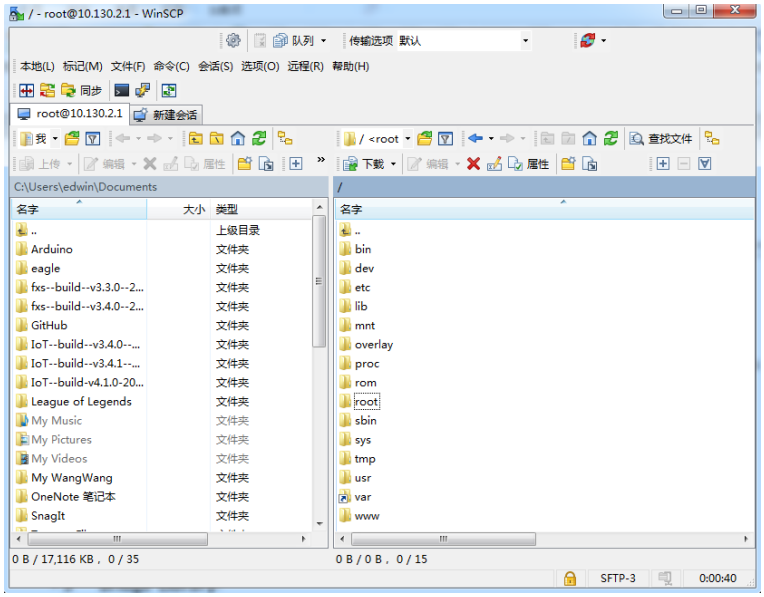
Password: **dragino** (default)

After log in, you will be in the Linux console and type command here.



## 10.2 Edit and Transfer files

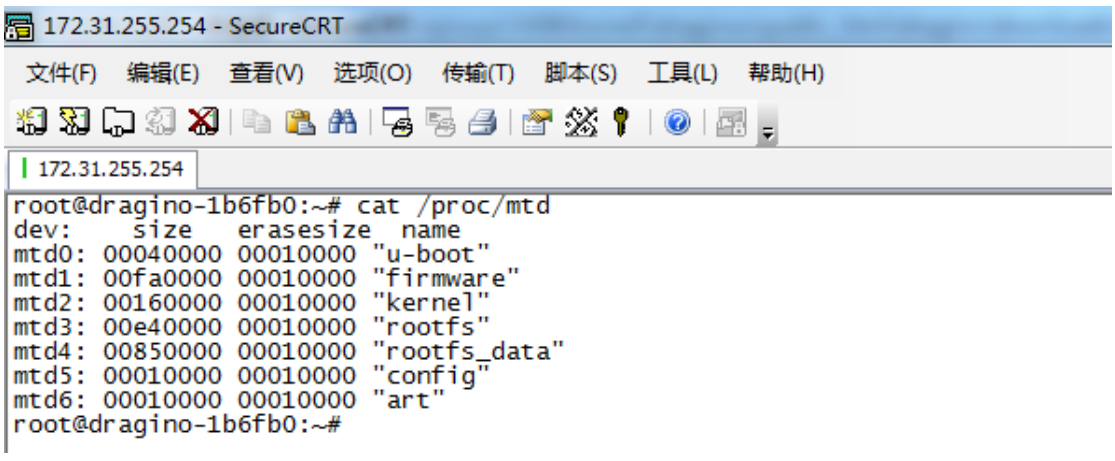
The LG02 support **SCP protocol** and has a built **SFTP server**. There are many ways to edit and transfer files using these two protocols. One of the easiest is through WinSCP utility. After access via WinSCP to the device, use can use a FTP alike window to drag / drop files to the LG02 or Edit the files directly in the windows. Screenshot is as below:




## 10.3 File System

The LG02 has a 16MB flash and a 64MB RAM. The /var and /tmp directory are in the RAM, contents stored in /tmp and /var will be erased after reboot the device. Other directories are in the flash and will keep after reboot.

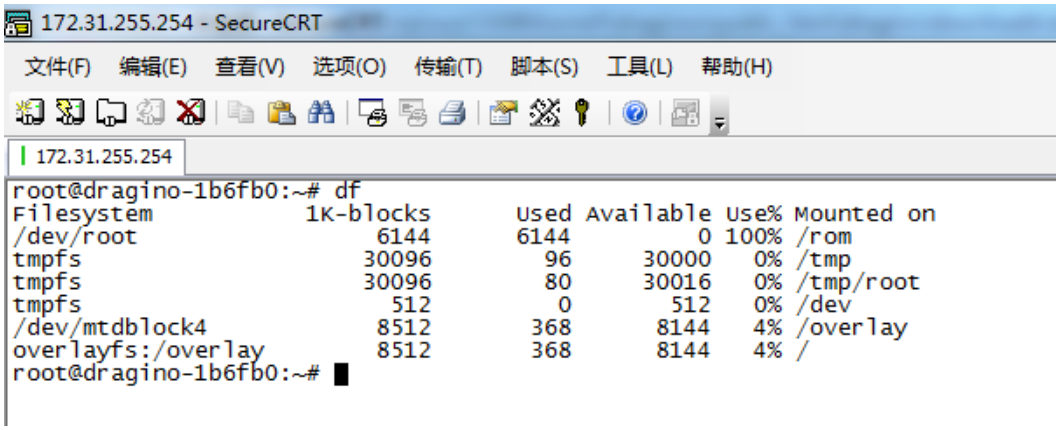
**Use cat /proc/mtd to see all blocks/partitions.**




- ✓ "u-boot" // for boot-loader
- ✓ "firmware" // combination of kernel & rootfs
- ✓ "kernel" // Linux kernel
- ✓ "rootfs" // Linux rootfs

- ✓ "rootfs_data" //inside rootfs, all data store here.
- ✓ "config" // a separate partition doesn't include file system
- ✓ "art" // radio data and board ID.

**Use df command to see available flash & RAM:**




**tmpfs 30096 96 30000 0% /tmp // RAM: reset after reboot,**

**/dev/mtdblock4 8512 368 8144 4% /overlay //Flash: Remain after reboot**

**Reset to factory default:**

mtd erase rootfs_data -r

Except /tmp and /var. all data will be store in flash. /tmp and /var are store in RAM

## 10.4 Package maintain system

LG02 uses **OPKG package maintain system**. There are more than 3000+ packages available in our package server for user to install for their applications. For example, if user wants to add iperf tool, they can install the related packages and configure LG02 to use iperf

Below is some examples opkgs command, more please refer **OPKG package maintain system**

In Linux Console run:

**root@dragino-169d30:~# opkg update** // to get the latest packages list

**root@dragino-169d30:~# opkg list** //shows the available packages

**root@dragino-169d30:~# opkg install iperf** // install iperf, it will auto install the required packages.

*root@dragino-169d30:/etc/opkg# opkg install iperf*

*Installing iperf (2.0.12-1) to root...*

*Downloading http://downloads.openwrt.org/snapshots/packages/mips_24kc/base/iperf_2.0.12-1_mips_24kc.ipk*

*Installing uclibcxx (0.2.4-3) to root...*

*Downloading*

*http://downloads.openwrt.org/snapshots/packages/mips_24kc/base/uclibcxx_0.2.4-3_mips_24kc.ipk*

*Configuring uclibcxx.*

*Configuring iperf.*

## 11. Upgrade Linux Firmware

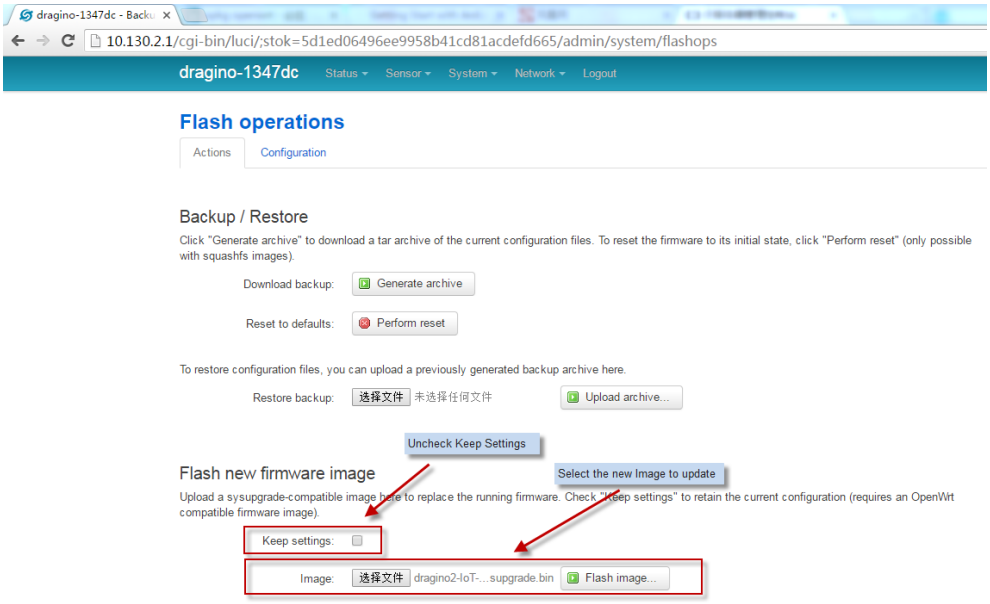
We keep improving the LG02 Linux side firmware for new features, bug fixes. The latest firmware can be found on LG02 Firmware & release note

The file named as **dragino-LG02_LG08----xxxxx-squashfs-sysupgrade.bin** is the upgrade Image. There are different methods to upgrade, as below:

### 11.1 Upgrade via Web UI

Go to the page: **Web --> System --> Back Up and flash firmware,** Select the image and click Flash Image, the image will be uploaded to the device and then click Process Update to upgrade.

System will auto boot to the new firmware after upgrade.




### 11.2 Upgrade via Linux console

SCP the firmware to the system **/var** directory and then run

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

note: it is important to transfer the image in the /var directory, otherwise it may exceed the flash size.

# 12. FAQ

## 12.1 Why there is 433/868/915 version LoRa part?

Different country has different rules for the ISM band for using the LoRa. Although the LoRa chip can support a wide range of Frequency, we provide different version for best tune in the LoRa part. That is why we provide different version of LoRa.

## 12.2 What is the frequency range of LG02 LoRa part?

The chip used in the LoRa part is:

| Version | LoRa IC | Support Frequency | Best Tune Frequency |
|---|---|---|---|
| **433** | Semtech SX1278 | Band2(LF): 410 ~525Mhz<br>Band3(LF): 137 ~175Mhz | 433Mhz |
| **868** | Semtech SX1276 | Band1(HF): 862 ~1020Mhz | 868Mhz |
| **915** | Semtech SX1276 | Band1(HF): 862 ~1020Mhz | 915Mhz |

User can set the LoRa within above frequency range in the software.

## 12.3 What does “Limited support on LoRaWAN”?

The base requirement to fully compatible with LoRaWAN protocol requires the gateway support 8 channels. The LG02 only support two channels and can only support limited LoRaWAN protocol. Below are limitations:

**1/ It works only on one frequency at a time**. It can support multiply end nodes, but all end nodes must transmit data at the same frequency so the LG02 can receive it. For example: if the End node transmits at 868.1Mhz, The LG02’s RX setting must be 868.1Mhz so to receive this packet.

**2/ It works only for one DR at a time.** DR specifies the Spreading Factor and Bandwidth. In LG02, even the rx frequency match , if DR doesn’t match, it still can’t get the sensor data.

**3/ LoRaWAN compatible issue**

In LoRaWAN protocol, the LoRaWAN end nodes send data in a hopping frequency. Since LG02 only supports one single frequency, it will only be able to receive the packets sent from the same radio parameters (frequency & DR) in LG02.

For example, in EU868, a standard LoRaWAN device may send the data in eight frequencies with different Frequency & SF, such as:

```
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);       // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
```

```
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7),    BAND_CENTI);       // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK,  DR_FSK),    BAND_MILLI);       // g2-band
```

So the LG02 will only able to receive the 868100000, SF7 packet and will not receive others. Means only one packet will arrive the TTN server in every 8 packet sent from the LoRaWAN end node.

If user wants to receive all packets from LoRaWAN end node, user needs to set up the LoRaWAN node to send packets in a single frequency.

**4/ Downlink & OTAA issue**

According to the LoRaWAN class A spec, the end node will open two receive windows to get the message from LoRaWAN server for OTAA or downlink function. These two receive windows are quite short (milliseconds), if LoRa packet from the gateway can't reach End Node in the receive window time, the end node won't get the rx message and Downlink / OTAA won't work.

In our LoRaWAN example, the Arduino LMIC library is modified to enlarge the RX window to let OTAA & downlink works.

### 12.4 Can I develop my own application for LG02?

Yes, the fastest way to develop own software is through the SDK. The instruction is here:

https://github.com/dragino/openwrt_lede-18.06/blob/master/README.md#how-to-develop-a-c-software-before-build-the-image

### 12.5 Can I make my own firmware for LG02? Where can I find the source code of LG02?

Yes, User can make own firmware for LG02 for branding purpose or add customized application.

The LG012source code and compile instruction can be found at:

https://github.com/dragino/openwrt_lede-18.06

### 12.6 Can I change the LoRa behavior of LG02?

Yes, Developer can change the LoRa control behavior of LG02. The source code of the LoRa controller in LG02 is here: https://github.com/dragino/dragino-packages/tree/lg02/lg02-pkt-fwd

And the compile method can refer: Compile an application.

### 12.7 On OTAA mode, if I use the other frequency, how should I modify in the library?

In page OTAA, We use frequency 904.6Mhz for sending. According the LoRaWAN protocol, if the device Join the network successfully, the server will downlink the reply. The different intervals of frequency, the receiving frequency of the end node is also different.

Ex1: We use 914.2Mhz frequency.

We can input the command: logread -f

```
Wed Sep 12 01:39:19 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:19 2018 daemon.info lg02_pkt_fwd[14341]: INFO (json): [down] {"txpk":{"imme":false,"tmst":2831770149,"freq":927.5,"rfch":0,"powe":20,"modu":"LORA","da
tr":"SF7BW500","codr":"4/5","ipol":true,"size":17,"ncrc":true,"data":"I1adG+uy4Y+L7RAfx5hIxOA="}}
Wed Sep 12 01:39:19 2018 daemon.info lg02_pkt_fwd[14341]: SF=0x07
Wed Sep 12 01:39:19 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:19 2018 daemon.info lg02_pkt_fwd[14341]: Transmit at SF7BW500 on 927.500000.
Wed Sep 12 01:39:20 2018 daemon.info lg02_pkt_fwd[14341]: SF=0x07
Wed Sep 12 01:39:20 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:20 2018 daemon.info lg02_pkt_fwd[14341]: Transmit at SF7BW500 on 927.500000.
Wed Sep 12 01:39:20 2018 daemon.info lg02_pkt_fwd[14341]: Donwlink done: count_us=2831770149
Wed Sep 12 01:39:21 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:21 2018 daemon.info lg02_pkt_fwd[14341]: INFO (json): [down] {"txpk":{"imme":false,"tmst":2833763738,"freq":927.5,"rfch":0,"powe":20,"modu":"LORA","da
tr":"SF7BW500","codr":"4/5","ipol":true,"size":17,"ncrc":true,"data":"IOGNtMK9p5Y1jF9BP1xbZvI="}}
Wed Sep 12 01:39:21 2018 daemon.info lg02_pkt_fwd[14341]: SF=0x07
Wed Sep 12 01:39:21 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:21 2018 daemon.info lg02_pkt_fwd[14341]: Transmit at SF7BW500 on 927.500000.
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]: SF=0x07
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]: Transmit at SF7BW500 on 927.500000.
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]: Donwlink done: count_us=2833763738
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]:
Wed Sep 12 01:39:22 2018 daemon.info lg02_pkt_fwd[14341]: Receive(HEX):40ad2a0126800000010a2fd88ae57fa9451d478e5a1e693d8b
```

We should modify this on <lorabase.h>, save and re-upload the sketch.

```
enum { US915_125kHz_UPFBASE = 914200000,
       US915_125kHz_UPFSTEP =    0,
       US915_500kHz_UPFBASE = 902320000,
       US915_500kHz_UPFSTEP =   0,
       US915_500kHz_DNFBASE = 927500000,   //receive
       US915_500kHz_DNFSTEP =    0
};
```

modify this frequency.

For the result:

| | | | | |
|---|---|---|---|---|
| 10:06:25 | 116 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:06:11 | 115 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:05:57 | 114 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:05:43 | 113 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:05:29 | 112 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |

## Ex2: We use 903.0Mhz frequency

We can input the command: logread -f

```
root@dragino-19a944:~# logread -f
Wed Sep 12 02:11:31 2018 daemon.info lgo2_pkt_fwd[20677]:
Wed Sep 12 02:11:31 2018 daemon.info lgo2_pkt_fwd[20677]: INFO (json): [down] {"txpk":{"imme":false,"tmst":468442152,"freq":923.3,"rfch":0,"powe":20,"modu":"LORA","datr":"SF7BW500","codr":"4/5","ipol":true,"size":17,"ncrc":true,"data":"IGlkYOueyJXLoMTFSovbRBg="}}
Wed Sep 12 02:11:31 2018 daemon.info lgo2_pkt_fwd[20677]: SF=0x07
Wed Sep 12 02:11:31 2018 daemon.info lgo2_pkt_fwd[20677]:
Wed Sep 12 02:11:31 2018 daemon.info lgo2_pkt_fwd[20677]: Transmit at SF7BW500 on 923.300000.
Wed Sep 12 02:11:32 2018 daemon.info lgo2_pkt_fwd[20677]: SF=0x07
Wed Sep 12 02:11:32 2018 daemon.info lgo2_pkt_fwd[20677]:
Wed Sep 12 02:11:32 2018 daemon.info lgo2_pkt_fwd[20677]: Transmit at SF7BW500 on 923.300000.
Wed Sep 12 02:11:32 2018 daemon.info lgo2_pkt_fwd[20677]: Donwlink done: count_us=468442152
Wed Sep 12 02:11:36 2018 daemon.info lgo2_pkt_fwd[20677]:
Wed Sep 12 02:11:36 2018 daemon.info lgo2_pkt_fwd[20677]: Receive(HEX):00ac2301d07ed5b370907cb65d67c64a00cd3586bb5c88
Wed Sep 12 02:11:36 2018 daemon.info lgo2_pkt_fwd[20677]:
Wed Sep 12 02:11:36 2018 daemon.info lgo2_pkt_fwd[20677]: INFO (JSON): [up] {"rxpk":[{"time":"2018-09-12T02:11:36.210520Z","tmst":472538265,"chan":0,"rfch":1,"freq":903.000000,"stat":1,"modu":"LORA","datr":"SF7BW125","codr":"4/5","lsnr":7.8,"rssi":-34,"size":23,"data":"AKwjAdB+1bNwkHy2XwfGSqDNNYa7XIq="}]}
```

| | | | | |
|---|---|---|---|---|
| 10:13:33 | 1 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:13:21 | | 0 | | |
| 10:13:20 | 0 | 1 retry | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:13:15 | | | dev addr: 26 01 20 71 | app eui: 70 B3 D5 7E D0 01 23 AC dev eui: 00 4A C6 67 5D B6 7C 90 |

If join the network successfully, it will send a reply.

We should modify this on <lorabase.h>, save and re-upload the sketch.

```
enum { US915_125kHz_UPFBASE = 903000000,
       US915_125kHz_UPFSTEP =    0,
       US915_500kHz_UPFBASE = 902320000,
       US915_500kHz_UPFSTEP =   0,
       US915_500kHz_DNFBASE = 923300000,   //receive
       US915_500kHz_DNFSTEP =    0
};
```

For the result:

| | | | | |
|---|---|---|---|---|
| 10:16:57 | 16 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:16:43 | 15 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:16:29 | 14 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:16:15 | 13 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:16:01 | 12 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |
| 10:15:47 | 11 | 1 | payload: | 68 65 6C 6C 6F 20 77 6F 72 6C 64 21 |

## 12.8 How can I reset the device to factory default?

User can reset the device to factory default in different ways:

Method 1:

Reset via Web UI. Click the button in Web UI --> System --> Back up / Flash firmware --> Perform Reset

Method 2:

Reset in Linux console, command is below:

root@dragino-1b8288:~# firstboot

This will erase all settings and remove any installed packages. Are you sure?

[N/y]

y

/dev/mtdblock4 is mounted as /overlay, only erasing files

root@dragino-1b8288:~# reboot

## 12.9 Can I control the LEDs?

Except the PWR LED and sensor LED, all other LEDs can be controlled by developer.

**Control Globe LED:**

ON: echo 1 > /sys/class/leds/dragino2\:red\:wlan/brightness

OFF: echo 0 > /sys/class/leds/dragino2\:red\:wlan/brightness

## 12.10 More FAQs about general LoRa questions

We keep updating more FAQs in our WiKi about some general questions. The link is here:

http://wiki.dragino.com/index.php?title=LoRa_Questions

# 13. Trouble Shooting

## 13.1 I get kernel error when install new package, how to fix?

In some case, when install package, it will generate kernel error such as below:

```
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.
```

In this case, user can use the –force-depends option to install such package.

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

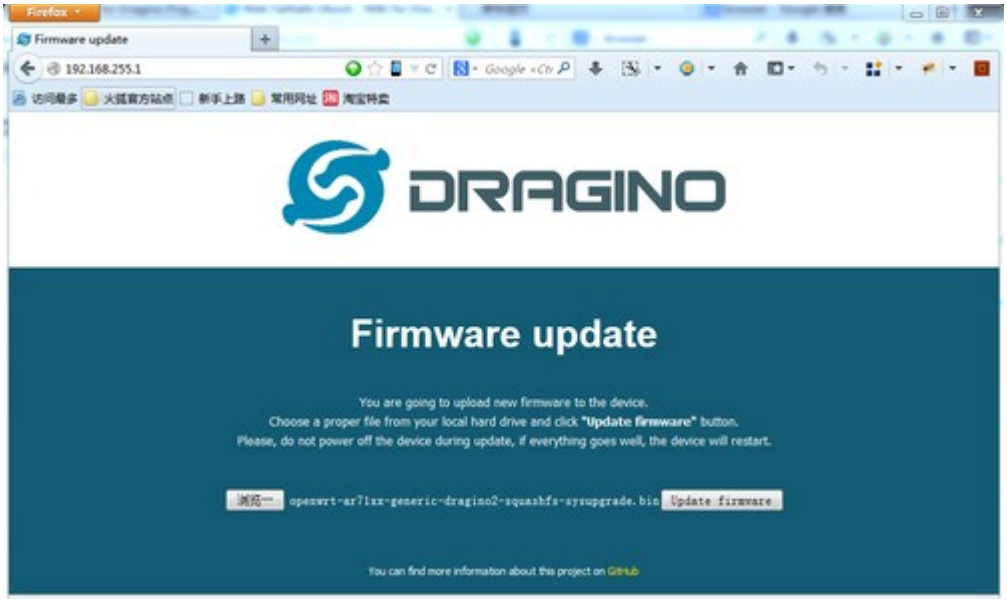
## 13.2 How to recover the LG02 if firmware crash

LG012provide user a full control on its Linux system, it is possible that the device will brick and can't boot after improper modification in some booting files.

In this case, user can recover the whole Linux system by uploading a new firmware via Web Failsafe mode.

Procedure is as below:

1. Use a RJ45 cable to connect the PC to LG02's port directly.
2. Set the PC to ip 192.168.255.x, netmask 255.255.255.0
3. Pressing the toggle button and power on the device
4. All LEDs of the device will blink, release the toggle button after four blinks
5. All LEDs will then blink very fast once, this means device detect a network connection and enter into the web-failsafe mode. Your PC should be able to ping 192.168.255.1 after device enter this mode.
6. Open 192.168.255.1 in web broswer
7. Select a squashfs-sysupgrade type firmware and update firmware.




Note: If user sees all LEDs blink very fast in Step 5. This means the network connection is established. If in this case, PC still not able to see the web page, user can check:

- ✓ Try different browser.
- ✓ Check if your PC is in 192.168.255.x
- ✓ Check if you have connected two RJ45 cable to device, If so, remove the unused one

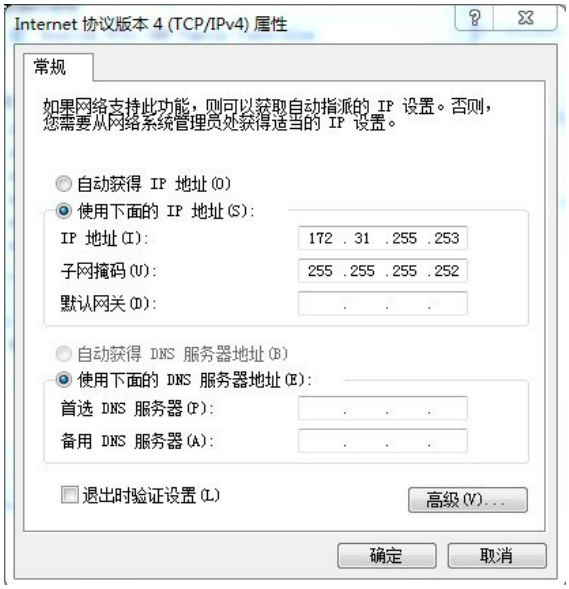
## 13.3 I configured LG02 for WiFi access and lost its IP. What to do now?

The LG01 has a fall-back ip in its LAN port. This IP is always enabled so user can use fall back ip to access LG01 no matter what the WiFi IP is. The fall back ip is useful for connect and debug the unit.

(Note: fallback ip can be disabled in the LAN and DHCP page)

Steps to connect via fall back IP:

1. Connect PC's Ethernet port to LG01's LAN port
2. Configure PC's Ethernet port has IP: 172.31.255.253 and netmask: 255.255.255.252
   As below photo:




3. In PC, use 172.31.255.254 to access LG01 via Web or Console.

## 14. Order Info

***PART:***

- ✓ ***Indoor version LG02-XXX-YYY***
- ✓ ***Outdoor version: OLG02-XXX-YYY***

***XXX: Frequency Band***

- **433**: For Bands: EU433, CN470.
- **868**: For Bands: EU868,IN865.
- **915**: For Bands: US915,AU915,AS923,KR920.

***YYY: 4G Cellular Option***

- **EC25-E**: EMEA, Korea, Thailand, India.
- **EC25-A**: North America/ Rogers/AT&T/T-Mobile.
- **EC25-AU**: Latin America, New Zeland, Taiwan
- **EC25-J**: Japan, DOCOMO/SoftBank/ KDDI

More info about valid bands, please see EC25-E product page.

## 15. Packing Info

**Package Includes**:

- ✓ LG02 or OLG02 LoRa Gateway x 1
- ✓ Stick Antenna for LoRa RF part. Frequency is one of 433 or 868 or 915Mhz depends the model ordered
- ✓ Power Adapter: EU/AU/US type power adapter depends on country to be used
- ✓ Packaging with environmental protection paper box

**Dimension and weight**:

- ✓ Device Size: 12 x 8.5 x 3 cm
- ✓ Device Weight: 150g
- ✓ Package Size / pcs : 21.5 x 10 x 5 cm
- ✓ Weight / pcs : 360g
- ✓ Carton dimension: 45 x 31 x 34 cm. 36pcs per carton
- ✓ Weight / carton : 12.5 kg

## 16. Support

- ➢ Try to see if your questions already answered in the wiki.
- ➢ Support is provided Monday to Friday, from 09:00 to 18:00 GMT+8. Due to different timezones we cannot offer live support. However, your questions will be answered as soon as possible in the before-mentioned schedule.

- Provide as much information as possible regarding your enquiry (product models, accurately describe your problem and steps to replicate it etc) and send a mail to

support@dragino.com

## 17. Reference

- Source code for LG02 LoRa Gateway
  https://github.com/dragino/openwrt_lede-18.06

- OpenWrt official Wiki
  http://www.openwrt.org/

- Download of this manual or Update version
  http://www.dragino.com/downloads/index.php?dir=UserManual/LG02_OLG02/

- LMIC library for Arduino LoRaWAN end device use with LG02.
  https://github.com/dragino/arduino-lmic

- Hardware Source
  LG02 = IBB + LoRa G Dual
  OLG02= PAN v2.0 + LoRa G Dual