AN01541

Set up LoRaWAN GW with RHF0M301

V1.8

Document information

Info	Content
Keywords	*RisingHF, LoRaWAN, GW, Module, Raspberry Pi, LorIOT server*
Abstract	This document is a simple guide to setup a LoRaWAN/LoRa gateway with RisingHF 8 channel LoRaWAN GW module.

Content

Figures

Tables

1 Preface

RHF0M301 is an 8 channel LoRa/LoRaWAN gateway module designed by RisingHF. With its small size and 24 pin DIP connector feature, customer could easily integrate the module into their own platform to design their own customized LoRa/LoRaWAN gateway.
This document would describe essential information about hardware design with this module. And a guide to set up LoRaWAN GW with Raspberry pi 2 is also included in this file.

2 Hardware set up

2.1 Absolute Maximum Ratings

Table 1 Absolute maximum ratings

Item	min	typ	max	Unit
Temperature	-40	+25	+85	℃
RF Input			-13	dBm
Supply Input	-0.3	+5	+6	V

2.2 Pin definition

A 2x14 DIP with pitch=2.54mm is used for this module. The pin definition is shown in Figure 2-1 below.

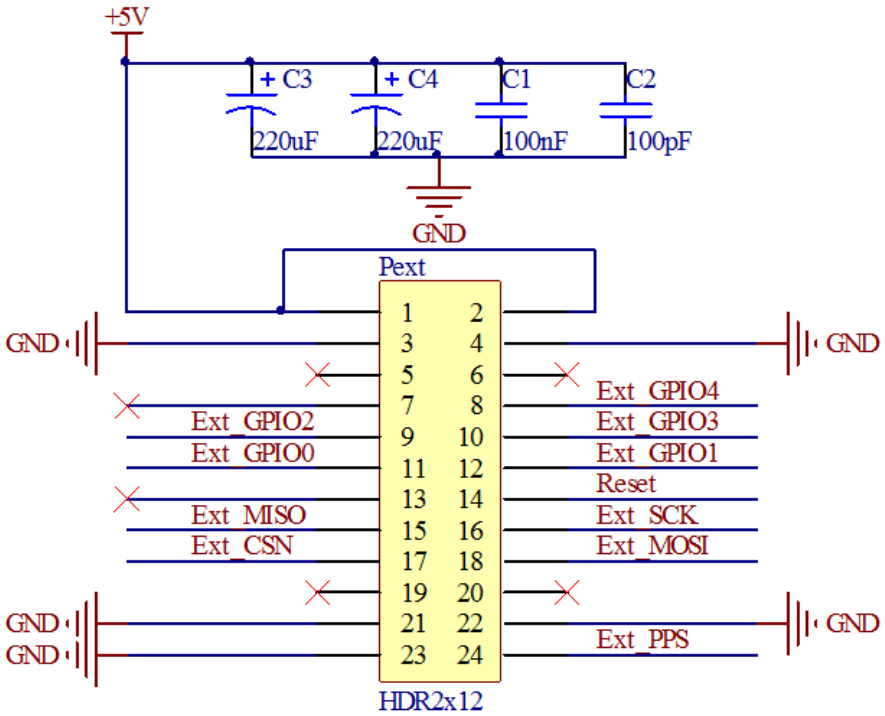

Figure 2-1 Pin definition of RHF0M301

Table 2 Pin definition and description

Pin	Definition	description
1	VCC5V	+5V Input
2	VCC5V	+5V Input
3	GND	Ground
4	GND	Ground
5	NC	No connection
6	NC	No connection
7	NC	No connection
8	SX1301_GPIO4	GPIO4 from SX1301
9	SX1301_GPIO2	GPIO2 from SX1302
10	SX1301_GPIO3	GPIO3 from SX1303
11	SX1301_GPIO0	GPIO0 from SX1304
12	SX1301_GPIO1	GPIO1 from SX1305
13	NC	No connection
14	Reset	Reset signal input to reset SX1301
15	MISO	MISO of SPI
16	SCK	SCK of SPI
17	CSN	CSN of SPI
18	MOSI	MOSI of SPI
19	NC	No connection
20	NC	No connection
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GPS_PPS	PPS signal input from GPS module

Note: 220uF//220uF//100nF//100pF is strongly suggested to put as close as to the input pin (Pin1 and Pin2) of the module when you layout !

2.3 Mechanical size of RHF0M301

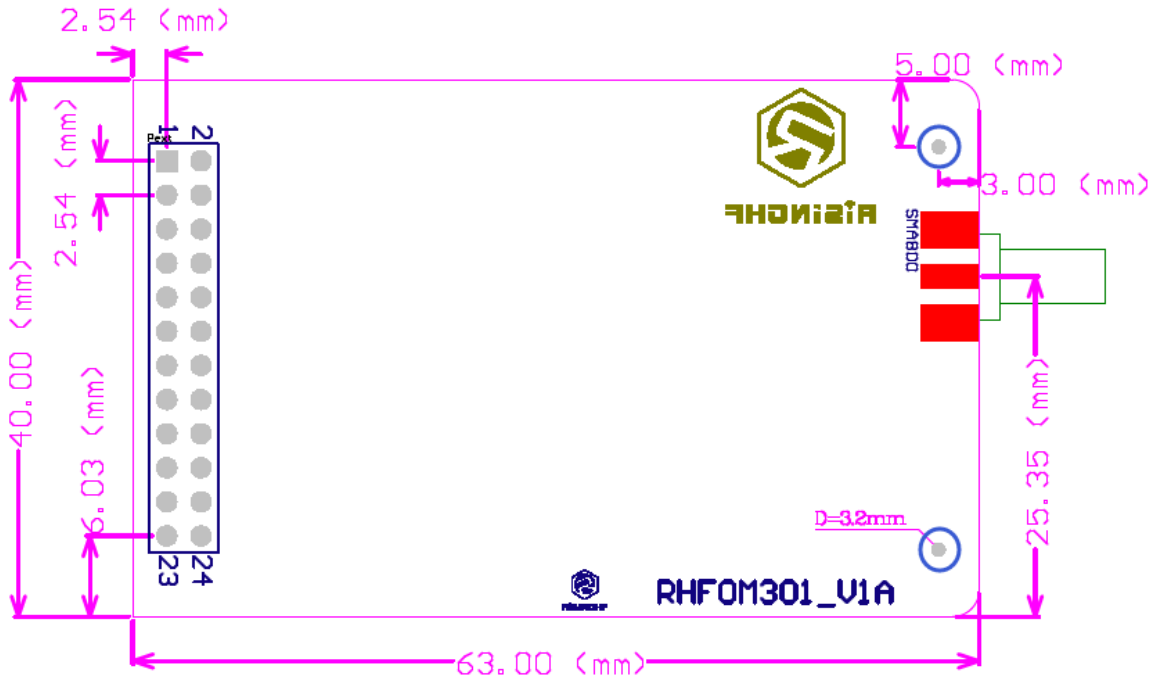

Figure 2-2 Mechanical size of RHF0M301

2.4 Hardware Requirement to setup LoRaWAN Gateway

Hardware platform: Raspberry pi 2 (for example here) or others similar platform.
Interface connection: SPI.
Supply Range: +4.5V to +5.5V.
Rated current suggested: at least 1.5A[(1)].

Note:

(1) The maximum current is about 660mA with max output power with 50R match. But peak current would be about 1A if the output port is mismatching, antenna is mismatch for example. So l.5A is only for RHF0M301.

3 Application information

3.1 Power consumption

Table 3 Power consumption of RHF0M301V1A

Status	Current	Unit
Normal 8 Rx CH ON PA ON	340	mA
Normal 8 Rx CH ON PA ON (Uplink) Average	590	mA
Normal 8 Rx CH ON PA ON (Uplink) Peak	660	mA
Normal Standby mode	40	mA
Test mode 8 Rx CH ON	340	mA
Test Mode TX continues 27dBm set	395	mA

Note: All the test data above is based on the RF port is matching with 50R impedance, RHF0M301V1A-434 used, 25℃ Temperature.

(1) 5V DC supply
(2) RF port is matched with 50Ω load
(3) RHF0M301V1A-434 used, 25℃ Temperature

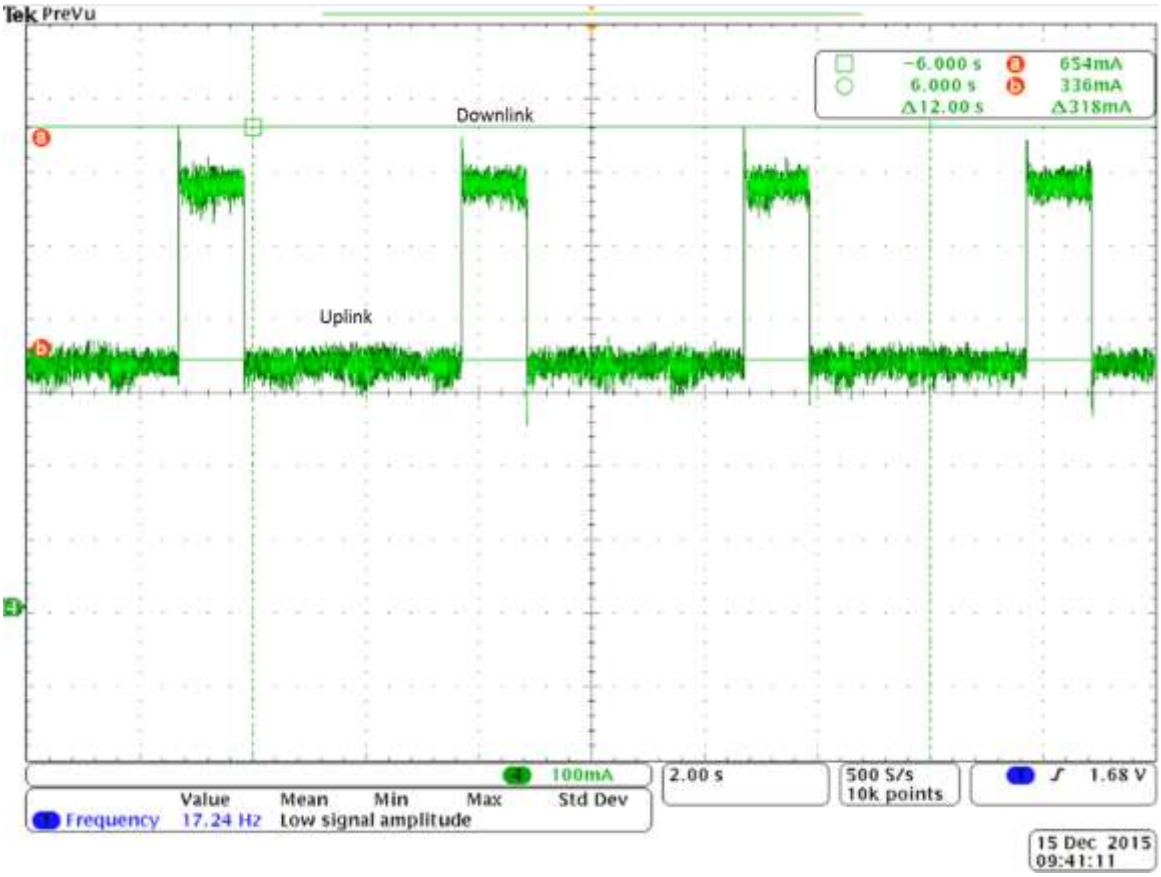

Figure 3-1 Consumption of RHF0M301V1A-434 when GW in Normal mode

3.2 Frequency response of RHF0M301-434

For RHF0M301V1A-434, from 430MHz to 437MHz band is available, which is related to the SAW filter and matching used in the module. Please refer to Figure 3-2 Figure 3-3 below.

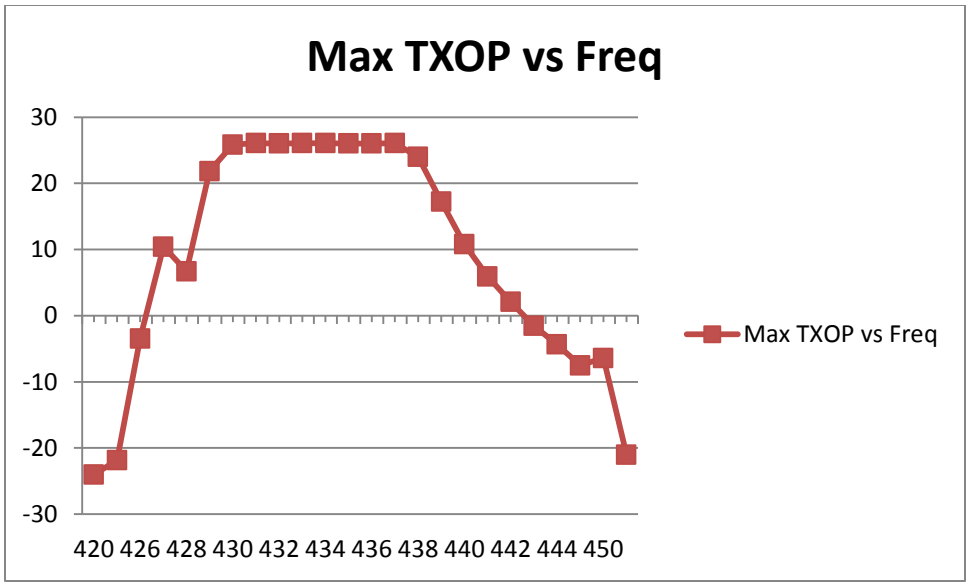

Figure 3-2 RHF0M301V1A-434 Max output power vs Frequency

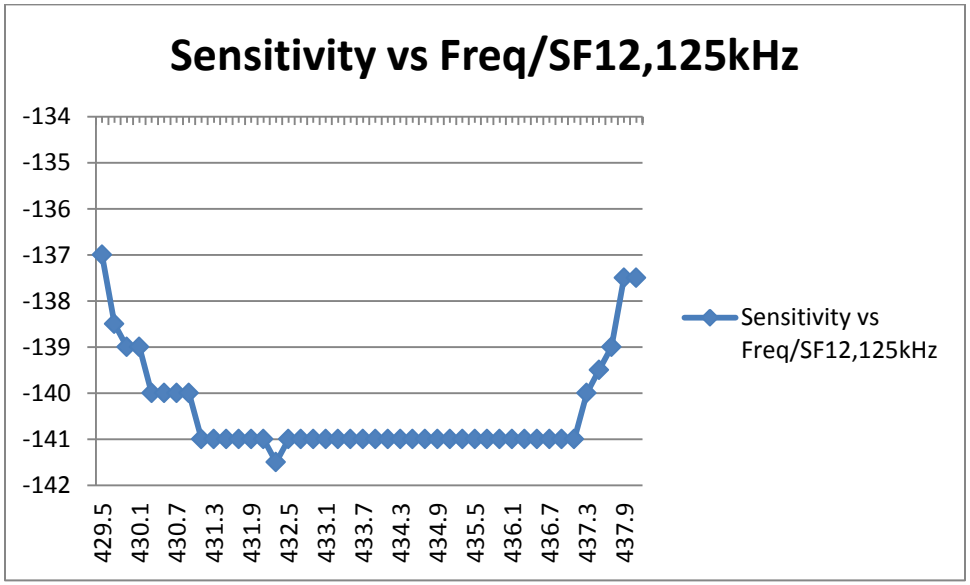

Figure 3-3 RHF0M301V1A-434 Sensitivity (SF12/125kHz) vs Frequency

3.3 Frequency response of RHF0M301-470

For RHF0M301V1A-470, from 470MHz to 490MHz band is available (Band from **470.5MHz to 485MHz** is strongly suggested to be used), which is related to the SAW filter and matching used in the module. Please refer to Figure 3-4 Figure 3-5 below.

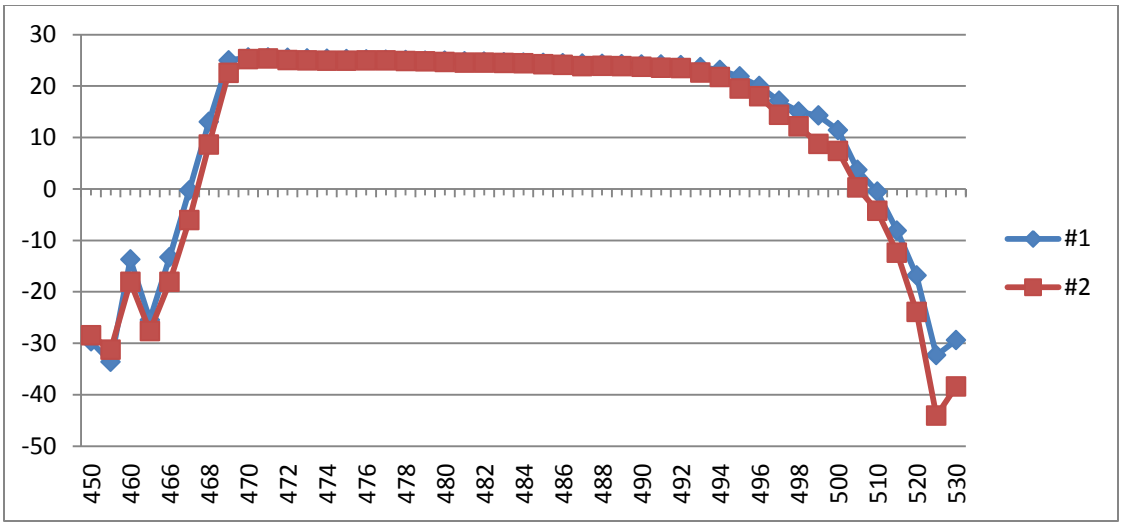

Figure 3-4 RHF0M301V1A-470 Max output power vs Frequency

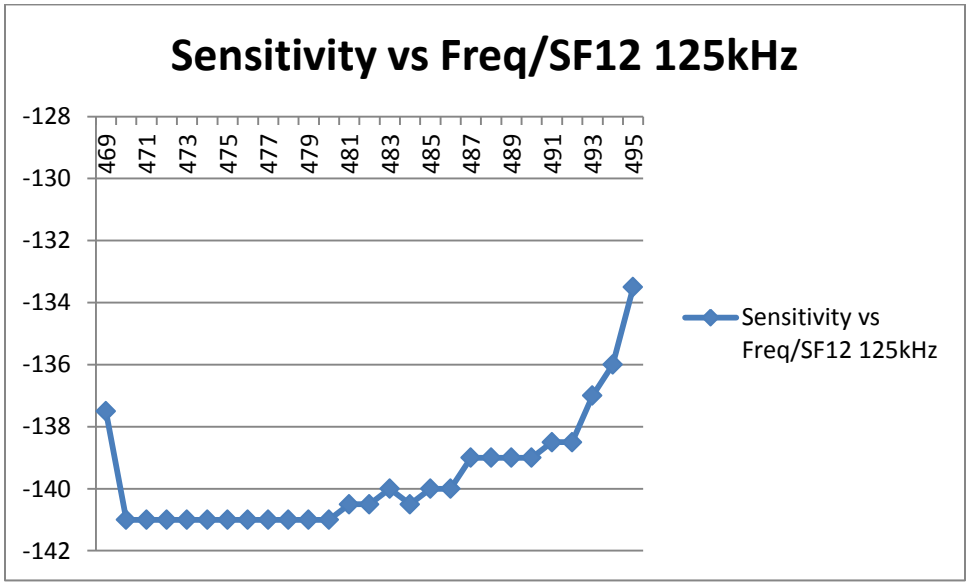

Figure 3-5 RHF0M301V1A-470 Sensitivity (SF12/125kHz) vs Frequency

3.4 Frequency response of RHF0M301-780

For RHF0M301V1A-780, from 779MHz to 787MHz band is available, which is related to the SAW filter and matching used in the module. Please refer to below for more details.

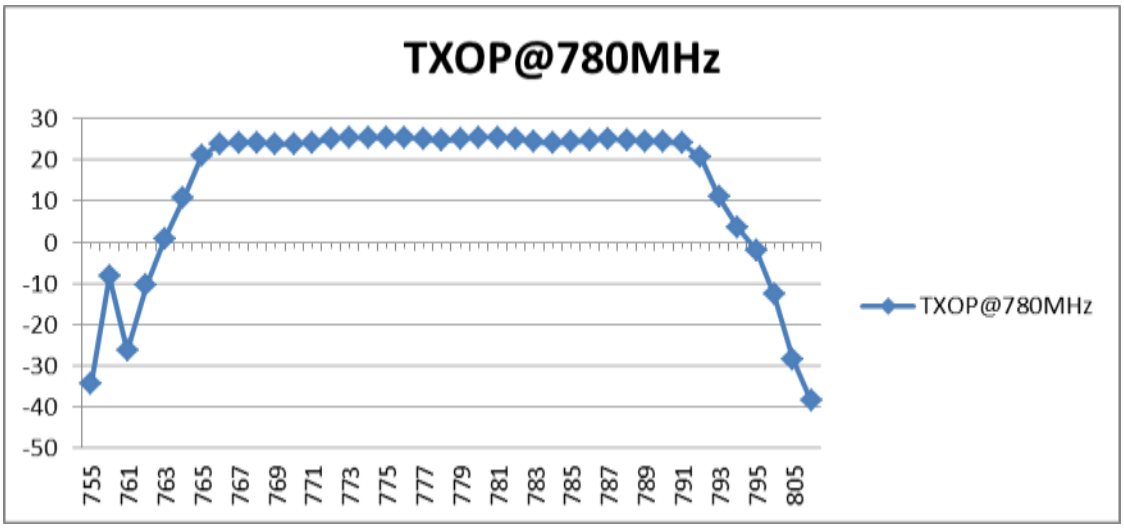

Figure 3-6 RHF0M301V1A-780 Max output power vs Frequency

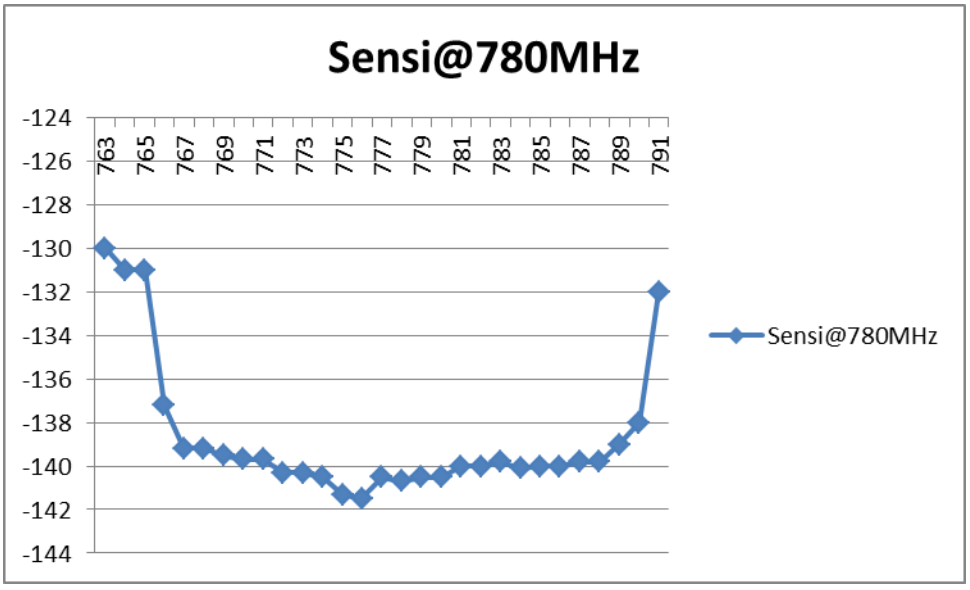

Figure 3-7 RHF0M301V1A-780 Sensitivity (SF12/125kHz) vs Frequency

3.5 Frequency response of RHF0M301-868

For RHF0M301-868, from 859MHz to 871MHz band is available, which is related to the SAW filter and matching used in the module. Please refer to below for more details.

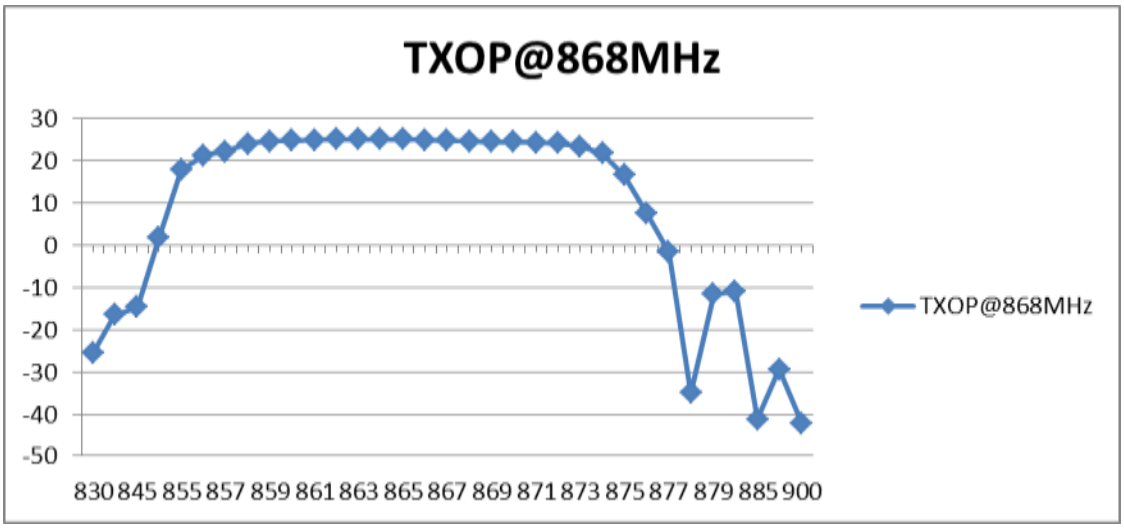

Figure 3-8 RHF0M301-868 Max output power vs Frequency

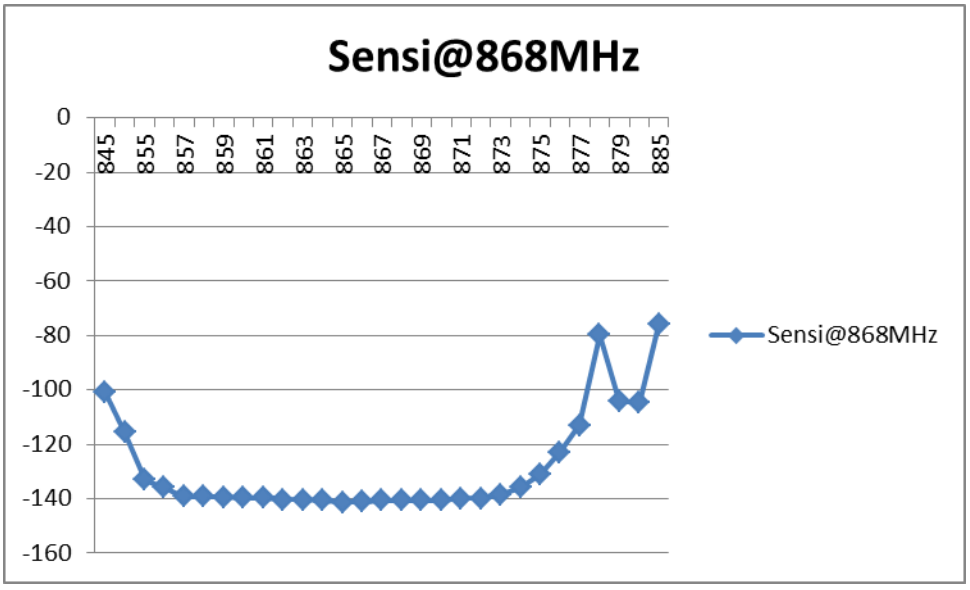

Figure 3-9 RHF0M301-868 Sensitivity (SF12/125kHz) vs Frequency

3.6 Frequency response of RHF0M301-915

For RHF0M301-915, from 900MHz to 930MHz band is available, which is related to the SAW filter and matching used in the module. Please refer to below for more details.

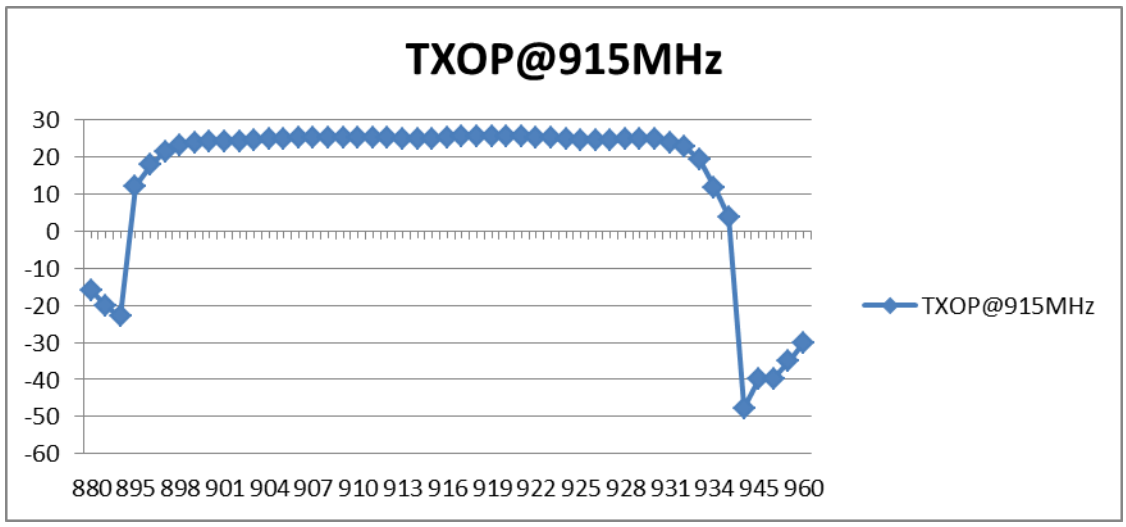

Figure 3-10 RHF0M301-915 Max output power vs Frequency

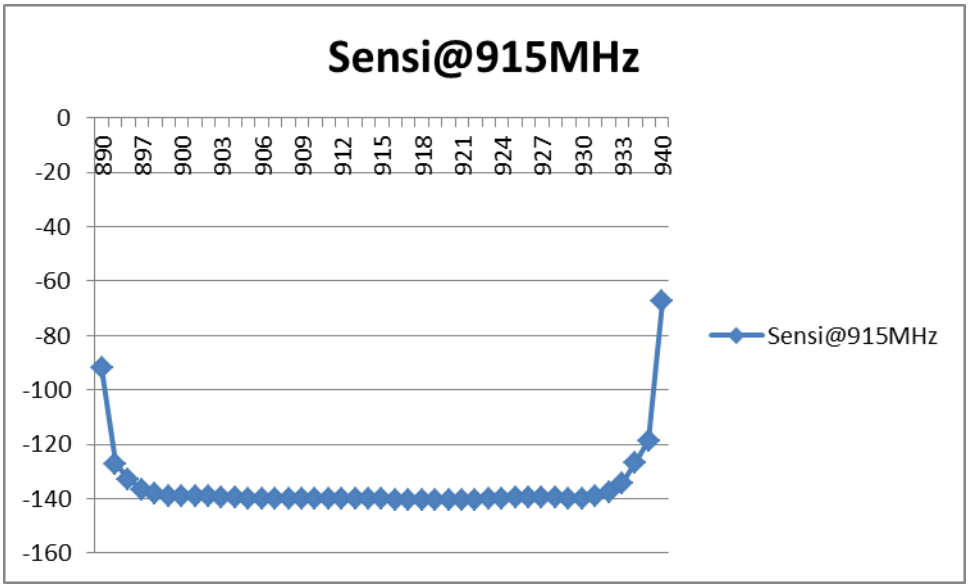

Figure 3-11 RHF0M301-915 Sensitivity (SF12/125kHz) vs Frequency

3.7 Test in temperature chamber

Table 4 RF performance in temperature chamber

Temperature	-40℃	+25℃	+85℃
Sensitivity/dBm	-141	-141	-139.5
TXOP/dBm	26.9	25.7	25.1
Frequency/MHz	433.9995	433.99975	434.00025

4 Use with Raspberry Pi

4.1 Prepare

- Raspberry Pi 2 or Raspberry Pi B+
- SD Card (at least 4G)
- 5V Adapter (>2A)
- RHF0M301 8 channel LoRaWAN Module
- Tools (Wires, Pin header, Soldering iron etc.)

Make sure you are using Raspberry Pi 2 (With 1G RAM) or Raspberry Pi 1 Model B+, some procedures may not apply to other Raspberry Pi boards.

4.2 Connection

Connect Raspberry Pi 2 with RHF0M301 according to the picture and table below. **Be careful about RHF0M301 power supply, do not use 5V of RPi, use external 5V power instead.**

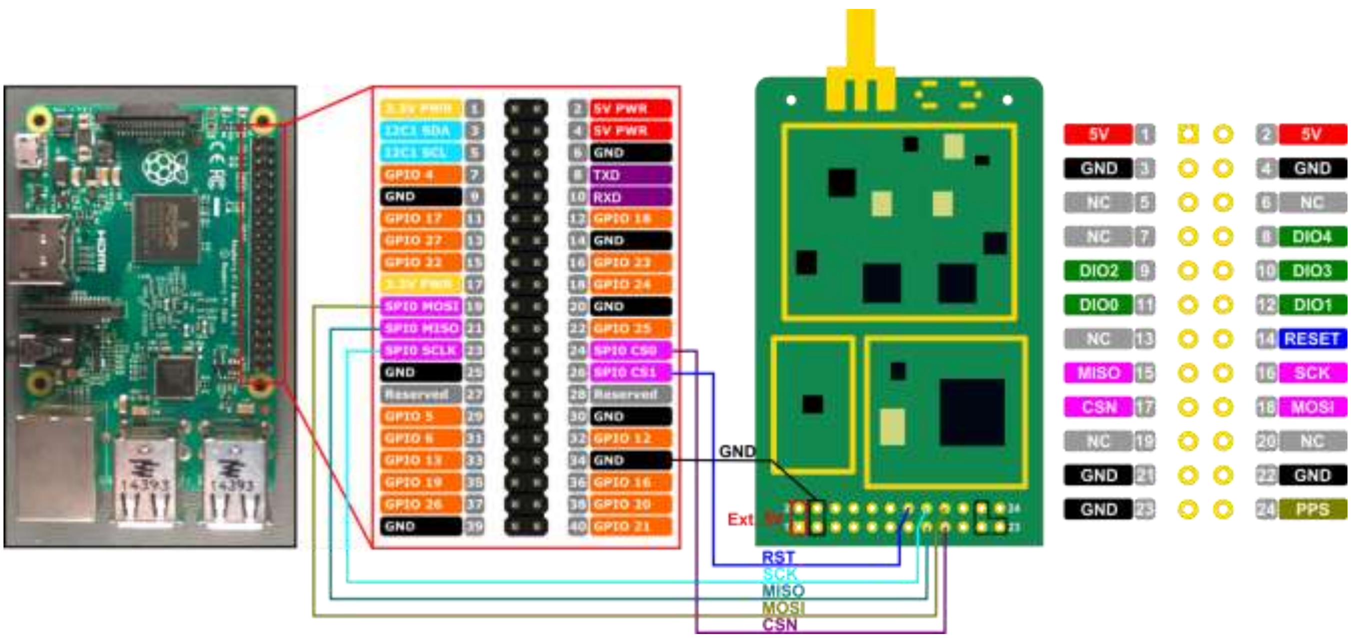

Figure 4-1 RHF0M301 and RPi Connection

4.3 Burn Raspberry Pi Image

1) Download

   Raspbian image download address:

   https://www.raspberrypi.org/downloads/raspbian/

   Choose RASPBIAN JESSIE or newer image.

2) Extract

   After image downloaded, get file 2015-11-21-raspbian-jessie.zip. Extract the file get file **2015-11-21-raspbian-jessie.img,** this is the file which need to be burned to a SD card to use.

3) Burn SD Card

Here only shows how to burn image under Windows platform, for other platforms please refer to this link.

a. Insert the SD card into your SD card reader and check which drive letter was assigned. You can easily see the drive letter (for example F:) by looking in the left column of Windows Explorer. You can use the SD Card slot (if you have one) or a cheap SD adaptor in a USB port.
b. Download the Win32DiskImager utility from the Sourceforge Project page (it is also a zip file); you can run this from a USB drive.
c. Extract the executable from the zip file and run the Win32DiskImager utility; you may need to run the utility as administrator. Right-click on the file, and select Run as administrator.

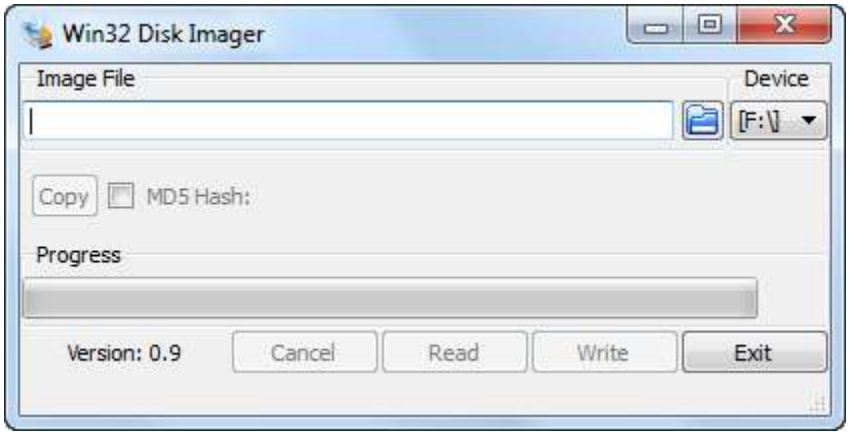

Figure 4-2 Win32 Disk Imager

d. Select the image file you extracted above.

Figure 4-3 Win32 Disk Imager Choose image

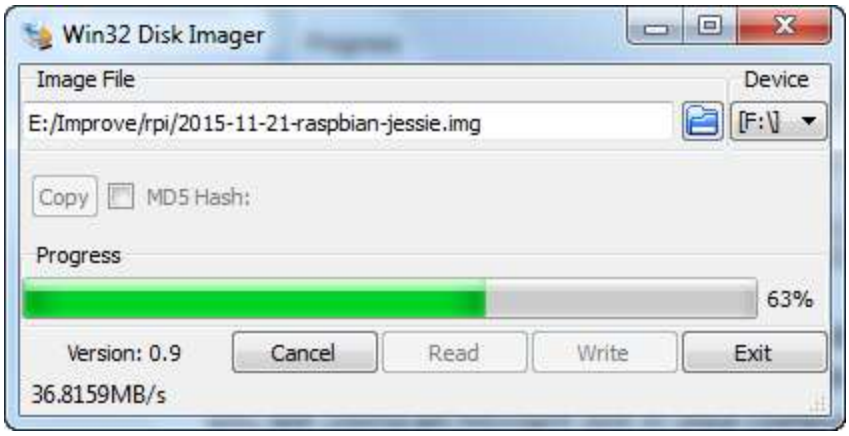

Figure 4-4 Win32 Disk Imager Writing

e. Select the drive letter of the SD card in the device box. Be careful to select the correct drive; if you get the wrong one you can destroy your data on the computer's hard disk! If you are using an SD card slot in your computer and can't see the drive in the Win32DiskImager window, try using a cheap SD adaptor in a USB port.

f. Click Write and wait for the write to complete.

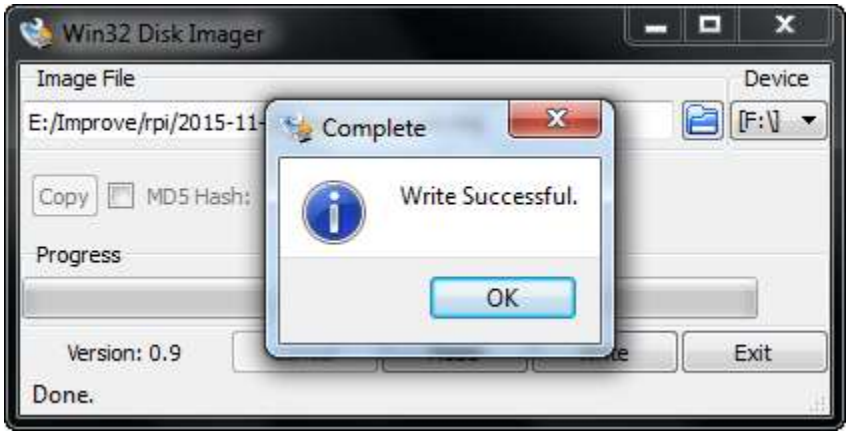

Figure 4-5 Win32 Disk Imager Write Successfully

g. Exit the imager and eject the SD card.

4.4 Start Raspberry Pi

After SD card is burned, now insert SD card to RPi, before power on you should choose a way to login RPi either use monitor and key board or UART terminal.

4.4.1 Monitor and Keyboard

Connect monitor with RPi through HDMI interface, and also connect a USB keyboard.

4.4.2 UART terminal

For those who don’t have enough resources could use UART terminal instead. An extra USB2Serial tool is needed. Pin 8 and pin 10 are Rpi UART interface. Follow below picture to connect. Baud rate is 115200 bps.

Figure 4-6 RPi and USB Serial Tool

4.4.3 Start

Now it is time to start RPi! This document will show how to control RPi through UART terminal, monitor and keyboard way is same.

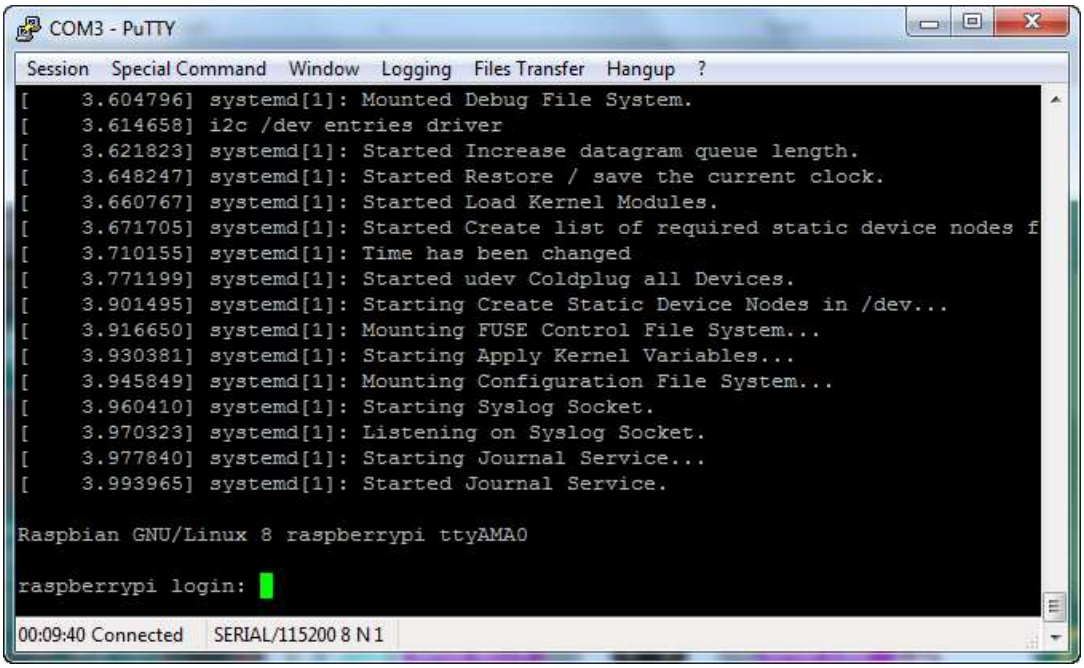

Figure 4-7 RPi Start Up Log

Use username pi and password raspberry to log in.

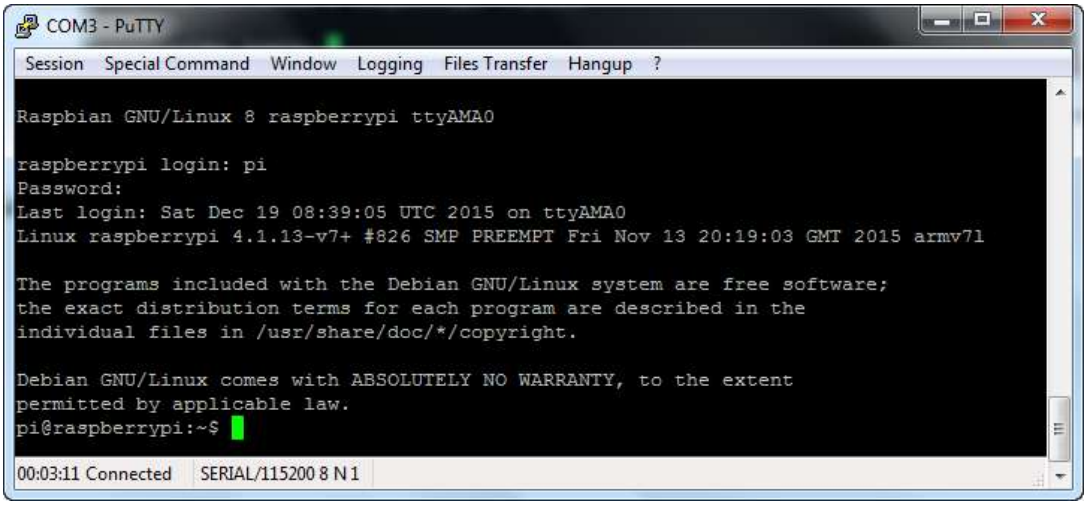

Figure 4-8 Logged in

4.4.4 Enable SPI

Backup and edit /boot/config.txt to add “dtoverlay=spi-bcm2708” and “dtparam=spi=on”, then restart your Raspberry Pi

```
$ cp /boot/config.txt ~/boot-config.txt.bak
$ sudo su
$ echo -e "\ndtoverlay=spi-bcm2708\ndtparam=spi=on\n" >> /boot/config.txt
$ reboot
```

Check if SPI device is enabled:

```
$ ls /dev/spi*
// /dev/spidev0.0  /dev/spidev0.1
```

Connect Ethernet cable and check if connection is OK:

```
$ ifconfig
$ sudo ping risinghf.com
```

After you RPi gets IP successfully, it is better to switch to use SSH log in, all commands after are executed under SSH, replace “192.168.1.117” with your ip.

```
ssh pi@192.168.1.117
```

Update RPi: (this takes time if your network is not good)

```
$ sudo apt-get update
```

4.4.5 Compile Semtech HAL

Run:

```
$ mkdir ~/risinghf
$ cd ~/risinghf
$ git clone https://github.com/Lora-net/lora_gateway.git
$ git clone https://github.com/Lora-net/packet_forwarder.git
$ mkdir -p lora_gateway/util_spectral_scan/obj

$ cd ~/risinghf/lora_gateway && make

$ cd ~/risinghf/packet_forwarder && make
```

4.4.6 Checking

Reset RHF0m301:

```
$ cd ~/risinghf/packet_forwarder/
$ sudo ./reset_pkt_fwd.sh start
```

Note: reset script need to run every time when other commands need to be done

Check if everything is OK:

```
$ ~/risinghf/lora_gateway/libloragw/test_loragw_reg

Output should be like this:
Beginning of test for loragw_reg.c
Start of register verification
```

```
+++MATCH+++ reg number 0 read: 0 (0) default: 0 (0)
+++MATCH+++ reg number 1 read: 0 (0) default: 0 (0)
...
...
...
+++MATCH+++ reg number 325 read: 0 (0) default: 0 (0)
End of register verification
IMPLICIT_PAYLOAD_LENGHT = 197 (should be 197)
FRAME_SYNCH_PEAK2_POS = 11 (should be 11)
PREAMBLE_SYMB1_NB = 49253 (should be 49253)
ADJUST_MODEM_START_OFFSET_SF12_RDX4 = 3173 (should be 3173)
IF_FREQ_1 = -1947 (should be -1947)
End of test for loragw_reg.c
```

4.4.7 Connect to LoRaWAN IoT server

4.4.7.1 loriot.io

Choose https://cn1.loriot.io/login.html to register and login.

Figure 4-9 loriot register form

Log in at https://cn1.loriot.io/home/login.html after the account is verified.

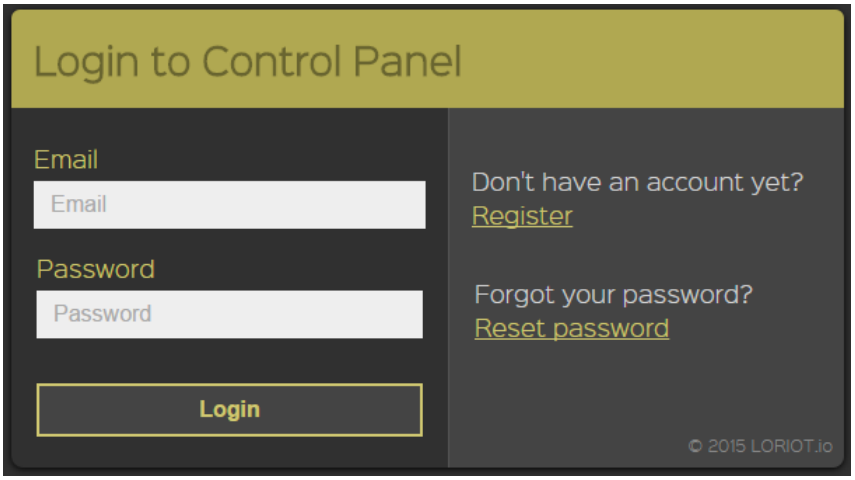

Figure 4-10 loriot.io log in

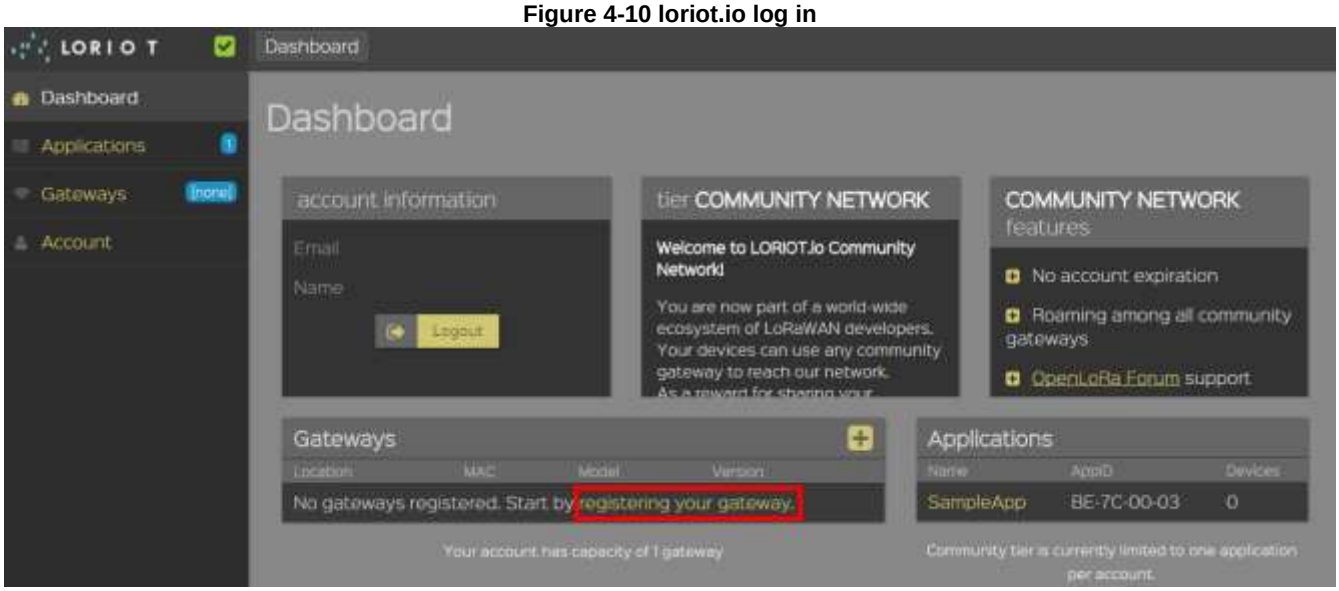

Figure 4-11 loriot main page

Click “register your gateway” link, and follow the guide to register your gateway.

1) Choose Raspberry Pi platform
2) Choose options “Raspberry Pi 2”
3) Choose concentrator model “China Reference”, if you are using (434MHz, 470MHz) board; or choose “SX1301 Reference” if you are using 868MHz, 915MHz module.

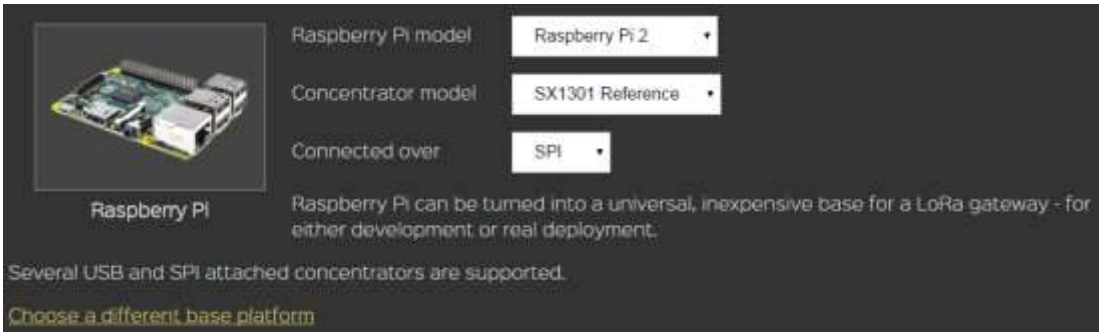

Figure 4-12 loriot gateway options

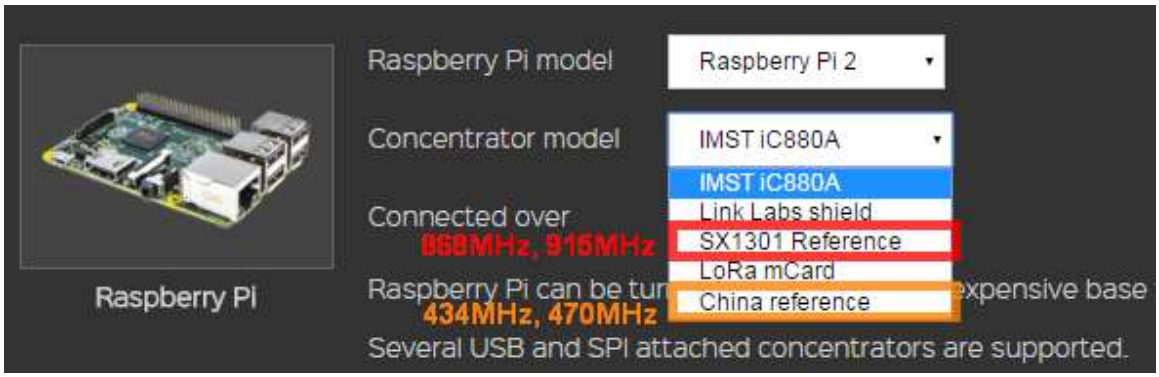

Figure 4-13 loriot Concentrator module

4) Fill mac address

Figure 4-14 loriot mac address

5) Fill gateway location

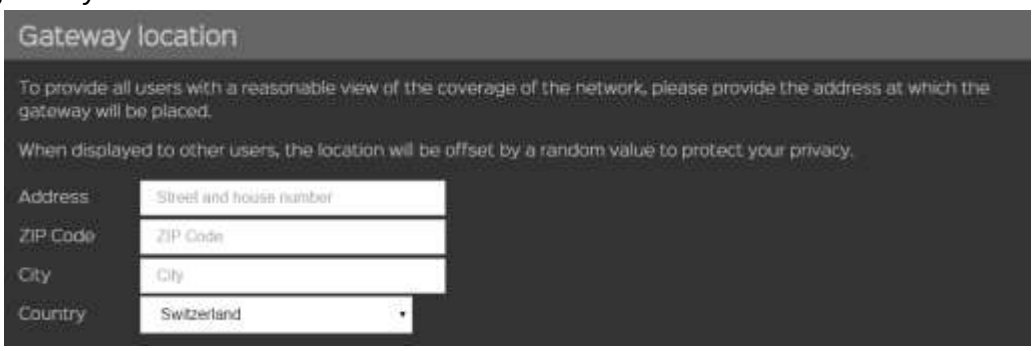

Figure 4-15 loriot gateway location

Figure 4-16 loriot China address

6) Click register button finish gateway registration
7) Download loriot gateway software package from gateway page(eg: loriot_pi_2_sx1301_ref_SPI_0.1.6.tgz), copy the file to RPi 2, extract and run it. After gateway is connected you could see the “online” status from gateway page.

```
$ ./loriot_pi_2_sx1301_ref_SPI_0.1.6
```

```
pi@raspberrypi:~/loriot $ ./loriot_pi_2_sx1301_ref_SPI_0.1.6
 INFO: LORIOT.io Gateway Version 0.1.6
 INFO: Acquired EUI B8-27-EB-FF-FF-1D-9E-02 from interface eth0
 INFO: Connecting to gateway configuration server ...

  CFG: HTTP response  HTTP/1.1 200 OK
  CFG: Content-type   application/json; charset=utf-8
  CFG: Timestamp      Tue, 22 Dec 2015 01:34:48 GMT

 INFO: Parsing configuration file ...
 INFO: 12 configuration parameters found
BOARD: running PUBLIC network, clock fed from radio #1
RADIO: radio 0 enabled, SX1257, center frequency 868200000, RSSI offset -166.0, TX enabled
RADIO: radio 1 enabled, SX1257, center frequency 869200000, RSSI offset -166.0, TX disabled
 INFO: LoRa Multi-SF channel 0 >> Radio 0, IF -100000 Hz, 125 kHz BW Enabled
 INFO: LoRa Multi-SF channel 1 >> Radio 0, IF 100000 Hz, 125 kHz BW Enabled
 INFO: LoRa Multi-SF channel 2 >> Radio 0, IF 300000 Hz, 125 kHz BW Enabled
 INFO: LoRa Multi-SF channel 3 >> Radio 1, IF -350000 Hz, 125 kHz BW Enabled
 INFO: LoRa Multi-SF channel 4 >> Radio 1, IF -150000 Hz, 125 kHz BW Enabled
 INFO: LoRa Multi-SF channel 5 >> Radio 1, IF 325000 Hz, 125 kHz BW Enabled
 INFO: no configuration for Lora multi-SF channel 6
 INFO: no configuration for Lora multi-SF channel 7
 INFO: Lora std channel> radio 0, IF 100000 Hz, 250000 Hz bw, SF 7
 INFO: FSK channel> radio 0, IF 100000 Hz, 250000 Hz bw, 50000 bps datarate
 INFO: Connecting to gateway update server ...

  CFG: HTTP response  HTTP/1.1 204 No Content
  CFG: Content-type   Dec 2015 01:34:48 GMT
 INFO: No update available for current version

 INFO: Gateway HAL Version: 3.1.0; Options: native;

 INFO: Starting LoRa Concentrator
 INFO: Concentrator started, daemonizing ...
pi@raspberrypi:~/loriot $
```

Figure 4-17 loriot gateway binary starts

Figure 4-18 loriot gateway binary download

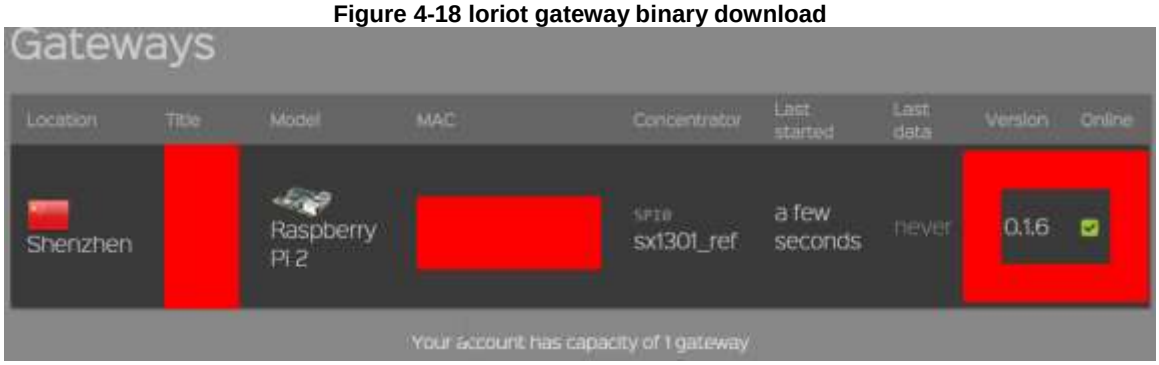

Figure 4-19 loriot gateway status

8) Register device. Fill all required information DevAddr, DevEui, NwkSKey, AppSKey etc and start use.

4.4.7.2 iotcn.semtech.com

Edit ~/risinghf/packet_forwarder/global_conf.json file, set server to iotcn.semtech.com and frequency channels.

A 433MHz global_conf.json file like this:

```
{
    "SX1301_conf": {
        "lorawan_public": true,
        "clksrc": 1,
        "radio_0": {
            "enable": true,
            "type": "SX1255",
            "freq": 433600000,
            "rssi_offset": -166.0,
            "tx_enable": true
        },
        "radio_1": {
            "enable": true,
            "type": "SX1255",
            "freq": 434400000,
            "rssi_offset": -166.0,
            "tx_enable": false
        },
        "chan_multiSF_0": {
            "enable": true,
            "radio": 1,
            "if": -300000
        },
        "chan_multiSF_1": {
            "enable": true,
            "radio": 1,
            "if": -100000
        },
        "chan_multiSF_2": {
            "enable": true,
            "radio": 1,
            "if": 100000
        },
        "chan_multiSF_3": {
            "enable": true,
            "radio": 1,
            "if": 300000
        },
        "chan_multiSF_4": {
            "enable": true,
            "radio": 0,
            "if": -300000
        },
        "chan_multiSF_5": {
            "enable": true,
            "radio": 0,
            "if": -100000
        },
        "chan_multiSF_6": {
            "enable": true,
            "radio": 0,
            "if": 100000
        },
        "chan_multiSF_7": {
            "enable": true,
            "radio": 0,
            "if": 300000
        },
        "chan_Lora_std": {
            "enable": false,
            "radio": 1,
            "if": -200000,
            "bandwidth": 250000,
            "spread_factor": 7
        },
        "chan_FSK": {
            "enable": false,
            "radio": 1,
            "if": 300000,
            "bandwidth": 125000,
            "datarate": 50000
        },
        "tx_lut_0": {
            /* TX gain table, index 0 */
            "pa_gain": 0,
            "mix_gain": 8,
            "rf_power": -6,
            "dig_gain": 0
        },
        "tx_lut_1": {
            /* TX gain table, index 1 */
            "pa_gain": 0,
            "mix_gain": 10,
            "rf_power": -3,
            "dig_gain": 0
        },
        "tx_lut_2": {
            /* TX gain table, index 2 */
            "pa_gain": 0,
            "mix_gain": 12,
            "rf_power": 0,
            "dig_gain": 0
        },
        "tx_lut_3": {
            /* TX gain table, index 3 */
            "pa_gain": 1,
            "mix_gain": 8,
            "rf_power": 3,
            "dig_gain": 0
        },
        "tx_lut_4": {
            /* TX gain table, index 4 */
            "pa_gain": 1,
            "mix_gain": 10,
            "rf_power": 6,
            "dig_gain": 0
        },
        "tx_lut_5": {
            /* TX gain table, index 5 */
            "pa_gain": 1,
            "mix_gain": 12,
            "rf_power": 10,
            "dig_gain": 0
        },
        "tx_lut_6": {
            /* TX gain table, index 6 */
            "pa_gain": 1,
            "mix_gain": 13,
            "rf_power": 11,
            "dig_gain": 0
        },
        "tx_lut_7": {
            /* TX gain table, index 7 */
            "pa_gain": 2,
            "mix_gain": 9,
            "rf_power": 12,
            "dig_gain": 0
        },
        "tx_lut_8": {
            /* TX gain table, index 8 */
```

```
            "pa_gain": 1,
            "mix_gain": 15,
            "rf_power": 13,
            "dig_gain": 0
        },
        "tx_lut_9": {
            /* TX gain table, index 9 */
            "pa_gain": 2,
            "mix_gain": 10,
            "rf_power": 14,
            "dig_gain": 0
        },
        "tx_lut_10": {
            /* TX gain table, index 10 */
            "pa_gain": 2,
            "mix_gain": 11,
            "rf_power": 16,
            "dig_gain": 0
        },
        "tx_lut_11": {
            /* TX gain table, index 11 */
            "pa_gain": 3,
            "mix_gain": 9,
            "rf_power": 20,
            "dig_gain": 0
        },
        "tx_lut_12": {
            /* TX gain table, index 12 */
            "pa_gain": 3,
            "mix_gain": 10,
            "rf_power": 23,
            "dig_gain": 0
        },
        "tx_lut_13": {
            /* TX gain table, index 13 */
            "pa_gain": 3,
            "mix_gain": 11,
            "rf_power": 25,
            "dig_gain": 0
        },
        "tx_lut_14": {
            /* TX gain table, index 14 */
            "pa_gain": 3,
            "mix_gain": 12,
            "rf_power": 26,
            "dig_gain": 0
        },
        "tx_lut_15": {
            /* TX gain table, index 15 */
            "pa_gain": 3,
            "mix_gain": 14,
            "rf_power": 27,
            "dig_gain": 0
        }
    },

    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",
        "server_address": "iotcn.semtech.com",
        "serv_port_up": 1680,
        "serv_port_down": 1680,
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false,
        /* GPS configuration */
        "gps_tty_path": "/dev/ttyAMA0"
    }
}
```

A 470MHz global_conf.json file like this:

```
{
    "SX1301_conf": {
        "lorawan_public": true,
        "clksrc": 1,
        "radio_0": {
            "enable": true,
            "type": "SX1255",
            "freq": 471800000,
            "rssi_offset": -166.0,
            "tx_enable": true
        },
        "radio_1": {
            "enable": true,
            "type": "SX1255",
            "freq": 472600000,
            "rssi_offset": -166.0,
            "tx_enable": false
        },
        "chan_multiSF_0": {
            "enable": true,
            "radio": 1,
            "if": -300000
        },
        "chan_multiSF_1": {
            "enable": true,
            "radio": 1,
            "if": -100000
        },
        "chan_multiSF_2": {
            "enable": true,
            "radio": 1,
            "if": 100000
        },
        "chan_multiSF_3": {
            "enable": true,
            "radio": 1,
            "if": 300000
        },
        "chan_multiSF_4": {
            "enable": true,
            "radio": 0,
            "if": -300000
        },
        "chan_multiSF_5": {
            "enable": true,
            "radio": 0,
            "if": -100000
        },
        "chan_multiSF_6": {
            "enable": true,
            "radio": 0,
            "if": 100000
        },
        "chan_multiSF_7": {
            "enable": true,
            "radio": 0,
            "if": 300000
        },
```

```
        "chan_Lora_std": {
            "enable": false,
            "radio": 1,
            "if": -200000,
            "bandwidth": 250000,
            "spread_factor": 7
        },
        "chan_FSK": {
            "enable": false,
            "radio": 1,
            "if": 300000,
            "bandwidth": 125000,
            "datarate": 50000
        },
        "tx_lut_0": {
            /* TX gain table, index 0 */
            "pa_gain": 0,
            "mix_gain": 8,
            "rf_power": -6,
            "dig_gain": 0
        },
        "tx_lut_1": {
            /* TX gain table, index 1 */
            "pa_gain": 0,
            "mix_gain": 10,
            "rf_power": -3,
            "dig_gain": 0
        },
        "tx_lut_2": {
            /* TX gain table, index 2 */
            "pa_gain": 0,
            "mix_gain": 12,
            "rf_power": 0,
            "dig_gain": 0
        },
        "tx_lut_3": {
            /* TX gain table, index 3 */
            "pa_gain": 1,
            "mix_gain": 8,
            "rf_power": 3,
            "dig_gain": 0
        },
        "tx_lut_4": {
            /* TX gain table, index 4 */
            "pa_gain": 1,
            "mix_gain": 10,
            "rf_power": 6,
            "dig_gain": 0
        },
        "tx_lut_5": {
            /* TX gain table, index 5 */
            "pa_gain": 1,
            "mix_gain": 12,
            "rf_power": 10,
            "dig_gain": 0
        },
        "tx_lut_6": {
            /* TX gain table, index 6 */
            "pa_gain": 1,
            "mix_gain": 13,
            "rf_power": 11,
            "dig_gain": 0
        },
        "tx_lut_7": {
            /* TX gain table, index 7 */
            "pa_gain": 2,
            "mix_gain": 9,
            "rf_power": 12,
            "dig_gain": 0
        },
        "tx_lut_8": {
            /* TX gain table, index 8 */
            "pa_gain": 1,
            "mix_gain": 15,
            "rf_power": 13,
            "dig_gain": 0
        },
        "tx_lut_9": {
            /* TX gain table, index 9 */
            "pa_gain": 2,
            "mix_gain": 10,
            "rf_power": 14,
            "dig_gain": 0
        },
        "tx_lut_10": {
            /* TX gain table, index 10 */
            "pa_gain": 2,
            "mix_gain": 11,
            "rf_power": 16,
            "dig_gain": 0
        },
        "tx_lut_11": {
            /* TX gain table, index 11 */
            "pa_gain": 3,
            "mix_gain": 9,
            "rf_power": 20,
            "dig_gain": 0
        },
        "tx_lut_12": {
            /* TX gain table, index 12 */
            "pa_gain": 3,
            "mix_gain": 10,
            "rf_power": 23,
            "dig_gain": 0
        },
        "tx_lut_13": {
            /* TX gain table, index 13 */
            "pa_gain": 3,
            "mix_gain": 11,
            "rf_power": 25,
            "dig_gain": 0
        },
        "tx_lut_14": {
            /* TX gain table, index 14 */
            "pa_gain": 3,
            "mix_gain": 12,
            "rf_power": 26,
            "dig_gain": 0
        },
        "tx_lut_15": {
            /* TX gain table, index 15 */
            "pa_gain": 3,
            "mix_gain": 14,
            "rf_power": 27,
            "dig_gain": 0
        }
    },

    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",
        "server_address": "iotcn.semtech.com",
        "serv_port_up": 1680,
        "serv_port_down": 1680,
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
```

```
        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false,
        /* GPS configuration */
        "gps_tty_path": "/dev/ttyAMA0"
    }
}
```

After global_conf.json file is configured done. Run below commands to connect the gateway to Semtech IoT server to test.

```
$ cd ~/risinghf/packet_forwarder/gps_pkt_fwd
$ sudo ../reset_pkt_fwd.sh start
$ ./gps_pkt_fwd
```

```
Log (470MHz):
*** GPS Packet Forwarder for Lora Gateway ***
Version: 2.2.0
*** Lora concentrator HAL library version info ***
Version: 3.2.0;
***
INFO: Little endian host
INFO: found global configuration file global_conf.json, parsing it
INFO: global_conf.json does contain a JSON object named SX1301_conf, parsing SX1301 parameters
INFO: lorawan_public 1, clksrc 1
INFO: Configuring TX LUT with 16 indexes
INFO: radio 0 enabled (type SX1255), center frequency 471800000, RSSI offset -166.000000, tx enabled 1
INFO: radio 1 enabled (type SX1255), center frequency 472600000, RSSI offset -166.000000, tx enabled 0
INFO: Lora multi-SF channel 0>  radio 1, IF -300000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 1>  radio 1, IF -100000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 2>  radio 1, IF 100000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 3>  radio 1, IF 300000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 4>  radio 0, IF -300000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 5>  radio 0, IF -100000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 6>  radio 0, IF 100000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 7>  radio 0, IF 300000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora standard channel 8 disabled
INFO: FSK channel 8 disabled
INFO: global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to AA555A0000000000
INFO: server hostname or IP address is configured to "localhost"
INFO: upstream port is configured to "1680"
INFO: downstream port is configured to "1680"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
INFO: GPS serial port path is configured to "/dev/ttyAMA0"
INFO: found local configuration file local_conf.json, parsing it
INFO: redefined parameters will overwrite global parameters
INFO: local_conf.json does not contain a JSON object named SX1301_conf
INFO: local_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to AA555A0000000101
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
WARNING: [main] impossible to open /dev/ttyAMA0 for GPS sync (check permissions)
INFO: [main] concentrator started, packet can now be received
```

4.4.8 Commands Summary

```
// Under uart terminal
$ cp /boot/config.txt ~/boot-config.txt.bak
$ sudo echo -e "\ndtoverlay=spi-bcm2708\ndtparam=spi=on\n" >> /boot/config.txt
$ sudo reboot

// Check ip and switch to SSH terminal(use either Putty or similar SSH tool)
$ ifconfig

// Under SSH terminal
$ sudo apt-get update

// Download source code
$ mkdir ~/risinghf
$ cd ~/risinghf
$ git clone https://github.com/Lora-net/lora_gateway.git
$ git clone https://github.com/Lora-net/packet_forwarder.git
$ mkdir -p lora_gateway/util_spectral_scan/obj

// Compile lora_gateway and packet_forwarder
$ cd ~/risinghf/lora_gateway
$ make
$ cd ~/risinghf/packet_forwarder
$ make

// Test if hardware connection is good
$ cd ~/risinghf/packet_forwarder/
$ sudo ./reset_pkt_fwd.sh start
$ cd ~/risinghf/lora_gateway/libloragw/
$ ./test_loragw_reg

// Semtech IoT server
$ cd ~/risinghf/packet_forwarder/gps_pkt_fwd
$ sudo ../reset_pkt_fwd.sh start
$ ./gps_pkt_fwd

// loriot server
$ ./loriot_pi_2_sx1301_ref_SPI_0.1.6
```

4.5 Troubleshooting

Q1: Hardware and connections are good, but software doesn't work?
A1: Check if /boot/config.txt file are configured properly. Below two lines must be added to /boot/config.txt. RPi need reboot after configured.

```
dtoverlay=spi-bcm2708
dtparam=spi=on
```

Q2: Can't compile lora_gateway successfully, error "**Fatal error: can't create obj/util_spectral_scan.o**"

A2: This error is caused by "obj" directory missing, run below command to fix it.

```
cd ~/risinghf
mkdir -p lora_gateway/util_spectral_scan/obj
```

Revision

V1.8 2016-05-25
+Update with spec of RHF0M301-868 and RHF0M301-915

V1.7 2016-04-05
+Update with spec of RHF0M301V1A-780

V1.6 2016-03-19
+Update with spec of RHF0M301V1A-470
+Update with baud rate for UART connection

V1.5 2016-03-17
+ Sync with latest loriot server

V1.4 2016-02-16
+ Fix typo
+ Fix /boot/config.txt modification wrong sequence
+ Validate with Raspberry Pi B+

V1.3 2015-12-22
+ Update with test results in Temperature Chamber
+ Improve some information of HW design
+ Detailed server configuration section

V1.2 2015-12-20
+ Add Chapter “Use with Raspberry Pi”
+ Review adjust tables

V1.1 2015-12-15
+ Update with data of RHF0M301V1A-434
+ Review adjust tables

V1.0 2015-11-25
+Draft creation

Please Read Carefully:

Information in this document is provided solely in connection with RisingHF products. RisingHF reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All RisingHF products are sold pursuant to RisingHF's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the RisingHF products and services described herein, and RisingHF assumes no liability whatsoever relating to the choice, selection or use of the RisingHF products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by RisingHF for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN RISINGHF'S TERMS AND CONDITIONS OF SALE RisingHF DISCLAIMS ANY EXPRESS OR IMPLIEDWARRANTY WITH RESPECT TO THE USE AND/OR SALE OF RisingHF PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIEDWARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWSOF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

RISINGHF PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE RISINGHF PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER'S SOLE RISK, EVEN IF RISINGHF HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY RISINGHF AS BEING INTENDED FOR "AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL" INDUSTRY DOMAINS ACCORDING TO RISINGHF PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of RisingHF products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by RisingHF for the RisingHF product or service described herein and shall not create or extend in any manner whatsoever, any liability of RisingHF.

RisingHF and the RisingHF logo are trademarks or registered trademarks of RisingHF in various countries.
Information in this document supersedes and replaces all information previously supplied.
The RisingHF logo is a registered trademark of RisingHF. All other names are the property of their respective owners.

http://www.risinghf.com