MQTT Forward Instruction

From Wiki for Dragino Project

Contents

INTRODUCTION

Dragino LoRa/LoRaWAN gateway support MQTT forwarding. It can forward the sensor data from LoRa network to MQTT server , and vice verse.

Support Devices

This MQTT forward instruction is for below devices:

- Firmware Version > LG02_LG08-5.3.1580178039 Firmware Download (http://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/Firmware/Release/)
- LG01N, OLG01N (Warning: LG01-P LG01-S use another instruction: MQTT for LG01-P/LG01S)
- LG02, OLG02
- LG308, DLOS8
- LPS8
- MS14 series if installed with the same firmware. (in this case, the MQTT forward will work , but no LoRa support)

Firmware Change Log for MQTT feature

This instruction is wrote start from LG02_LG08-5.3.1580178039. Below is related change log since this version of firmware.

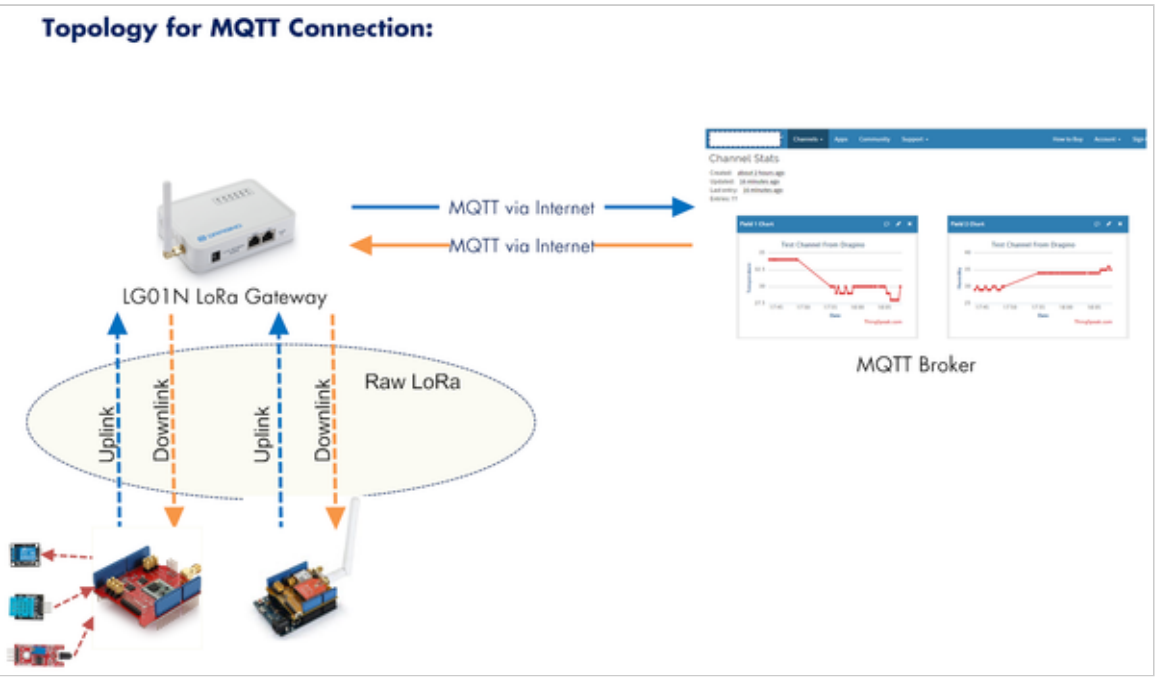
- LG02_LG08-5.3.1580178039
  - Initiate version

MQTT forward operating principle

Network Structure

Below shows the network structure for MQTT forwarding.

- For Uplink: The sensor sends data to LoRa Gateway via LoRa wireless, The gateway will process these data and forward to remote MQTT Broker via Internet.
- For Downlink: The gateway subscribe a topic in the MQTT broker, when there is update on the topic, the gateway will know and broadcast the data to Local LoRa network,



General MQTT structure

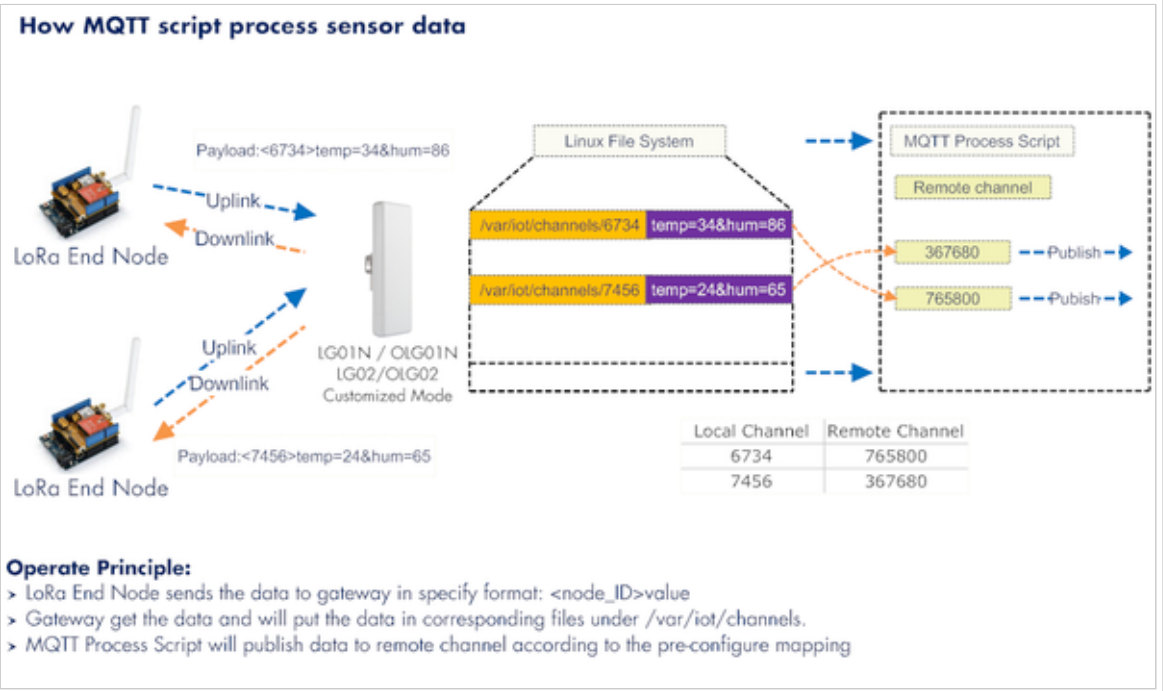
How sensor data is forwarded

In this MQTT forward feature, the key point is how the gateway process the sensor data.

Upstream

Assume there are two sensor nodes, their ID are Node1 ID: 6734 , Node2 ID: 7456. In the remote MQTT broker there are two topics: Topic1: /channel/765800, Topic2: /channel/367860. We can set up in the gateway to map Node1 to Topic1 and Node2 to Topic2. So when there is a sensor data from Node1, the gateway will forward the data to Topic1, when there is sensor data from Node2, the gateway will forward to Topic2.

The data flow works as below diagram.



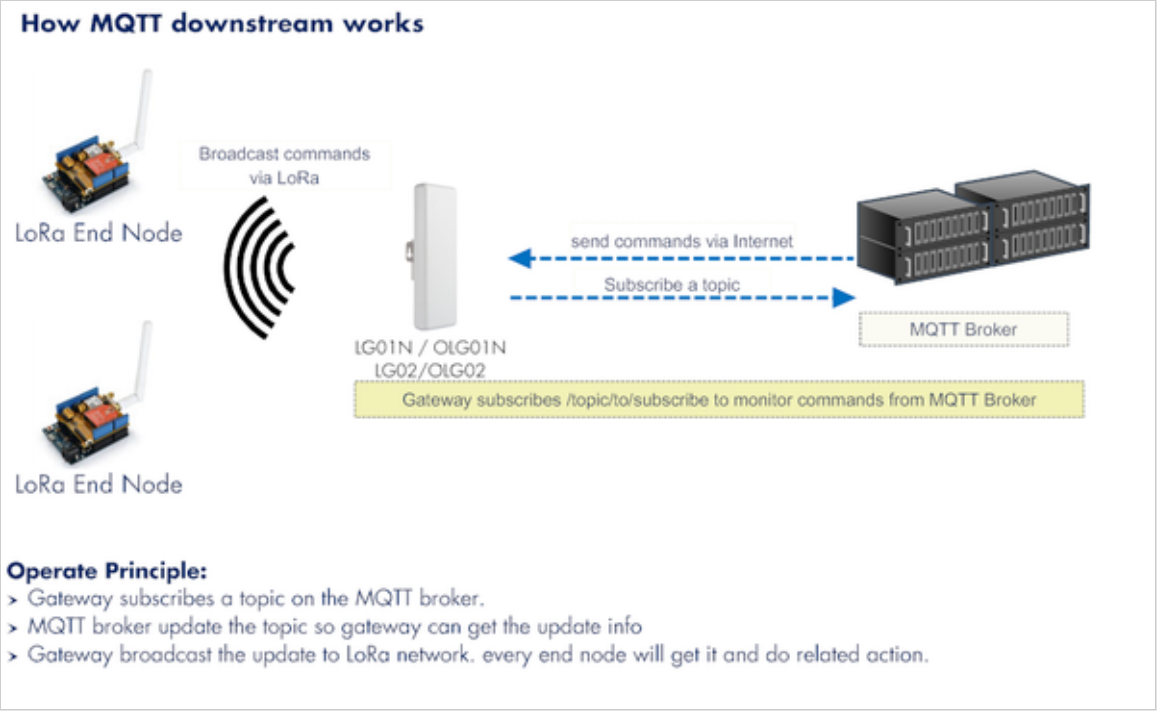
Upstream path

Note: The sensor data can base or LoRa or other method, as long as there are data on the file /var/iot/channels. /span>

Downstream

The gateway subscribes to a topic of the remote MQTT broker topic. When there is some one publish a value on this topic. The gateway will get it and broadcast to local LoRa Network.

Below are the data flow for downstream.



Downstream path

Macro Definition

The MQTT publish command use Macro settings to generate flexible upstream payload for MQTT publish.

Currently the -t (topic) and -m (message) support Macros.

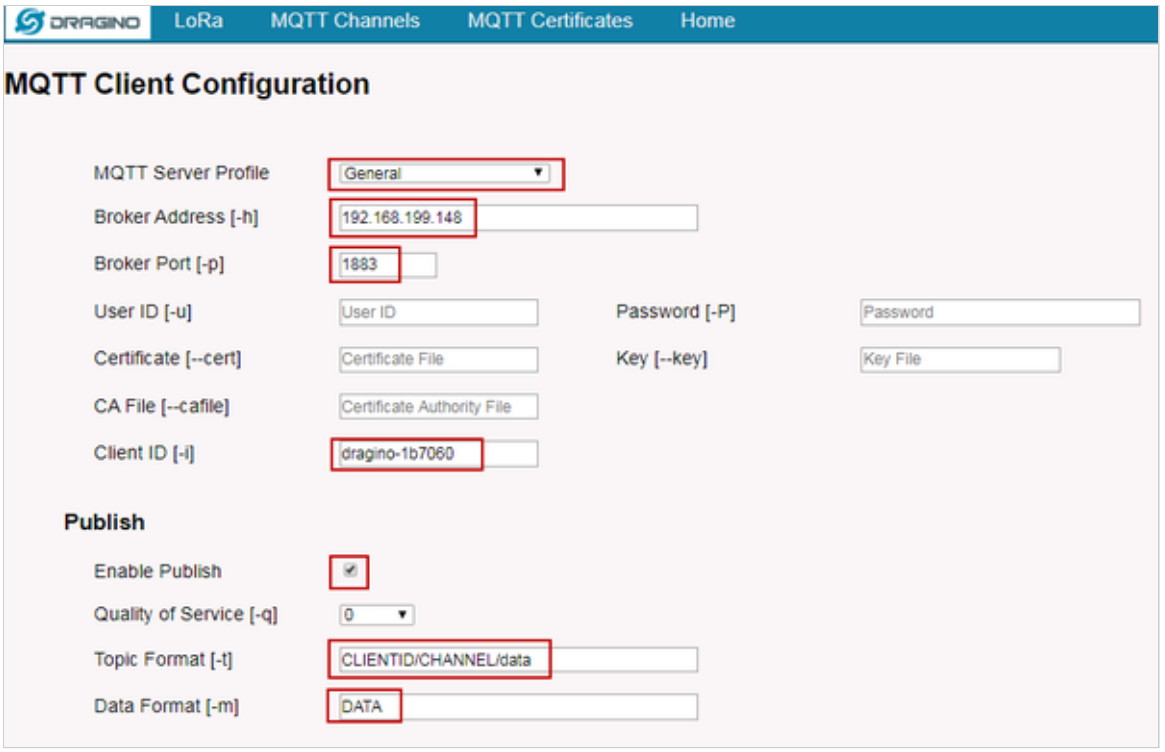
-t topic macro

- CHANNEL: Remote Channel ID
- CLIENTID: Client ID , Same as -i
- WRITE_API: Remote Channel Write API
- USERNAME: User ID (-u)
- HOSTNAME: Device Hostname

-m message macro

- HOSTNAME: Device Hostname
- CHANNEL: Remote Channel ID
- DATA: Sensor Data without time stamp and rssi
- META: Completely sensor data with time stamp and rssi
- JSON: Convert META to json format.

Example for Macro

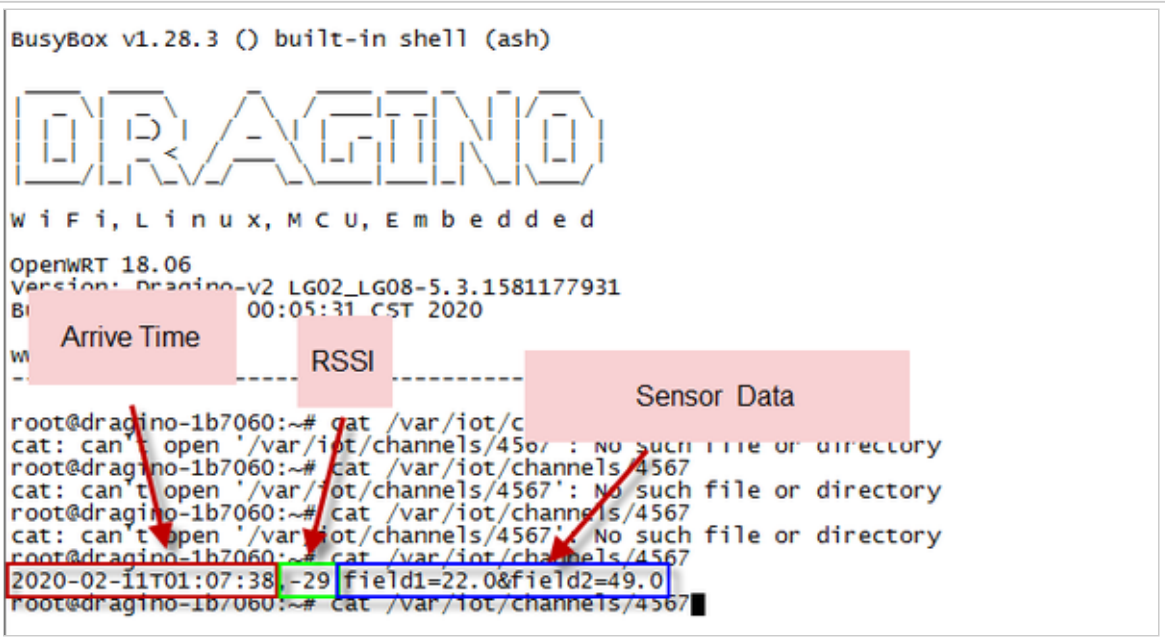


MQTT Publish configure

Above screen shots shows below format:

- -t: CLIENTID/CHANNEL/data
- -m: DATA

When there is a LoRa sensor arrive. it will be store at the /var/iot/channels as below:



Sensor Data

According to above macro. Gateway will publish field1=22.0&field2=49.0 to topic: dragino-1b7060/78901/data, where 78901 is the remote channel for this node ID.

Example to communicate to a simple MQTT server

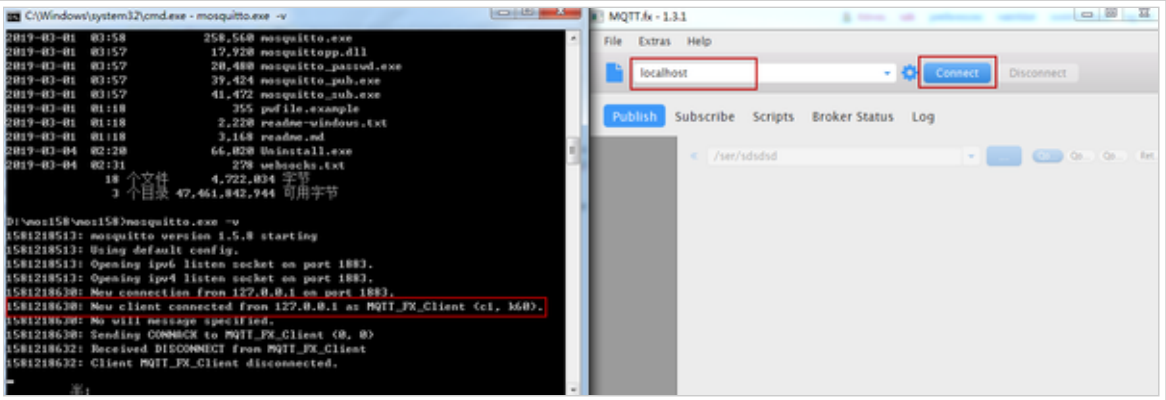
Overview

This section is an example to show how to set up LG01-N to communicate with a MQTT server. The MQTT server is a simple utility set up in a local PC. Note: User can set up same server via this instruction (http://www.steves-internet-guide.com/install-mosquitto-broker/).

Simulate via MQTT.fx utility

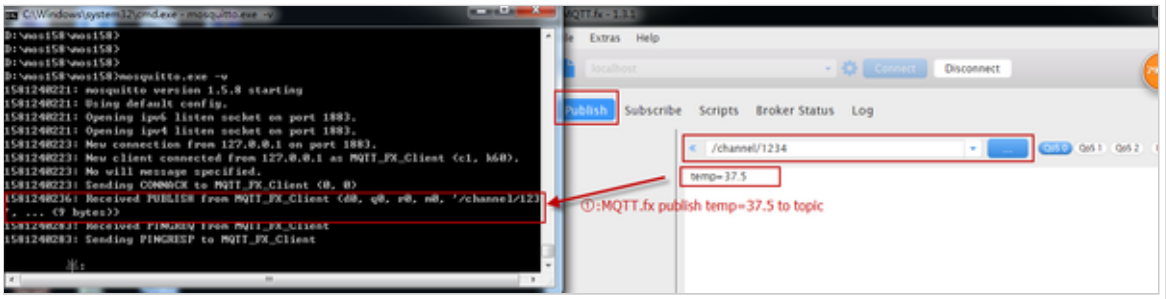
The MQTT.fx (http://mqttfx.jensd.de/index.php/download) is a MQTT client tool. We can use this to simulate a MQTT connection to our MQTT broker first to make sure the MQTT broker works. This will also help us understand how it works.

In this test, the MQTT broker and MQTT.fx are installed in the same PC, so the MQTT server address in MQTT.fx should be localhost. Below shows how to connect to the server.



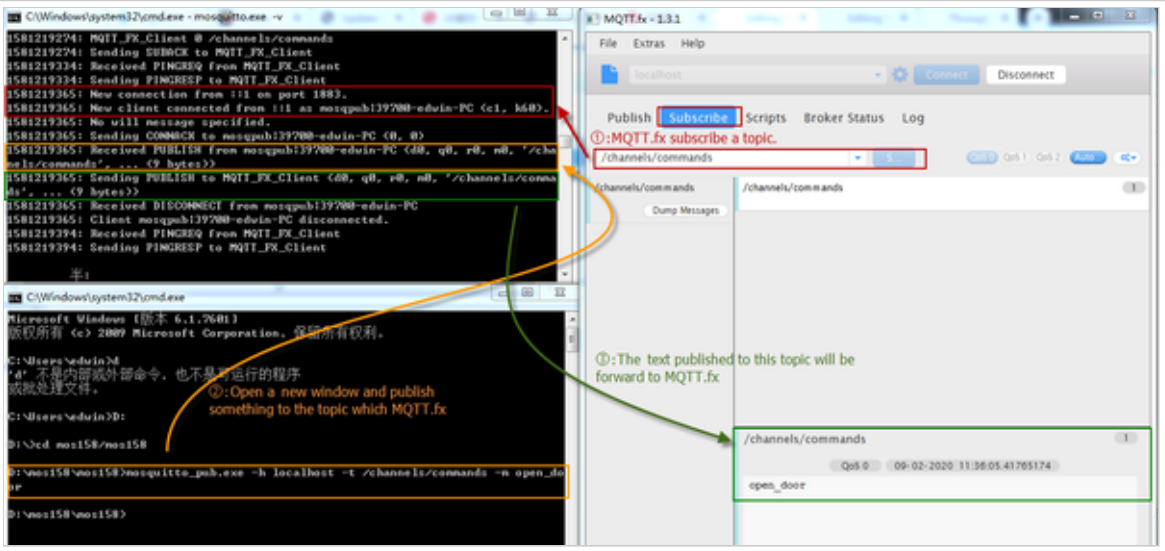
Connect to MQTT Broker

After connected, use publish to public some thing to MQTT server. This to simulate upsteam



Upstream: Publish message to MQTT Broker

To simulate a downstream, use MQTT.fx to subscribe a topic, and publish something to this topic. as Below:



Downstream: Subscribe a topic to get downstream

Simulate via Dragino Command Line

For first try of MQTT connection, simulate via command line is recommend, there are many servers / connection type for MQTT. They are using different connection parameters. Simulating the connection via command line will help us rapidly connect to server and debug.

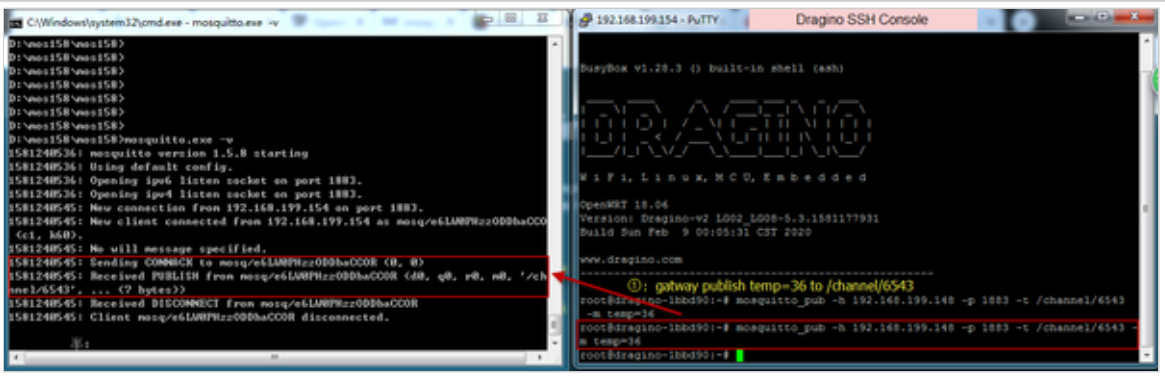
In the Dragino Gateway, we use mosquitto client (https://mosquitto.org/) for MQTT connection.

For Upstream

command is mosquitto_pub (https://mosquitto.org/man/mosquitto_pub-1.html)

Example: mosquitto_pub -h 192.168.199.148 -p 1883 -t /channel/6543 -m temp=36

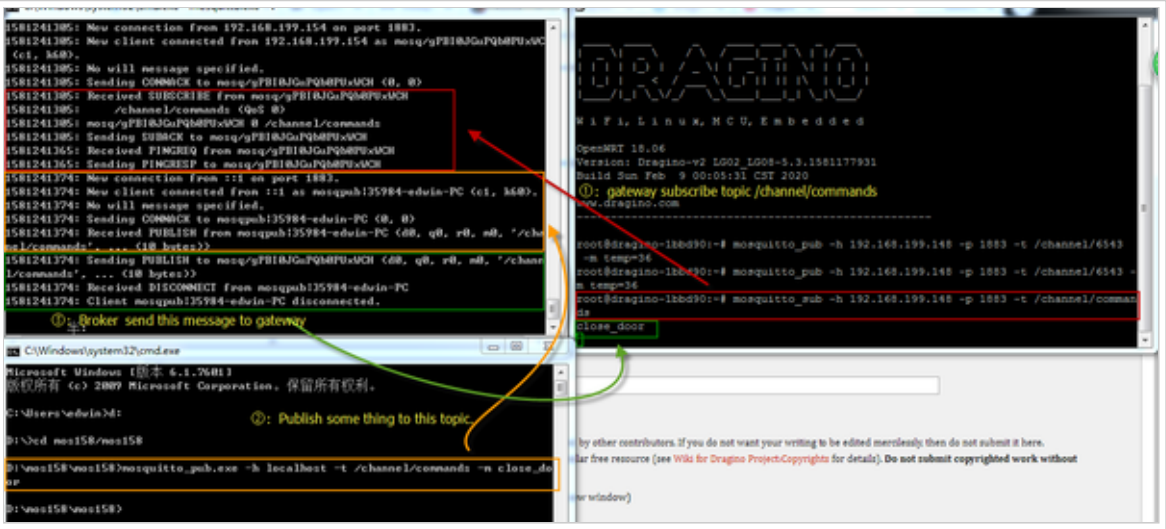
Note: 192.168.199.148 is MQTT broker address, the gateway and the MQTT broker PC are in the same network.



mosquitto_pub

For Downstream

Use mosquitto_sub (https://mosquitto.org/man/mosquitto_sub-1.html) to subscribe the change on the topic.

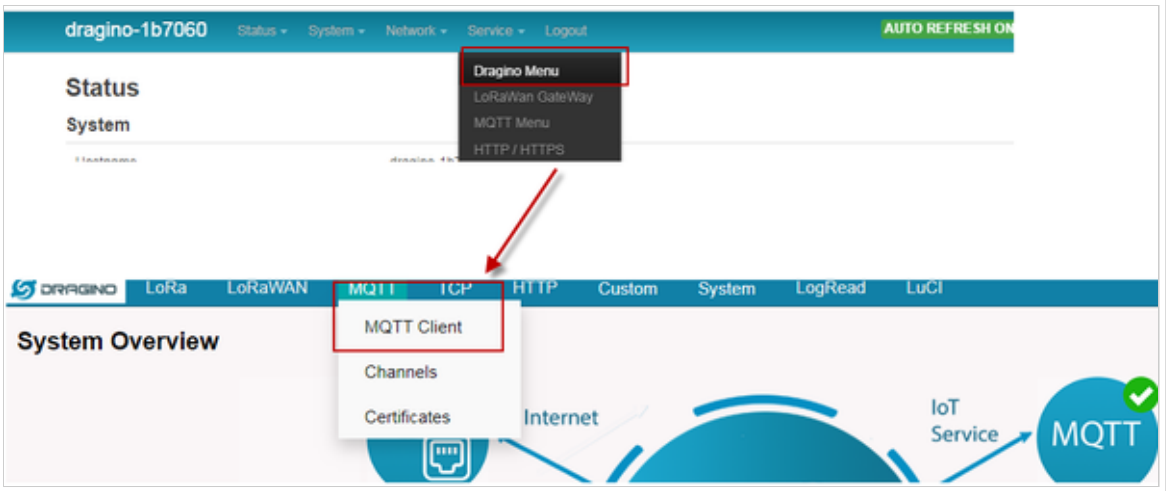


mosquitto_sub

Configure Dragino UI for MQTT connection

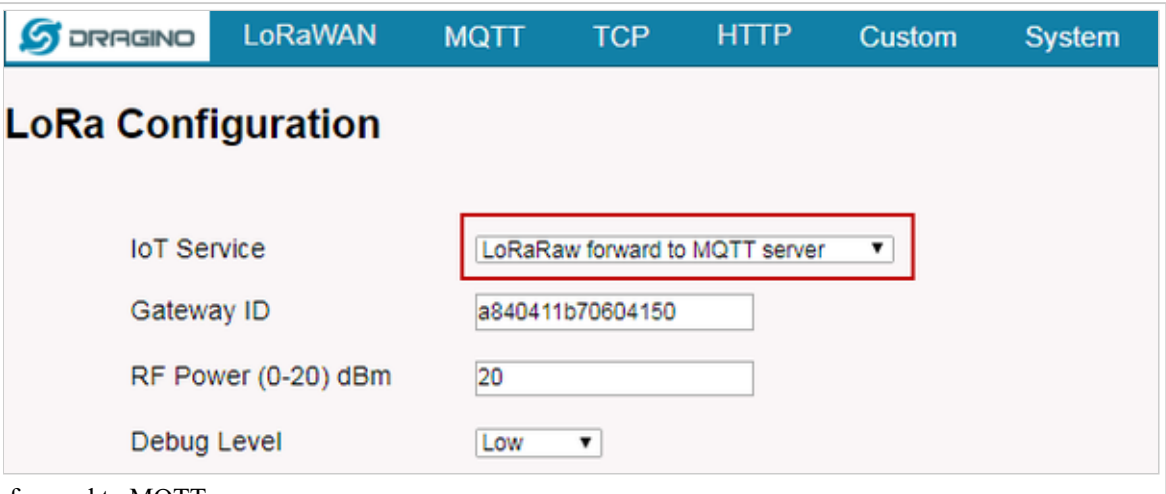
This chapter are step by step to show to configure the Dragino Menu for MQTT auto connection.

Go to Dragino Menu --> MQTT Client



go to mqtt configure menu

Select Forward to MQTT server. Notice: This option is removed from the latest firmware, in the latest firmware, if user submit "SAVE & APPLY" in MQTT page, the gateway will use MQTT service.

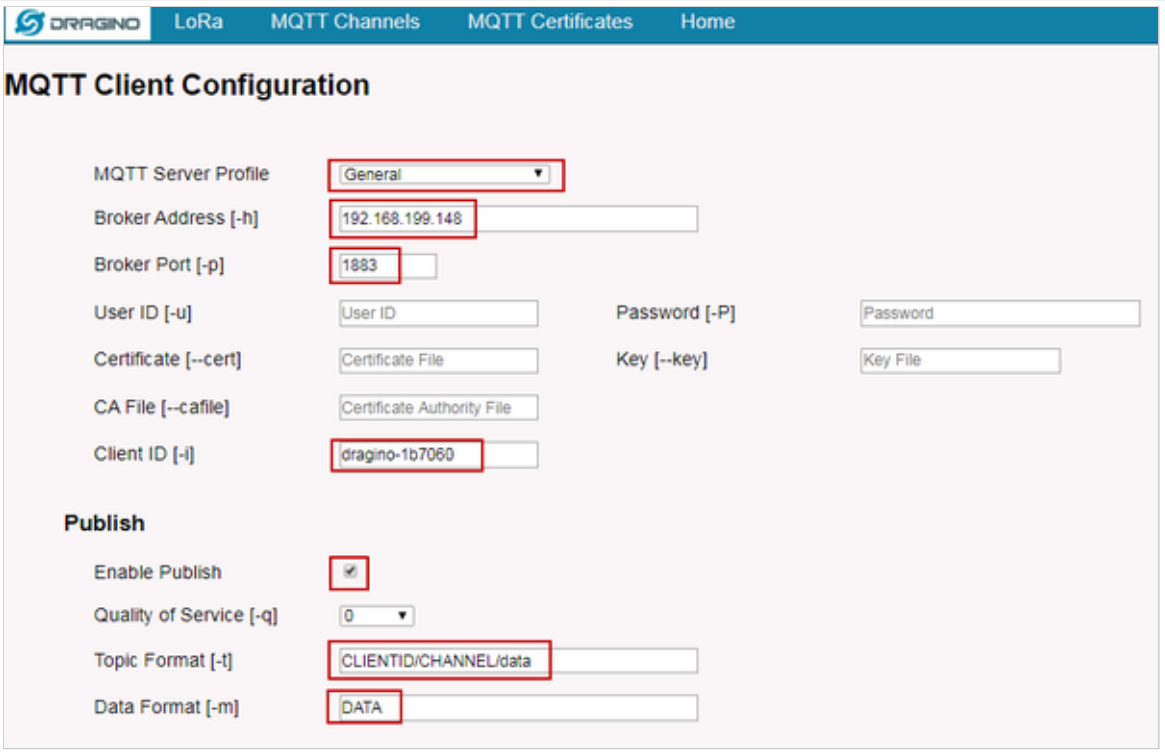


forward to MQTT

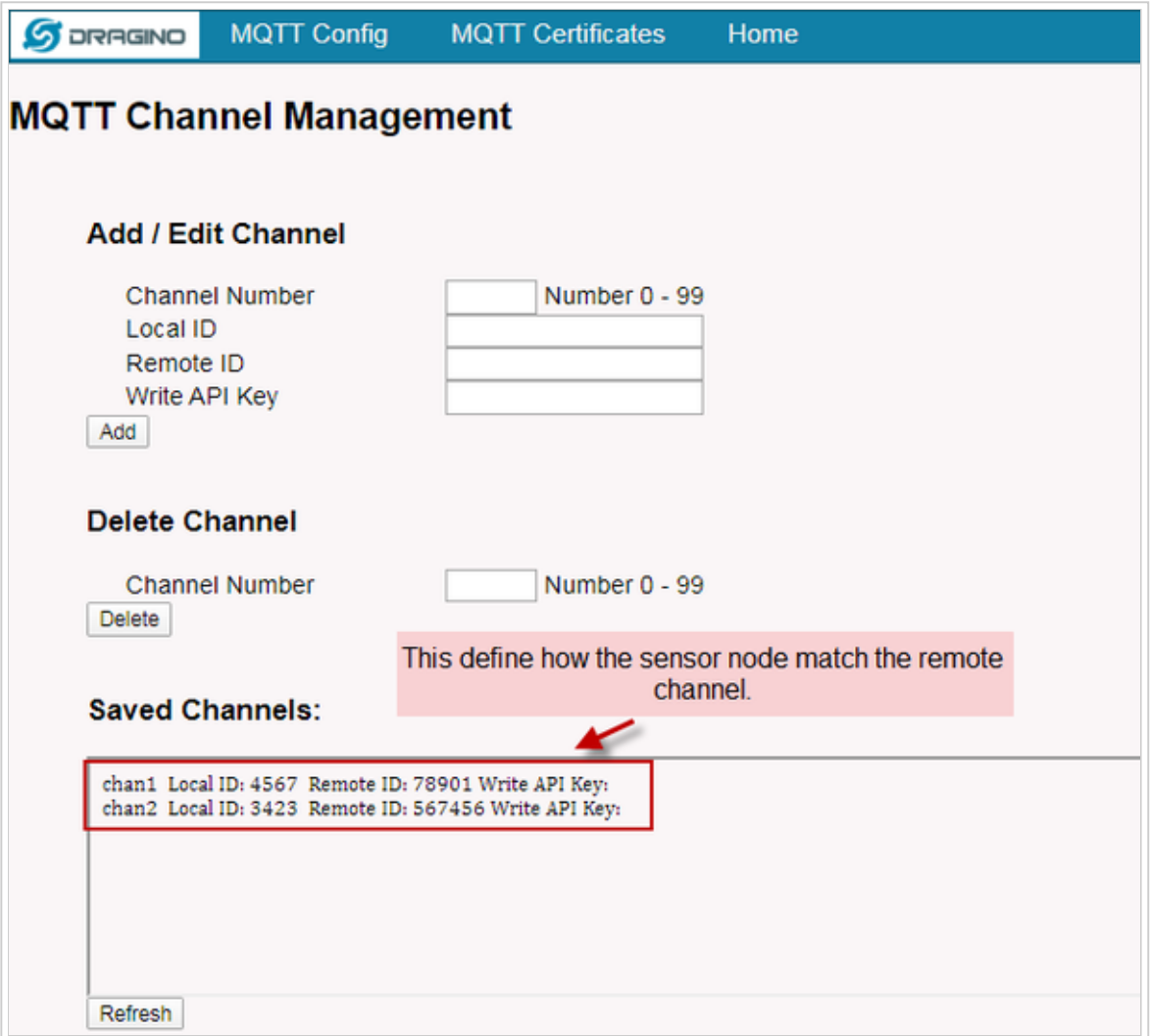
Configure the MQTT Client for Upstream

Below screenshot is same as the publish command:

```
mosquitto_pub -h 192.168.199.148 -p 1883 -i dragino-1b7060 -t CLIENTID/CHANNEL/data -m DATA
//where the CLIENTID, CHANNEL & DATA are macro. represent for
//CLIENTID: dragino-1b7060
//CHANNEL: Remote ID in Channel settings; here is 78901 or 567456
//DATA: The data stores in /var/iot/channels/
```



MQTT Publish configure



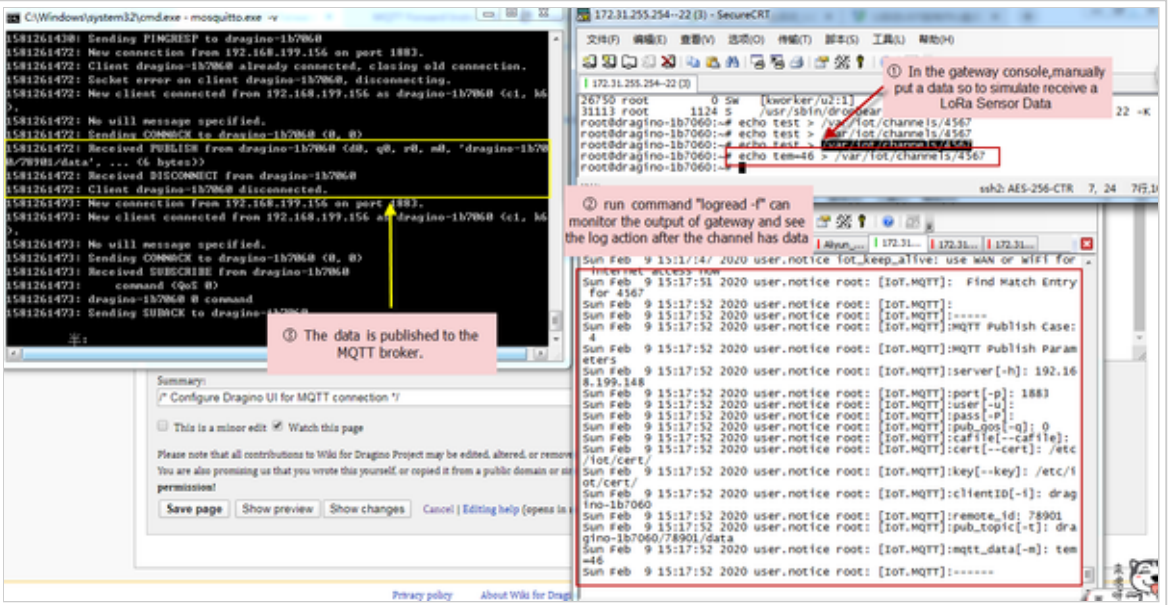
MQTT Channel settings

For example, if we put a data(temp=46) on the file /var/iot/channels/4567, because 4567 match the remote channel 78901. the gateway will run this command:

```
mosquitto_pub -h 192.168.199.148 -p 1883 -i dragino-1b7060 -t dragino-1b7060/78901/data -m temp=46
```

to MQTT broker.

Below is a simulation to put this data to active the MQTT publish.

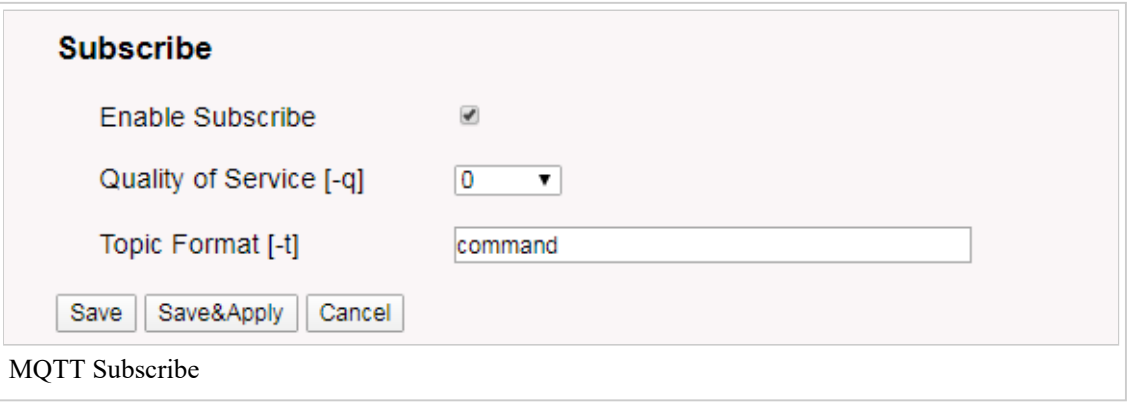


MQTT Publish

Configure the MQTT Client for Downstream

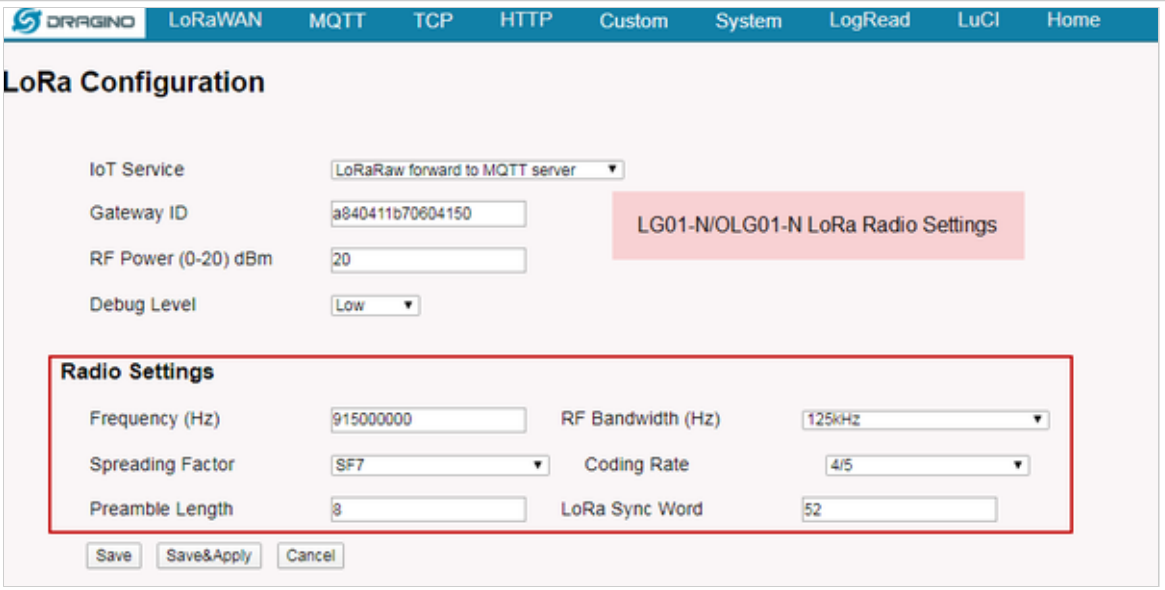
Below screen shot equal to this subscribe command:

```
mosquitto_sub -h 192.168.199.148 -p 1883 -i dragino-1b7060 -t command.
```



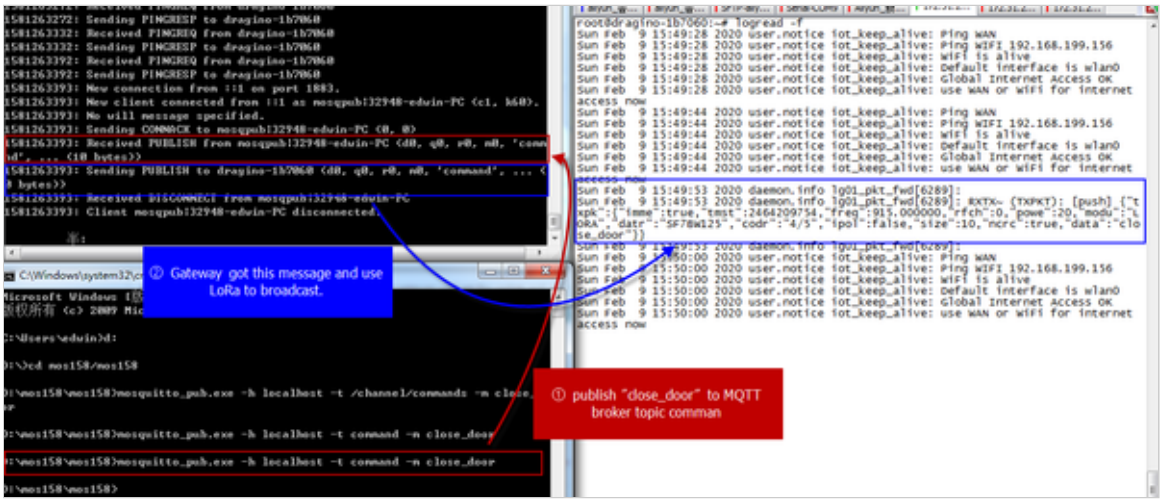
MQTT Subscribe

When MQTT broker receive a update on this topic, the gateway will get the update and use LoRa radio to broadcast this message. The LoRa parameters used for update is:



LoRa Broadcast parameters.

And below is the subscribe simulation:



downstream simulation

Add LoRa support to communicate with remote sensor

In above section, we have configured the UI to support MQTT upstream and downstream. We can simulate via Linux command. In this section, we will guide how to communicate with remote LoRa End Node for upstream and downstream.

Use LoRa Raw protocol for communication -- For LG01/LG02

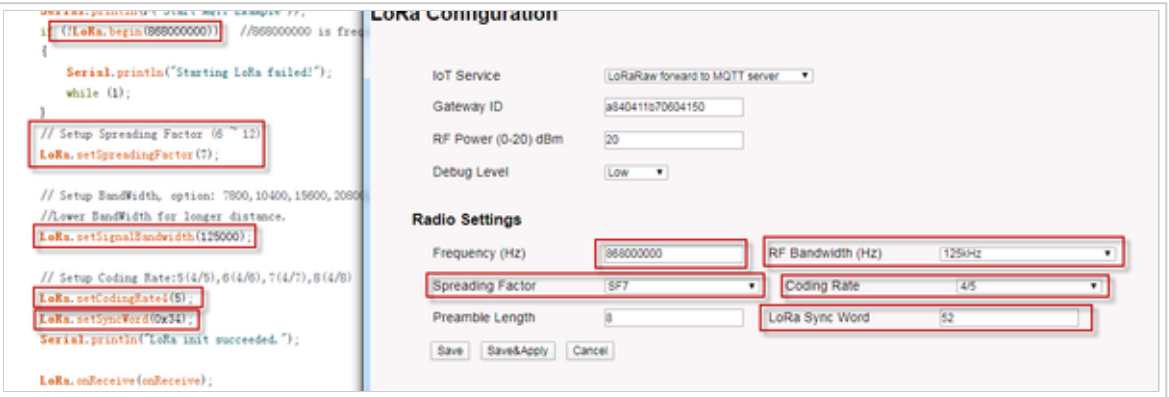
We can use LoRa Shield (http://www.dragino.com/products/lora/item/102-lora-shield.html) to send LoRa Raw data to Gateway and receive data from gateway.

The example Sketch for LoRa Shield +Arduino is here: LoRa_Shield_Sketch_For_MQTT (http://www.dragino.com/downloads/index.php?dir=LoraShield/)

And this link is the required library: arduino-LoRa-master (http://www.dragino.com/downloads/index.php?dir=LoraShield/). Unzip this library and put in Arduino library location.

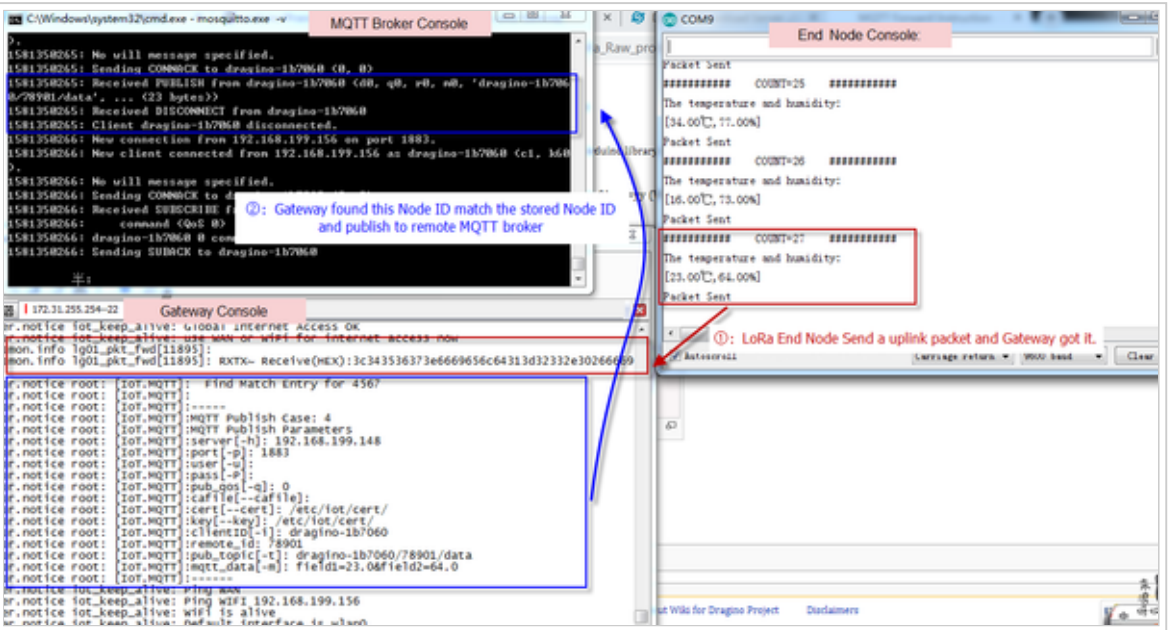
What does the Arduino Sketch do? The Arduino Sketch will:

- Upstream: Keep sending a LoRa Message every minutes with this payload : <4567>tem=xx&hum=yy (Where xx and yy are temperature and humidity value generated randomly).
- Downstream: Listening broadcast message from gateway, and print it in console.
- The LoRa parameter settings in Arduino should match the LoRa settings in gateway, as below:

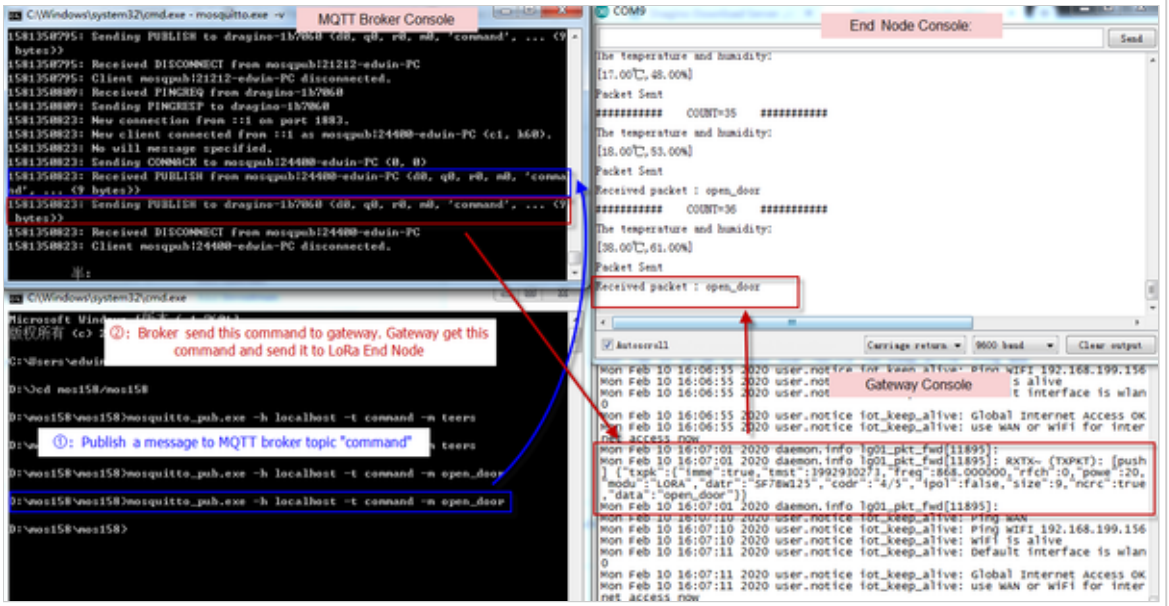


LoRa Parameter should match

Below is the test result after the Arduino Sketch is running.



Upstream Data Flow



Downstream Data Flow

Use LoRaWAN Protocol for communication -- For LG308/LPS8/DLOS8

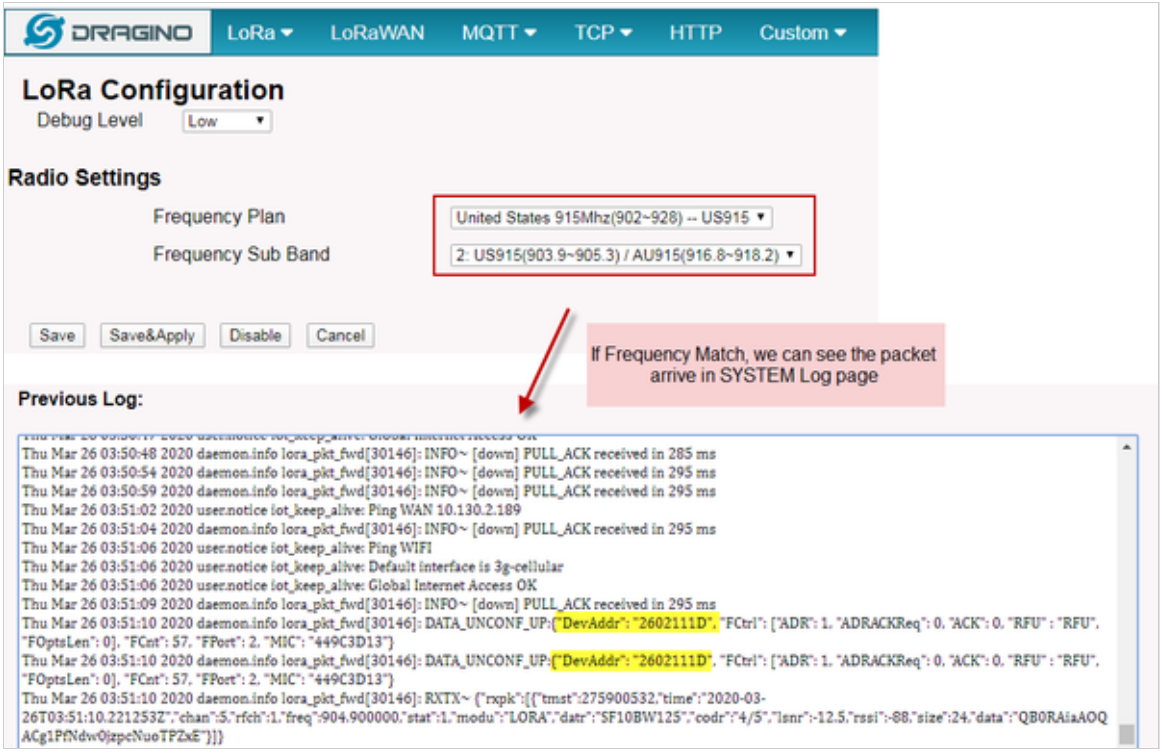
Since firmware LG02_LG08--build-v5.3.1585192026-20200326-1109, Dragino LoRaWAN gateways support the communication to LoRaWAN ABP end node locally without the need of LoRaWAN server. This feature allow us to integrate MQTT in the gateway to support LoRaWAN to MQTT forwarding or visa verse.

When use test this feature, please use the version higher then : LG02_LG08--build-v5.4.1593400722-20200629-1120, in this version, the upload format is changed and readable, which is easier for integration.

Video Instruction:https://youtu.be/qJTY441-t90

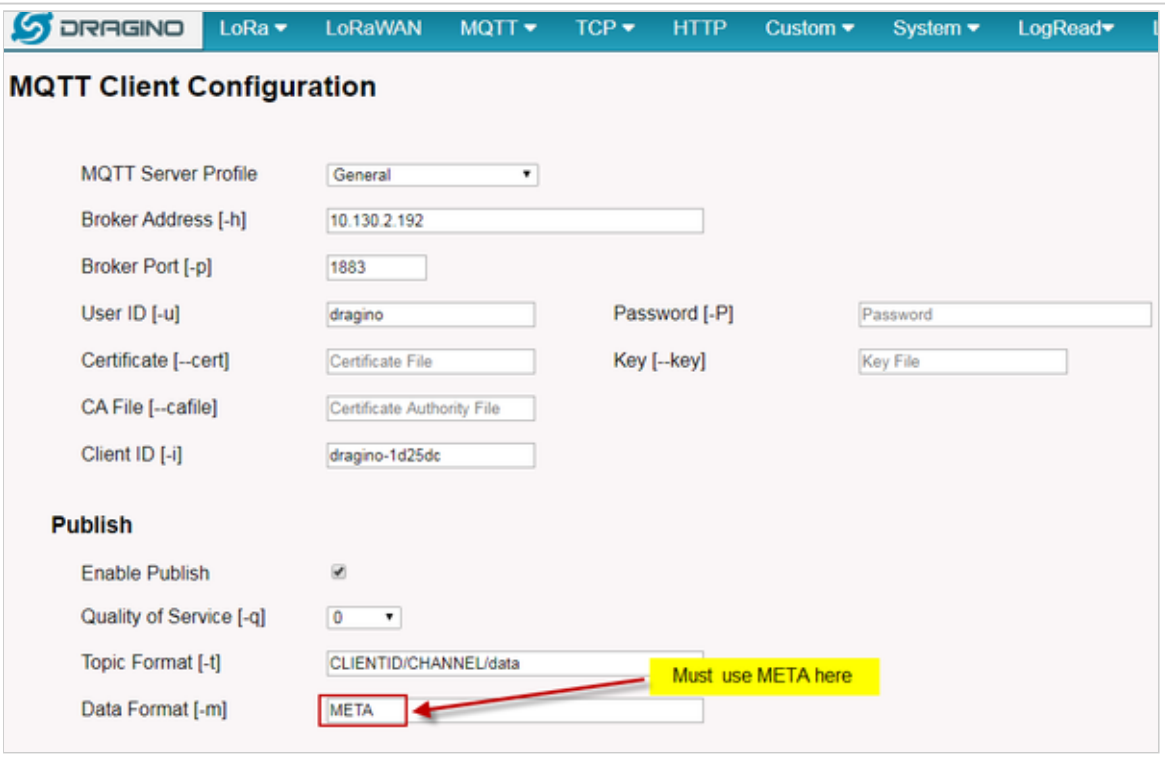
Step 1: Refer Communicate with ABP End Node to know how to set up LG308 to work with LoRaWAN End node.

Step 2: Make sure your Radio settings match the End Node settings.



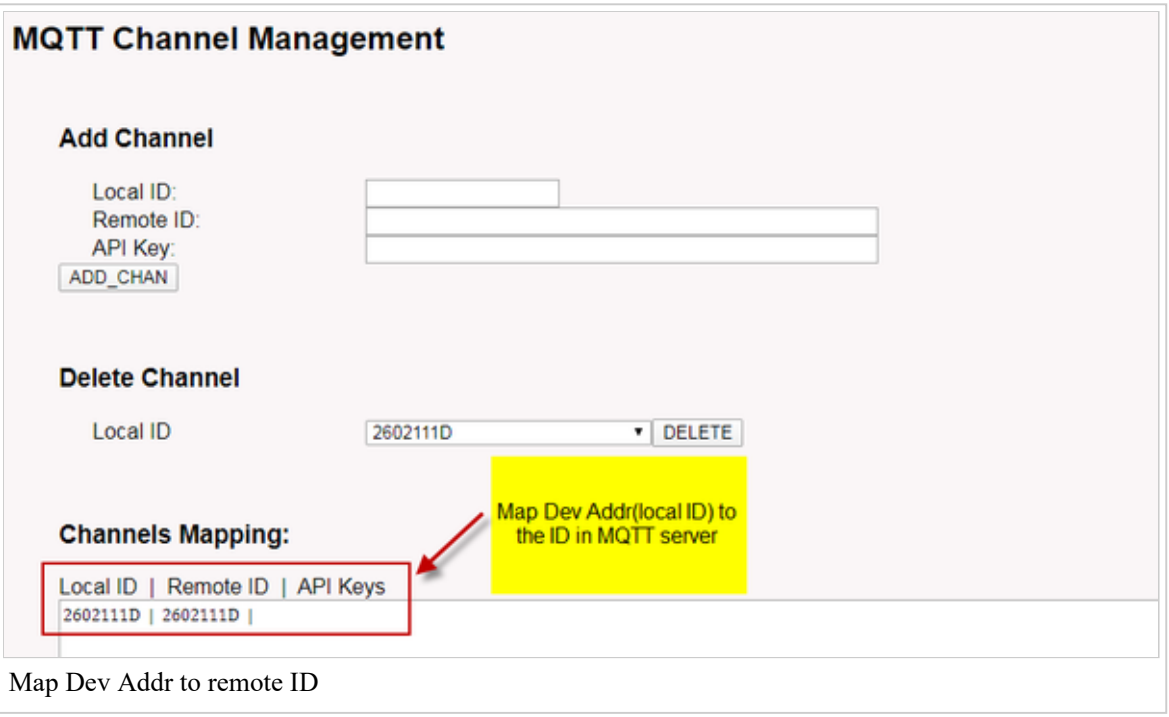
Use Same Frequency Band as End Node

Step 3: Set up publish format and MQTT channel. The LG308 will store the Data from End node in HEX format in the file. And we need to config the format to META



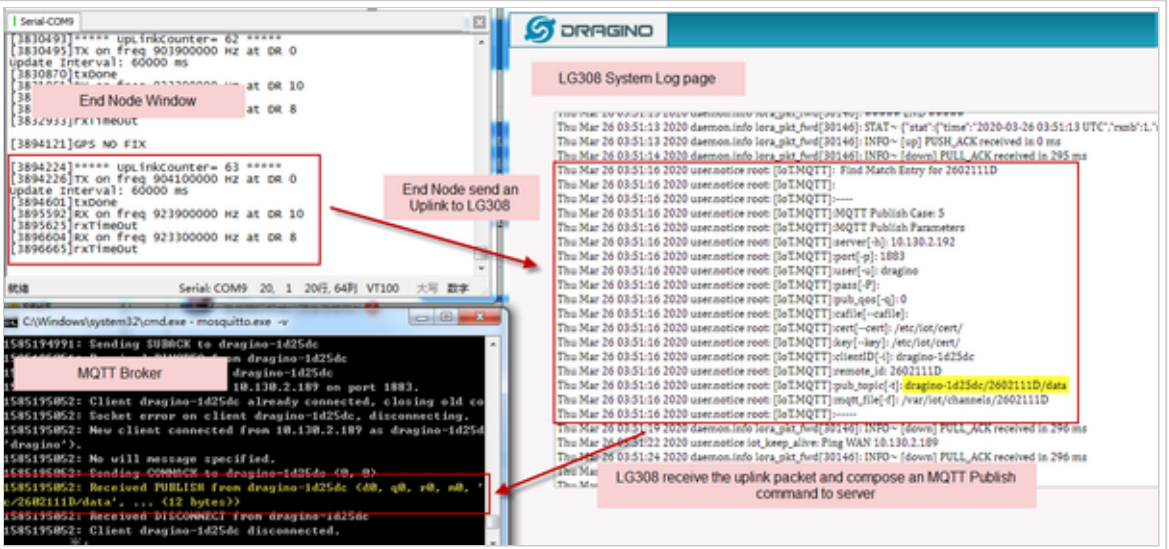
Publish mush use META as data format for LG308

Step 4: Map the Device Address to Remote ID in MQTT server.

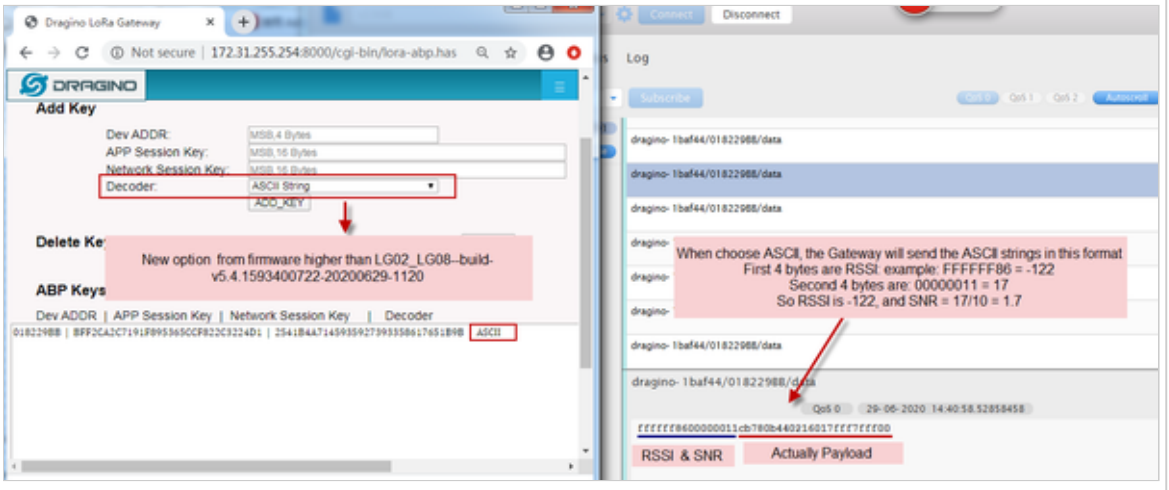


Map Dev Addr to remote ID

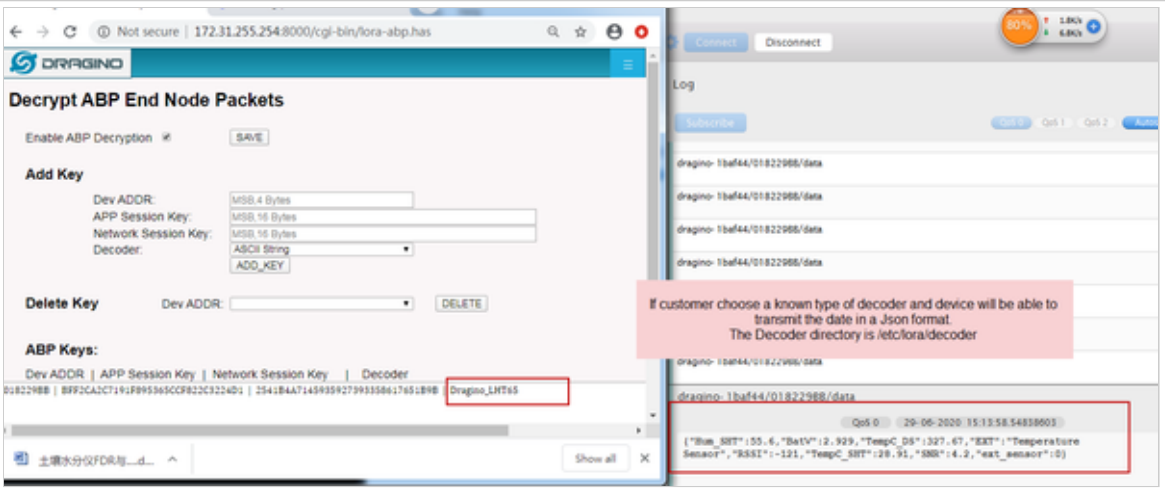
Step 5: Upstream: Save the change, we can see the log info via "sytem log", End Node and MQTT Server



Upstream Process

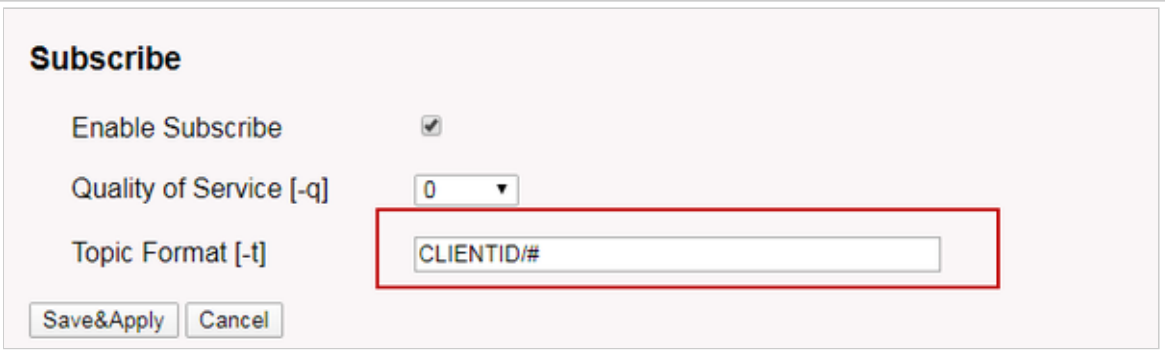


Choose ASCII Format



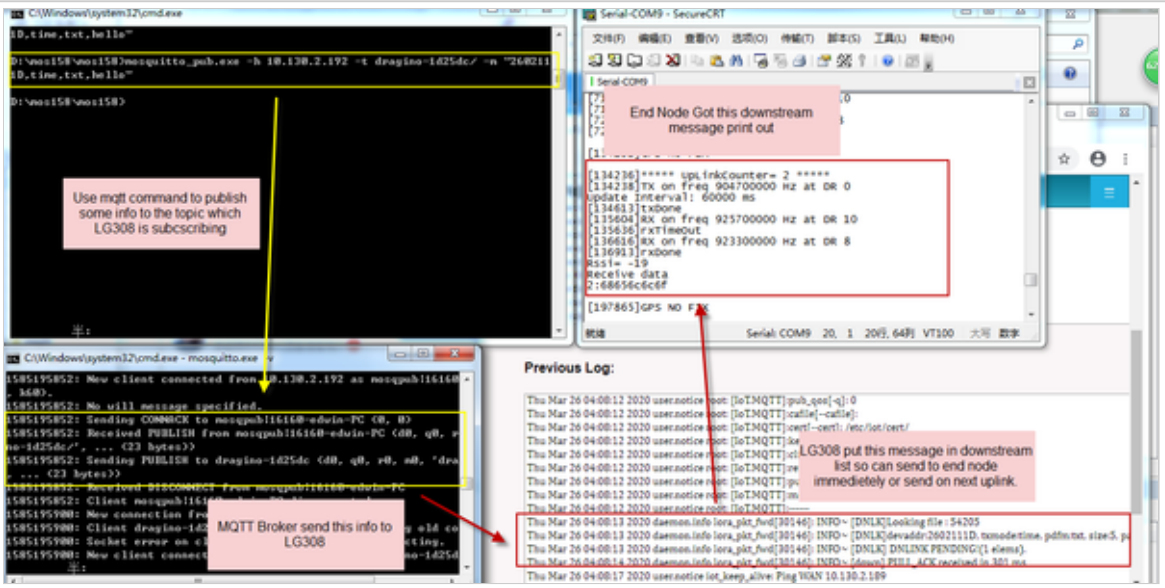
LHT65 Decoder

Step 6: Set up subscribe: Subscribe a topci for downstream.



Subscribe to a topic

Step 7: Downstream: Save the change, we can see the log info via "sytem log", End Node and MQTT Server.



Downstream Flow

Notice: The text use for Downstream must meet the requirement from LG308 Downstream Payload

Example For Different MQTT Servers

		
ThingSpeak Server (http://www.thingspeak.com/)	**乐联网平台 (https://www.lewei50.com/)**	**AWS-IOT (https://aws.amazon.com/cn/iot-platform/how-it-works/)**
Examples	**lewei Example**	**AWS Examples**

===End===

Retrieved from "http://wiki.dragino.com/index.php?title=MQTT_Forward_Instruction&oldid=6997"

- This page was last modified on 21 August 2020, at 09:00.