GW300 User Manual

V1.2

Document information

Info	Content
Keywords	*RisingHF, LoRaWAN, Installation, Lighting surge protection*
Abstract	This document describes how to install and use RisingHF GW300 LoRaWAN IoT Gateway.

1 Purpose

GW300 is an IoT gateway/concentrator which integrates POE, LTE/4G, GPS and LoRa, designed and manufactured by Shenzhen Friendcom Technology Development Co., Ltd.

This document will describes how to use and configure GW300, for end-user and developer usage. Covers:

- Get started
  - Log in
  - Device file system structure introduction
- How to use GW300LTE/4G cellular network?
  - Auto APN configuration
- How to use GW300 LoRa function, and how to do secondary development?
- How to use GW300 GPS module?
- How to connect to LoRaWAN server?
  - General Packet Forwarder server
  - Loriot LoRaWAN server
  - Aisenz Loraflow server
- Semtech Packet Forwarder advanced usage
  - Packet Forwarder json configuration file details
- How to deploy GW300 cross compile environment?
  - Hello world example
- How to use Friendcom supplied reverse SSH tool to get help
- FAQs

2 Get Started

New GW300 firmware use Systemd structure. User could use systemctl and journalctl to manage the integrated services and check log.

2.1 Log In

Default account information:

User: rxhf

Password: risinghf

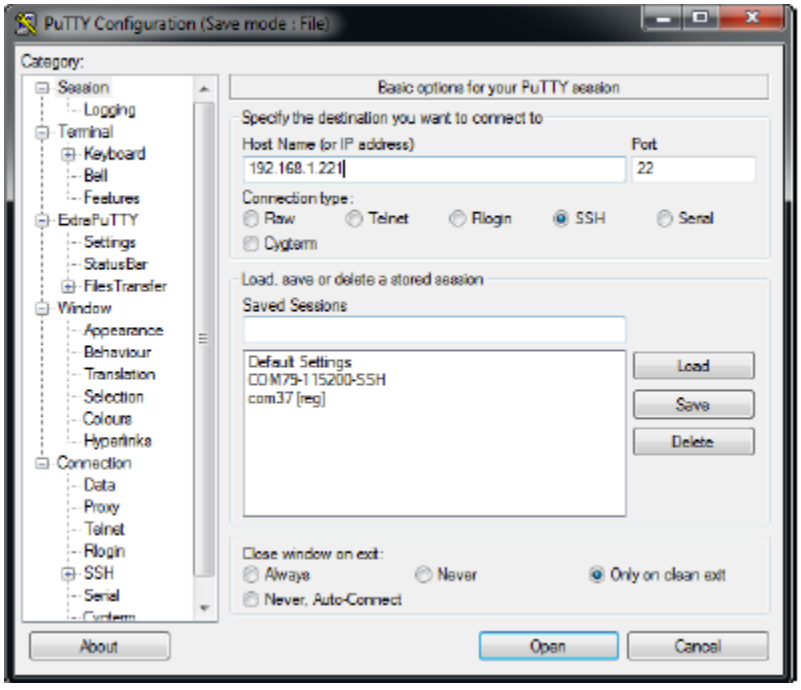
Connect GW300 with DHCP router, log in router or scan IP to get GW300 ip. Hostname of GW300 is "rhf2s008".

After get ip, user could use SSH tool to log the device check or set configuration.

For example, use PuTTY under Windows and use ssh command under Linux. SSH port is 22.

To log in device of which ip is 192.168.1.221.

Windows:



Linux

ssh rxhf@192.168.1.221

2.2 Device File System

/home/rxhf and /opt are very important directory for user. /opt contains working directory of most internal services (Eg. gw, pktfwd, pingloop). /home/rxhf contains original software package and working directory of services which is not contained by /opt folder (Eg. lorabridge and lrgateway of

Aisenze/loraflow.io), and lora_gateway and packet_forwarder source code from Semtech etc.

2.2.1 /home/rxhf

2.2.1.1 /home/rxhf/aisenz

Working directory of lrgateway and lorabridge services for Aisenz loraflow.io server. And more, /home/rxhf/aisenz/pktfwd contains json configuration files of Aisenz packet forwarder (Default: CN470). User could create new configuration to use different frequency plan.

2.2.1.2 /home/rxhf/loriot

Loriot server related service working directory.

2.2.1.3 /home/rxhf/risinghf

Legacy folder contains LTE module ME909 dial script, which is replaced by lte service. And lora testing programs. It is replaced, does not suggest user to use it anymore.

2.2.1.4 /home/rxhf/semtech Semtech Github source code included

https://github.com/Lora-net/lora_gateway.git https://github.com/Lora-net/packet_forwarder.git

2.2.1.5 /home/rxhf/utilities

GW300 services install package. In case of situations user can't restore the configuration of a service, related below packages can be used to restore it.

aisenz-rhf2s008-install-package	aisenz loraflow.io software package
loriot-rhf2s008-install-package	loriot software package
pktfwd-rhf2s008-install-package	Semtech Packet Forwarder software package
pingloop-install-package	pingloop software package
rssh-rxhf-install-package	RisingHF Reverse SSH software package
lte-install-package	LTE/4G cellular network software package
tool-install-package	System command software package, include gwrst/gpsctl tools

2.2.1.6 /home/rxhf/changelog.md

Firmware change log.

2.2.1.7 /home/rxhf/version

Firmware version.

cat /home/rxhf/version

2.2.2 /opt

/opt contains working directory of most internal services (Eg. gw, pktfwd, pingloop)

2.2.2.1 /opt/risinghf

/opt/risinghf contains packet forwarder services pktfwd working directory.

2.2.2.2 /opt/loriot

/opt/loriot contains loriot service and loriot-gw working directory.

2.2.2.3 /opt/pingloop.sh

pingloop service working directory.

2.3 LoRaWAN Server Solutions

According to the different request, user could choose which server to connect to. GW300 integrates several third party manufacture's server like Loriot.io and Loraflow.io, and also preinstall the packet forwarder program for customer's convenience.

NOTE: At the same time, a gateway can only connect to only one server. When user needs to switch between servers, it must be sure that service which is not used is closed.

2.3.1 Packet Forwarder

Packet Forwarder is a bridge program for lora gateway and server which is provided by Semtech, supplies basic LoRa packet forwarding feature for simple LoRaWAN network. Packet Forwarder itself doesn't handle LoRaWAN frame analysis. GW300 integrates two versions of Packet Forwarder, the legacy gps_pkt_fwd (v1) and lora_pkt_fwd (v2).

Application	PROTOCOL.TXT Version	Frame Header Version
gps_pkt_fwd	V1.2	1
lora_pkt_fwd	V1.3	2

TIPS: PROTOCOL.TXT is accessible at GitHub https://github.com/Lora-net/packet_forwarder, checkout different commit to check the specified version.

Preinstalled Packet Forwarder locates at /opt/risinghf/pktfwd directory.

```
├── global_conf_as920.json
├── global_conf_cn433.json
├── global_conf_cn470.json
├── global_conf_cn780.json
├── global_conf_eu434.json
├── global_conf_eu868.json
├── global_conf_jp920.json
├── global_conf.json -> global_conf_eu868.json
├── global_conf_ttn920.json
├── global_conf_us915.json
├── gps_pkt_fwd
├── install.sh
├── local_conf.json -> local_conf_localhost_1680.json
├── local_conf_localhost_1580.json
├── local_conf_localhost_1680.json
├── lora_pkt_fwd
├── pktfwdbin -> gps_pkt_fwd
├── pktfwd.service
├── pktfwd.sh
└── update_gwid.sh
```

It depends on the server which packet forwarder to support. User need be responsible to choose which protocol to support. *.json files under pkt service working directory are gps_pkt_fwd / lora_pkt_fwd configuration files. Among the json files, global_conf.json and local_conf.json are files which is used by Packet Forwarder, when using it user could use symbol link to point global_conf.json and

local_conf.json to specified json files to choose different frequency plan and server.

global_conf.json and local_conf.json are in the same format. local_conf.json has higher priority than global_conf.json, Packet Forwarder will use configurations contained by local_conf.json overwrite the global_conf.json ones.

json configuration file example: (868MHz)

```
{
    "SX1301_conf": {
        "lorawan_public": true,
        "clksrc": 1,
        "antenna_gain": 0,
        "radio_0": {
            "enable": true,
            "type": "SX1257",
            "freq": 867500000,
            "rssi_offset": -166.0,
            "tx_enable": true,
            "tx_freq_min": 863000000,
            "tx_freq_max": 870000000
        },
        "radio_1": {
            "enable": true,
            "type": "SX1257",
            "freq": 868500000,
            "rssi_offset": -166.0,
            "tx_enable": false
        },

        // SX1301 MultiSF, standard LoRa and FSK channels
        "chan_multiSF_0": { "enable": true, "radio": 1, "if": -400000 },
        "chan_multiSF_1": { "enable": true, "radio": 1, "if": -200000 },
        "chan_multiSF_2": { "enable": true, "radio": 1, "if": 0 },
        "chan_multiSF_3": { "enable": true, "radio": 0, "if": -400000 },
        "chan_multiSF_4": { "enable": true, "radio": 0, "if": -200000 },
        "chan_multiSF_5": { "enable": true, "radio": 0, "if": 0 },
        "chan_multiSF_6": { "enable": true, "radio": 0, "if": 200000 },
        "chan_multiSF_7": { "enable": true, "radio": 0, "if": 400000 },
        "chan_Lora_std":  { "enable": true, "radio": 1, "if": -200000, "bandwidth": 250000, "spread_factor": 7 },
        "chan_FSK":       { "enable": true, "radio": 1, "if": 300000, "bandwidth": 125000, "datarate": 50000 },

        // RHF0M301-868
        "tx_lut_0":  { "rf_power": -1, "dig_gain": 0, "mix_gain": 8,  "pa_gain": 1 },
        "tx_lut_1":  { "rf_power": 2,  "dig_gain": 0, "mix_gain": 10, "pa_gain": 1 },
        "tx_lut_2":  { "rf_power": 5,  "dig_gain": 0, "mix_gain": 12, "pa_gain": 1 },
        "tx_lut_3":  { "rf_power": 6,  "dig_gain": 0, "mix_gain": 8,  "pa_gain": 2 },
        "tx_lut_4":  { "rf_power": 8,  "dig_gain": 0, "mix_gain": 9,  "pa_gain": 2 },
        "tx_lut_5":  { "rf_power": 9,  "dig_gain": 0, "mix_gain": 10, "pa_gain": 2 },
        "tx_lut_6":  { "rf_power": 11, "dig_gain": 0, "mix_gain": 11, "pa_gain": 2 },
        "tx_lut_7":  { "rf_power": 12, "dig_gain": 0, "mix_gain": 12, "pa_gain": 2 },
        "tx_lut_8":  { "rf_power": 14, "dig_gain": 0, "mix_gain": 13, "pa_gain": 2 },
        "tx_lut_9":  { "rf_power": 15, "dig_gain": 0, "mix_gain": 8,  "pa_gain": 3 },
        "tx_lut_10": { "rf_power": 17, "dig_gain": 0, "mix_gain": 9,  "pa_gain": 3 },
        "tx_lut_11": { "rf_power": 18, "dig_gain": 0, "mix_gain": 10, "pa_gain": 3 },
        "tx_lut_12": { "rf_power": 20, "dig_gain": 0, "mix_gain": 11, "pa_gain": 3 },
        "tx_lut_13": { "rf_power": 22, "dig_gain": 0, "mix_gain": 12, "pa_gain": 3 },
        "tx_lut_14": { "rf_power": 23, "dig_gain": 0, "mix_gain": 13, "pa_gain": 3 },
        "tx_lut_15": { "rf_power": 25, "dig_gain": 0, "mix_gain": 15, "pa_gain": 3 }

    },

    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",
```

```
        /* change with default server address/ports, or overwrite in local_conf.json */
        "server_address": "localhost",
        "serv_port_up": 1680,
        "serv_port_down": 1680,

        /* adjust the following parameters for your network */
        "keepalive_interval": 10,
        "stat_interval": 3600,
        "push_timeout_ms": 100,
        "autoquit_threshold": 5,

        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false,

        /* GPS reference coordinates */
        "ref_latitude": 0.0,
        "ref_longitude": 0.0,
        "ref_altitude": 0,

        /* Beaconing parameters */
        //"gps tty path": "/dev/ttyAMA0",
        "beacon_period": 128,
        "beacon_freq_hz": 869525000
    }
}
```

Packet Forwarder configuration itself is a json object (enclosed by "{" and "}"), the root object contains 2 sub-objects SX1301_conf and gateway_conf. SX1301_conf object is for LoRa radio configuration, gateway_conf is for network communication configuration (server address, gateway id etc).

Because global_conf.json file already contains all configurations of Packet Forwarder, local_conf.json is mostly used for items which need to be changed frequently (Eg. gateway_ID), like below:

local_conf.json example:

```
{
    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",

        /* change with default server address/ports, or overwrite in local_conf.json */
        "server address": "localhost",
        "serv_port_up": 1680,
        "serv_port_down": 1680
    }
}
```

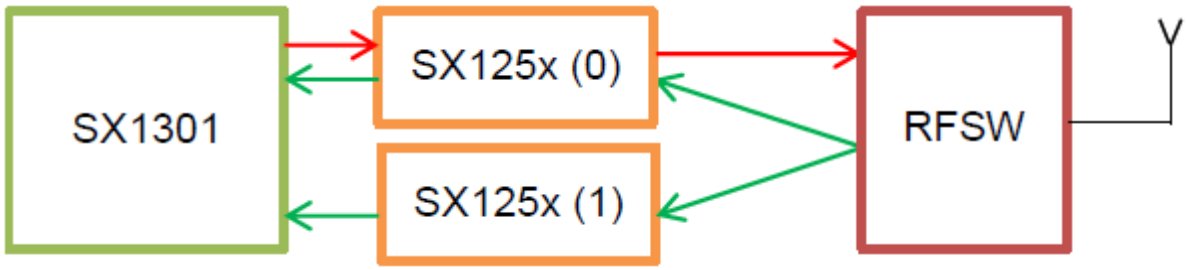
In combination with description above, SX1301 has below features:

1. One SX1301 supports 2 Radios. SX1255 or SX1257 can be used.
2. One SX1301 could support maximum 10 channels, of which 8 Multi-SF channels, 1 standard channel and 1 FSK channel.
3. One single SX125x chip has below limitations.

Lower Side Channel Bandwidth	Upper Side Channel Bandwidth	SX125x Bandwidth
125KHz	125KHz	925KHz
250KHz	250KHz	1MHz
500KHz	500KHz	1.1MHz
125KHz	250KHz	962.5KHz
250KHz	125KHz	962.5KHz
500KHz	250KHz	1.05MHz

Lower Side Channel Bandwidth	Upper Side Channel Bandwidth	SX125x Bandwidth
250KHz	500KHz	1.05MHz
500KHz	125KHz	1.0125MHz
125KHz	500KHz	1.0125MHz

LoRa Module hardware structures:



2.3.1.1 Customized channels

User could follow below principles to customize different frequency channel.

1. Split defined channels to two group for radio_0(radio_a) and radio_1(radio_b), calculate central frequency. Set SX1301_conf.radio_0.freq and SX1301_conf.radio_1.freq the new value.
2. Choose $S_{X1301\_conf.radio\_0.type}$ and SX1301_conf.radio_1.type depends on the new radio frequency. If it is higher than 520MHz set radio type SX1257, if it is less than 520MHz set radio type SX1255. (Note: real products have a narrow band support, please follow specified specification for the details)
3. Refer to the RHF-DS01603 document to get device configuration parameters, include RSSI offset and radio frequency TX power. Set SX1301_conf.radio_0.rssi_offset and SX1301_conf.radio_1.rssi_offset and tx_lut_xxx value.
4. Calculate IF channel offset value. Modify SX1301_conf.chan_multiSF_xxx, SX1301_conf.chan_Lora_std and SX1301_conf.chan_FSK object. Channel configuration follows below rules.
   a. Each IF channel can be associated with either radio0 or radio1 freely.
   b. Each IF channel offset can't be out of range the SX125x bandwidth limitation, or the IF channel setting will be invalid.
   c. Each IF channel can be enable or disable independently.
5. So the example json file contains below channels:
   a. CH0 868.1MHz Multi-SF
   b. CH1 868.3MHz Multi-SF
   c. CH2 868.5MHz Multi-SF

d. CH3 867.1MHz Multi-SF
e. CH4 867.3MHz Multi-SF
f. CH5 867.5MHz Multi-SF
g. CH6 867.7MHz Multi-SF
h. CH7 867.9MHz Multi-SF
i. CH8 868.3MHz LoRa Standard SF7/250KHz
j. CH9 868.8MHz FSK 50Kbps

2.3.1.2 Configure server address

Edit local_conf.json file. Modify server_address object to configure server address, modify serv_port_up object to configure uplink port, modify serv_port_down object to configure downlink port.

// local_conf.json file example:

```
{
    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",
        "server_address": "localhost",
        "serv_port_up": 1680,
        "serv_port_down": 1680
    }
}
```

Some known packet forwarder serer address list:

Address	Uplink Port	Downlink Port	Packet Forward Version	Supplier	Band
lot.semtech.com	1680	1680	V1/V2	Semtech	EU868
us01-iot.semtech.com	1780	1780	V1/V2	Semtech	US915
cn1.loriot.io	1780	1780	V1/V2	Loriot	All
ap1.loriot.io	1780	1780	V1/V2	Loriot	All
au1.loriot.io	1780	1780	V1/V2	Loriot	All
eu1.loriot.io	1780	1780	V1/V2	Loriot	All
sa1.loriot.io	1780	1780	V1/V2	Loriot	All
us1.loriot.io	1780	1780	V1/V2	Loriot	All
router.eu.thethings.network	1700	1700	V1	TTN	EU433 EU868
router.us.thethings.network	1700	1700	V1	TTN	US915
router.cn.thethings.network	1700	1700	V1	TTN	CN470 CN780
router.au.thethings.network	1700	1700	V1	TTN	AU915

2.3.1.3 Choose preinstalled frequency plan

Use ln command to make global_conf.json file points to other json file to choose preinstalled frequency plan. User can also create their own json file depends on the previous defined json file.

sudo ln -sf global_conf_xxx.json global_conf.json

NOTE:

1. Replace global_conf_xxx.json with the real file name
2. Use `ls -l /opt/risinghf/pktfwd` command to check which frequency plan is contained in /opt/risinghf/pktfwd
3. Physical frequency which is supported by RHF2S008P4G is decided by the hardware. For example RHF2S008P4G-470 only support 470MHz frequency band, doesn't support 433MHz, 868MHz etc...
4. Other undocumented channels plans, user could calculate each channel frequency according to the previous description.

CH	eu868	us915	eu433	cn780	as920	cn470
0	867.1	902.3	433.175	779.5	923.2	470.3
1	867.3	902.5	433.375	779.7	923.4	470.5
CH	eu868	us915	eu433	cn780	as920	cn470
2	867.5	902.7	433.575	779.9	923.6	470.7
3	867.7	902.9	433.775	780.1	923.8	470.9
4	867.9	903.1	433.975	780.3	924.0	471.1
5	868.1	903.3	434.175	780.5	924.2	471.3
6	868.3	903.5	434.375	780.7	924.4	471.5
7	868.5	903.7	434.575	780.9	924.6	471.7
8	868.3 BW250 SF7	903.0 BW500 SF8	OFF	OFF	OFF	OFF
9	868.8 FSK 50Kbps	OFF	OFF	OFF	OFF	OFF

2.3.1.4 Choose Packet Forwarder version

sudo su

cd /opt/risinghf/pktfwd

Choose V1 packet forwarder

sudo ln -sf gps_pkt_fwd pktfwdbin

Choose V2 packet forwarder

sudo ln -sf lora_pkt_fwd pktfwdbin

2.3.1.5 Start Packet Forwarder service

Execute commands below to start pktfwd service:

sudo systemctl enable pktfwd

sudo systemctl restart pktfwd

2.3.1.1 Stop Packet Forwarder service

Execute commands below to stop pktfwd service:

sudo systemctl disable pktfwd

sudo systemctl stop pktfwd

2.3.2 Loriot Server (loriot.io)

Loriot is a LoRaWAN server supplier based at Switzerland, Loriot server supply free test account for user. Main site http://www.loriot.io. Please note, free account has some limitation like active downlink, OTAA etc.

Please read online documentation before get started. (Subdomain name will be different if you use other region servers)

https://cn1.loriot.io/home/documentation.html

2.3.2.1 Register GW300 Gateway

(1) Get MAC address, which is in format xx:xx:xx:xx:xx:xx. MAC address is sticky on the side of the GW300 device.

(2) Access http://cn1.loriot.io register account, log in directly if you have already gotten one.

(3) Click Dashboard -> Gateways -> Add Gateway, choose GW300

(4) Radio front-end configuration. Match the device type. Available options:

a) 868/915 MHz (SX1257)

b) 434/470/780 MHz (SX1255)

c) Note: RHF2S008P4G-780MHz gateway need choose "868/915 MHz (SX1257)"

(5) Scroll down, fill in MAC address, and set gateway location information.

(6) Click "Register GW300" finish register

(7) Click "Go to the gateway detail page" or click "gateway xx:xx:xx:xx:xx:xx" from plane at the left to enter gateway configuration page.

(8) According to the gateway frequency band to choose a band.

(9) Register gateway finish.

(10) Connect Ethernet cable, power up GW300.

(11) Log in to the gateway and start Loriot service and start test.

2.3.2.2 Start Loriot Service

GW300 connect to CN1 server by default (not auto start), switch to server manually when you need. Example to switch to AP1 server:

sudo su

cd /opt/loriot/bin

ln -sf ap1 lrt

Replace ap1 to af1 ap1 au1 cn1 eu1 sa1 us1 can choose a different region server

Set auto start:

sudo systemctl enable loriot-gw

Start:

sudo systemctl start loriot-gw

Stop:

sudo systemctl stop loriot-gw

Disable auto start:

sudo systemctl disable loriot-gw

Note: Once you set to auto start the service, please make sure disable the auto start of loriot-gw service. In case of the gateway service collision.

2.3.2.3 Configure Gateway Frequency

Set gateway frequency, open loriot console find gateway page. Choose frequency plan. Check online documentation for more details.

https://cn1.loriot.io/home/documentation.html#docu/frequency-plan

2.3.2.4 Loriot Firmware Upgrade

Please download loriot-risinghf-rhf2s008-xxxxxx-SPI-0-latest.bin file, and replace /opt/loriot/bin/loriot-gw

sudo su

cd /opt/loriot/bin

wget URL -O loriot-gw

Please get the actual URL from the gateway page.

2.3.3 Aisenz Server (loraflow.io)

Loraflow.io is a Chinese LoRaWAN server. Official website http://loraflow.io.

Check loraflow online document first before get started,.

https://loraflow.io/static/docs/zh/index.html

2.3.3.1 Register GW300 Gateway

(1) Get MAC address, which is in format xx:xx:xx:xx:xx:xx. MAC address is sticky on the side of the GW300 device.

(2) Access https://loraflow.io register account, log in directly if you have already gotten one.

(3) Click gateway management page, click add button to register a new gateway.

(4) Gateway type GW300

(5) Gateway name, set a customized one

(6) MAC Address, fill in the address read from the device

(7) (Optional) Fill in gateway location

(8) Click gateway to configure

2.3.3.2 Start lorabridge and lrgateway service

sudo systemctl enable lorabridge

sudo systemctl enable lrgateway

sudo systemctl restart lrgateway

sudo systemctl restart lorabridge

2.3.3.3 Start/Stop loraflow SDK

Start:

sudo systemctl start lorabridge

sudo systemctl start lrgateway

Stop:

sudo systemctl stop lorabridge

sudo systemctl stop lrgateway

Disable auto start:

sudo systemctl disable lorabridge

sudo systemctl disable lrgateway

Enable auto start:

sudo systemctl enable lorabridge

sudo systemctl enable lrgateway

2.3.3.1 Configure Gateway Frequency

loraflow.io default working directory /home/rxhf/aisenz/pktfwd.

cd /home/rxhf/aisenz/pktfwd

/home/rxhf/aisenz/pktfwd contains different frequency plans. Among them Loraflow.io SDK reuse Semtech packet forwarder to control gateway, check 2.3.1 Packet Forwarder about the detailed configuration.

2.3.3.2 loraflow SDK upgrade

Download lorasdk.zip. Refer to loraflow online document to upgrade SDK
https://loraflow.io/static/docs/zh/gateway/SDKInstall.html

3 Advanced Usage

3.1 Cellular Network (LTE/4G)

GW300 device embedded service which is named lte, and is an auto start service. Use below command to check log of lte service:

sudo journalctl -f -n 200 -u lte

Feature:

- ✓ Auto generate APN information according to the SIM card
- ✓ SIM card hot plug
- ✓ Not support roaming

After dial-up success, system will generate ppp0 device.

```
rxhf@rhf2s008:~$ ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.65.6.136  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:56 (56.0 B)  TX bytes:80 (80.0 B)
```

When connected, ltestatus command could be used to check the signal quality

```
sudo ltestatus
OK, -85, 460, 01, 012,          // -85 is the signal strength
```

If not connected, it shows:

ERROR: Celluar network disconnected

3.1.1 pingloop

When using cellular network, to prevent LTE/4G/3G network to enter sleep mode, we supply a possible solution, the pingloop service. This service ping a specified network several seconds to keep cellular network active, but please note this possible solution takes more data flow.

/opt/pingloop.sh:

```
#!/bin/bash

ADDRESS="gw.risinghf.com"
ping -q -W 8 -i 2 -s 8 -c 10 $ADDRESS
exit $?
```

Edit /opt/pingloop.sh and replace the target address to the one you need.

Start pingloop service and enable auto start:

sudo systemctl enable pingloop

sudo systemctl restart pingloop

Check pingloop service log

sudo journalctl -f -n 50 -u pingloop

3.2 Use GPS module

GW300 embedded on board GPS module (MAX-7Q), which is necessary for LoRaWAN Class B network.

Access gps module through /dev/serial1 device, default baud rate 9600.

Simply test GPS module by executing below commands:

```
sudo gpsctl on
stty -F /dev/serial1 9600 -raw -echo
cat < /dev/serial1
```

It returns below messages:

```
$GPRMC,095521.00,V,,,,,,,100417,,,N*74

$GPVTG,,,,,,,,,N*30

$GPGGA,095521.00,,,,,0,00,99.99,,,,,,*6C

$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30

$GPGSV,4,1,13,01,70,146,,03,09,158,,07,75,257,,08,36,031,*77

$GPGSV,4,2,13,09,15,217,,11,80,034,,16,06,101,,17,14,253,*7D

$GPGSV,4,3,13,22,17,134,,23,05,183,,27,06,051,,28,20,314,*7B

$GPGSV,4,4,13,30,41,313,*4C

$GPGLL,,,,,095521.00,V,N*40
```

3.3 Remote Support

GW300 integrates an rssh service, with which customer could enable Friendcom to access your GW300 device remotely to supply technical support.

Note: The service is closed by default, and can only be used by Friendcom to support customer when necessary.

One time start rssh service (Invalid after restarting)

```
sudo systemctl restart rssh
```

Check log to get port number, execute below command and wait after a while:

```
sudo journalctl -f -n 50 -u rssh
```

Then it shows log like below, include "Allocated port â€¦" message

When asking remote support, please send your screenshot to support@friendcom.com, and explain your trouble, Friendcom technical support will help you diagnose.

4 Build Development Environment

This chapter will describe how to compile "Hello World" program and build GW300 development environment.

Hello World c program:

```c
#include <stdio.h>

int main(int argc, char **argv)
{
        printf("Hello world!\n");
        return 0;
}
```

4.1 Compile Natively

GW300 gateway embedded GCC tool chains natively. User could upload the program to GW300 to compile.

1. Log in GW300
2. Save Hello world code to main.c
3. Compile by executing below commands

   ```
   gcc main.c -o main
   ```

4. Run main program
   ./main

```
root@rhf2s008:/tmp# ./main
Hello world!
```

4.2 Cross compile

Cross compile environment must be built under Linux system, other OS is not supported.

1. Download tool chain. Address: https://github.com/raspberrypi/tools
   ```
   git clone --depth 1 https://github.com/raspberrypi/tools
   ```
2. Add tool chain to system path
   32bits machine
   ```
   export PATH=$PATH:/path/to/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin
   ```
   64bits machine
   ```
   export PATH=$PATH:/path/to/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin
   ```
   Add above command to ~/.bashrc file if you need add the tool chain to you OS permanently. This make sure the terminal adds toolchain to PATH environment variable each time it is opened.
3. Open terminal
   ```
   arm-linux-gnueabihf-gcc --version
   ```
   If you get outputs like below, the tool is installed successfully.
   ```
   arm-linux-gnueabihf-gcc (crosstool-NG linaro-1.13.1+bzr2650 - Linaro GCC 2014.03) 4.8.3 20140303 (prerelease)
   Copyright (C) 2013 Free Software Foundation, Inc.
   This is free software; see the source for copying conditions.  There is NO
   warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
   ```
4. Save Hello world code to main.c
5. Compile the code by
   ```
   arm-linux-gnueabihf-gcc main.c -o main
   ```
6. Upload the main file to GW300 gateway
   ```
   ./main
   ```
7. Run main program.

```
root@rhf2s008:/tmp# ./main
Hello world!
```

Note: "Native Compile" and “Cross Compile" may have different gcc version, this makes the target bin has different in size, it is normal if you observe the same.

5 Factory Default Restore

5.1 Download image

Contact support@friendcom.com to get image address.

5.2 Bootloader Mode

Set GW300 to entern Bootloader mode

1) Cut off GW300 power
2) Connect Micro USB cable let GW300 to connect with PC
3) Restart GW300
4) Bootloader mode enabled

5.3 Program

Check online document to burn GW300 firmware.

6 Hardware Structure

6.1 Appearance

Figure 2-1 shows the appearance of the GW300.

NOTE The actual device appearance may be different from the following device appearance, but these differences will not affect device functions.

Figure 2-1 GW300 appearance

CAUTION

 There is a scald warning label attached on the device, warning you not to touch the device after the device has been operating for a long time.

6.2 Accessories in Package

Before unpacking the carton, ensure that the packing carton is intact and not damaged or soaked. Stop unpacking if the equipment is rusted or soggy. Then, investigate causes and contact the supplier.

After unpacking, check items in the carton against the packing list. If any item is missing, contact the supplier or agent.

Usually, the packing carton contains all items listed in the packing list shown in Table 3-1.

Table 3-1 package list

Material	PN	Qty
GW300	GW300-xxx	1
PoE injector	PoE30G-AT	1
LoRaWAN Antenna (N-type)	RXHF-ANTxxx-GF	1
Fixed collar for LoRaWAN antenna	-	1
Screw for collar	M3x6	4
4G Antenna (N-type)	RXHF-ANT4G	1
GPS Antenna (N-type, 70cm)	RXHF-ANTGPS	1
Wire for Ground	1.5m length	2
Fixture	-	1
Screw to fix GW (Inner hexagonal M5)	Inner hexagonal M6x8	4
Screw to fix the auxiliary fixture	M5x10	4
Screw for ground	M5x10	2
Surge protector	N-JK-G-Y-6	1
RF cable (connect the antenna to GW)	N (Male) - KSR200 (80cm) – N (Female)	1

CAUTION

Only the package list accessories are permitted to install.

6.3 Lightening Protection Cautions

Lighting protection include direct lighting protection and indirect lighting protection. We should avoid fix the device to face the direct lighting protection. Below we just list some essential and useful methods to protect from indirect lighting, induction lighting or surge.

1. The lightning rod should be fixed above the tower with device or gateway. And the gateway should be in the protection area of the lightning rod.
2. When the gateway is fixed to the building, the gateway should be in the protection area of the lightning rod which is on the top of the building. If there is no lightning rod on the top of the building, please make sure that the gateway would not be in the influence area of the lightning, or you must fix a lightning rod above the antenna and connect to the ground of the building or lightning protection network.
3. The small lightning rod should be made of circular steel tube with diameter more than 16mm. The lightning rod should be higher than the top of the antenna 1m or more.
4. Lightning rod ground down lead should be no less than 8 mm diameter galvanized round steel or cross-sectional area not less than 48 mm squared multi-strand copper wire. When using multiple strands of copper wire for grounding, please make steel tube to prevent the mechanical damage.
5. When the gateway is put nearby the lightning protection area, please make sure all the device include antenna put below the lighting protection area.
6. Don’t put or fix the cable to the lightning protection line or area.
7. Please ground the fixture of the gateway.

6.4 Product positioning

GW300 is an IOT gateway based on LoRaWAN and target to LPWAN network. It is an IEEE 802.3 af/at compatibility PD, which could be powered by PoE. Both Ethernet and LTE-4G are supported to connect to the cloud server. With an integrated GPS module, the GW could support LoRaWAN Class B protocol with the synchronous clock from GPS PPS signal. This device integrate an high performance CPU ARM Cortex-A53 core, one pcs of baseband processor SX1301, that it could support 8 multi-SF channel (SF12 to SF7), 1 single-SF channel and 1 GFSK channel. Sensitivity is as low as -141dBm@300bps. With specified payload length and transmit period, one GW could support 10k nodes.GW300 is a smart device but with high reliability, that it could work outdoor or in a complexity environment.

GW300 only for industrial/commercial use.

7. Technical Specifications

7.1 Basic Specifications

Table 4-1 Basic Specifications

Item		Description
Technical specifications	Dimensions (H x W x D)	145 mm x 95 mm x 40 mm
	Weight	715g
	Installation	Derrick installation, Fixed on the wall
System Configuration	Core	ARM Cortex-A53
	Main Frequency	1.2GHz
	RAM	1Gbytes
	Flash	4Gbytes eMMC
Communication	Mobile Cellular	4G Wireless connection
	LoRaWAN	Long Range Wireless Communication
Electrical Specification	Power supply input	PoE +48 Input IEEE 802.3 af/at
	LoRa Sensitivity	-141dBm@SF12,BW=125kHz
Sensor	Temperature	Monitor device internal temperature
User Interface (External)	Mobile Cellular 4G Antenna	Connect gateway with internet
	LoRaWAN Antenna	LoRaWAN transceiver, IoT data collection
	GPS	GPS function
User Interface	Micro SIM Card Slot	Support Micro SIM Card
Operating Range	Operational temperature range	-40 to +75°C
	Memory temperature range	-40 to +85°C

8 Order information

GW300 include several part number, different part number would be used in different band and area, please contact with sales@friendcom.com for detailed information.

To place an order, contact the Shenzhen Friendcom Technology Development Co., Ltd. local office or authorized dealers.

Part Number	Description
GW300-915	Assembling Components, GW300 Mainframe(PoE injector, LoRaWAN Antenna (N type), Fixed collar for LoRaWAN antenna, 4G Antenna (N-type), GPS Antenna (N-type, 70cm))

9 FAQ

Q1: Customized global_conf.json file canâ€™t be recognized by packet forwarder?

A1: Check below rule. Every object must end with comma, except the last member of an object or array. (// and /* */ are special comment format, no need consider it)

Q2: One gateway receives 2 same packets at almost the same time

A2: If the main channel receive packets at a high signal strength, in such case SX1301 chip is possible to receive mirror packet, server should compare the packets to filter the fake packet.

Semtech official explain: https://github.com/Lora-net/lora_gateway/issues/48

Q3: Gateway TX packet is received back

A3: SX1301+SX125x is full duplex chips with half duplex design. When gateway switch to TX mode the receiver is not closed, only switch is controlled, it is possible to receive such packet.

Q4: Cannot dial in cellular network.

A4: Make sure below items OK.

1) lte service is enabled
2) SIM card is inserted correctly (Be careful about the SIM card direction, SIM direction mark outside)
3) SIM card is OK

Q5: ADR issue when connecting with Loriot server

A5: Loriot server assumes that device enables 8 channels, if device has less channels it is possible to lead to the problem.

FCC Statement

This device comply with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

● This device does not cause harmful interference.

● This device must accept any interference received, including interference that may cause undesired operation.

If this device is modified without authorization from Friendcom, the device may no longer comply with FCC requirements for Class B digital devices. In that a case, your right to use the device may be limited by FCC regulations. Moreover, you may be required to correct any interference to radio or television communications at your own expense.

This device has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

This device generates, uses and radiates radio frequency energy. If it is not installed and used in accordance with the instructions, it may cause harmful interference to radio communications.

However, there is no guarantee that interference will not occur in a particular installation. If this device does cause harmful interference to radio or television reception, which can be determined by turning the device off and on, the user may take one or more of the following measures:

● Reorient or relocate the receiving antenna.

● Reinforce the separation between the device and receiver.

● Connect the device into an outlet on a circuit different from that to which the receiver is connected.

● Consult the dealer or an experienced radio or TV technician for assistance.

FCC Radiation Exposure Statement

This equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body.