F8L10GW-02 LoRaWAN Gateway User Manual	Document Version	Classification
	V1.0.3	
	Product Name：F8L10GW-02	Total 98 pages

F8L10GW-02 LoRaWAN Gateway User Manual

This manual is applicable to the following product models:

Model	Product Category
F8L10GW-02433-XXX	Frequency Band：410-441MHz
F8L10GW-02470-XXX	Frequency Band：470-510MHz
F8L10GW-02868-XXX	Frequency Band：850-890MHz
F8L10GW-02915-XXX	Frequency Band：895-935MHz
F8L10GW-02433-MZZ	Frequency Band：410-441MHz
F8L10GW-02470-MZZ	Frequency Band：470-510MHz
F8L10GW-02868-MZZ	Frequency Band：850-890MHz
F8L10GW-02915-MZZ	Frequency Band：895-935MHz

Four-Faith 四信

Customer Hotline：400-8838 -199
Phone：+86-592-6300320
Fax：+86-592-5912735
Website：www.four-faith.com
Address：Building A06, 11th Floor, Phase III of Xiamen Jimei Software Park

Document Revision History

Date	Version	Explanation	Author
2022-10-18	V1.0.0	Initial Version	YSL
2022-11-1	V1.0.1	Modify the communication field "devEUI" to "devEui."	SGK
2024-02-05	V1.0.3	English Version	YYL

Copyright Statement

Trademark Statement

Four-Faith、四信、 、 、 All are registered trademarks of Xiamen Four-Faith Communication Technology Co., Ltd. Without prior written permission, no one is allowed to use the Four-Faith name and Four-Faith trademarks or symbols in any way.

Pole Mounting Diagram

Wall Mounting Diagram

Note: There may be differences in accessories and interfaces for different models. Please refer to the actual product for details.

Content

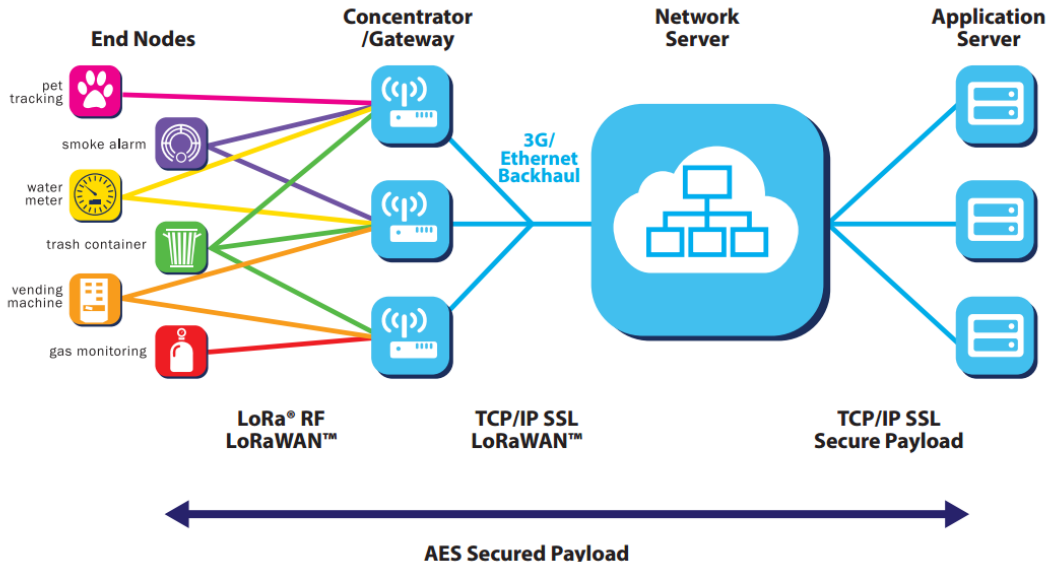
Chapter 1 Product Introduction

1.1 Product Overview

The F8L10GW-02 series device is a wireless communication base station based on the LoRaWAN protocol. It connects to various application nodes of LoRaWAN terminals, transmitting terminal information to the cloud through 4G or wired Ethernet. It supports WiFi wireless configuration management and online upgrades, GPS positioning, and can be powered by 220V AC or optional POE+.

The F8L10GW-02 gateway complies with the standard LoRaWAN protocol and supports multiple modes, including Embedded Network Server mode (Network Server deployed inside the gateway), Basicstation mode (connecting to an external Basicstation protocol server), and Semtech UDP GWMP Protocol mode (connecting to an external NS server via GWMP UDP protocol).

This product has been widely used in the M2M industry of the IoT ecosystem, including applications in smart meters, smart disasters, smart sensing, smart photovoltaics, smart grids, smart transportation, industrial automation, smart buildings, firefighting, public safety, environmental protection, meteorology, digital healthcare, remote sensing, military, space exploration, agriculture, forestry, water management, coal mining, petrochemicals, and other fields.



1.2 Product Features

Industrial-Grade Application Design:

- Utilizes high-performance industrial-grade wireless communication modules.
- Incorporates high-performance industrial-grade multi-channel LoRaWAN base station radio frequency chips.
- Features an aluminum alloy casing with an IP65 protection rating, ensuring robust protection against environmental factors.
- Supports AC220V power, POE+ Power optional

Stable and Reliable:

- WDT watchdog design ensures system stability.
- Adopts a comprehensive anti-dropout mechanism to ensure data terminals are always online.
- Ethernet interface with built-in 1.5KV electromagnetic isolation protection.
- SIM/UIM card interface with built-in 15KV ESD protection.
- Power interface with reverse polarity protection, overvoltage protection.
- Lightning protection for antenna interface.

1.3 Product Performance Parameters

- Business Channel: Uses a simple star network
- LoRaWAN Protocol Support: Class A, Class B*, Class C
- Operating Frequencies: EU433, CN470-510, CN779-787, EU863-870, US902-928, AU915-928, AS923, KR920-923
- Urban Communication Distance: 9km
- Maximum Transmit Power: 26±1dBm
- Maximum Receive Sensitivity: -142dBm @LoRa
- Communication Bandwidth: 125kHz, 250kHz, 500kHz
- 8 uplink channels, 1 downlink channel
- Implements secure, reliable, low-latency wireless transmission technology
- Communication Rate: Adaptive link rate
- Operating Modes: Supports asynchronous and synchronous frequency transmission
- Positioning Function: GPS, Beidou
- Server Reporting Methods: 4G, Wired Ethernet
- Wireless Management: WiFi wireless management and upgrades
- Local Storage: Supports TF card local storage
- Operating Temperature: -35~+75℃
- Overall Dimensions: 289.4223.62115 mm
- Waterproof and Dustproof: IP65
- Power Supply: AC220V, POE+ (optional)
- Total Power Consumption: <6W

◆ Electrical Performance

Number	Parameters	Technical Specifications
1	Rated Input Voltage	100~240VAC
2	Rated Output Voltage	12V
3	Rated Output Current	3A
4	Input Undervoltage Protection	None
5	Output Overvoltage Protection	Yes
6	Output Overcurrent Protection	Yes
7	Short Circuit Protection	Yes
8	Surge Voltage Resistance	Line-line, line-ground both are 6KV
9	Lightning Protection Level	3KA
10	Input Side Wire Diameter	Recommend 5-7mm
11	POE Power Supply	POE input, Support 10/100 Base-T Adaptive.
12	Supports POE Standards	IEEE802.3af/IEEE802.3at

◆ Power consumption

Average operating voltage V（V）	Average operating current I（mA）	Power Consumption (W)	Note
12.00	TX≦460 RX≦120	6	4G module connected to the Internet with GPS, LoRa communication
12.00	TX≦243 RX≦120	3.5	4G module connected to the Internet without GPS, with LoRa

Note：* Indicates that it is under development.

Chapter 2 Installation

2.1 Overview

F8L10GW-02 must be correctly installed to achieve its designed functionality. Typically, the installation of the device should be carried out under the guidance of engineers approved by our company.

- *Precautions：*

1、Please do not install F8L10GW-02 while it is powered.

2、Do not tamper with the plugs, power ports, antenna ports, and other interfaces of F8L10GW-02.

2.2 Packing List

2.2.1. Wall-mounted packing list

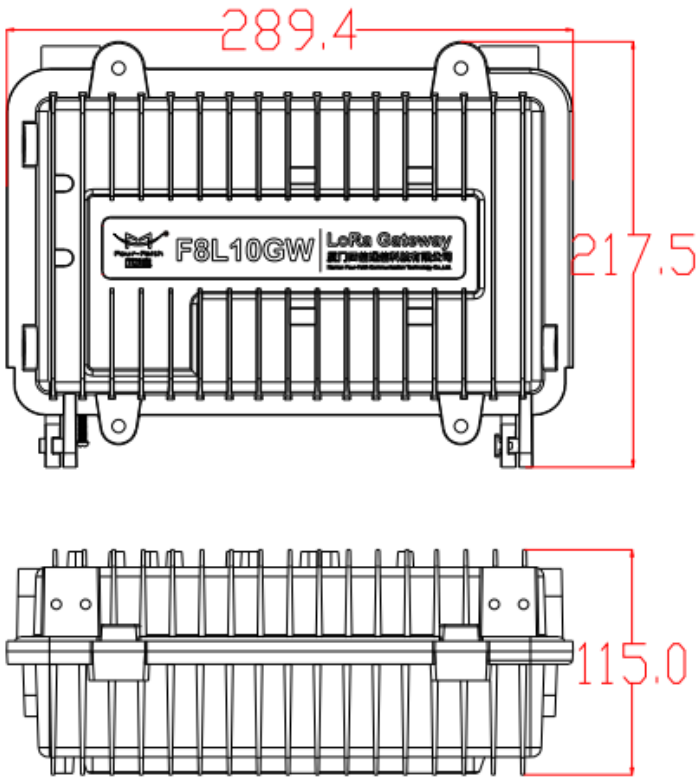
Name	Quantity	Note
F8L10GW-02 main unit	1	
4G fiberglass omnidirectional antenna	1	Optional
WiFi fiberglass omnidirectional antenna	1	
GPS fiberglass omnidirectional antenna	1	
LoRa fiberglass omnidirectional antenna	1	
Bracket	1	
Expansion screws ø14mm	3	
Power cord	1	Optional
Product certificate	1	
Product warranty card	1	

2.2.2. Pole-mounted packing list

Name	Quantity	Note

F8L10GW-02 main unit	1	
4G fiberglass omnidirectional antenna	1	Optional
WiFi fiberglass omnidirectional antenna	1	
GPS fiberglass omnidirectional antenna	1	
LoRa fiberglass omnidirectional antenna	1	
Bracket	2	
Power cord	1	Optional
Product certificate	1	
Product warranty card	1	

2.3 Device Scene Installation



F8L10GW-02 Size

2.3.1 SIM/UIM Card Installation

1、Turn off the device power.

2、Unscrew the M6 screws inside the casing, as shown in Figure 2.3.1.

3、When installing the SIM/UIM card, pay attention to the direction of the SIM/UIM card, as shown in Figure 2.3.2. Insert the SIM/UIM card into the slot and press it down until it is securely in place.

4、To remove the SIM/UIM card, gently push it downward, and it will automatically pop out.

5、Tighten the M6 screws on the outer casing until they are securely fastened.

Precautions:

1、Do not install the SIM/UIM card with power on!

2、Ensure that the M6 screws are securely tightened and immovable!

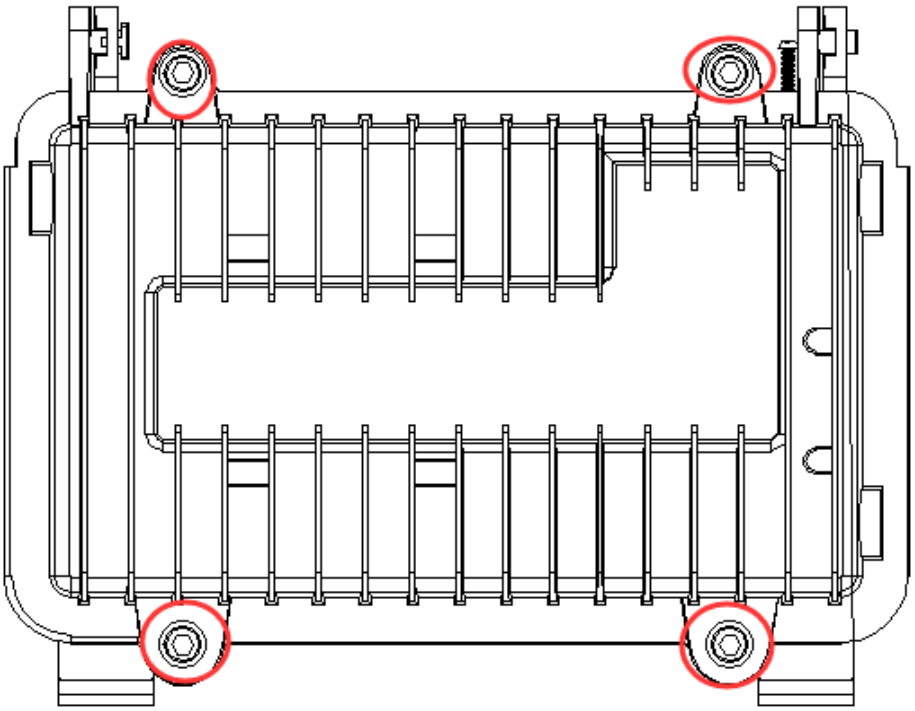


Figure 2.3.1

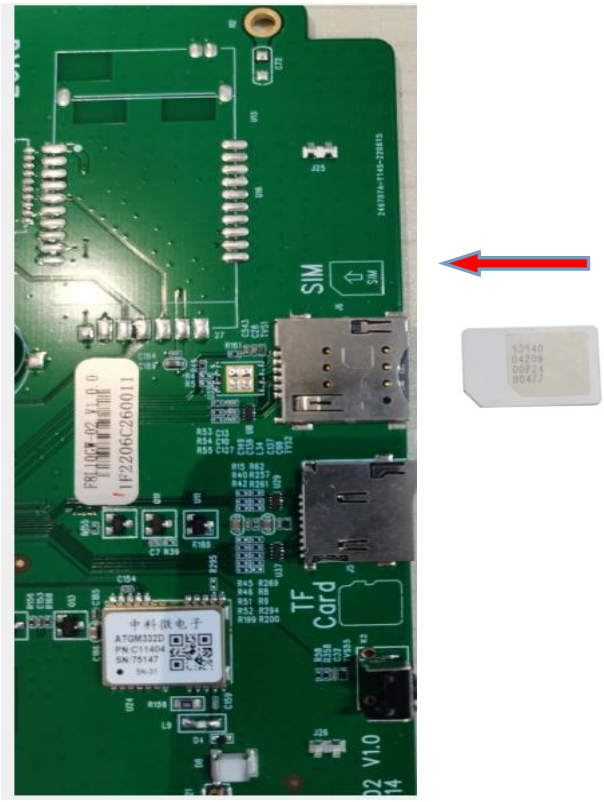


Figure 2.3.2

2.3.2 Wall-mounted installation

Step One: Find a suitable wall, ensuring that the wall is relatively flat. Install the device in a location close to an open area. There should be no obstructions within 5 meters around the LoRa antenna. Based on the positions of the bracket installation holes, drill three ø14mm holes in the wall, with a depth of approximately 60mm (the length of the expansion screws is around 50mm), as shown in the diagram 2.3.3.

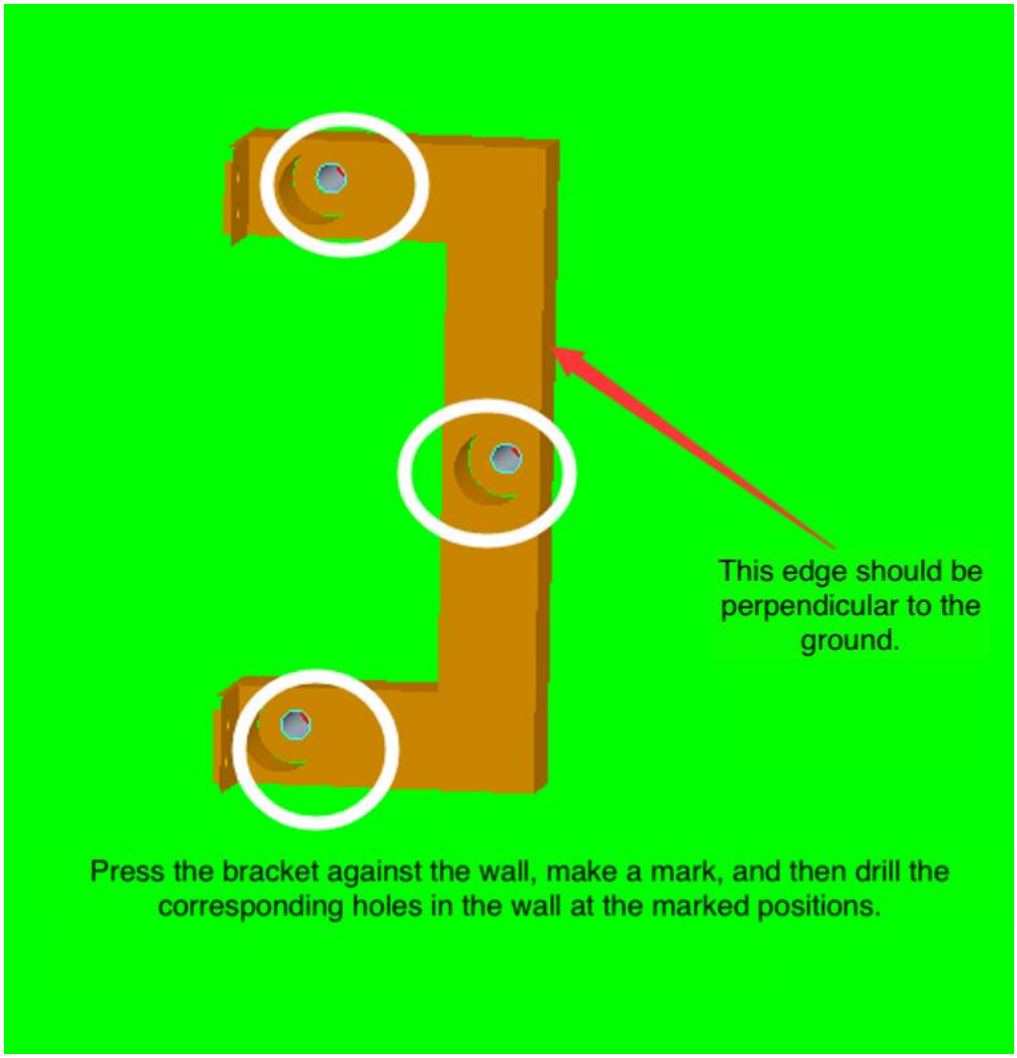


Figure 2.3.3

Step Two: Secure the bracket using the provided expansion screws, as shown in the diagram 2.3.4.

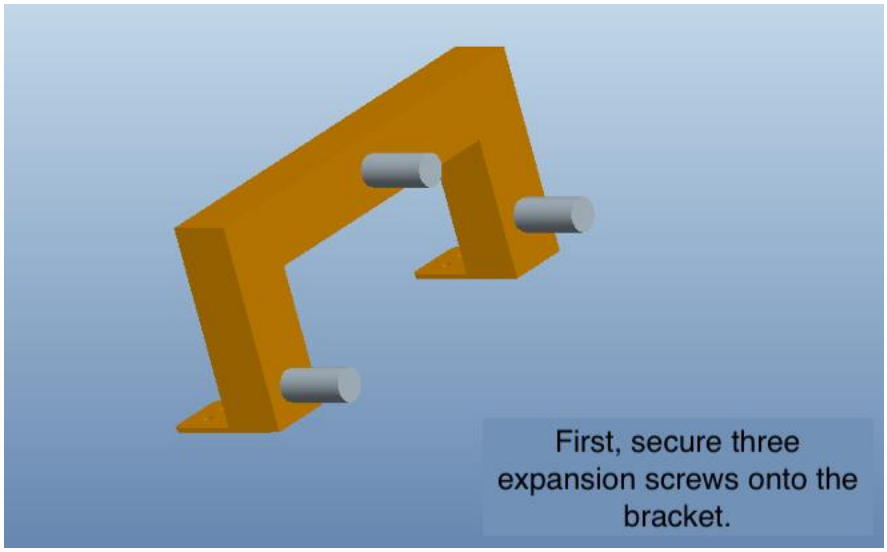


Figure 2.3.4

Step Three: Secure the bracket of the locked expansion screw to the wall and tighten the screws as shown in Figure 2.3.5.

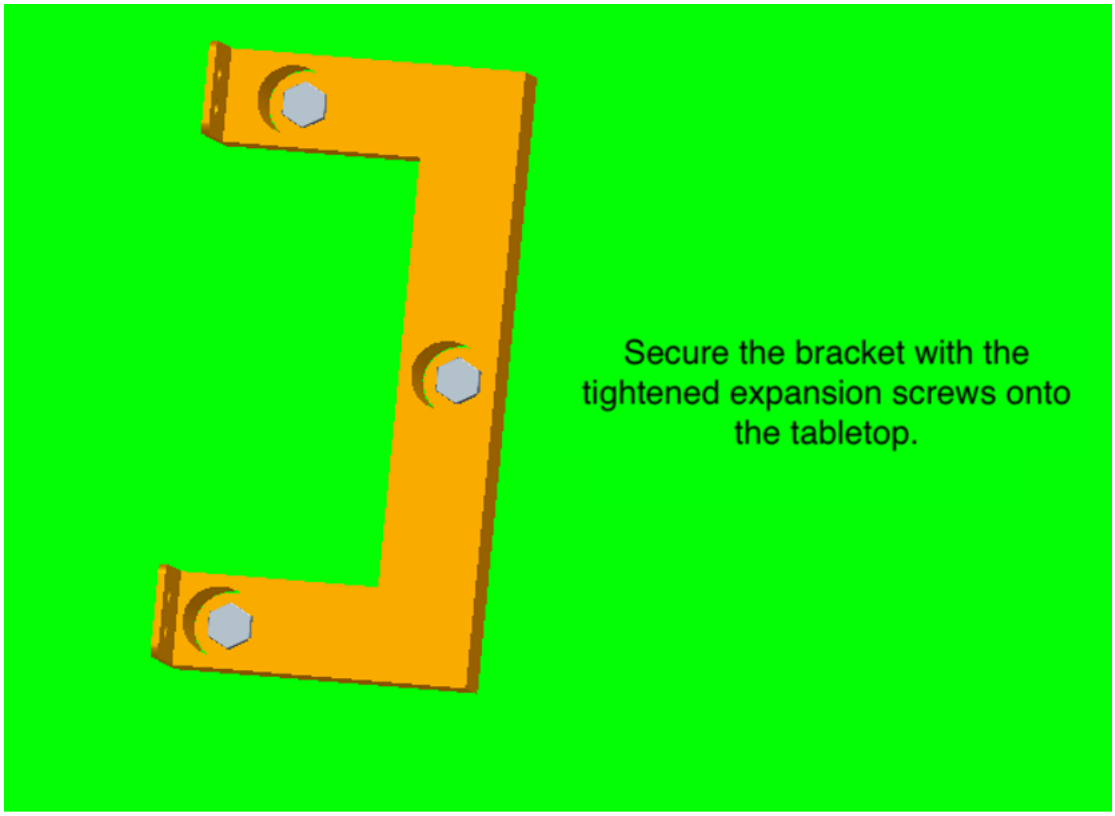


Figure 2.3.5

Step Four: Once the bracket is securely fixed, use the corresponding M5 wrench to tighten the four screws on the base station. Afterward, proceed to install the antenna, as shown in the diagram 2.3.6.

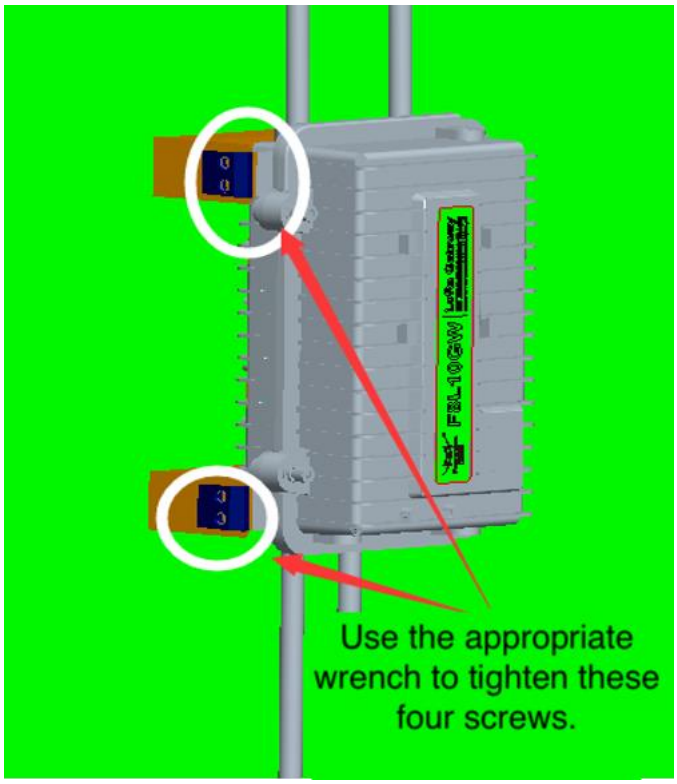
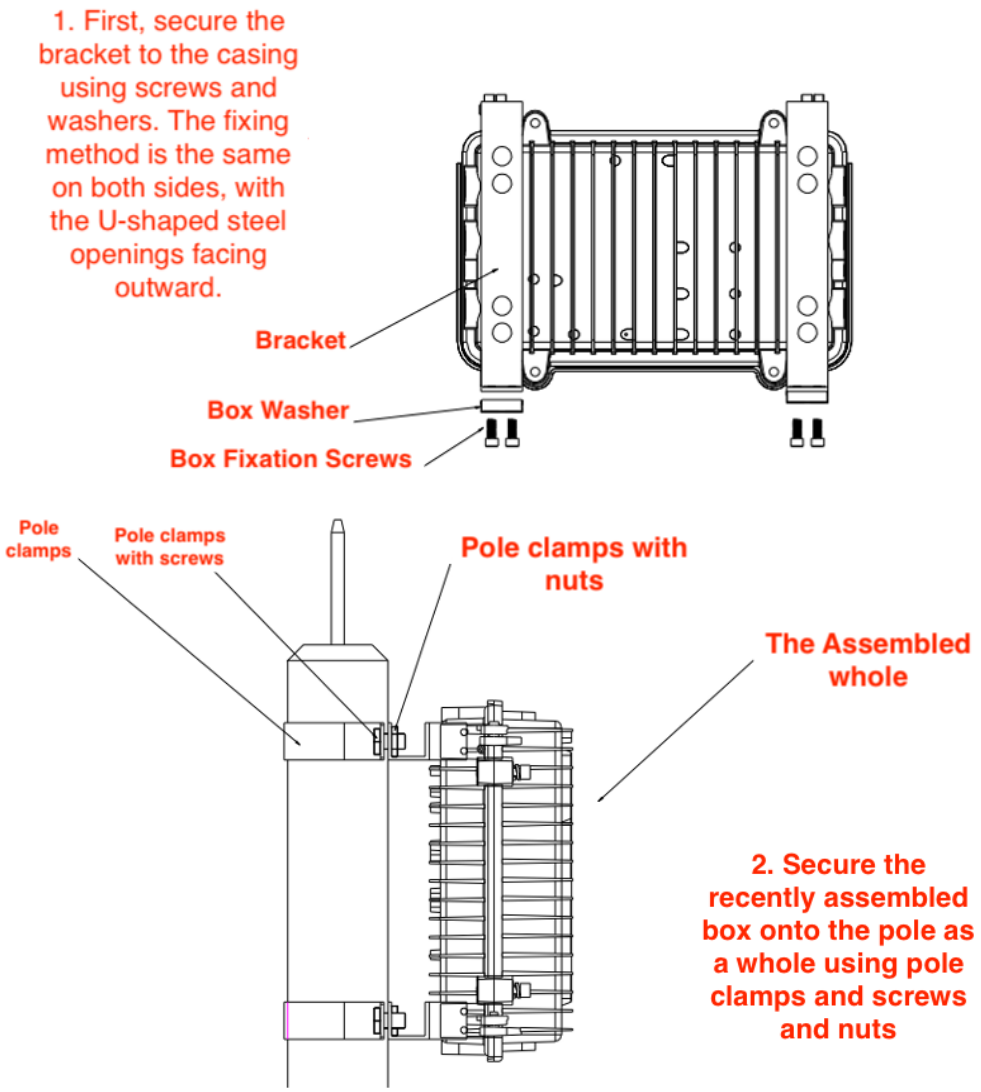


Figure 2.3.6

2.3.3 Pole-mounted installation

Step One: Prepare a suitable pole, and the diameter of the pole should be in the range of ø70~90mm. Install it in a location close to an open area, and ensure there are no obstructions within 5 meters around the LoRa antenna.

Step Two: Use the provided pole clamps (2 pieces) and slide them onto the pole, corresponding to the appropriate positions on the backplate. Insert bolts through the clamp fixation holes and use a wrench to securely tighten them onto the screw holes on the backplate. For ease of installation, you can first secure the backplate before mounting the antenna. Ensure a firm and secure fixation.



Note: Pole installation does not include pole clamps. We can provide specifications or assist in sampling.

2.3.4 Antenna Installation

After mounting the equipment on the wall or pole, connect the corresponding antenna interfaces of the matching fiberglass omnidirectional antennas. Each antenna is labeled, and the antenna connector is a standard coaxial N-type male connector. Simply screw it onto the corresponding coaxial N-type female connector on the equipment, ensuring a tight fit to avoid compromising signal quality and waterproofing. Refer to Figure 2.3.7 below.

Note: Ensure that each antenna is not connected in reverse and tighten with tools to prevent any impact on functionality and waterproofing.

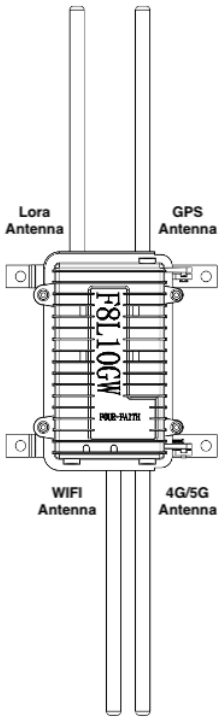


Figure 2.3.7

2.4 Indicator Light Instructions

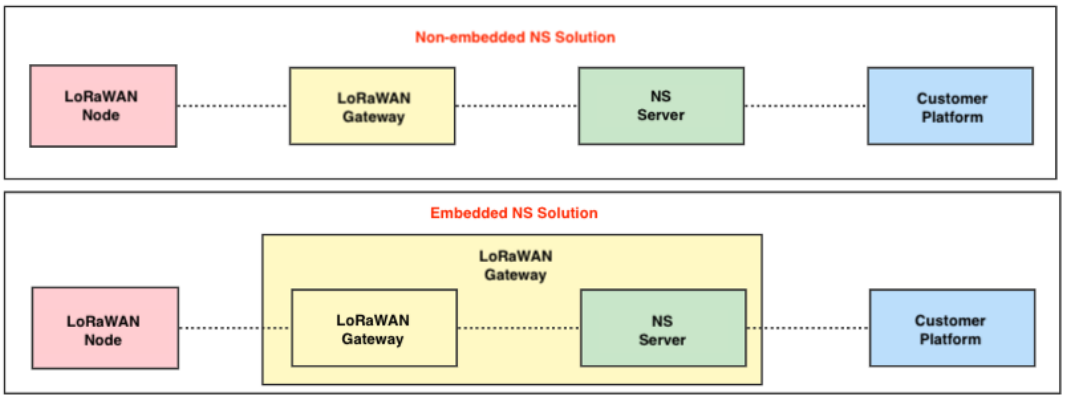
The F8L10GW-02 provides the following indicator lights: "Power," "System," "WiFi," "LoRa," "Signal Strength," "Online." The status explanations for each indicator light are as follows:

Lights	Status	Explanation	Note
PWR	Red Light On	Device power is normal	Located on the outer side of the PCB board.
	Red Light Off	device is not powered on	
SYS	Yellow Light Flashing	System is running normally	
	Yellow Light Off	System is not normal	
WIFI	Blue Light On	WiFi is activated	
	Blue Light Off	WiFi is not activated	
LoRa	Green Light On	LoRa module connection is normal	
	Green Light Off	LoRa module connection is abnormal	
	Green Light Flashing	LoRa module is in data communication (only indicates data packet transmission)	
Online	Green Light Off	Device is not logged into the network	
	Green Light On	device is logged into the network	

Chapter 3 Quick Start Guide

3.1 Brief Introduction to the Solution Architecture

3.1.1 Difference Between Embedded and Non-Embedded

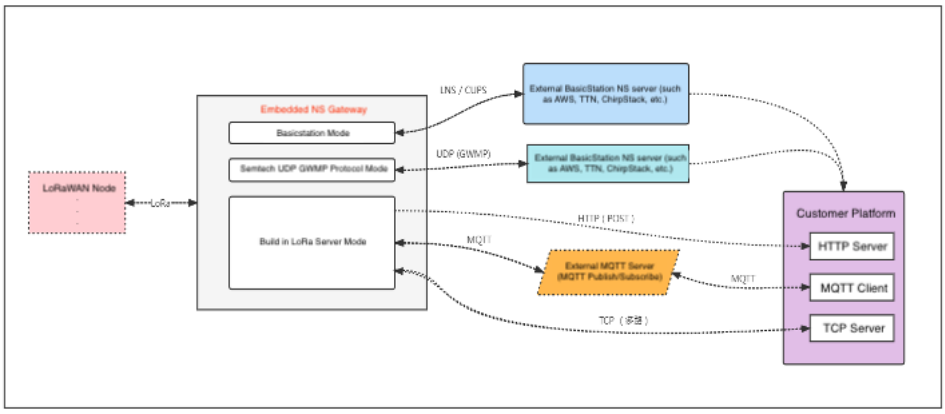


As shown in the above diagram, the main difference between the embedded and non-embedded solutions lies in the position of the NS. In the non-embedded solution, the NS is generally deployed on a server, while in the embedded solution, the NS is deployed within the gateway.

- Embedded Solution (Embedded Mode): The advantage is that there is no need to deploy NS on an external server, reducing operational costs and quickly setting up the entire LoRaWAN system. The drawback is that the performance and storage size of the gateway system are relatively poor compared to a server, limiting the number of nodes and the ability to cache a large amount of information.

- Non-Embedded Solution (External Mode): The advantage is that the server has strong performance, large storage, and can manage a large number of gateways and nodes. It can be deployed in a clustered manner, significantly improving system performance and availability. The drawback is that an additional server is required to deploy NS, which involves maintenance and increases project costs. The setup of the system and the time spent on problem-solving will be more extensive.

3.1.2 System Framework



The gateway communicates with devices or terminals, and the data flow direction is determined based on web configuration:

- Basics Station (basicstation mode): In this mode, data will communicate bidirectionally with the corresponding connected server. The gateway serves only as a data forwarding function. Device management, data encryption/decryption, and integration with the customer platform need to be handled on the server side.

- Semtech UDP GWMP Protocol (external NS mode): Data will communicate with an external NS using the standard UDP protocol. In this mode, device management, data encryption/decryption, and client integration will be handled on the external NS server, such as commonly used integration with the Four-Faith cloud NS server.

- Build-in LoRa Server (embedded NS mode): Data will flow to the built-in NS server in the gateway. In this mode, device management, data encryption/decryption, and client integration will be handled on the built-in NS server (LoRa Network Server). Clients can achieve data push functionality through HTTP server configuration (HTTP POST supports only uplink push, not downlink data), or through MQTT and TCP for both uplink and downlink data.

As an embedded NS serving as the core of the LoRaWAN, this product theoretically supports a large number of gateway and node connections. It manages LoRaWAN devices, including network joining, data encryption/decryption, data uplink/downlink, and data push. For uplink data from devices, after LoRaWAN decryption, it establishes a relationship with the client through an interface and sends the uplink data to the customer platform. Clients can send downlink data via MQTT or TCP after encrypting it through LoRaWAN to the specified device.

3.2 Login Configuration Interface

3.2.1 Access the Web Management Platform

1、 Method One: After the gateway is powered on, the default WiFi name is Four-Faith, and there is no default password. After successfully connecting to the WiFi, the gateway's LAN address is default to 192.168.1.1. You can then log in at http://192.168.1.1 (or simply enter 192.168.1.1).

2、 Method Two: If you know the gateway's WAN address (e.g., set to a static IP like 192.168.1.88), you can directly access http://192.168.1.88.

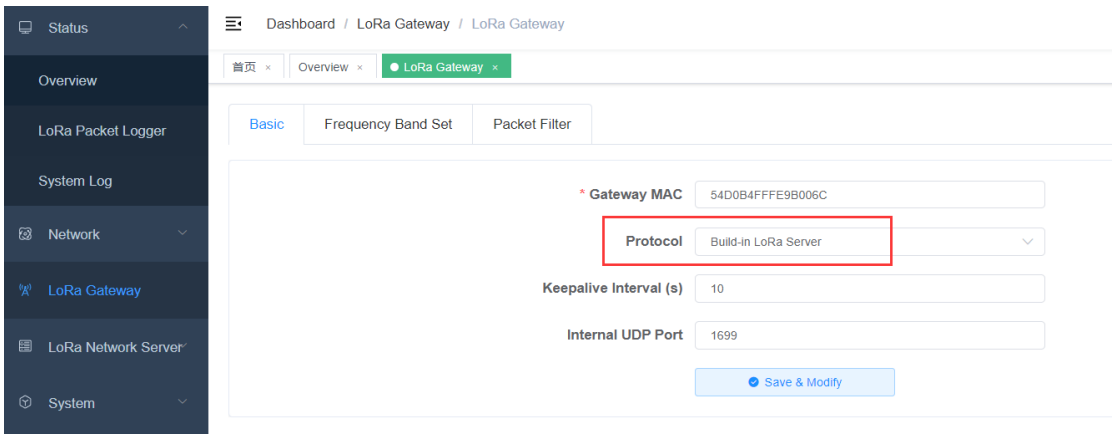
3、 Log in with the default account: admin, and the default password: admin. Click "Login" to access the Web Management Platform.

Note: Please use Google Chrome browser; other browsers may have compatibility issues.

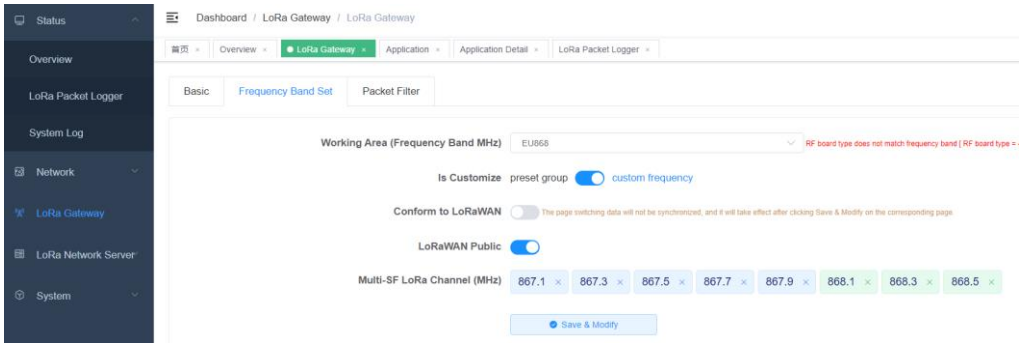
3.2.2 Adding Devices in Embedded Mode

1、 **Identify the device's frequency band and corresponding frequency points** (e.g., for a standard EU868 terminal, frequency points: 868.1MHz, 868.3MHz, 868.5MHz).

2、 **Confirm whether it is in embedded mode** (default is embedded mode). If not, change it to embedded mode.
<Path: LoRa Gateway → Basic>

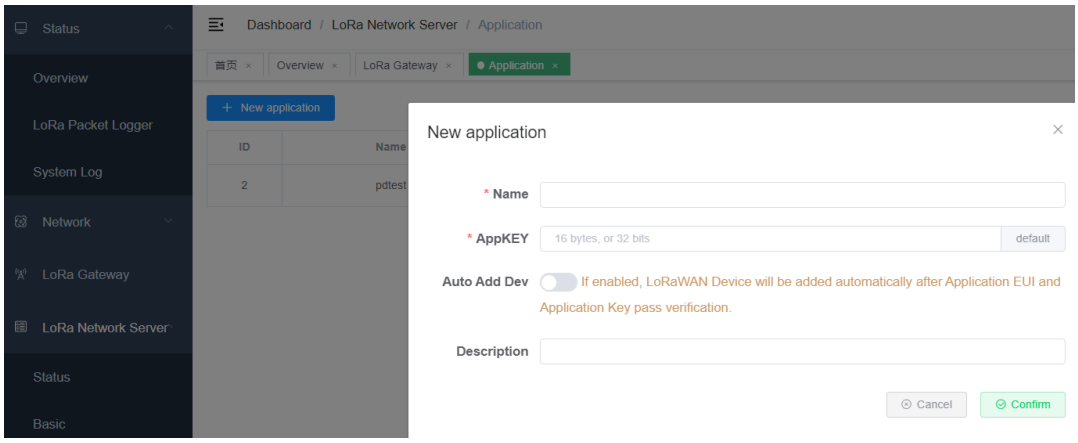


3、 **Check if the gateway's frequency band and frequency points are consistent** (default frequency points follow the regional parameter specified points). If not, modify them to match.
<Path: LoRa Gateway → Frequency Band Set>

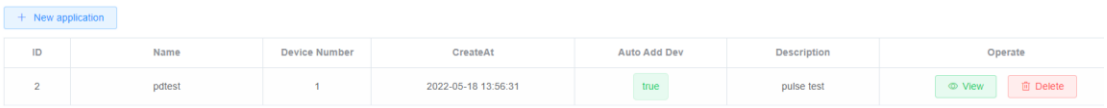


4、 **Add an application** (configure it for automatic device addition in network joining mode).

<Path: LoRa Network Server => Application => New application>



In the above figure, both AppKEY and AppEUI are automatically generated by clicking on the "default" on the right side (this value is the default value for Four-Faith; for devices from other manufacturers, modify it to the corresponding values). Choose ClassA or ClassC based on the device type, then click "Confirm" to add. After adding, the following page will appear:



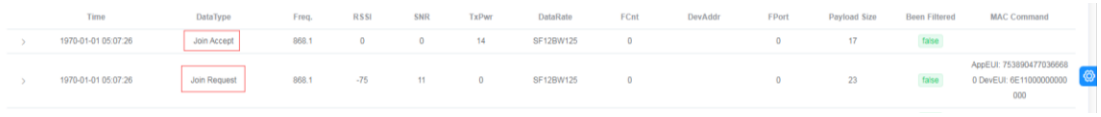
5、 Device Network Joining

The device initiates a network joining request, and check whether the network joining is successful. If it cannot join the network successfully, troubleshoot as follows:

Confirm whether the gateway can receive the network joining request from the device (you can use a packet capture tool, path: Status → LoRa Message Logger).

If the gateway can receive the network joining request but does not see the network joining response (Join Accept), it is generally caused by the inconsistency between the

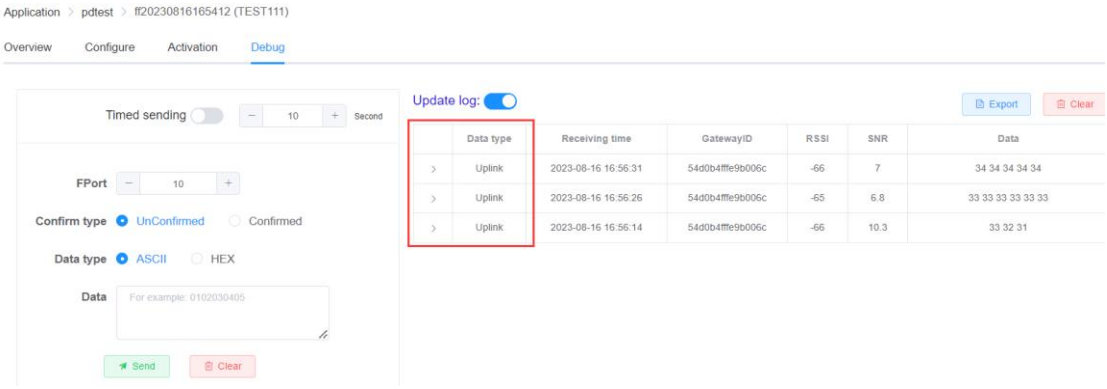
AppKey or AppEUI configured in the application and that of the device.



6、 Device Uplink Data

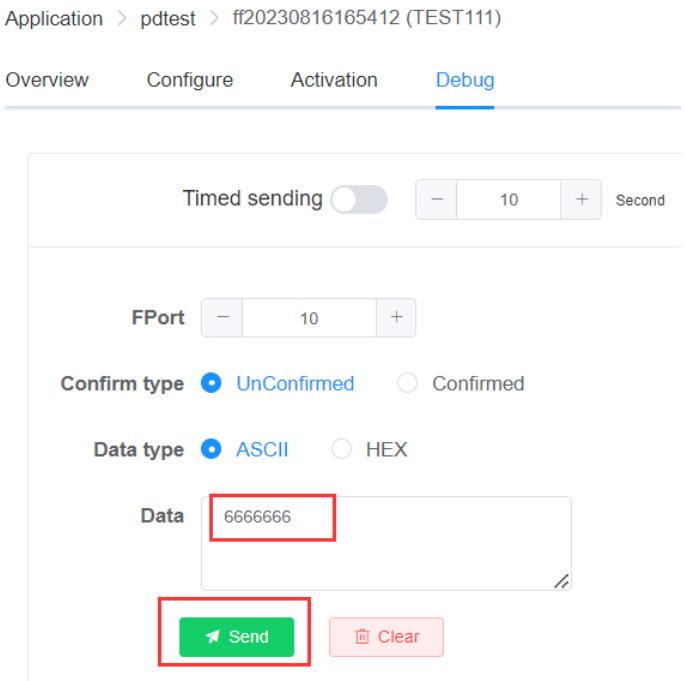
After the device successfully joins the network, instruct the device to report any data. At this time, in the application's device list, find the corresponding device for viewing.

<Path: LoRa Network Server → Applications → Corresponding Application (click to view) → Find the corresponding device (click to view) → Online Debugging>

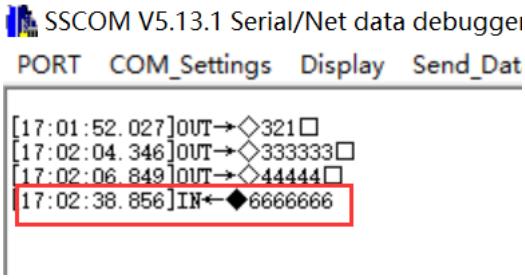


7、 Send Data to the Device

Send data to the device on the device's online debugging page, as shown in the following figure:



The Four-Faith module receives data as follows:



Attention:

The device types are divided into ClassA and ClassC, and the data reception methods are as follows:

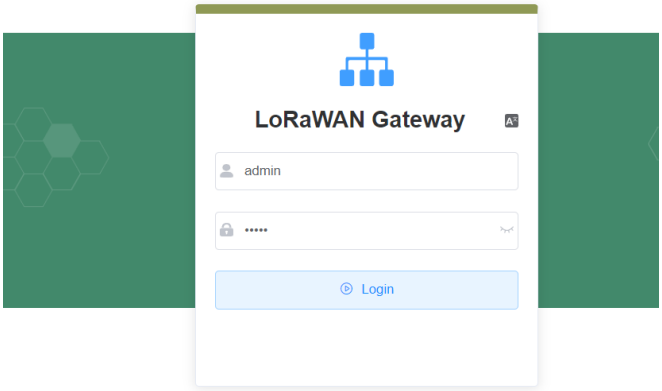
- In ClassA mode, after sending data, it will not be directly sent to the device. You need to wait for the device to send uplink data again before the sent data is delivered to the device.
- In ClassC mode, data sent will be directly delivered to the device. If the device does not receive the data, please check whether the type configured in NS matches the type configured for the device. If not, after modification, you need to rejoin the network before conducting data communication tests.

Chapter 4 Detailed Introduction to the Function Pages

4.1 Interface Management Configuration

4.1.1 Web Management Platform

- Method One: After the gateway is powered on, the default WiFi name is Four-Faith, and there is no default password. After successfully connecting to the WiFi, the gateway's LAN address is default to 192.168.1.1. You can then log in at http://192.168.1.1 (or simply enter 192.168.1.1).
- Method Two: If you know the gateway's WAN address (e.g., set to a static IP like 192.168.1.88), you can directly access http://192.168.1.88.
- Log in with the default account: admin, and the default password: admin. Click "Login" to enter the Web Management Platform.



Note: Please use Google Chrome browser; other browsers may have compatibility issues.

4.1.2 Directory Details

Here is an introduction to the functionality of each page based on the directory order:

- **Status**
  - **Overview:** Displays statistics for data monitored by the gateway and system parameter information.
  - **LoRa Message Logger:** Shows received data and downstream data for the gateway.

- **System Logs:** Records logs of the gateway's operational processes.

- **Network**
- **WAN Interface:** Configures the gateway's WAN settings. Network information, such as DHCP or static IP, can be configured here.
- **Wi-Fi:** Configures WiFi parameters and security settings.
- **Network Diagnostics:** Includes commands like Ping, Traceroute, Nslookup.
- **Firewall:** Configures basic parameters for the firewall.

- **LoRa Gateway**
- Configuration of gateway mode, frequency point parameters, packet filtering, etc.

- **LoRa Network Server**
- **Status:** Displays statistics for the embedded NS.
- **Basic Settings:** Configures NS-related parameters such as ADR switch, RX2 parameter settings, etc.
- **Gateway:** Displays gateway information.
- **Applications:** Displays application information, including device lists.
- **Multicast:** Manages multicast.
- **Interfaces:** Configures protocol types for customer platform integration, data conversion, heartbeat settings, etc.

- **System**
- **System:** Displays embedded NS version information, system time settings, etc.
- **Change Password:** Modifies the Web Management Platform password.
- **Restart:** Restarts the gateway.
- **Factory Reset:** Performs a factory reset.

4.1.3 Management Configuration

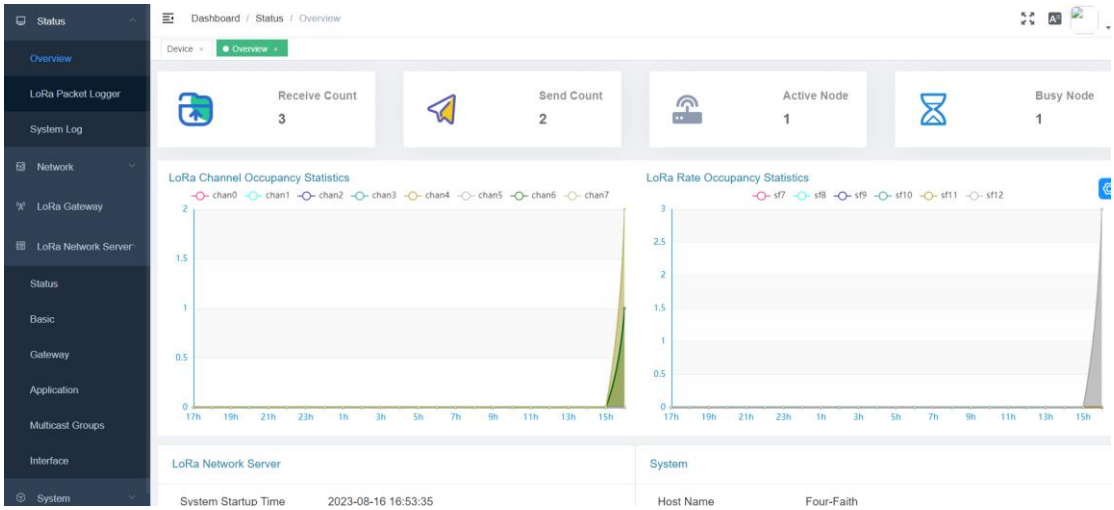
4.1.3.1 Status

1. Overview

- **Path:** Status → Overview
- **Function:** Displays communication statistics for the gateway, making it convenient to analyze the RF environment around the gateway. This helps in assessing device communication status or identifying the presence of interfering devices.
- **Details:**
  - **Received Messages:** Number of messages received since system startup.
  - **Sent Messages:** Number of messages sent since system startup.
  - **Active Nodes:** Number of upstream nodes received by the gateway.

- **Busy Nodes:** Nodes that have transmitted twice within 10 seconds are considered busy nodes. This section provides a count within the last hour.
- **LoRa Channel Occupancy Statistics:** Channel occupancy details for various time periods in the last 24 hours.
- **LoRa Rate Occupancy Statistics:** Rate occupancy details for various time periods in the last 24 hours.
- **LoRa Network Server:** Includes system startup time, LoRa protocol details, device count, NS device uplink count, NS device downlink count, and NS MQTT connection status.
- **System:** Includes host name, LAN MAC, WAN MAC, wireless MAC, WAN IP, LAN IP, and WAN protocol.
- **Wireless:** Includes wireless switch status, mode, network mode, name, channel, and transmission power.

- **Preview:**



2. LoRa Message Logger

- **Path:** Status → LoRa Message Logger

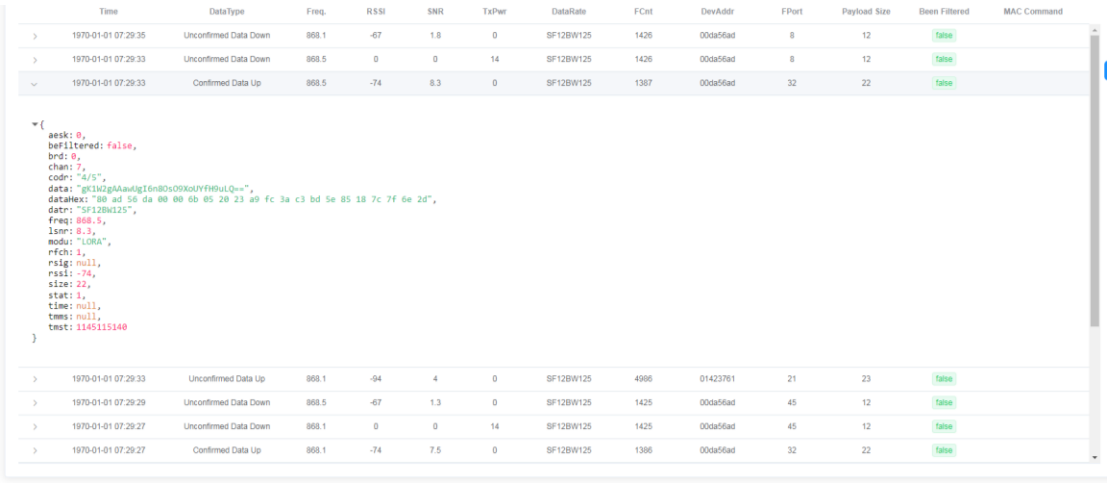
- **Function:**
  - Displays LoRaWAN data received and sent by the gateway.
  - Useful for analyzing communication between the gateway and devices, allowing for the identification of issues such as unanswered join requests, missing downlink data, and communication quality.

- **Details:**
  - **Update Log Switch:** Defaulted to ON; when turned off, data is expanded for viewing, and during this period, data is received but not listed. When turned on again, the data is automatically updated to the list.
  - **LoRaWAN Data Type Selection:** Facilitates analysis of communication issues

based on data type.

- **Packet Filtering Status:** Indicates whether the packet is filtered. Filtered data won't be reported to the NS server. Filter configurations can be found in LoRa Gateway → Packet Filter. Options include:
  - 0: All - Show both filtered and unfiltered data
  - UnFiltered - Show only unfiltered data
  - BeFiltered - Show only filtered data
- **devAddr:** Search by short address.
- **Time:** Timestamp of data reception.
- **DataType:** Data type, including:
  - ALL
  - Join Request
  - Join Accept
  - Unconfirmed Data Up
  - Unconfirmed Data Down
  - Confirmed Data Up
  - Confirmed Data Down
- **Freq:** Communication frequency.
- **RSSI:** Signal strength.
- **SNR:** Signal-to-noise ratio.
- **TxPwr:** Transmission power. This is 0 for uplink data.
- **FCnt:** Frame counter, useful for determining packet loss or retransmission.

- **Preview:**



3. LoRa Message Logger

- **Path:** Status → LoRa Message Logger
- **Function:**The log is useful for analyzing the overall operation of the gateway, abnormal device communication, and other anomalies.
- **Details:**

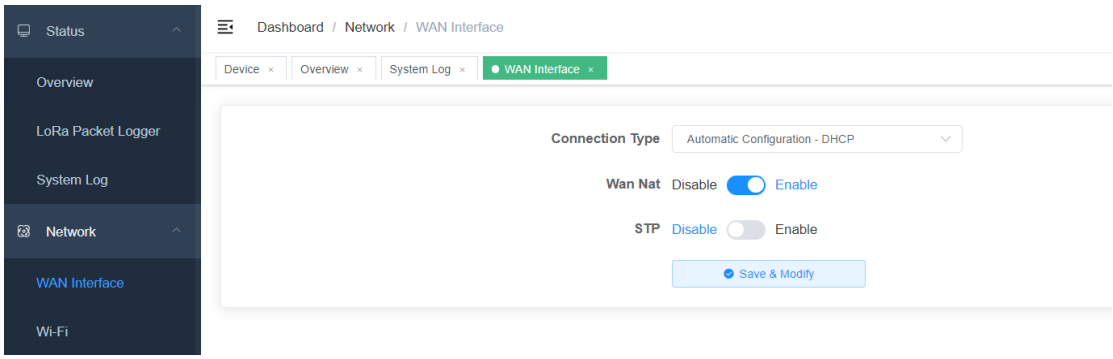
- **Switch:** Defaulted to ON; when paused, new data is stored in the browser cache and updates upon reactivation.

- **Preview:**

4.1.3.2 Network

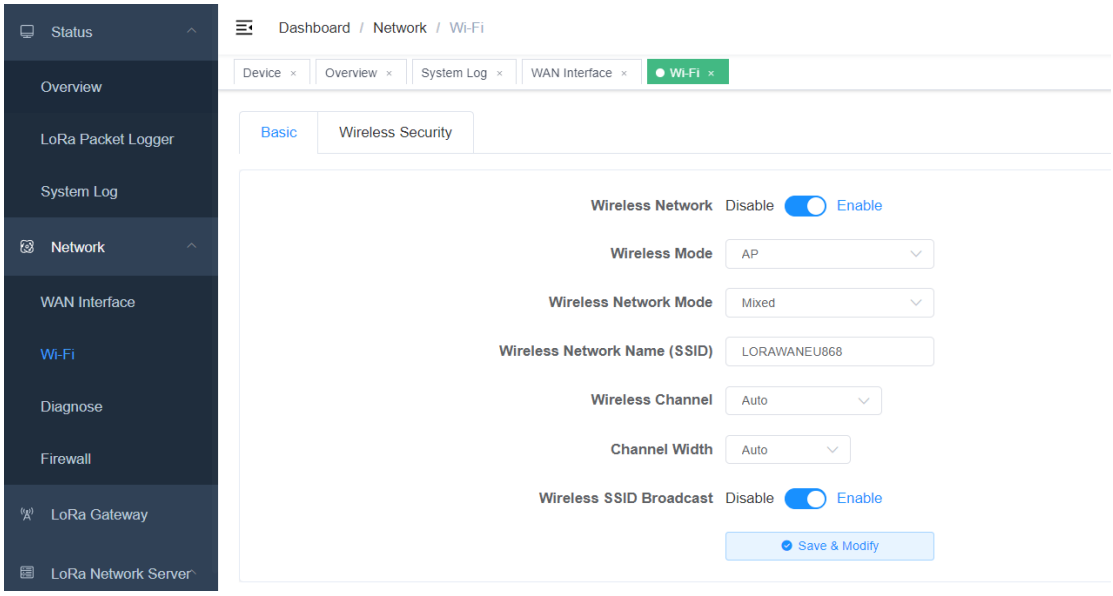
1. Network

- **Path:** Network → WAN Interface
- **Function:** Used to configure network parameters, such as setting up static IP or DHCP.
- **Details:**
  - Configure various modes based on mode parameters.
- **Preview:**：



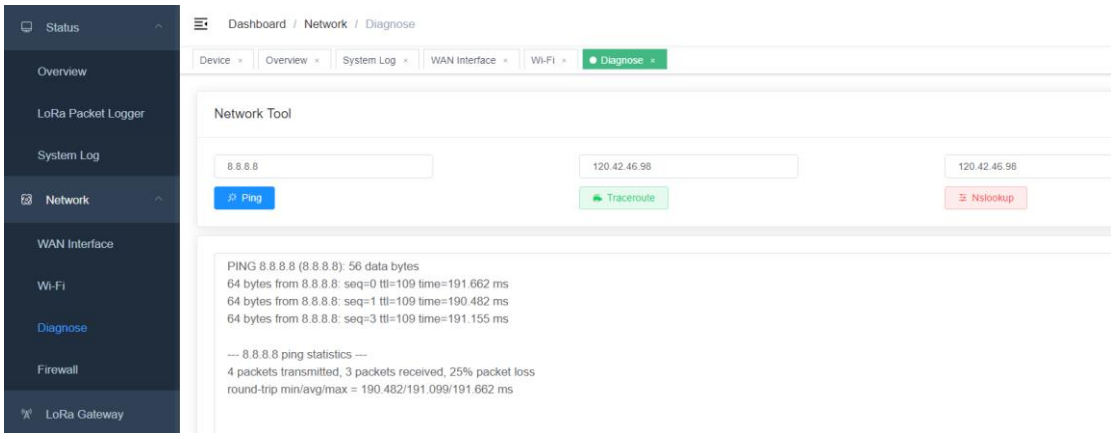
2. WiFi

- **Path:** Network → WiFi
- **Function:** Configures WiFi parameters and security settings.
- **Details:**
  - Configure various modes based on mode parameters.
- **Preview:**：



3. Network Diagnostics

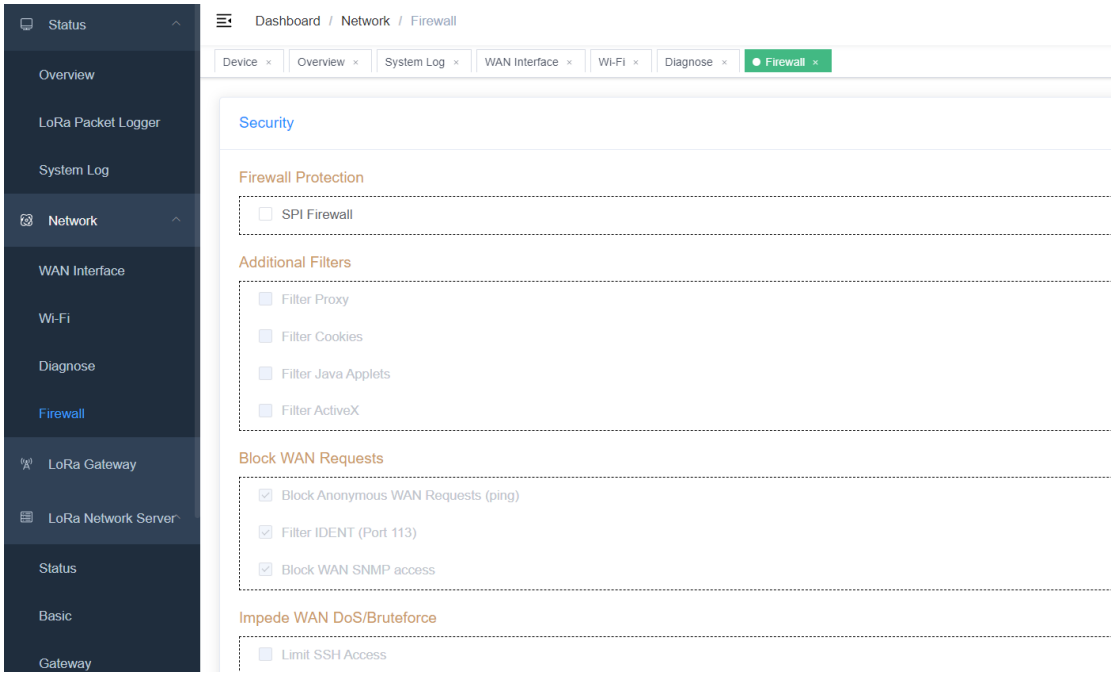
- **Path:** Network → Network Diagnostics
- **Function:** Supports Ping, Traceroute, and NsLookup commands.
- **Details:**
  - **Ping:** A program used to test network connectivity.
  - **Traceroute:** A command that uses ICMP protocol to locate all routers between your computer and the target computer.
  - **NsLookup:** A command-line tool to monitor whether DNS servers in the network can correctly perform domain name resolution.
- **Preview:**



4. Firewall

- **Path:** Network → Firewall
- **Function:** Configuration of firewall-related parameters.
- **Details:**
  - Configure parameters based on the displayed page.

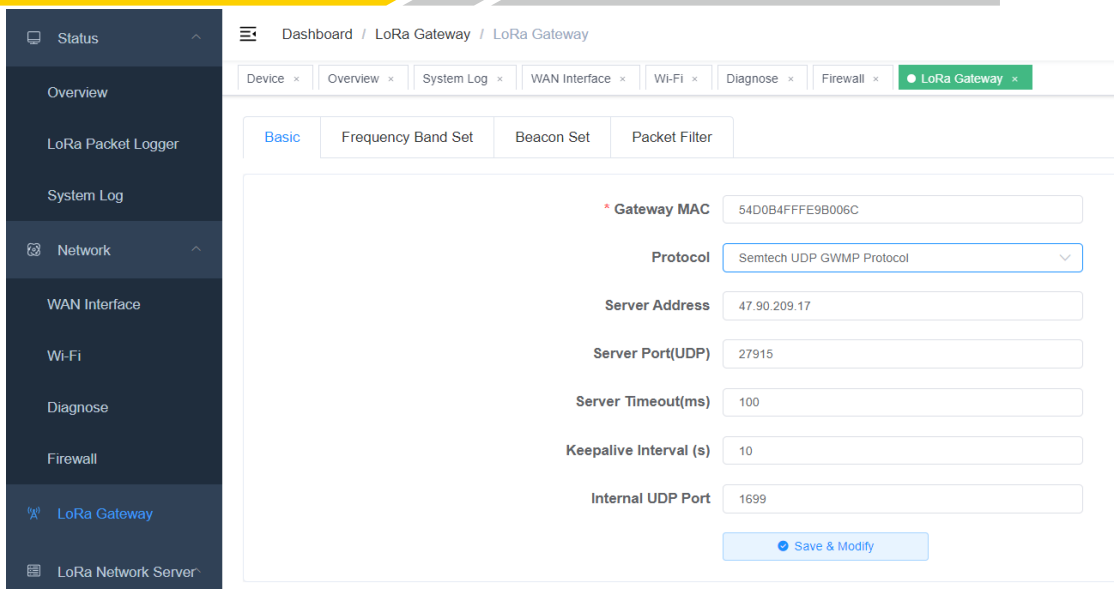
- **Preview:**



4.1.3.2 LoRa Gateway

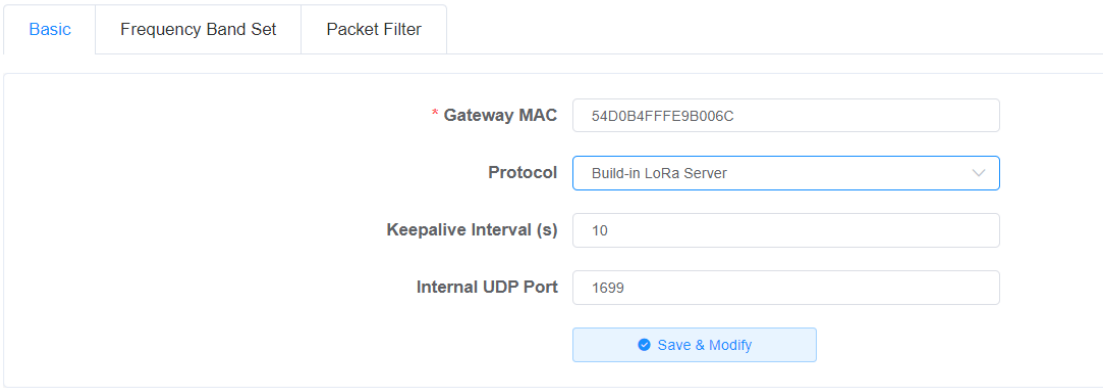
1. Basic Settings

- **Path:** LoRa Gateway → Basic Settings
- **Function:**Configuration of gateway protocols, allowing for settings such as Build-in LoRa Server, Semtech UDP GWMP Protocol, and Basics Station modes.
- **Details:**
  - **Semtech UDP GWMP Protocol** - GWMP Forwarding Mode
    - Gateway MAC: Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
    - Protocol: UDP GWMP protocol, connecting to an external NS server, with the gateway acting as a data forwarding role.
    - Server Address: IP or domain name.
    - Server Port: Port number (e.g., 1700).
    - Server Timeout (ms): Timeout duration for waiting for responses to data reporting, typically not modified.
    - Keepalive Interval (s): Interval for the pull_data command in the protocol, typically not modified.
    - Internal UDP Communication Port: In a cascaded application where the gateway acts as a server, this port number is configured as the server port for the sub-gateway.



➢ **Build-in LoRa Server** - Internal NS Mode

- ❖ Gateway MAC: Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
- ❖ Protocol: Internal NS mode, equivalent to having the NS deployed within the gateway.
- ❖ Keepalive Interval (s): Interval for the pull_data command in the protocol, typically not modified.
- ❖ Internal UDP Communication Port: In a cascaded application where the gateway acts as a server, this port number is configured as the server port for the sub-gateway.

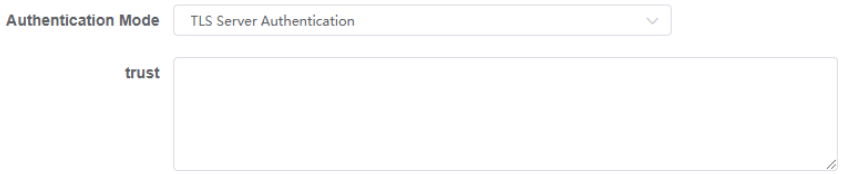


➢ **Basics Station** - More Secure and Reliable Protocol (Connected to NS via WebSocket or HTTP)

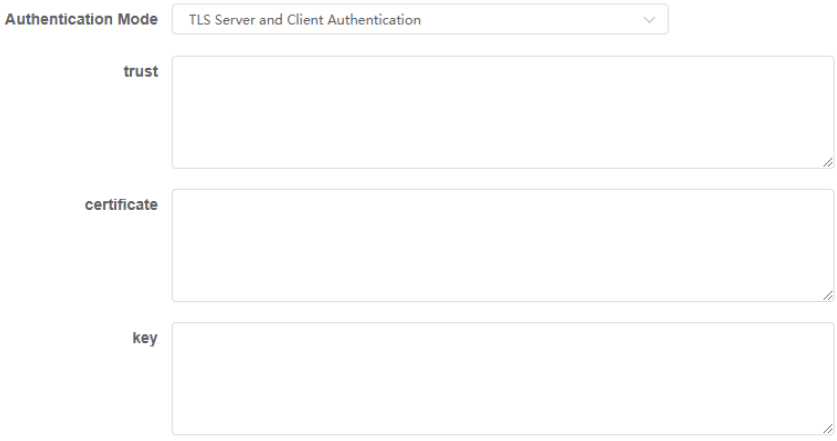
- ❖ **Gateway MAC:** Unique identifier for the gateway, 8 bytes in length (16 bits), typically not modified.
- ❖ **Protocol:** Basicstation mode.
- ❖ **Server:** LNS protocol (select this option for regular data communication) or CUPS protocol (adds gateway upgrade-related protocols).

- **URI:** Server address for connection (IP or domain name).
- **Port:** Corresponding port for the server.
- **Authentication Mode:** Security authentication mode (detailed scenarios for each mode will be explained when connecting to the platform). The following are brief introductions to each mode:
  - **No Authentication:** Establishes a regular WebSocket or HTTP connection without requiring identity verification (e.g., ChirpStack is configured in this mode for integration with the TTN platform).

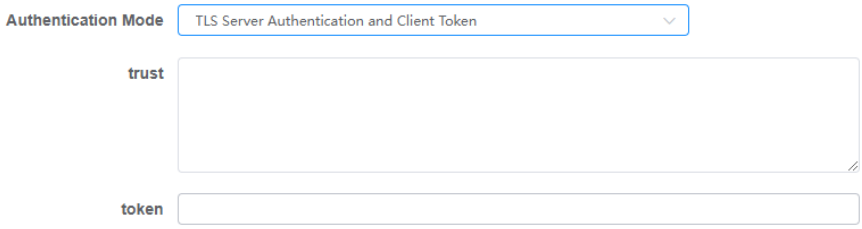
  - **TLS Server Authentication:** Authenticates the server (LNS or CUPS) through establishing a TLS connection (wss, https). （e.g. ChirpStack is configured in this mode)



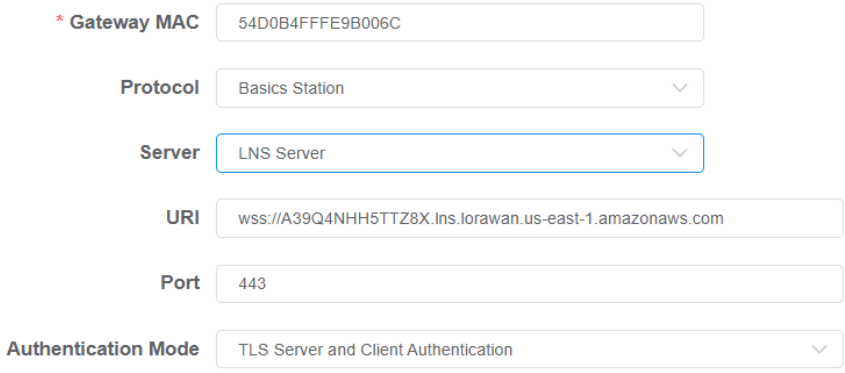
  - **TLS Server and Client Authentication:** Authenticates both the server (LNS or CUPS) and the client (gateway) through establishing a TLS connection (wss, https). The gateway authenticates the server by verifying its certificate, and the server authenticates the gateway by requesting its certificate along with a signature using a private key. (e.g. This mode is used when interfacing with the AWS platform.)



  - **TLS Server Authentication and Client Token:** In this mode, the gateway authenticates the server (LNS or CUPS) by establishing a TLS connection (wss, https). The server, on the other hand, verifies the identity of the gateway by examining the security token provided by the gateway.(e.g. This mode is used when interfacing with The Things Network (TTN) platform.)

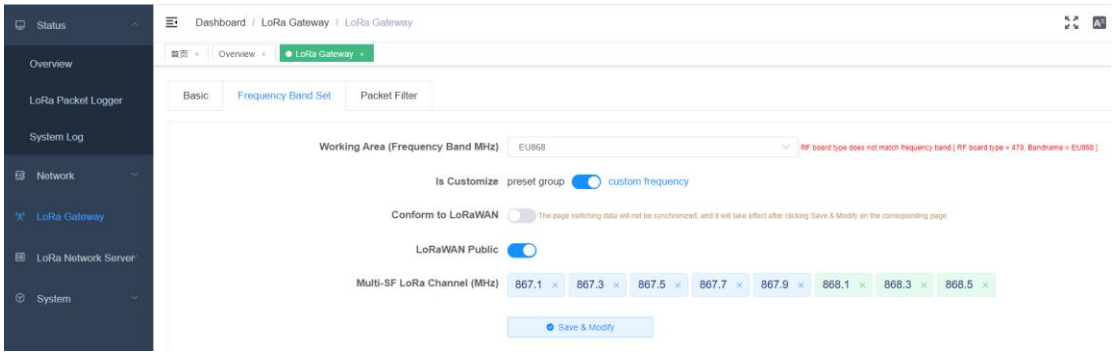


- **Preview:**

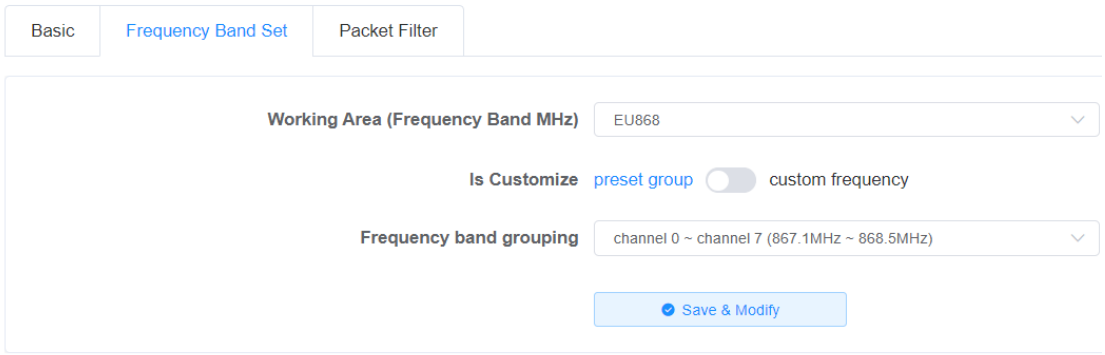


2. Frequency Configuration

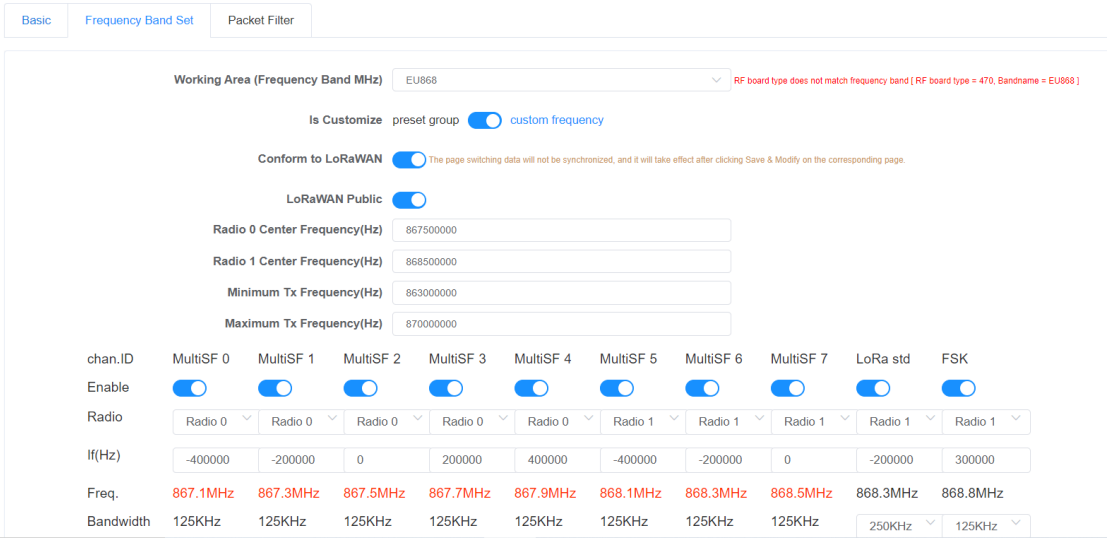
- **Path:** LoRa Gateway → Frequency Configuration
- **Function:** Configuration of gateway frequencies, supporting modes such as Semtech UDP GWMP Protocol or Build-in LoRa Server. For Basics Station mode, frequency settings are configured on the NS server.
- **Details:**
  - Frequency configuration supports three main methods.
    - **Custom Frequency Method:** This method provides a simple and intuitive view of allocated frequencies. In the example below, frequencies on the left (e.g., 867.1) can be deleted, while those on the right (e.g., 868.1) are essential and cannot be removed. Clicking the '×' next to a frequency deletes it, and clicking the '+ Add' on the far right adds a new frequency.



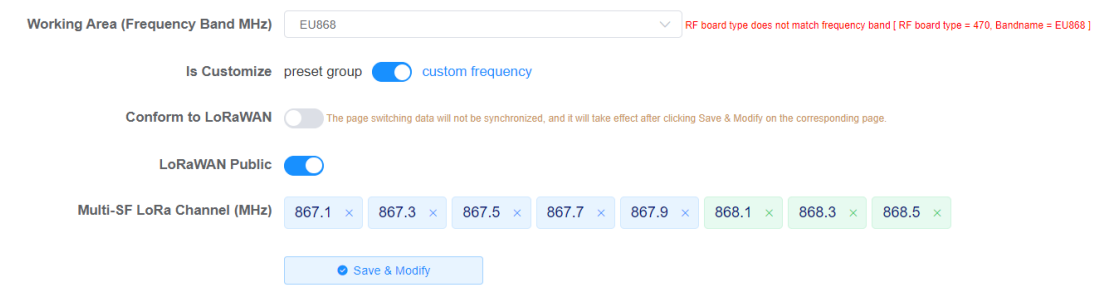
    - **Pre-set Group Method:** This method is the most convenient. Choose the corresponding group as needed, and it will display the starting and ending frequencies. Typically, there is a 0.2MHz interval between each, totaling 8 frequencies.



❖ **Custom Frequency + Conform to LoRaWAN Method:** This method is the most in line with the gateway configuration file structure and is the most comprehensive configuration method. Use this method when the other two methods cannot meet the requirements.



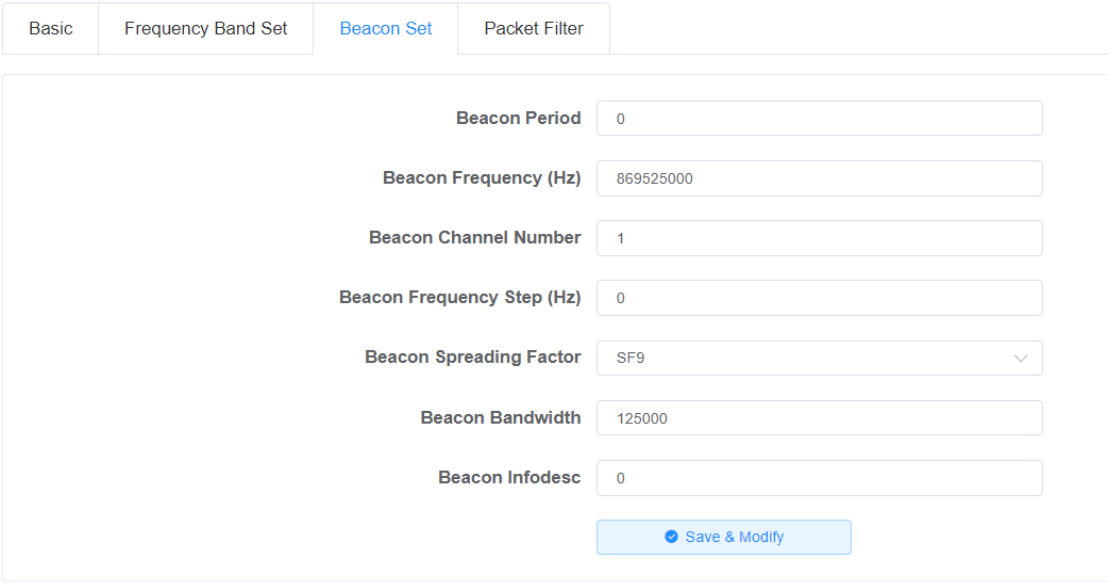
- **Preview:**



3. Beacon Set

- **Path:** LoRa Gateway → Beacon Set
- **Function:** Configure the gateway's ClassB parameters, available in Semtech UDP GWMP Protocol mode.

- **Details:**
  - **Beacon Period:** Period, set to 0 means it is turned off.
  - **Beacon Frequency (Hz):** Frequency point.
  - **Beacon Spreading Factor:** Spreading factor.
  - **Beacon Bandwidth:** Beacon packet bandwidth.
  - **Beacon Tx Power:** Transmit power.
- **Preview:**



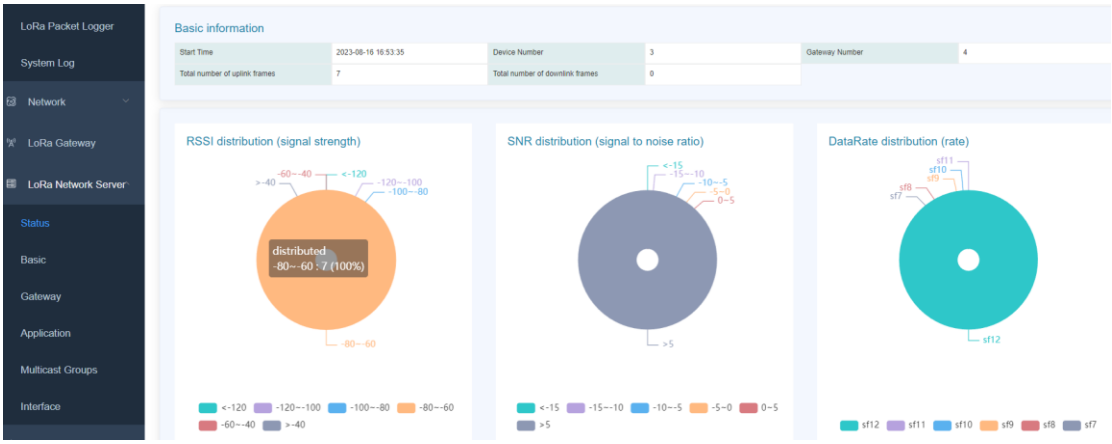
4. Packet Filter

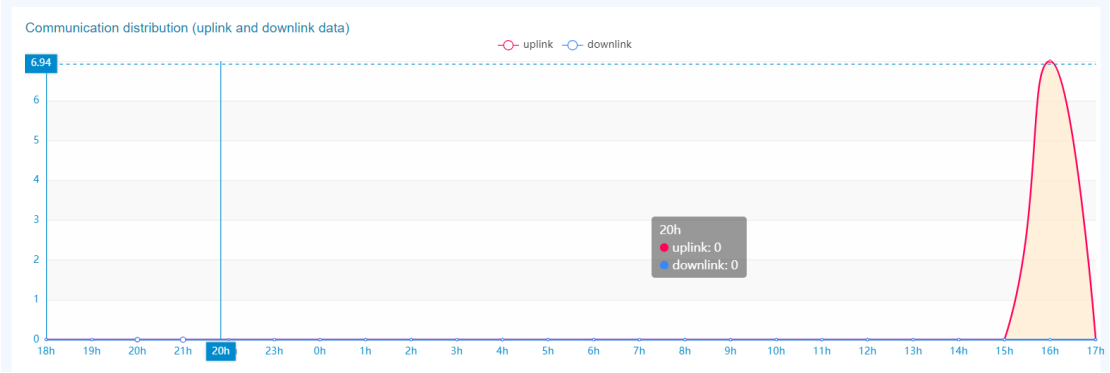
- **Path:** LoRa Gateway → Packet Filter
- **Function:** On the gateway side, filter out some packets based on configured rules to reduce the amount of invalid data transmitted to the NS server, alleviate the processing pressure on the NS. Modes available: Semtech UDP GWMP Protocol or Build-in LoRa Server.
- **Details:**
  - **Supports configuration of NetID and JoinEUI.**
  - **NetID:** Network number filtering. The short address assigned during device joining is associated with the network number. By configuring this value, it can effectively filter out non-join packets and interference data, especially in embedded mode. This value can be configured as the network ID of the gateway to avoid interference from other device data.
  - **JoinEUI (AppEUI):** JoinEUI filtering, one of the triplets of the terminal. Multiple range values can be set here. Once set, JoinEUI outside the range will be filtered.
- **Preview:**

4.1.3.3 LoRa Network Server

1. Status

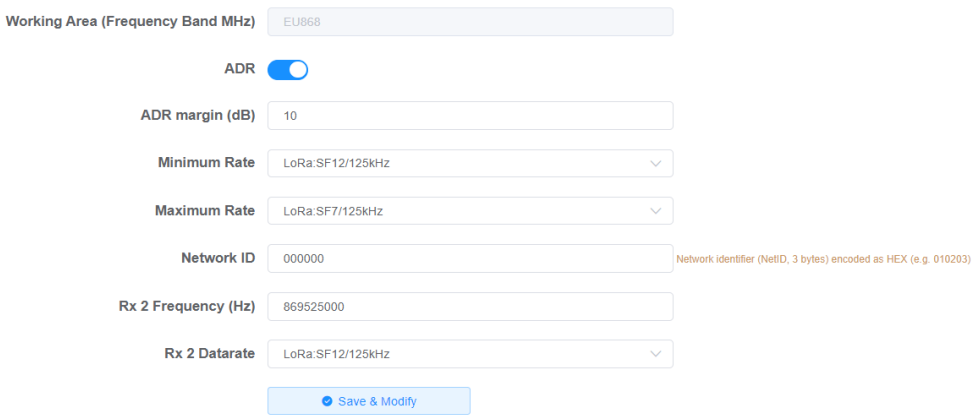
- **Path:** LoRa Network Server → Status
- **Function:** Displays statistics flowing into the built-in NS server.
- **Details:**
  - **Basic Information:** Includes the number of gateways, devices, and statistics for device uplink and downlink data.
  - **RSSI, SNR, DataRate Distribution:** Used to analyze the communication quality between gateways and nodes.
  - **Communication Distribution:** Curve graph of uplink and downlink communication, analyzing whether the distribution of uplink and downlink data matches expectations.
- **Preview:**





2. Basic Settings

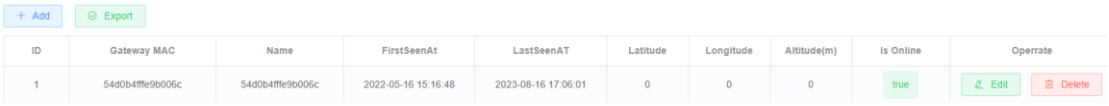
- **Path:** LoRa Network Server → Basic Settings
- **Function:** Configures parameters related to the NS server.
- **Details:**
  - **Work Area:** Corresponds to the region parameter table's frequency band; it cannot be configured here and should match the configuration in LoRa Gateway → Frequency Configuration → Work Area.
  - **Enable Dynamic Data Rate Adjustment (ADR):** Indicates whether ADR functionality is enabled.
  - **ADR Margin:** This value affects the sensitivity of ADR adjustments. A higher value makes adjustments less aggressive, while a lower value makes adjustments more aggressive.
  - **Minimum Data Rate:** The minimum data rate for ADR adjustments.
  - **Maximum Data Rate:** The maximum data rate for ADR adjustments.
  - **Network ID:** Parameter for producing device short addresses, can be configured in filtering parameters to avoid interference.
  - **Rx2 Frequency:** Frequency corresponding to the Rx2 window.
  - **Rx2 Datarate:** Data rate corresponding to the Rx2 window.
  - **Downlink Transmit Power (dBm):** Configures the transmit power. When set to -1, it will follow the specifications in the region parameter table.
- **Preview:**



3. Gateway

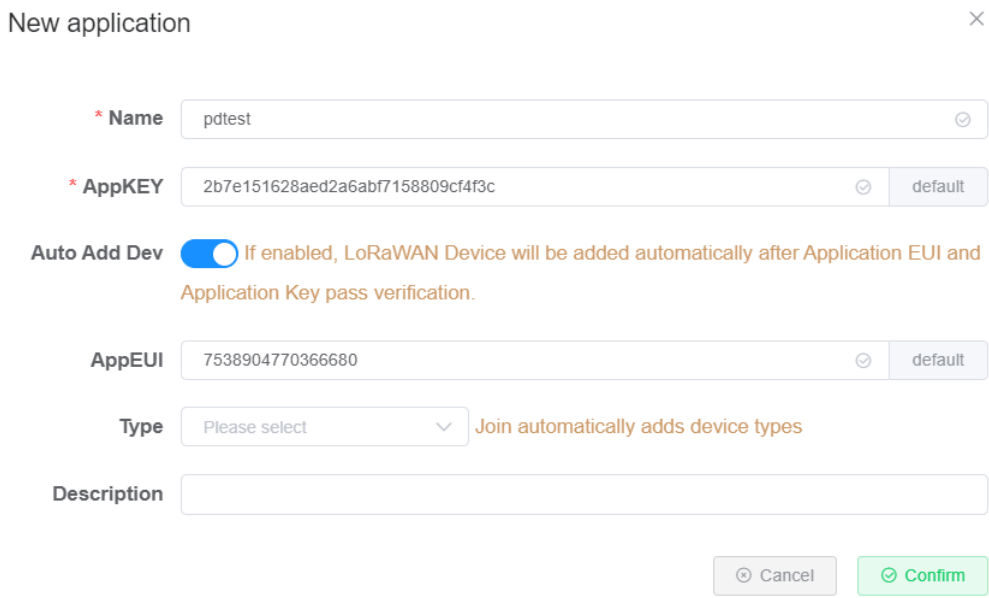
- **Path:** LoRa Network Server → Gateway

- **Function:** Manages the addition, deletion, modification, and viewing of gateways integrated with NS. Gateways generally do not need to be manually added; they will be added automatically when connected.
- **Details:**
  - Displays the list of gateways with detailed information, including online status.
- **Preview:**

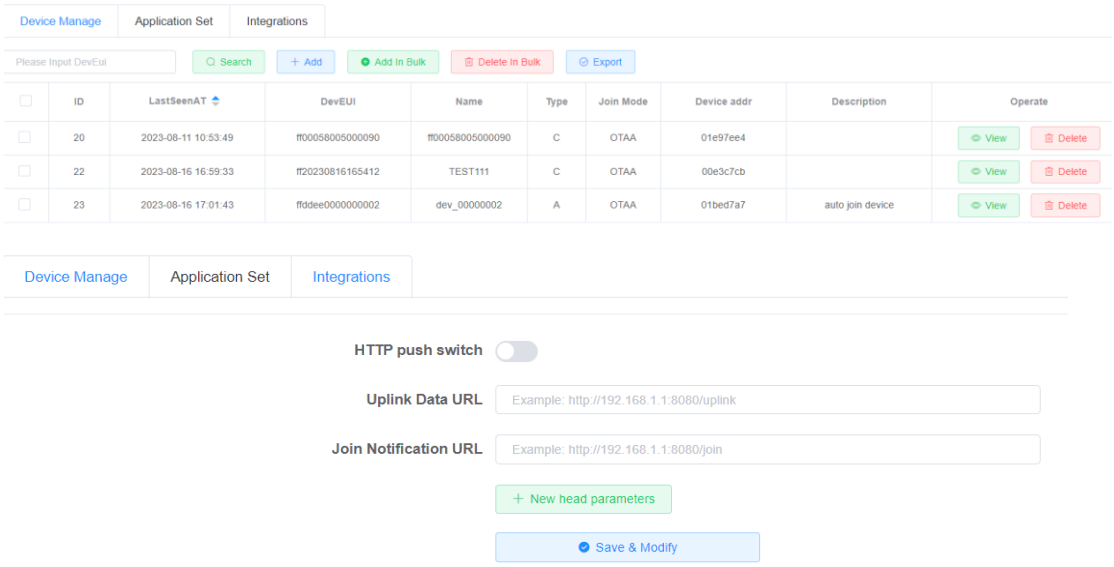


4. Application

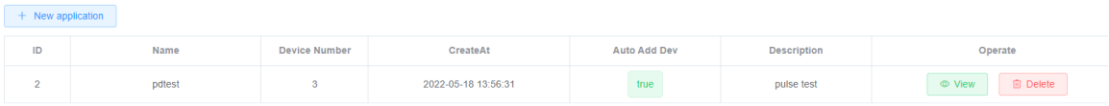
- **Path:** LoRa Network Server → Application
- **Function:** Functions similarly to grouping, where different groups correspond to different application scenarios for easier management.
- **Details:**
  - **Clicking the add button opens the following page.**
    - **Name:** Equivalent to the group name for easy identification.
    - **AppKEY:** Corresponds to the AppKEY of the terminal; this value needs to be verified when adding devices automatically (clicking on the default on the right will change it to the default value of Four-Faith).
    - **Auto-add Devices**: When checked, devices can be added without prior manual addition. After AppKEY and AppEUI validation, devices will be added automatically.
    - **AppEUI (JoinEUI):** One of the triplets; configuration is required when enabling automatic device addition (clicking on the default on the right will change it to the default value of Four-Faith).
    - **Type:** Device type corresponding to the automatic addition of devices: ClassA or ClassC.
    - **Description:** Descriptive information.



- **Delete:** Cannot be deleted if there are devices associated with the application; devices must be deleted first.
- **View:** Entering the application allows access to the device list and more.
  - **Device Management:** Detailed explanation of the device's add, delete, modify, and query functionalities.
  - **Application Settings:** Similar to the initial creation, this allows modifications to existing applications.
  - **Interface Management:** Configuration for HTTP POST; enabling this function will push data from all devices under this application to a specified address using the HTTP POST method.

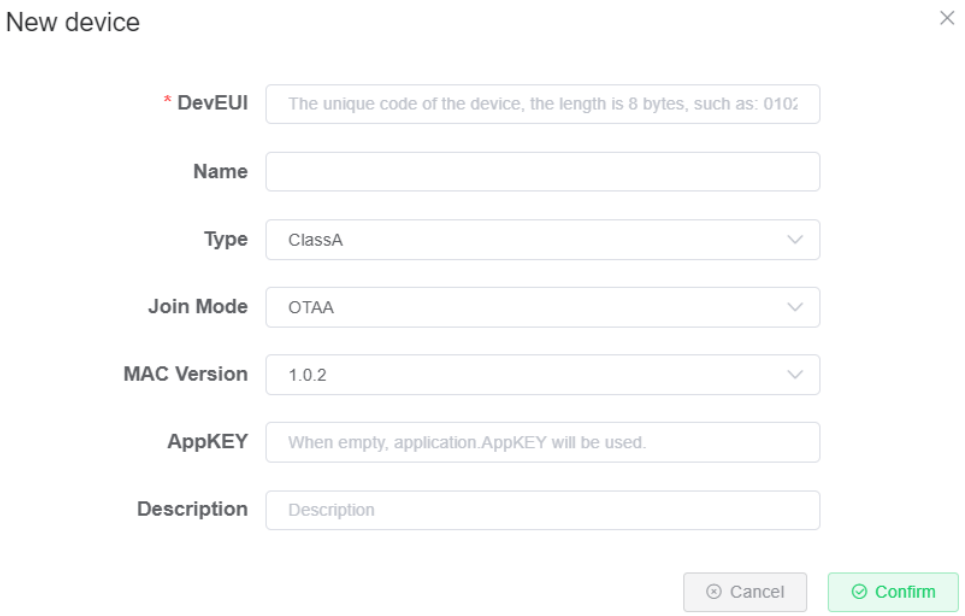


- **Preview:**

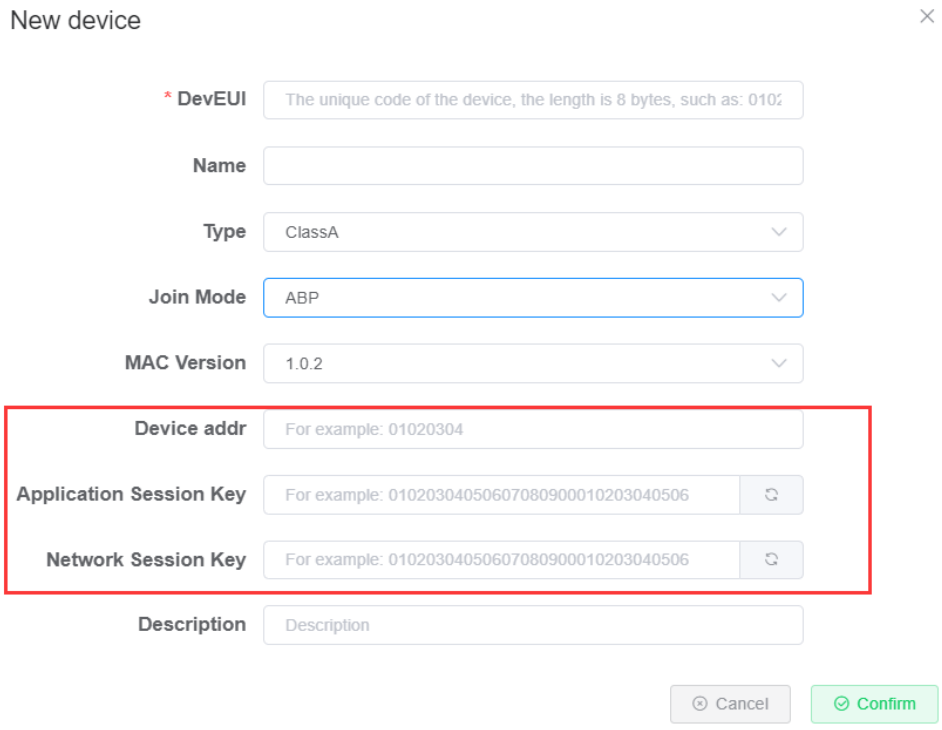


5. Device

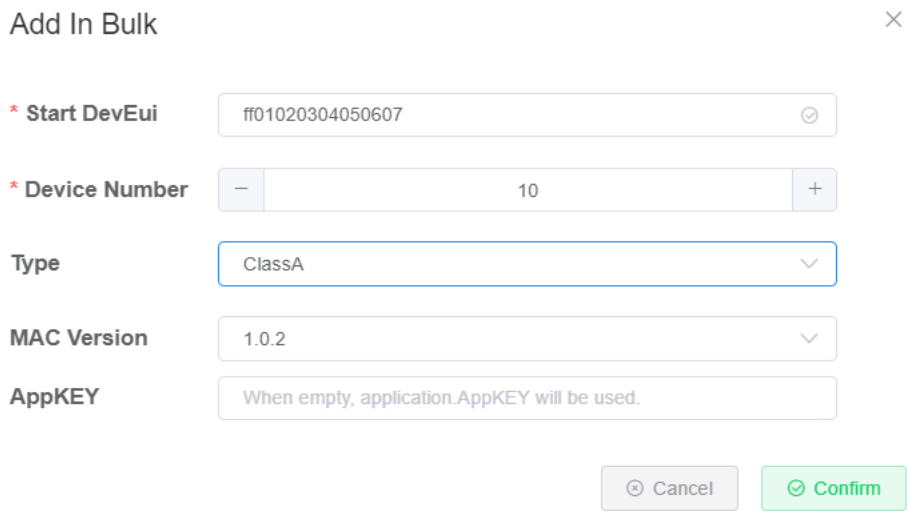
- **Path:** LoRa Network Server → Device
- **Function:** Adding, deleting, modifying, and querying devices. Web entry: LoRa Network Server → Application → View → Device Management
- **Details:**
  - **Add:** Setting basic parameters for the device. Network entry methods include OTAA (device initiates network joining) or ABP (no need for network joining). Specific AppKEY can be entered here when it is different from the AppKEY of the application.



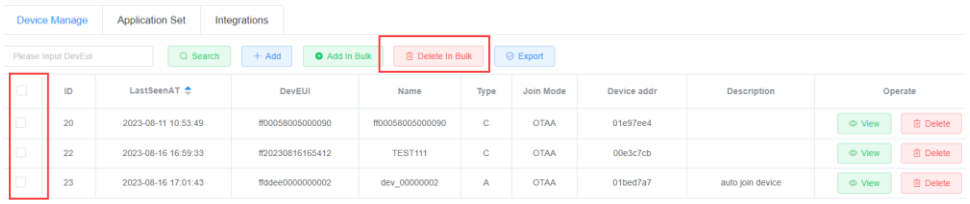
- **ABP Mode:** In this mode, you need to add the short address and session key information in the specified fields.



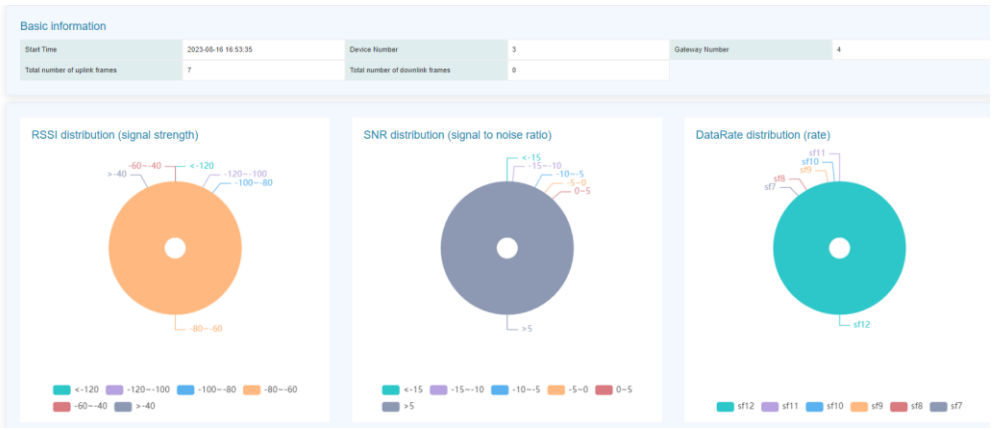
- **Bulk Add:** The parameters for bulk addition are essentially the same as for adding a single device. However, bulk addition is only applicable for OTAA devices.



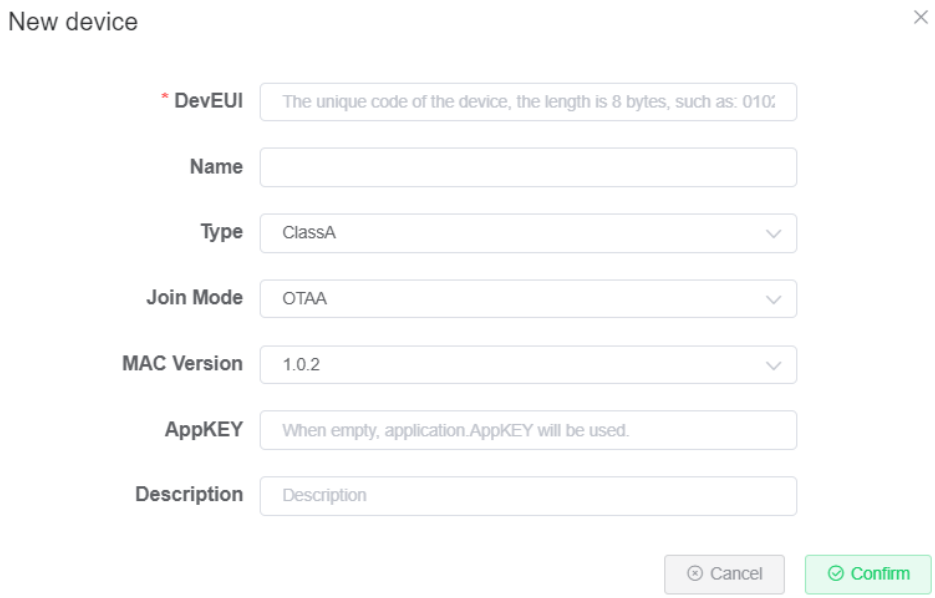
- **Bulk Delete:** First, select the devices you want to delete, then click "Bulk Delete" to remove them in batches.



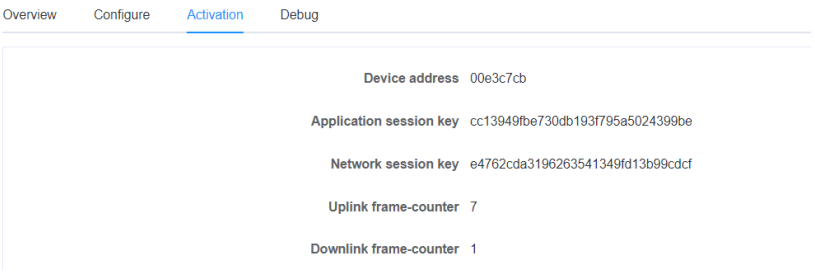
- **Export:** Export data in Excel format for backup and management purposes.
- **Device Details:** Click on "View" on the right side of the corresponding device to enter the details page.
  - **Overview:** Displays the uplink information and relevant statistics for the device, useful for analyzing packet loss and other issues.



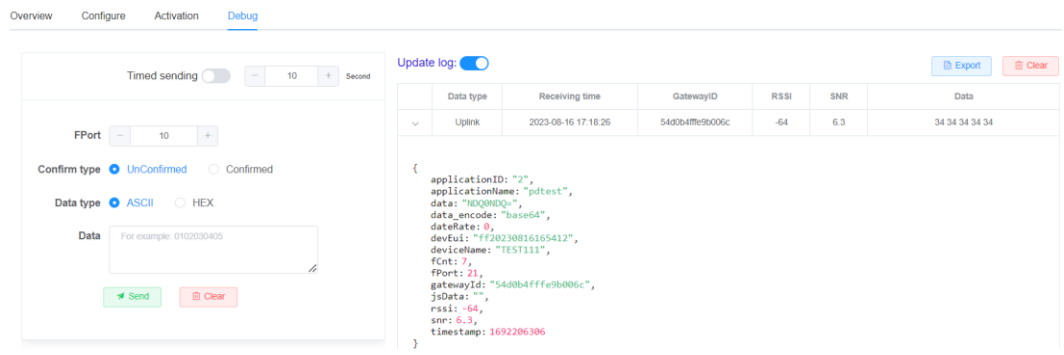
❖ **Configuration:** Adjust the parameters of the device.



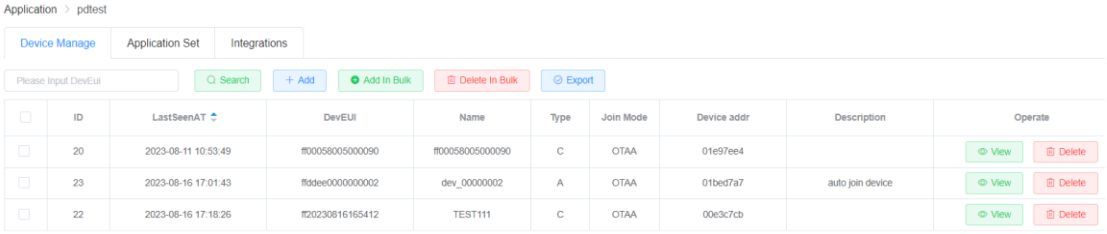
❖ **Activation Information:** Display of parameters after device activation.



❖ **Online Debugging:** Allows for data downlink (scheduled), and displays uplink data, etc.

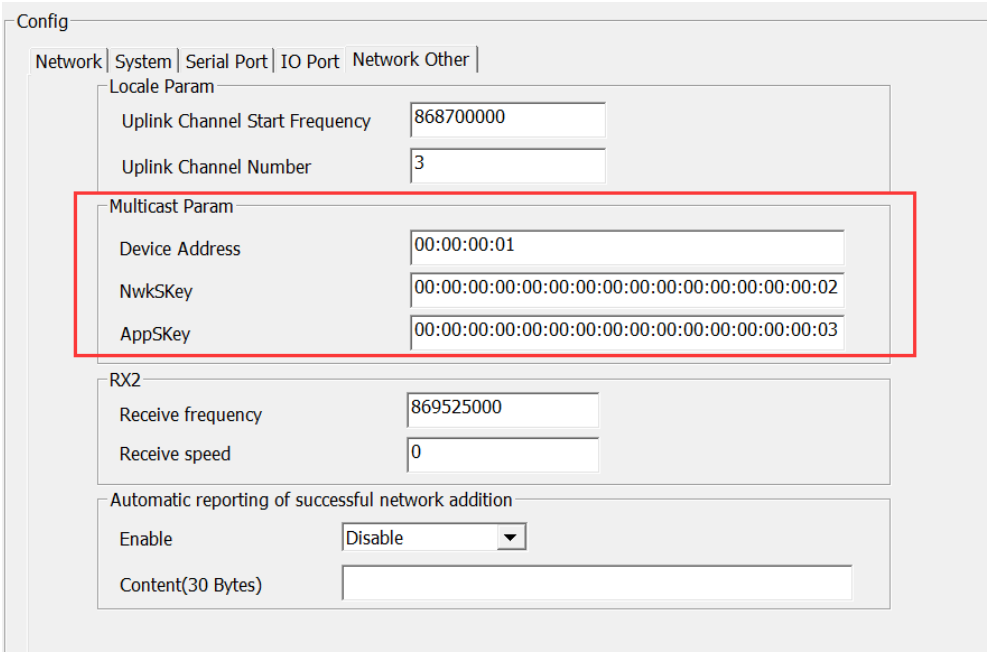


- Preview:

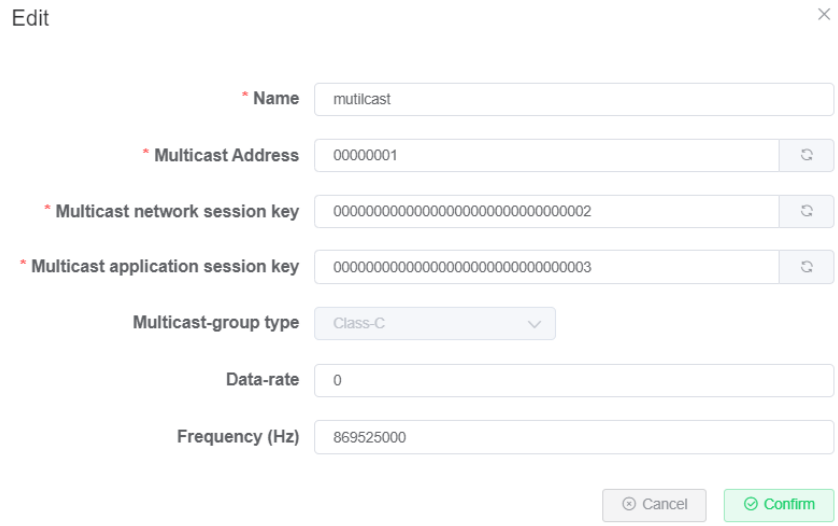


6. Multicast

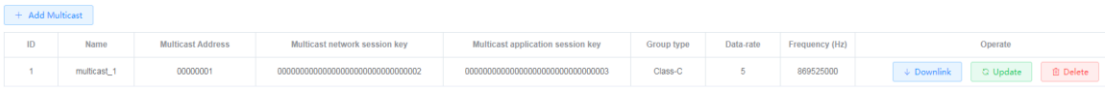
- **Path:** LoRa Network Server → Multicast
- **Functionality:** Multicast here refers to all terminals with the same parameters corresponding to this NS. The downlink of multicast data can be sent via MQTT (the webpage here also supports sending tests).
- **Details:**
  - **Add Multicast:** The values corresponding to Four-Faith terminals are shown below.



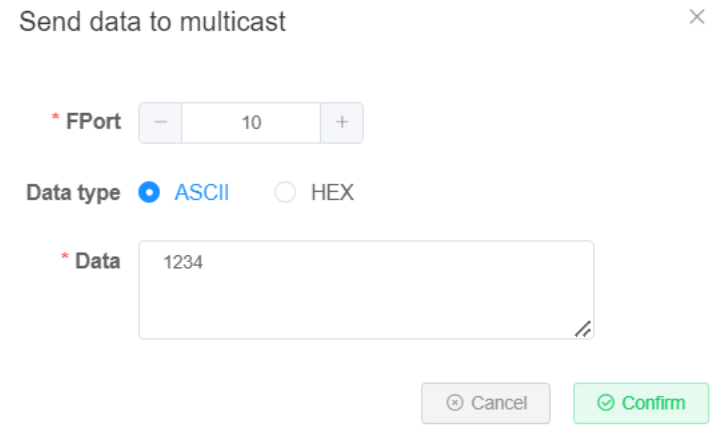
  - Configure multicast parameters based on the values above.

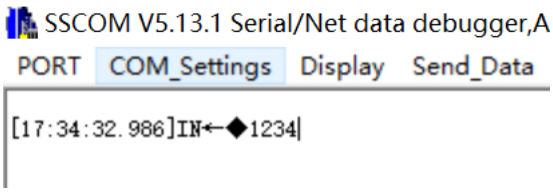


➢ After creation, you will be able to see the following multicast list information.



➢ Sending Data Test: Click "Downlink" to open the data transmission page.



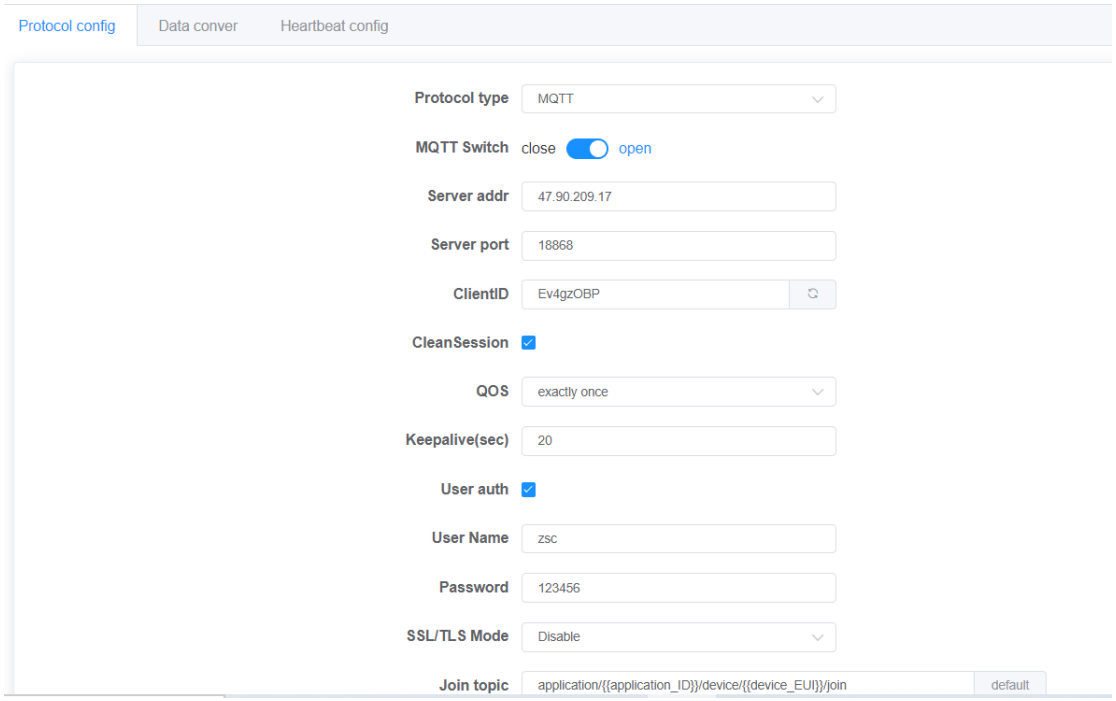


➢ When officially in use, multicast data can be sent via MQTT or TCP. Refer to the data format for specific details.

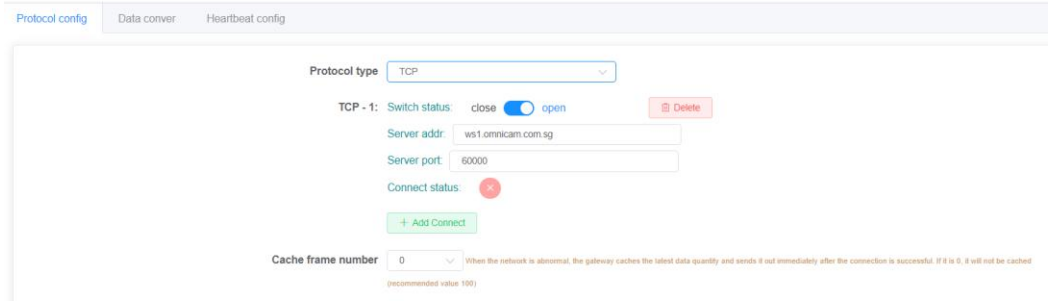
7. Interfaces

- **Path:** LoRa Network Server → Interfaces
- **Function:** Configuration page for integrating NS with client platforms, supporting both MQTT and TCP communication methods. Data can be transformed using JavaScript functions, and heartbeat configurations are supported.
- **Details:**

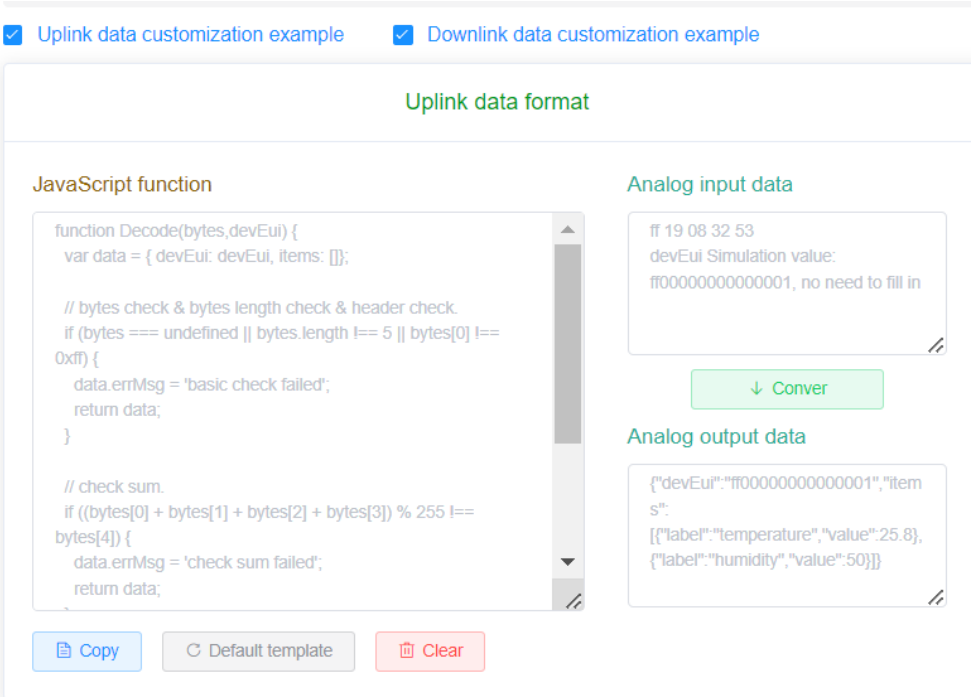
- **Protocol Configuration**
  - **NONE:** Not enabled
  - **MQTT:** Configuration of MQTT parameters. Specific topics and data formats are detailed in the data format section.



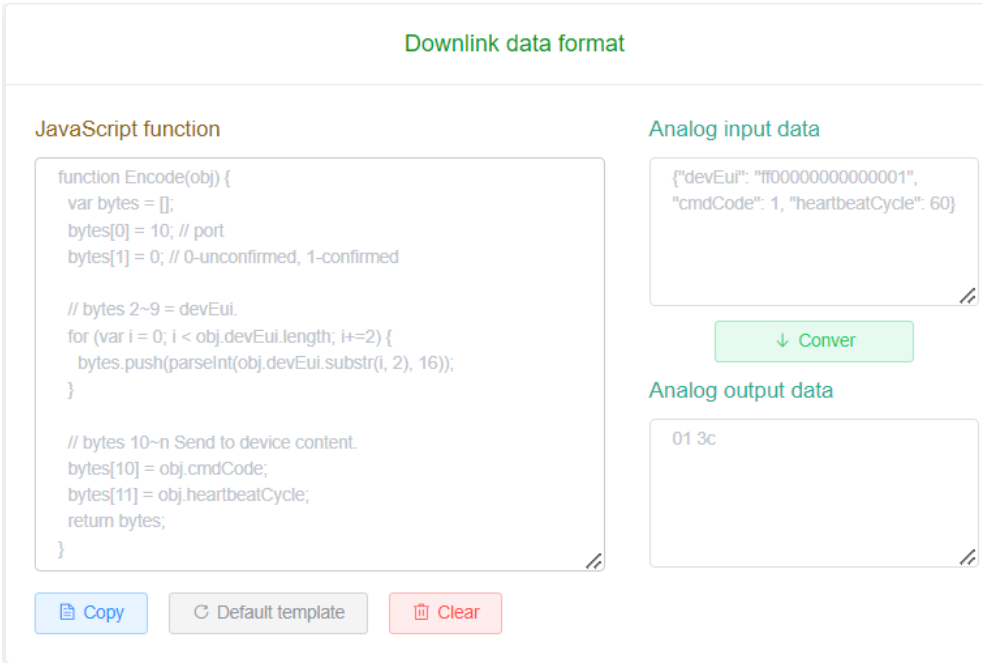
  - **TCP:** Connects to a TCP server, allowing simultaneous connections to multiple servers. Connection status can be monitored to assess the connectivity situation.



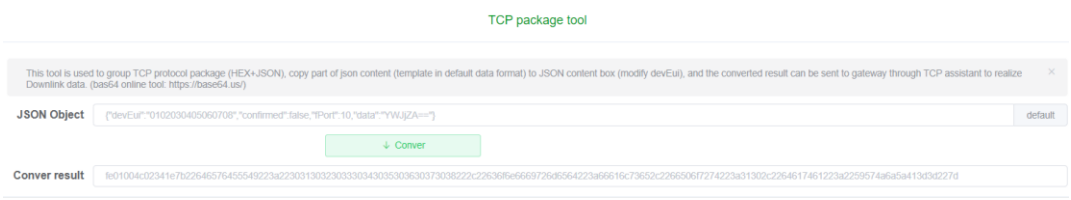
- **Data Transformation:** If not configured here, the default data format will be used for communication. If data transformation is required, functions can be configured for conversion. Upon arrival at the gateway, both uplink and downlink data can be transformed using specified functions before forwarding.
  - **Upstream Transformation**



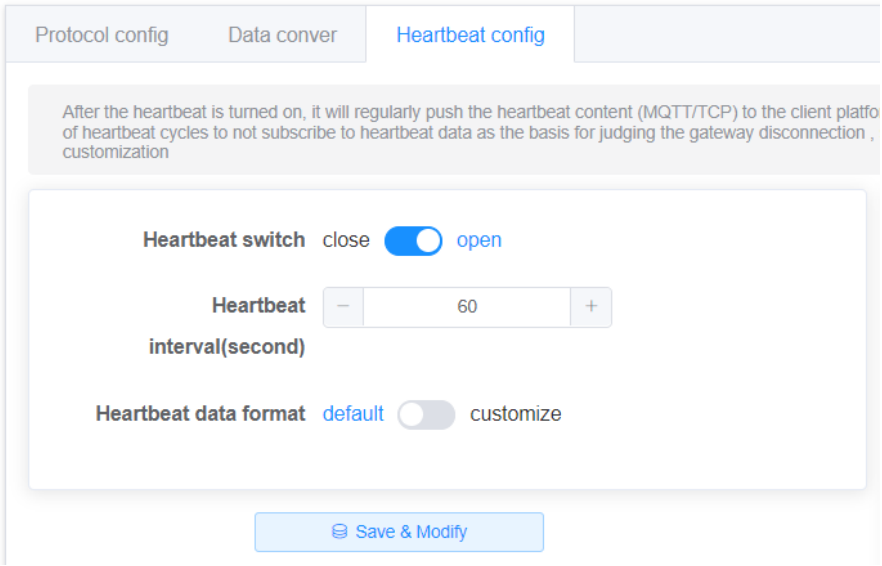
❖ **Downstream Transformation**



❖ **TCP Packet Assembly Tool:** During the testing phase when connecting to a TCP server, this tool can be used to generate corresponding data for testing through the TCP server. In a normal project, a program can be developed to generate this data.



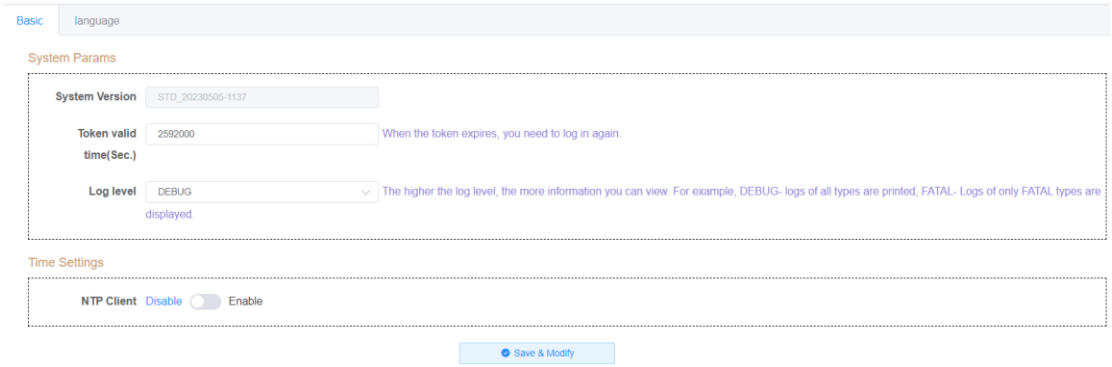
- **Heartbeat Configuration:** You can configure the heartbeat switch, heartbeat interval time, and heartbeat data format. It supports configuration as a custom string. Heartbeat is mainly used for regularly reporting status information. The gateway can also use heartbeat to determine the connection status with the MQTT server.



4.1.3.4 System

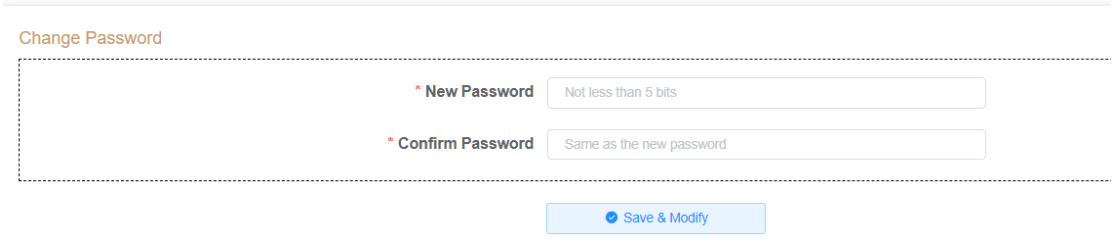
1. **System:**

- **Path:** System → System
- **Functionality:** View program version, configure token duration, time settings, and language switch.
- **Details:**
  - **System Program Version:** Use version information to troubleshoot related issues.
  - **Token Valid Duration:** The shorter the time, the more frequent the need to log in to the web page.
  - **NTP Time Configuration:** Configure.
- **Preview:**



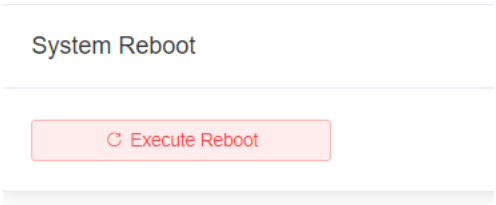
2. **Change Password:**

- **Path:** System → Change Password
- **Functionality:** Modify the password for the gateway system, with a length range of 5-32 characters.
- **Details:**
  - Enter the new password and confirm it. After modification, exit the system. When logging in again, use the new password.
- **Preview:**



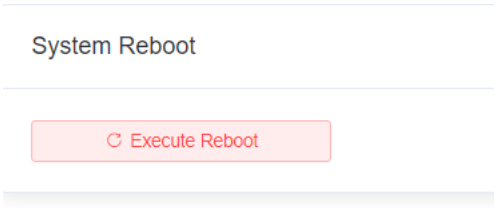
3. **Restart:**

- **Path:** System → Restart
- **Functionality:** Reboot the gateway.
- **Details:**
  - Click to initiate the restart of the gateway.
- **Preview:**



4. **Factory Reset:**

- **Path:** System → Factory Reset
- **Functionality:** Clicking this button will restore the gateway parameters, mainly router-related parameters such as network settings (LoRa-related parameters like device lists and network information will not be deleted).
- **Details:**
  - Click to execute the factory reset.
- **Preview:**



4.1.4 Data Format

4.1.4.1 Data Explanation

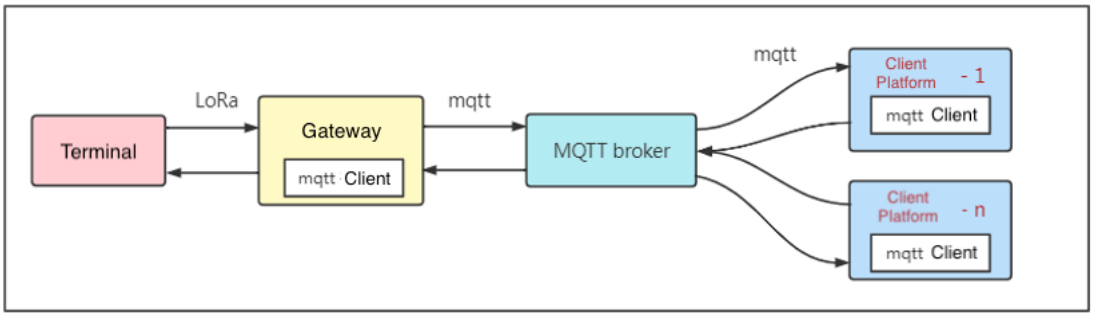
1. Data Format Explanation:

To communicate with the client, the protocols include MQTT, TCP, and HTTP, where MQTT and TCP support bidirectional communication. However, HTTP only supports the gateway pushing data to the client via the POST method and does not support downstream communication. The data formats for each protocol are as follows:

- **MQTT Data:** Topic + JSON Content
- **TCP Data:** Data Header + JSON Content
- **HTTP Data:** URL + JSON Content

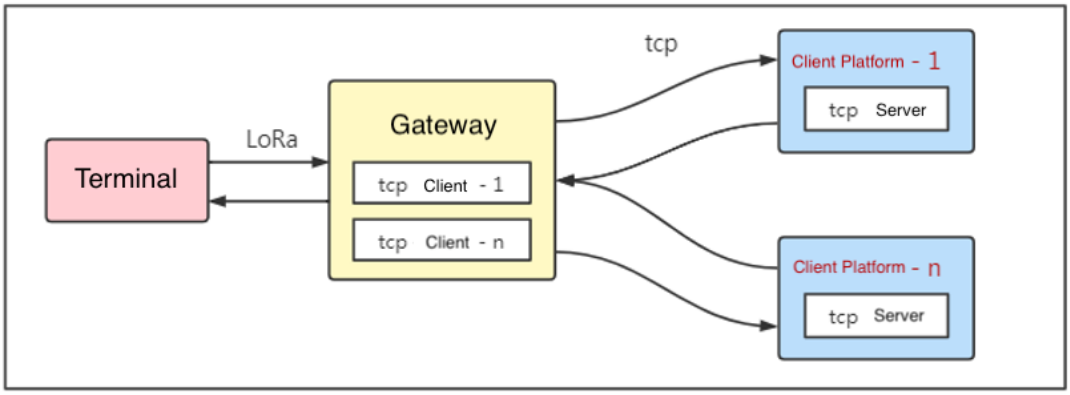
Note: The JSON content data format is entirely consistent within the same type. If JavaScript function conversion is applied, it will be universally applied to all formats.

2. MQTT Data Flow:



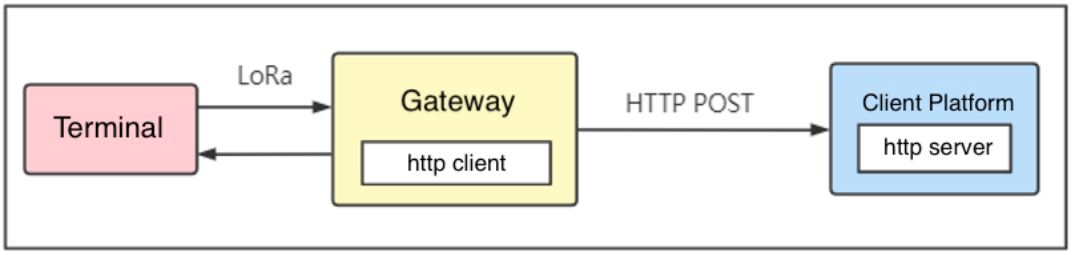
As shown in the above diagram, in this mode, you need to deploy an MQTT Broker first. Both the gateway and the client platform establish connections with it and subscribe to corresponding topics according to the topic format. If the client needs multiple sets of data, this can be achieved by connecting multiple clients and subscribing. In comparison to the TCP mode, this method involves an additional step of deploying an external MQTT server.

3. TCP



As shown in the diagram above, in this mode, the client platform opens a TCP server, and the gateway is configured in TCP mode, pointing to the IP and port of the corresponding server. This allows multiple TCP connections to be established simultaneously.

4. HTTP

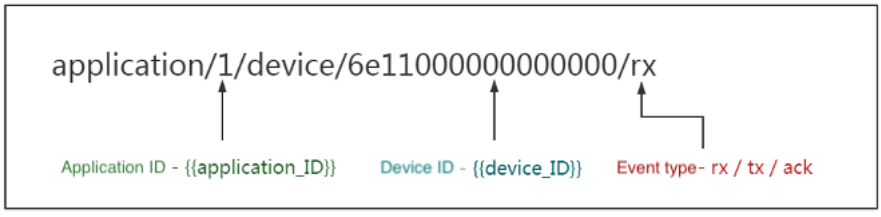


As shown in the diagram above, the configuration for this mode is in the interface settings of each application. This mode only supports data push and does not support downstream data.

4.1.4.2 MQTT Data Format

1. MQTT Topic and Data Format

- Default MQTT Topic Format



- MQTT format primarily includes topics and data content, which can be displayed and modified in the interface.
- The default topic contains {{application_ID}} and {{device_EUI}}.

- {{application_ID}}: Application ID, which will be replaced with the corresponding application ID when reporting data (e.g., application/1/device/6e11000000000000/rx). It also needs to be replaced with the actual application ID when sending downlink data (e.g., application/1/device/6e11000000000000/tx).
- {{device_EUI}}: Device's unique identifier, which will be replaced with the device's EUI when reporting data (e.g., application/1/device/6e11000000000000/rx). It also needs to be replaced with the actual device EUI when sending downlink data (e.g., application/1/device/6e11000000000000/tx). If this field is present in the topic, the JSON content of the downlink data may omit the device's unique identifier.

- Modification Instructions:

  - Topics can be modified, such as changing it to lorawan/uplink.
  - {{application_ID}}: Can be deleted. If removed, only the application ID will be excluded.
  - {{device_EUI}}: If removed, the topic won't recognize the corresponding device. In this case, the JSON content of downlink data must include the device's unique identifier (as explained in the subsequent content).
- Examples of Subscribing to Topics:

  - Subscribe to a specific event for a single device: application/1/device/6e11000000000000/rx
  - Subscribe to all events for a single device: application/1/device/6e11000000000000/+
  - Subscribe to a specific event for all devices under an application: application/1/device/+/rx
  - Subscribe to all events for all devices under an application: application/1/device/#
  - Subscribe to a specific event for all devices under all applications: application/+/device/+/rx
  - Subscribe to all events for all devices under all applications: application/+/device/+/+ or application/##

- The data content follows the JSON format, and the specific format is detailed later. It's essential to note that if the {{device_EUI}} is removed from the downlink topic, the topic won't recognize the specific device. In such cases, you need to look for the "devEui" field in the data content. If it also doesn't exist, the device-specific data will be lost.

2. Uplink Data:

- **Execution Condition:** Forwarded upon receiving business data reports from devices

that have joined the network.

- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/rx
- **Example Default Topic:** application/1/device/6e11000000000000/rx
- **Example Default JSON Data Content:**

```
{
  "applicationID": "1",
  "applicationName": "temperature",
  "deviceName": "dev_00000000",
  "devEui": "6e11000000000000",
  "rxInfo": [{
    "gatewayID": "ff0000000000000a",
    "name": "ff0000000000000a",
    "time": "", // Only applicable when the gateway can receive GPS signals to provide actual values.
    "rssi": -76,
    "loRaSNR": 7.5,
    "location": {
      "latitude": 0,
      "longitude": 0,
      "altitude": 0
    }
  }],
  "txInfo": {
    "frequency": 868100000,
    "dr": 0
  },
  "adr": false,
  "fCnt": 6,
  "fPort": 32,
"data": "MTQ1OTYzNTgy" // Base64 encoding, as explained in the later section "Base64 Encoding and
Decoding."
```

```
}
```

3. **Join Data:**

- **Execution Condition:** Pushed upon receiving a device's join request and responding to the join accept packet.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/join
- **Example Default Topic:** application/1/device/6e11000000000000/join
- **Example Default Data Content:**

```
{
  "applicationID": "1",
  "applicationName": "temperature",
  "deviceName": "dev_00000000",
  "devEui": "6e11000000000000",
  "devAddr": "01b0e489"
}
```

4. **Downlink Data:**

- **Execution Condition:** Sent when business data needs to be delivered to the device.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/tx
- **Example Default Topic:** application/1/device/6e11000000000000/tx
- **Example Default Data Content:**

```
{
  "devEui": "6e11000000000000",
  "confirmed": true,
  "fPort": 12,
  "data": "MTIzNA==" // Base64 encoding, as explained in the later section "Base64 Encoding and Decoding." In this context, it corresponds to "1234."
}
```

- **Convenient Test Data**(The data above contains spaces which may lead to sending failures)

```
{"devEui":"6e11000000000000","confirmed":true,"fPort":12,"data":"MTIzNA=="}
```

5. **Downlink Acknowledgment Response:**

- **Execution Condition:** After receiving the downlink acknowledgment, push pending device responses.
- **Default Topic Format:** application/{{application_ID}}/device/{{device_EUI}}/ack
- **Default Topic Example:** application/1/device/6e11000000000000/ack
- **Default JSON Data Content Example:**

```
{
  "applicationID": "1",
  "applicationName": "temperature",
  "deviceName": "dev_00000000",
  "devEui": "6e11000000000000",
  "acknowledged": true
}
```


6. Downlink Multicast Data:

- **Execution Condition:** To send multicast information to devices with the same triplets in the multicast group.
- **Default Topic Format:** mcast_group/{{mcast_ID}}/tx
- **Default Topic Example:** mcast_group/1/tx
- **Default JSON Data Content Example:**


```
{
  "multicastGroupId": 1,
  "fPort": 10,
  "data": "YWJjZA==" // base64 Encoding
}
```


- **Convenient Test Data Example:**


```
{"multicastGroupId":1,"fPort":10,"data":"YWJjZA=="}
```


7. Heartbeat Data:

- **Execution Condition:** Heartbeat switch is turned on, heartbeat interval > 0, and heartbeat content is non-empty.
- **Default Topic:** lorawan/heartbeat
- **Default JSON Data Content Example:**


```
{
  "gateways": [{
    "gatewayID": "ff0000000000000a",
    "gatewayName": "ff0000000000000a",
    "lastSeenAt": "2022-04-29 14:18:36",
    "isOnline": true,
    "longitude": 0,
    "latitude": 0
```

```
    }],
    "applications": [{
      "applicationID": 1,
      "name": "app",
      "deviceNum": 1,
      "activatNum": 1,
      "isAutoJoin": false
    }]
}
```

4.1.4.3 TCP Data Format

1. TCP Data Format

Offset	Bytes	Function	Identifier	Value Example
0	1	Frame Header	header	0xFE
1	1	Version Number (Current V1)	version	0x01
2	2	JSON Data Length (Big Endian)	length	0x0001
4	1	Data Type	type	0x00- Heartbeat Packet
5	2	Random Key (Big Endian)	random	0x1234
7	n	JSON Content	JSON Object	{...}

- The first 7 bytes represent the TCP data header, and starting from the 7th byte is the JSON content, which is the same as MQTT and HTTP.

2. Uplink Data

Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01

2	2	length	0x018A
4	1	type	0x01
5	2	random	0x1234
7	394	JSON object	{ "applicationID": "2", "applicationName": "app1", "deviceName": "dev_00000001", "devEui": "ff00000000000001", "rxInfo": [{ "gatewayID": "54c345fffed5a1e3", "name": "54c345fffed5a1e3", "time": "2021-11-19T01:51:01.136686Z", "rssi": -107, "loRaSNR": 7.5, "location": { "longitude": 118.03394, "latitude": 24.48405, "altitude": 89 } }], "txInfo": { "frequency": 923400000, "dr": 4 }, "adr": false, "fCnt": 4,

"fPort": 32,

"data": "YWJjZA=="

}

	Description	Type
applicationID	Application ID	string
applicationName	Application Name	string
deviceName	Device Name	string
devEui	Device EUI	string
rxInfo	Gateway Information for Received Data	Struct Array
- gatewayID	Gateway EUI	string
- name	Gateway Name	string
- time	GPS Time	string
- rssi	Signal Strength	float64
- loRaSNR	Signal-to-Noise Ratio	float64
- location	GPS Location (Empty when no GPS signal)	
- longitude	Longitude	float64
- latitude	Latitude	float64
- altitude	Altitude	float64
TxInfo	Device Data Transmission Parameters	
- frequency	Frequency	uint32
- dr	Rate	uint8
adr	ADR Request Status	bool
fCnt	Uplink Frame Counter	uint32
fPort	Uplink Port	uint8
data	Business Data (Base64 encoded format)	string

3. Join Data

<table>
<tr><th>Offset</th><th>Bytes</th><th>Function</th><th>Value or Description</th></tr>
<tr><td>0</td><td>1</td><td>header</td><td>0xFE</td></tr>
<tr><td>1</td><td>1</td><td>version</td><td>0x01</td></tr>
<tr><td>2</td><td>2</td><td>length</td><td>0x007B</td></tr>
<tr><td>4</td><td>1</td><td>type</td><td>0x03</td></tr>
<tr><td>5</td><td>2</td><td>random</td><td>0x1234</td></tr>
<tr><td>7</td><td>123</td><td>JSON object</td><td>{
"applicationID": "2",
"applicationName": "app1",
"deviceName": "dev_00000001",
"devEui": "ff00000000000001",
"devAddr": "032013ac"
}
<table>
<tr><th></th><th>Description</th><th>Type</th></tr>
<tr><td>applicationID</td><td>Application ID</td><td>string</td></tr>
<tr><td>applicationName</td><td>Application Name</td><td>string</td></tr>
<tr><td>deviceName</td><td>Device Name</td><td>string</td></tr>
<tr><td>devEui</td><td>Device Unique Identifier</td><td>string</td></tr>
<tr><td>devAddr</td><td>Short Address Assigned by Device Joining</td><td>string</td></tr>
</table>
</td></tr>
</table>

4. Downlink Data

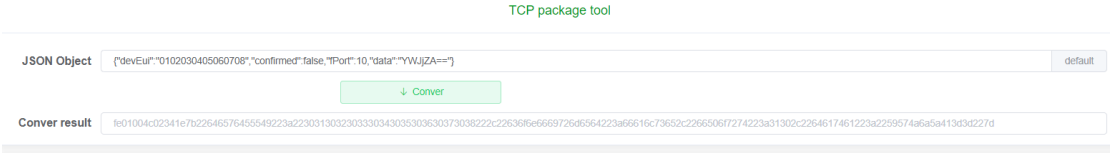
Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x004D
4	1	type	0x02
5	2	random	0x1234
7	77	JSON	{

<table>
<tr><td></td><td></td><td>object</td><td>"devEui": "ff00000000000001",
"confirmed": false,
"fPort": 10,
"data": "YWJjZA=="
}
<table><tr><th></th><th>Description</th><th>Type</th></tr><tr><td>devEui</td><td>Device EUI</td><td>string</td></tr><tr><td>confirmed</td><td>Whether to acknowledge the packet
(default is false)</td><td>bool</td></tr><tr><td>fPort</td><td>port (default is 10)</td><td>uint8</td></tr><tr><td>data</td><td>business data sent (base64 encoded)</td><td>string</td></tr></table></td></tr>
</table>

- Convenient test data (the data above contains spaces, which may cause transmission failures at times)

```
{"devEui":"ff00000000000001","confirmed":true,"fPort":10,"data":"MTIzNA=="}
```

Note: You can use the TCP Packet Tool on the webpage (Path: LoRa Network Server -> Interface -> Data Transformation -> TCP Packet Tool) to generate corresponding data for testing, as shown below:



The data that can be used for testing with TCP server (devEui may need to be modified during testing) includes:：
fe01004b0204427b22646576455549223a22666630303030303030303030303030303031222c22636f6e6669726d6564223a747275652c2266506f7274223a31302c2264617461223a224d54497a4e413d3d227d

5. Downlink Acknowledgment Packet Response

Offset	Bytes	Function	Description or Value
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x0000

4	1	type	0x05
5	2	random	0x1234
7	77	JSON object	{ "applicationID": "1", "applicationName": "app1", "deviceName": "dev_00000000", "devEui": "6e00000000000000", "acknowledged": true }

	Description	Type
applicationID	Application ID	String
applicationName	Application Name	String
deviceName	Device Name	String
devEui	Device Unique Identifier	String
acknowledged	Response Status (Success: true)	Bool

6. Downlink Multicast Data

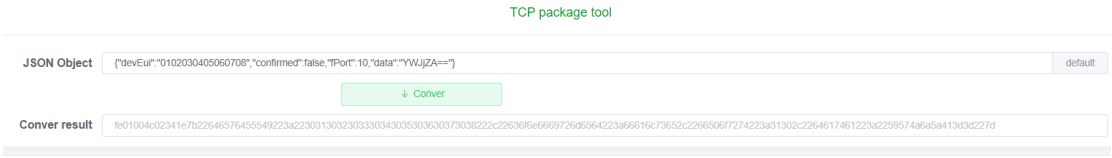
Offset	Bytes	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x0000
4	1	type	0x04
5	2	random	0x1234
7	77	JSON object	{ "multicastGroupId": 1, "fPort": 10, "data": "YWJjZA==" }

	Description	Type
multicastGroupId	Multicast ID	int
fPort	Port（Default 10）	uint8
data	Sending business data (Base64 encoded format)	string

- **Convenient test data**

```
{"multicastGroupId":1,"fPort":10,"data":"YWJjZA=="}
```

Note: You can utilize the TCP Packet Tool on the web page (Path: LoRa Network Server → Interface → Data Conversion → TCP Packet Tool) to generate corresponding data for testing, as shown below:



The converted result can be used for testing by sending it to the TCP server. The content of the example above is:

```
fe01003304341e7b226d756c74696361737447726f75704964223a312c2266506f7274223a31302c2264617461
223a2259574a6a5a413d3d227d
```

7. Heartbeat Data

<table>
<tr><th>Offset</th><th>Bytes</th><th>Functions</th><th>Value or Description</th></tr>
<tr><td>0</td><td>1</td><td>header</td><td>0xFE</td></tr>
<tr><td>1</td><td>1</td><td>version</td><td>0x01</td></tr>
<tr><td>2</td><td>2</td><td>length</td><td>0x01BC</td></tr>
<tr><td>4</td><td>1</td><td>type</td><td>0x00</td></tr>
<tr><td>5</td><td>2</td><td>random</td><td>0x1234</td></tr>
<tr><td>7</td><td>n</td><td>JSON object</td><td>
<pre>{
    "gateways": [{
        "gatewayID": "54D0B4FFFE3AB6CE",
        "lastSeenAt": "2021-11-18 15:34:02",
        "isOnline": true,
        "longitude": 118.03394,
        "latitude": 24.48405
    }],
    "applications": [{
        "applicationID": 1,
        "name": "烟感",
        "deviceNum": 10,
        "activatNum": 7,
        "isAutoJoin": false
    }]
}</pre>
<table>
<tr><th></th><th>Description</th><th>Types</th></tr>
<tr><td>gateways</td><td>Gateway information array</td><td></td></tr>
<tr><td>- gatewayID</td><td>Gateway unique code</td><td>string</td></tr>
<tr><td>- lastSeenAt</td><td>Last uplink time of the gateway</td><td>string</td></tr>
<tr><td>- isOnline</td><td>Online status (true: online, false: offline)</td><td>bool</td></tr>
<tr><td>- longitude</td><td>Longitude</td><td>float64</td></tr>
<tr><td>- latitude</td><td>Latitude</td><td>float64</td></tr>
<tr><td>applications</td><td>Application information array</td><td></td></tr>
<tr><td>- applicationID</td><td>Application ID</td><td>int</td></tr>
<tr><td>- name</td><td>Application name</td><td>string</td></tr>
<tr><td>- deviceNum</td><td>Total number of devices under this application</td><td>int</td></tr>
<tr><td>- activatNum</td><td>Number of activated (joined) devices</td><td>int</td></tr>
<tr><td>- isAutoJoin</td><td>Whether the application allows automatic device addition during network activation</td><td>bool</td></tr>
</table>
</td></tr>
</table>

4.1.4.4 HTTP Push Data Format

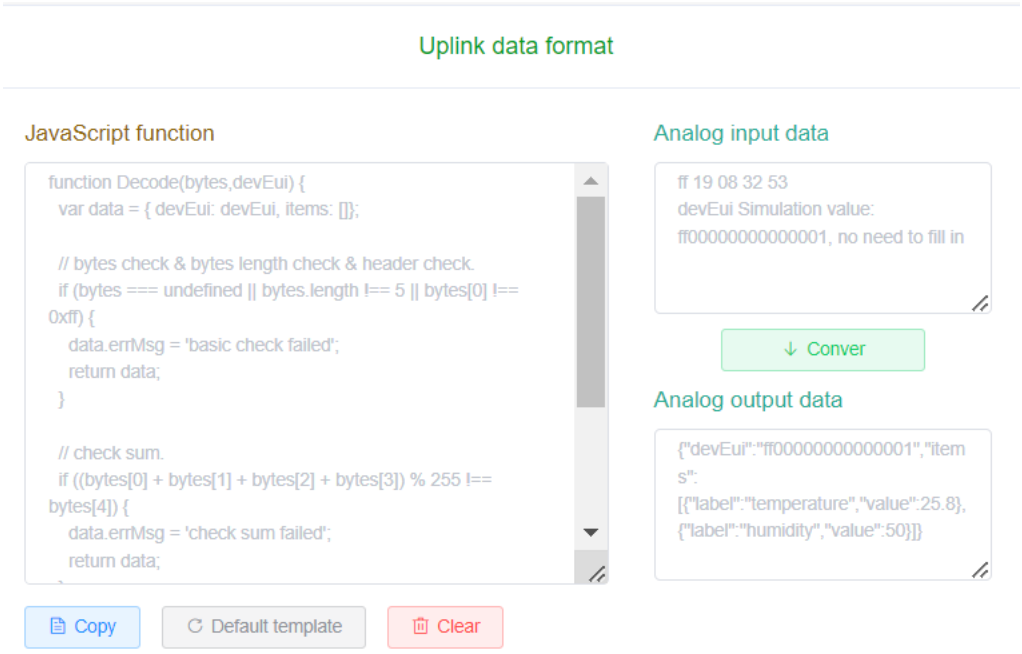
- HTTP is configured for each application at the path: LoRa Network Server → Application → View (corresponding APP) → Interface Management.
- The data content pushed by HTTP is in JSON format, consistent with the JSON content of MQTT and TCP methods (please refer to the previous two chapters).
- When JavaScript function parsing is configured, the JSON data will no longer use the default data format but will use the converted data format.
- HTTP only supports pushing and does not support downstream data.

4.1.4.5 JavaScript Function Transformation Method

- The function of this transformation method:
  - When receiving uplink data, it converts the hexadecimal data (or string) reported by the device into JSON format corresponding field data. This allows integration with specific platforms without the need for customization.
  - When receiving downlink data, it converts the JSON data sent by the customer platform into corresponding hexadecimal data, and then sends this hexadecimal data to the device.
- The transformation function is not configured by default, and the default data format is used when not configured.
- The gateway supports function transformation for both uplink and downlink data, and it is disabled by default.

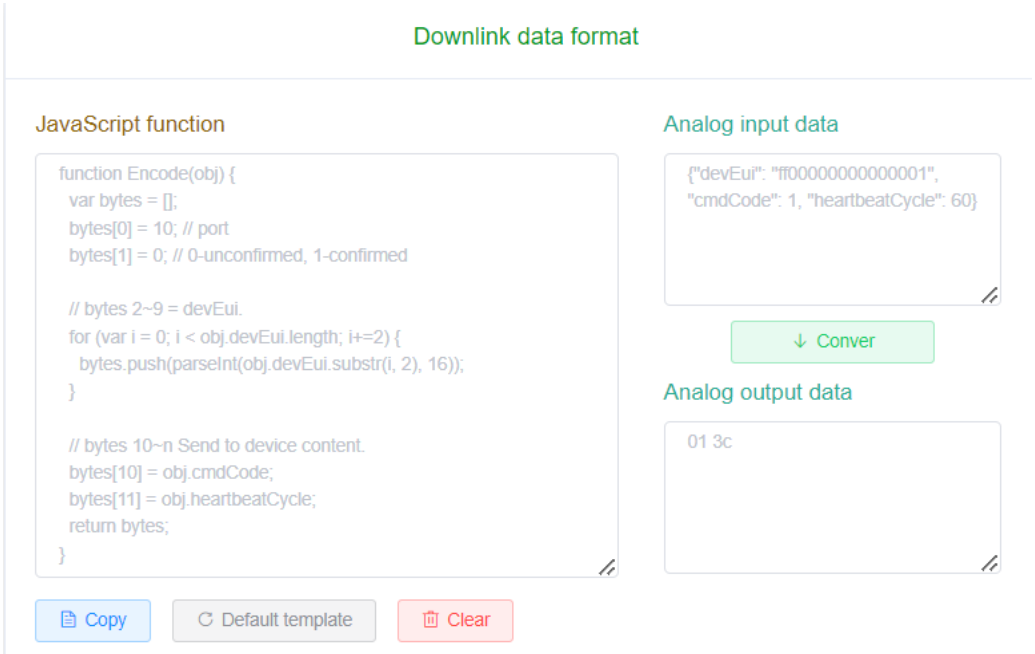
1. Uplink Data Transformation

- When the device reports data as the hexadecimal number ff 19 08 32 53, it can be transformed into the following JSON data: {"devEui":"ff00000000000001","items":[{"label":"temperature","value":25.8},{"label":"humidity","value":50}]} (where ff is the fixed header of the protocol, 19 is the integer part of the temperature value, 08 is the decimal part of the temperature value, 32 is the humidity value, and 53 is the checksum). After successful configuration, the JSON-formatted data received by the client will be as described above.
- **Configuration Path:** LoRa Network Server → Interface → Data Conversion → Upstream Data Format



2. Downlink Data Transformation

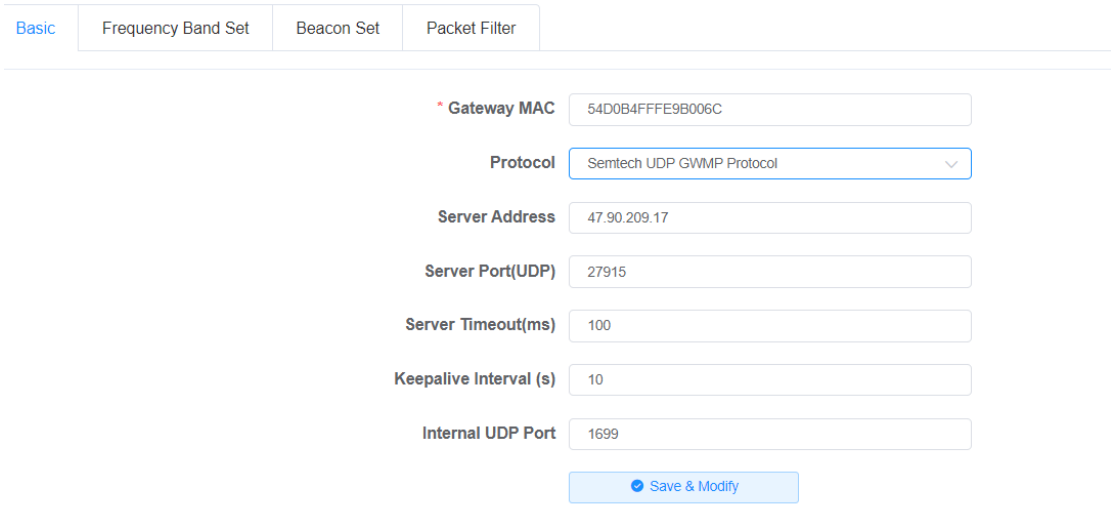
- When the device sends data {"devEui": "ff00000000000001", "cmdCode": 1, "heartbeatCycle": 60} (after function transformation) it will be converted to 01 3c (01-command code for heartbeat cycle configuration, 3c-heartbeat cycle value), which will be sent to the terminal.
- Configuration Path: LoRa Network Server → Interface → Data Transformation → Downlink Data Format



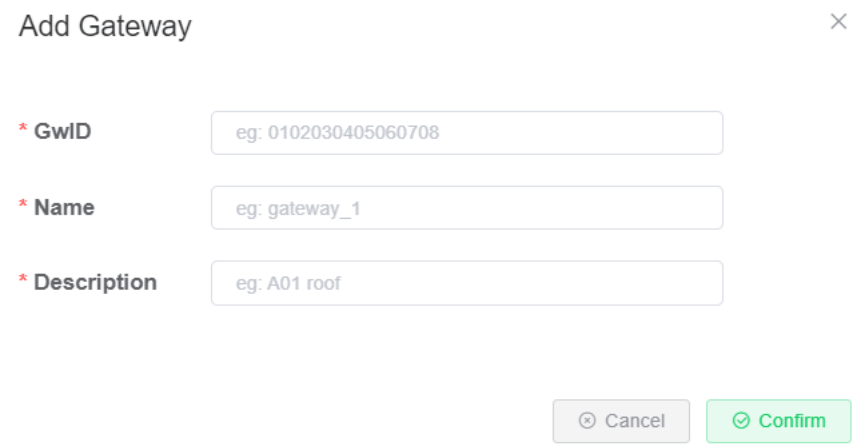
4.1.5 Common Platform Integration

4.1.5.1 Four-Faith Cloud NS

- The standard NS used by Four-Faith Cloud adopts the Semtech UDP GWMP Protocol.
- In this mode, the gateway implements data forwarding functionality.
- Configuration path: LoRa Gateway → Basic Settings. The main configurations include protocol, server address, and server port (UDP). The specific configurations are as follows:



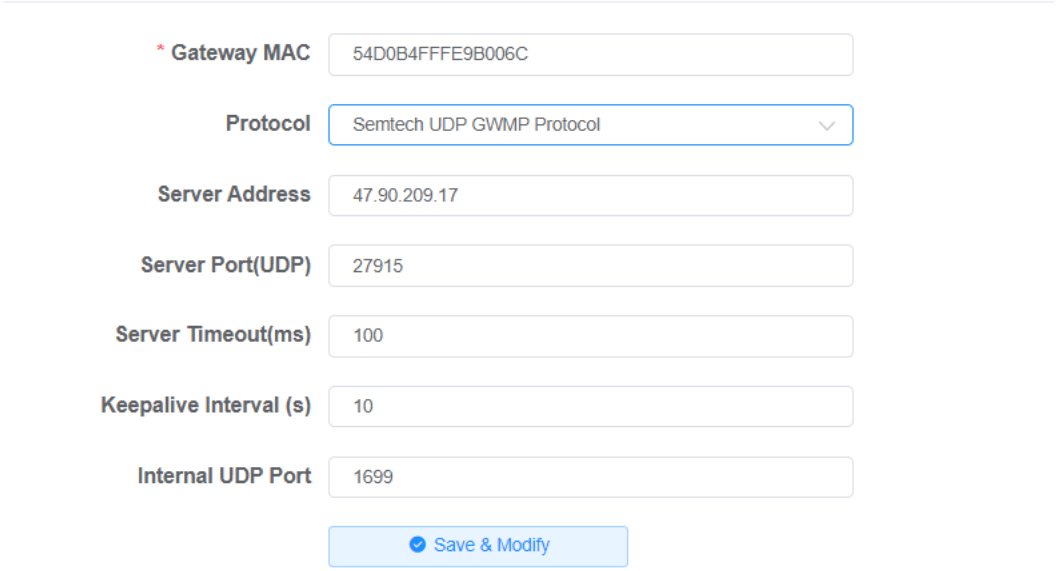
- **Open CSTool:** http://47.90.209.17:51868/#/ns/gateways
- **Create a gateway.**



- Check the gateway status, as shown in the figure below, indicating that the gateway is already online.

4.1.5.2 ChirpStack Platform（GWMP）

- ChirpStack is a general open-source NS that supports multiple access methods, commonly used for GWMP protocol access.
- **Configuration path:** LoRa Gateway → Basic Settings, mainly configuring the protocol, server address, and server port (UDP). The specific configuration is as follows:

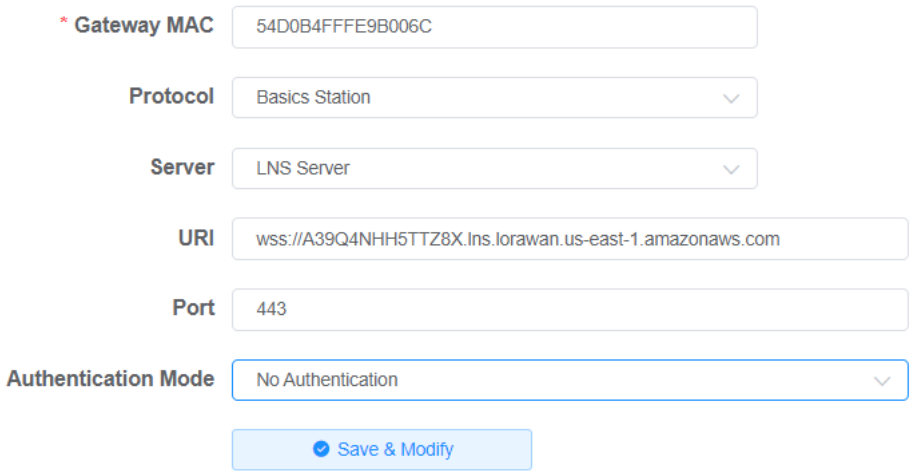


4.1.5.3 ChirpStack Platform（LNS）

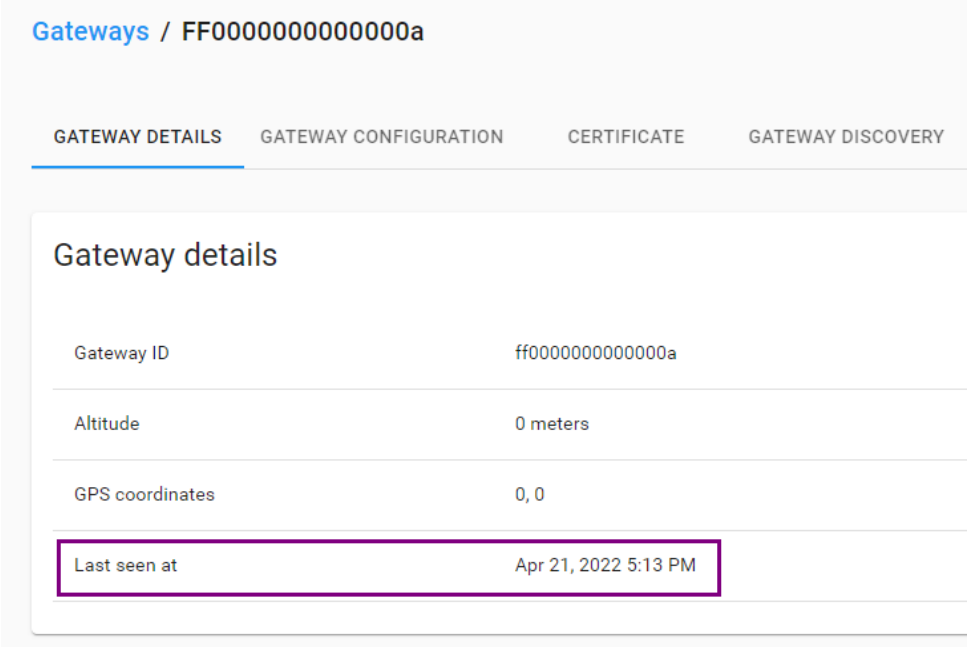
ChirpStack can be configured for Basicstation protocol access, which is generally used as LNS. It supports modes such as No Authentication or TLS Server Authentication. The following provides examples for configuring access in both ways.

1. LNS - No Authentication

- By configuring the protocol, server protocol type, URI, port, and mode selection, you can modify the settings successfully.

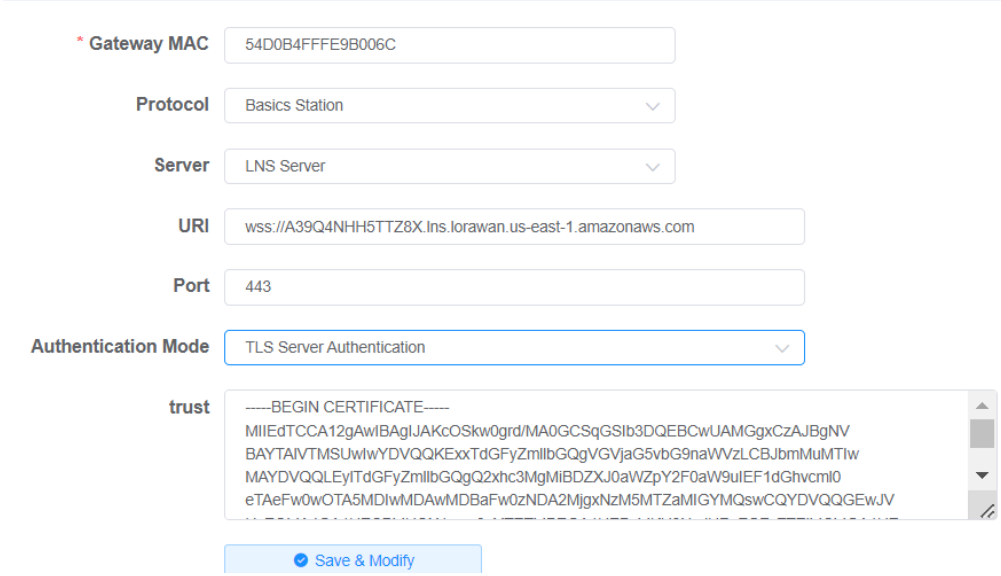


- On the platform, the "Last seen at" for the gateway indicates the connection status of the gateway.

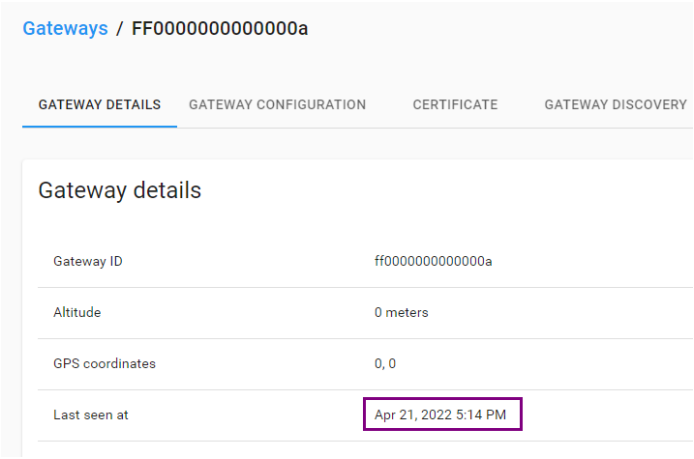


2. LNS - TLS Server Authentication

- When configuring the gateway, the URI requires the corresponding domain name of the server, and the "trust" content is derived from the server's .pem file.



- The "Last seen at" of the gateway on the platform indicates the connection status of the gateway.



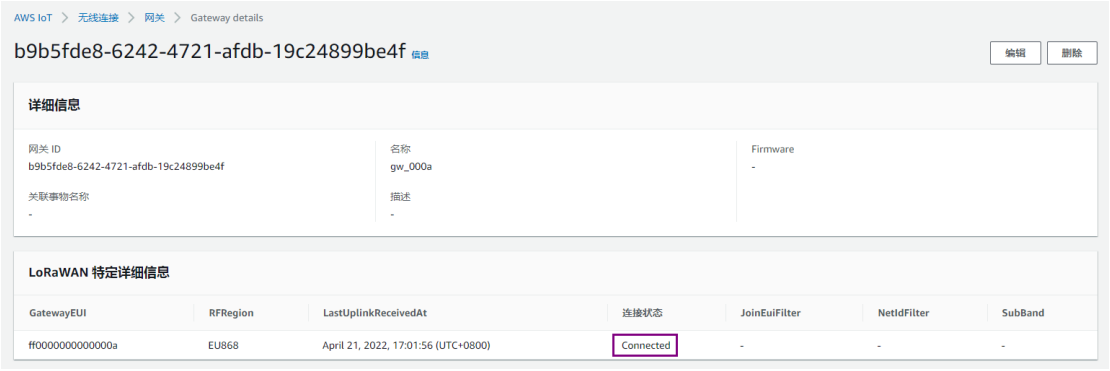
4.1.5.4 AWS Platform（LNS）

- Creating a gateway on the AWS platform

- Download the corresponding key generated by the gateway and configure the corresponding parameters for the gateway. Select the mode as TLS Server and Client Authentication. Boxes of the same color in the following image represent identical content.

- Continue to complete the creation of the gateway.

- After successfully configuring the gateway, you can see the connection status of the gateway on the AWS platform.



4.1.5.5 AWS Platform（CPUS）

- Create a gateway on the AWS platform.

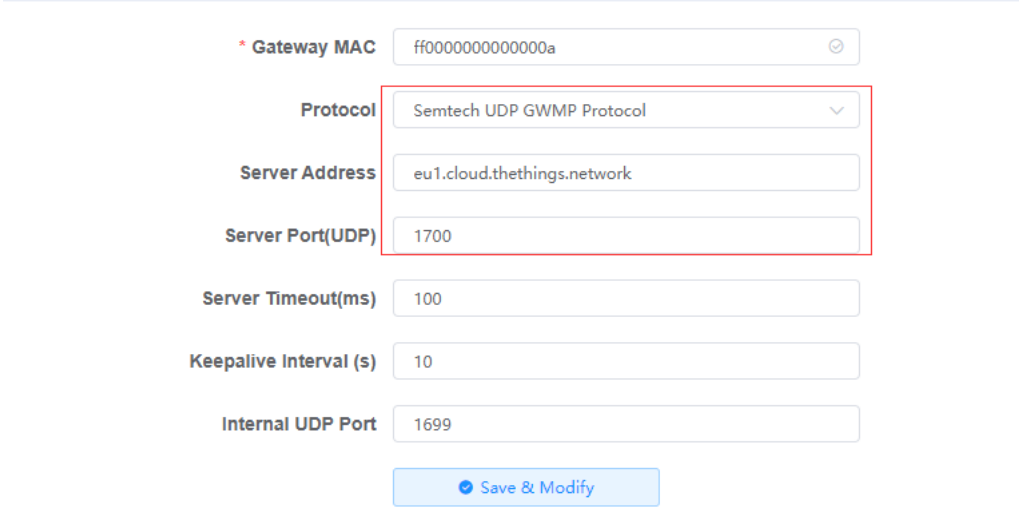
- Download the corresponding key generated by the gateway and configure the corresponding parameters for the gateway. Select the mode as TLS Server and Client Authentication. The boxes with the same color in the following image represent identical content.

- Continue to complete the creation of the gateway.

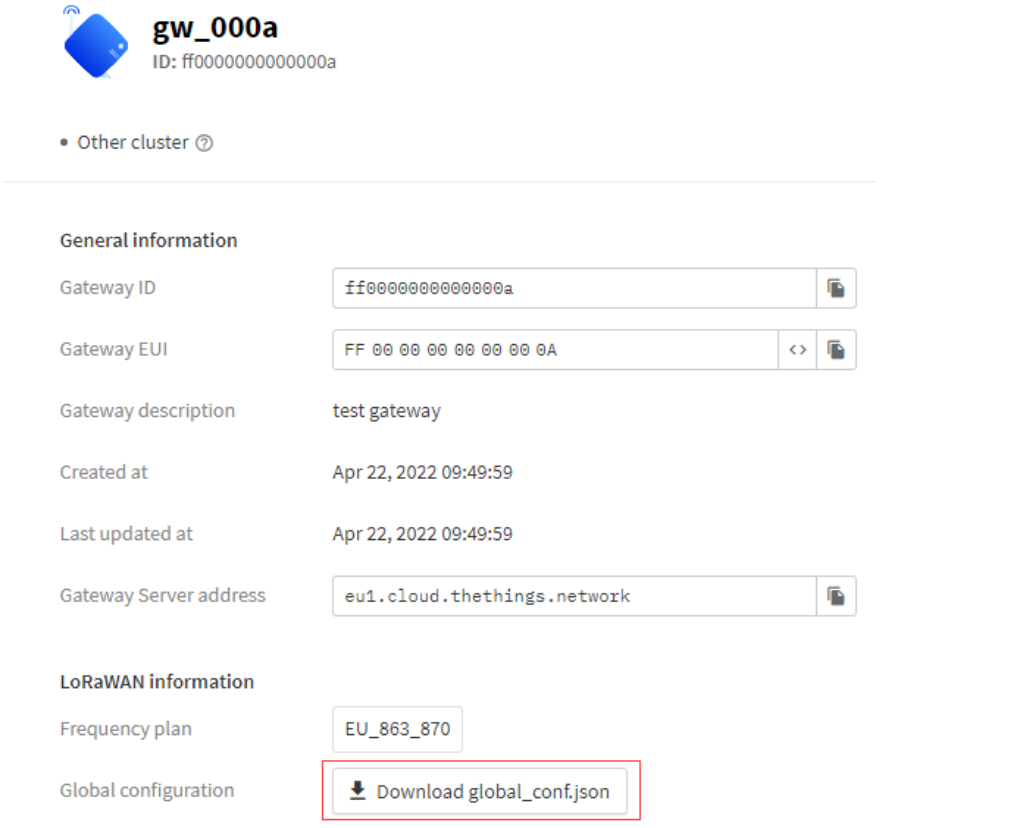
- After successfully configuring the gateway, you can view the gateway's connection status on the AWS platform.

4.1.5.6 TTN Platform（GWMP）

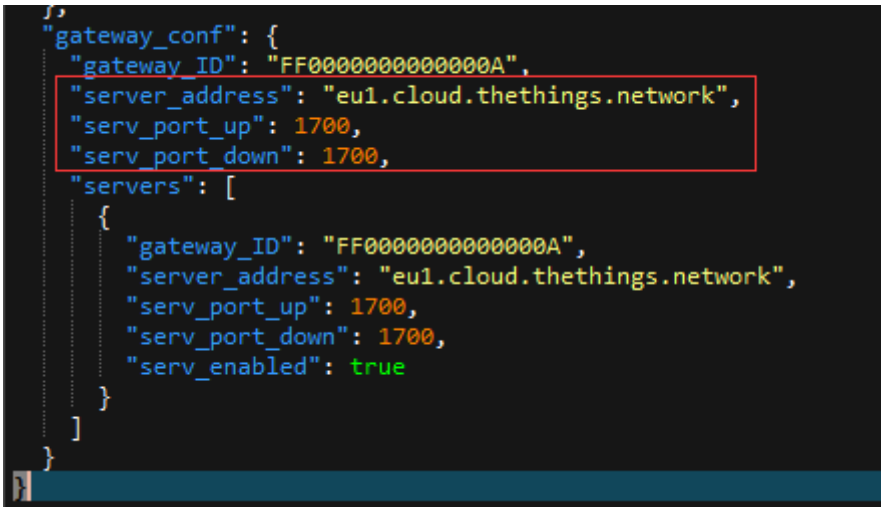
- The TTN platform supports both GWMP and Basicstation modes of access.
- When using the GWMP protocol, the configuration is the same as other platforms, requiring only the server IP and port settings.



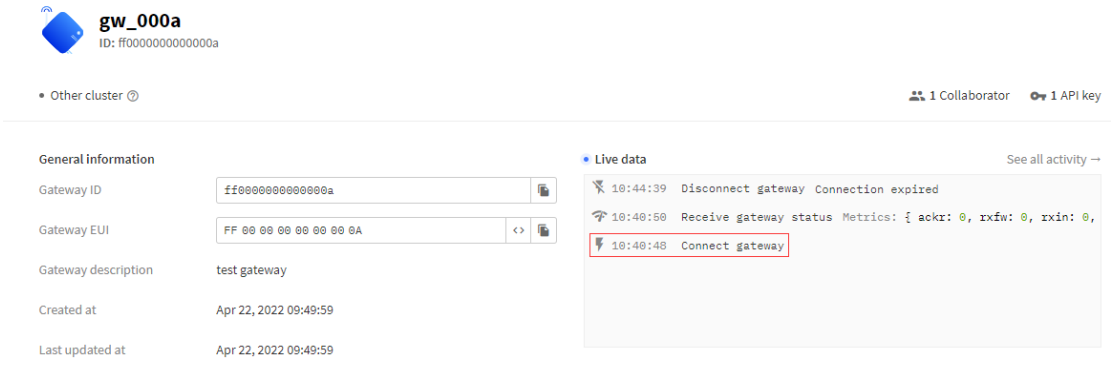
- The server address and port information can be obtained from the global_conf.json file, which can be downloaded from the TTN platform.



- The server address and port can be found at the end of the file.



- After successfully configuring the gateway, you can view the connection information on the TTN platform.



4.1.5.7 TTN Platform（LNS）

- The TTN platform also supports connection using the Basicstation's LNS protocol, with the mode set to TLS Server Authentication and Client Token.
- Add the gateway.

Add gateway

General settings

Owner *

sugk

Gateway ID *

ff0000000000000a

Gateway EUI

FF 00 00 00 00 00 00 0A

Gateway name

gw_000a

Gateway description

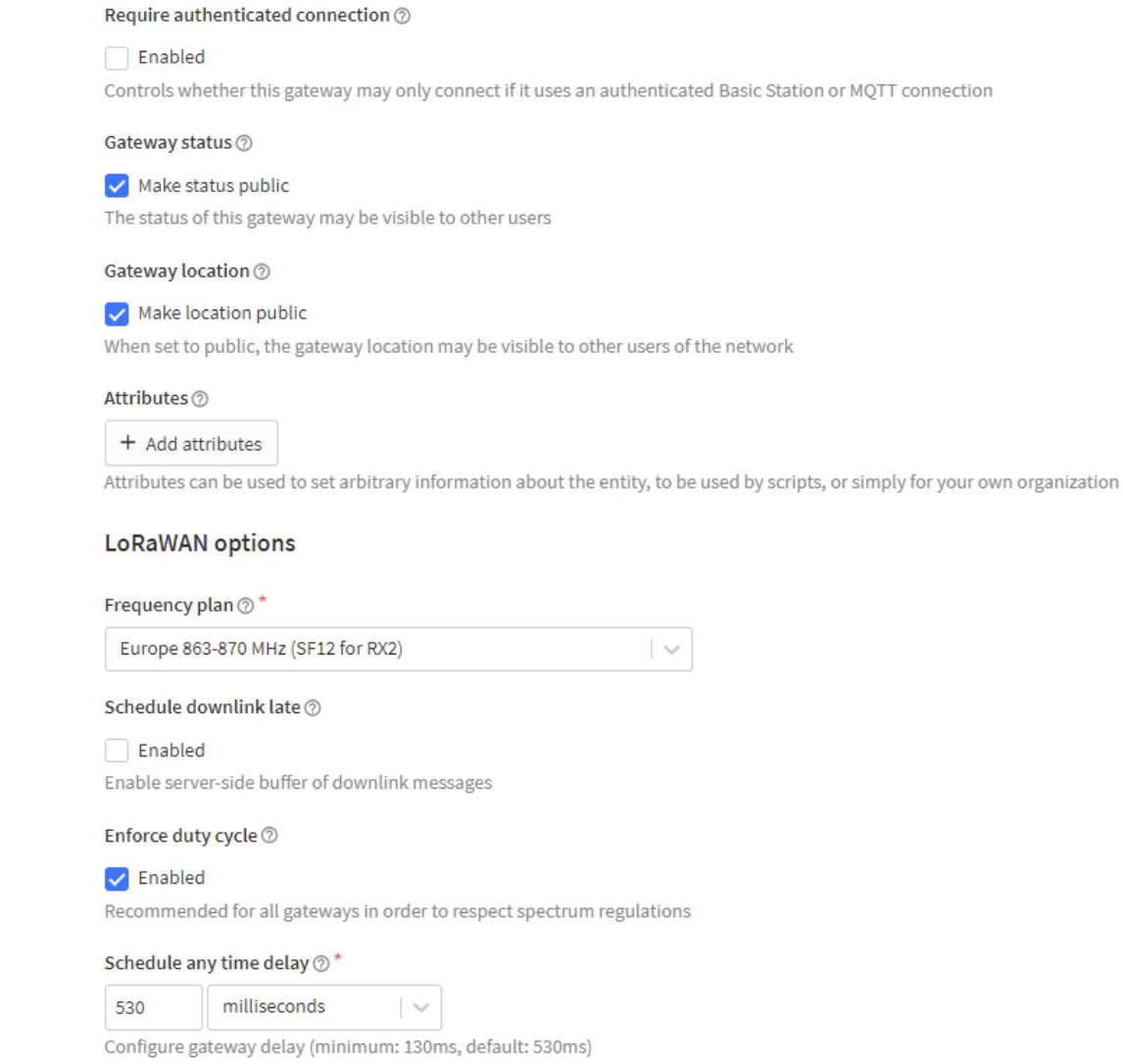
test gateway

Optional gateway description; can also be used to save notes about the gateway

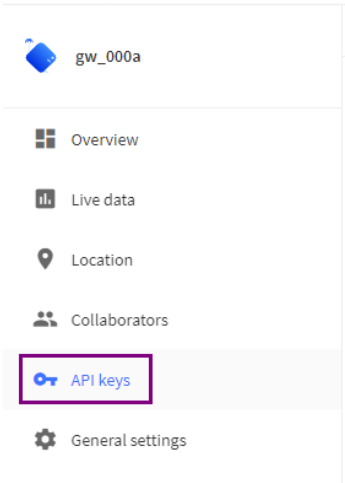
Gateway Server address

eu1.cloud.thethings.network

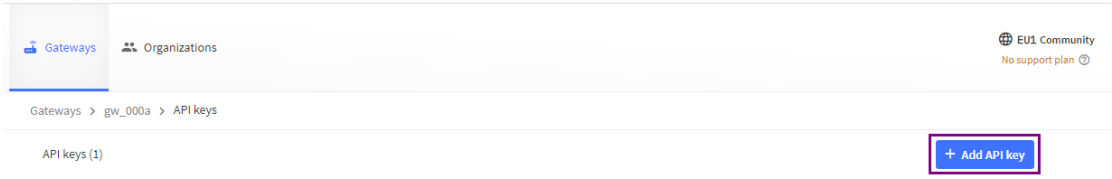
The address of the Gateway Server to connect to



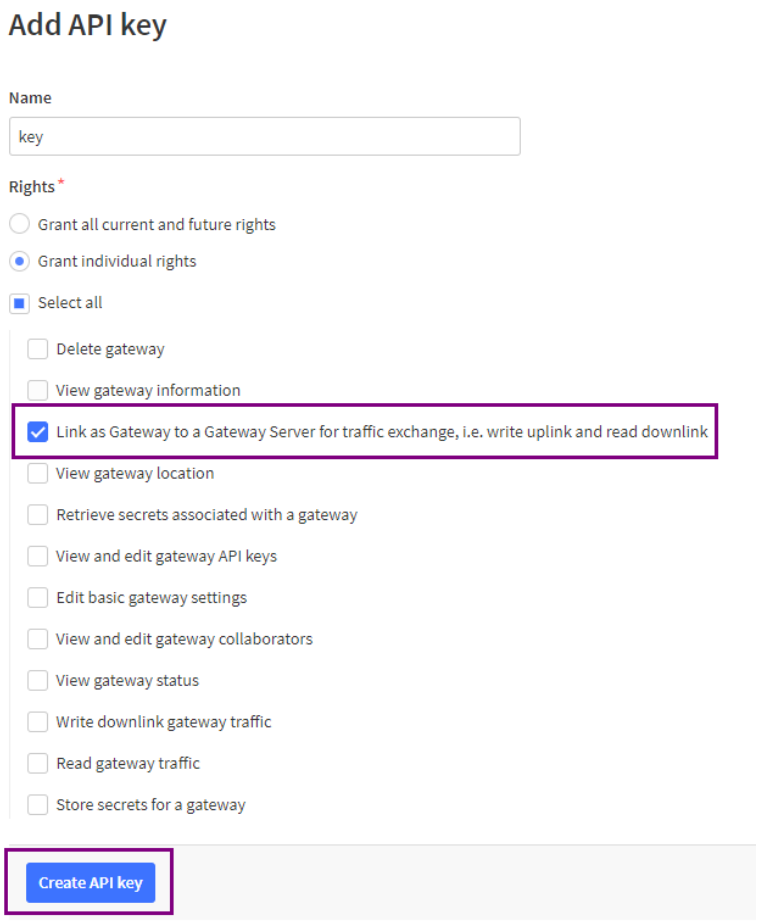
- Obtain the token value.



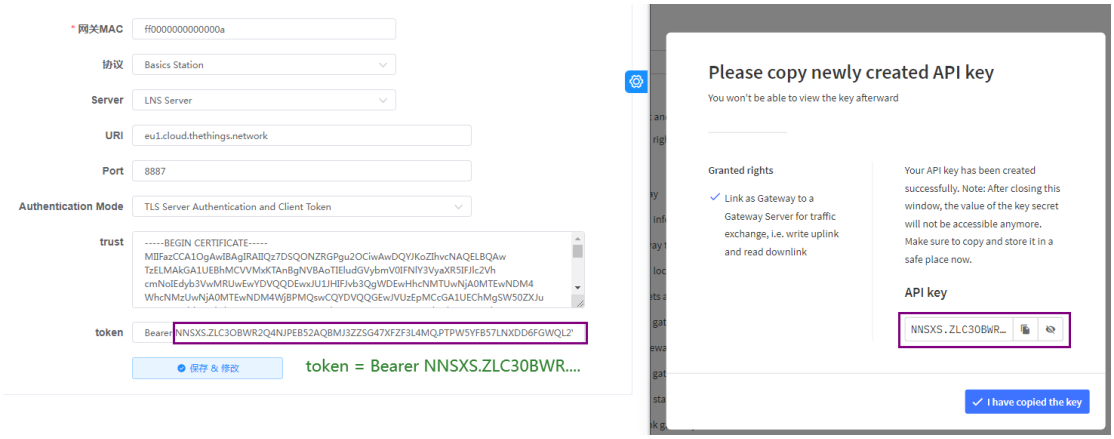
- Add API key



- Select according to the following image and create an API key.

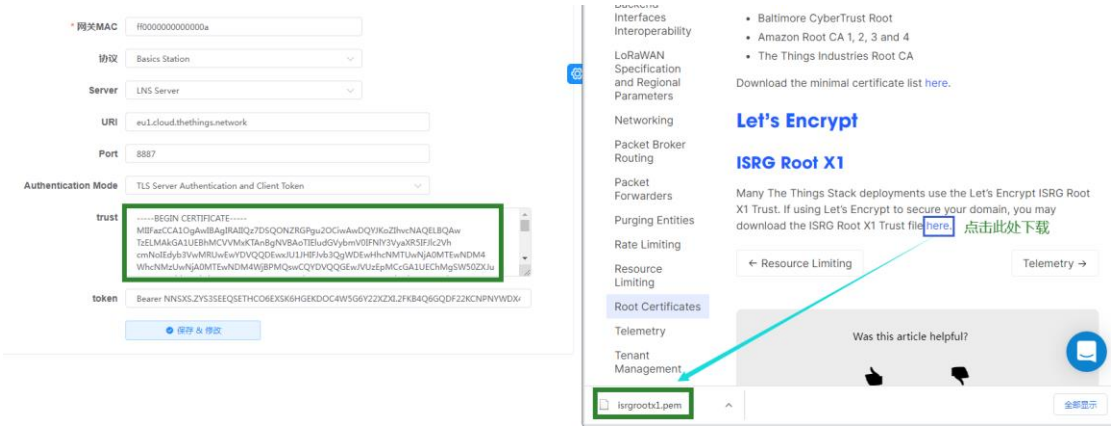


- Token Explanation: token = Bearer + space + API key, for example, Bearer NNSXS...

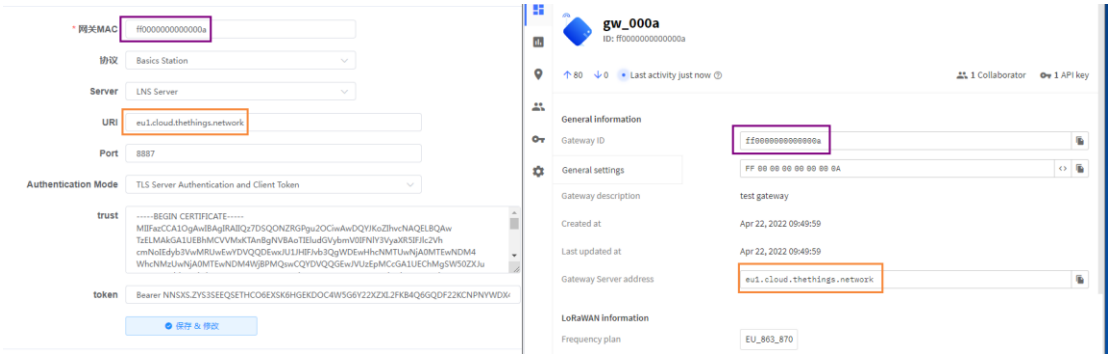


- **Trust Explanation:** This content is from the file isrgrootx1.pem, which can be downloaded from the TTN platform. The file download path is:

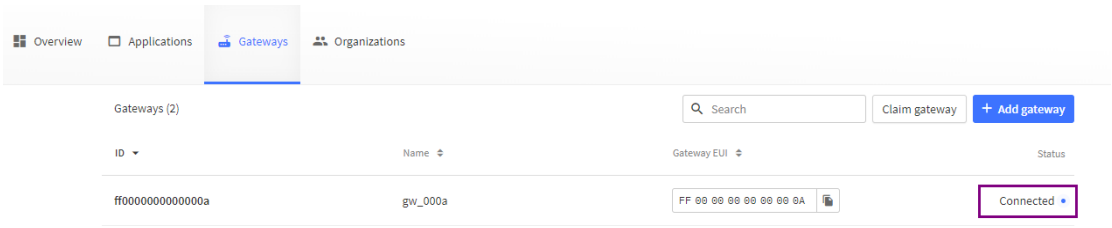
https://www.thethingsindustries.com/docs/reference/root-certificates/#lets-encrypt



- URI Configuration



- **Port Configuration:** Fixed to 8887
- After the gateway configuration is successful, you can check the gateway's connection status to determine whether the connection is successful.



4.1.6 Common Issues:

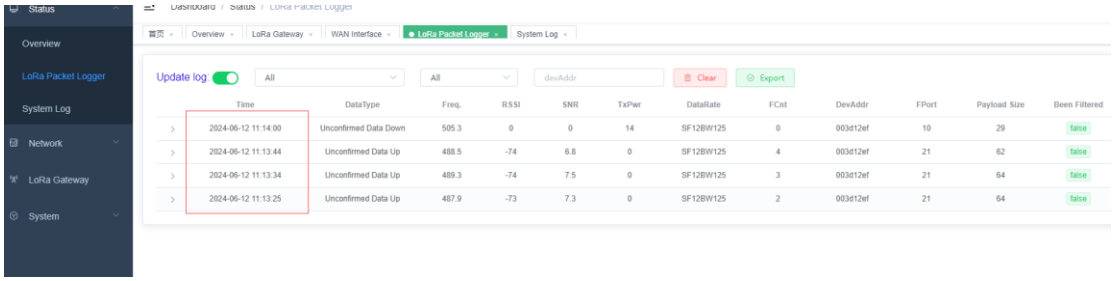
4.1.6.1 Gateway Status

1. Internal Program Status Troubleshooting

When using Semtech UDP GWMP Protocol or Build-in LoRa Server, you can check the logs for the presence of PullData and PullACK. If there is no response after waiting for 30 seconds, it indicates an issue with the gateway.

2.Can the gateway receive RF data

- Open the LoRa Packet Logger and transmit data or initiate device activation with a device configured on the same frequency as the gateway. If the LoRa Packet Logger can capture logs, it indicates that the RF module is functioning properly.



4.1.6.2 Communication Device

1. Abnormal Reception of Uplink Data

- **Antenna Verification:** Ensure that the antennas on both the gateway and the terminal are set to the correct frequency band. Are the antennas installed correctly? Is the feeder line of the gateway installed correctly?
- **Frequency Point Confirmation:** Compare the frequency points configured on the device with those configured on the gateway to ensure consistency.
- **Gateway LoRa Packet Logger:** Open the LoRa packet logger on the gateway, have the terminal send data, or initiate network joining to see if the gateway can listen to the terminal's data.

2. Not receiving downlink data

- Confirm the correctness of antennas on both the gateway and the device. Check if they are operating on the correct frequency band and if the antenna connections are secure.
- Examine the packet logger to determine if there are logs indicating downlink data.
  - For Class A devices, downlink data transmission occurs after an uplink

transmission from the device.

- For Class C devices, downlink data is sent immediately.

- Verify if the frequency and data rate of the downlink data match the frequency and data rate that the device is listening on. (For Four-Faith modules, you can set DBL=2 to observe this.)
- Ensure that the device type is consistent between the device and the server:
  - For Class A devices, if the server is Class C, the data is sent immediately, but the device may not be in the receive window, resulting in data loss.
  - For Class C devices, if the server is Class A, the sent data won't be understood by the device until it sends an uplink again. Without an uplink, the device won't receive the data.

  After adjusting the device or server device type, the device needs to be reconnected to synchronize.

4.1.6.3 Device Joining Exception

- First, check whether the gateway can receive the join request packet initiated by the device. If it cannot be received, please refer to the "Communication with Devices Troubleshooting."
- Embedded NS
  - Check whether the device has been entered into the embedded NS or if automatic device addition is enabled.
  - For automatic device addition, verify if the AppEUI and AppKey are consistent.
  - For devices that have been entered, confirm if the AppKey is consistent.
- External NS Server
  - Check whether the device has been added to the platform.
  - Verify if the AppEUI and AppKey of the device are consistent. AppKey is a mandatory verification field, and AppEUI verification depends on the platform requirements.
- If the gateway can see the Join Accept downlink packet but the terminal does not receive it, check if the device's frequency band matches the NS frequency band. Inconsistency can result in the listening frequency or rate not matching the downlink, causing data to be unable to be received properly.

Note: The failure of device joining is not related to inconsistent device types. For example, if the device is class A and the server is class C, the joining process can still be successful.

4.1.6.4 Customer Platform Integration

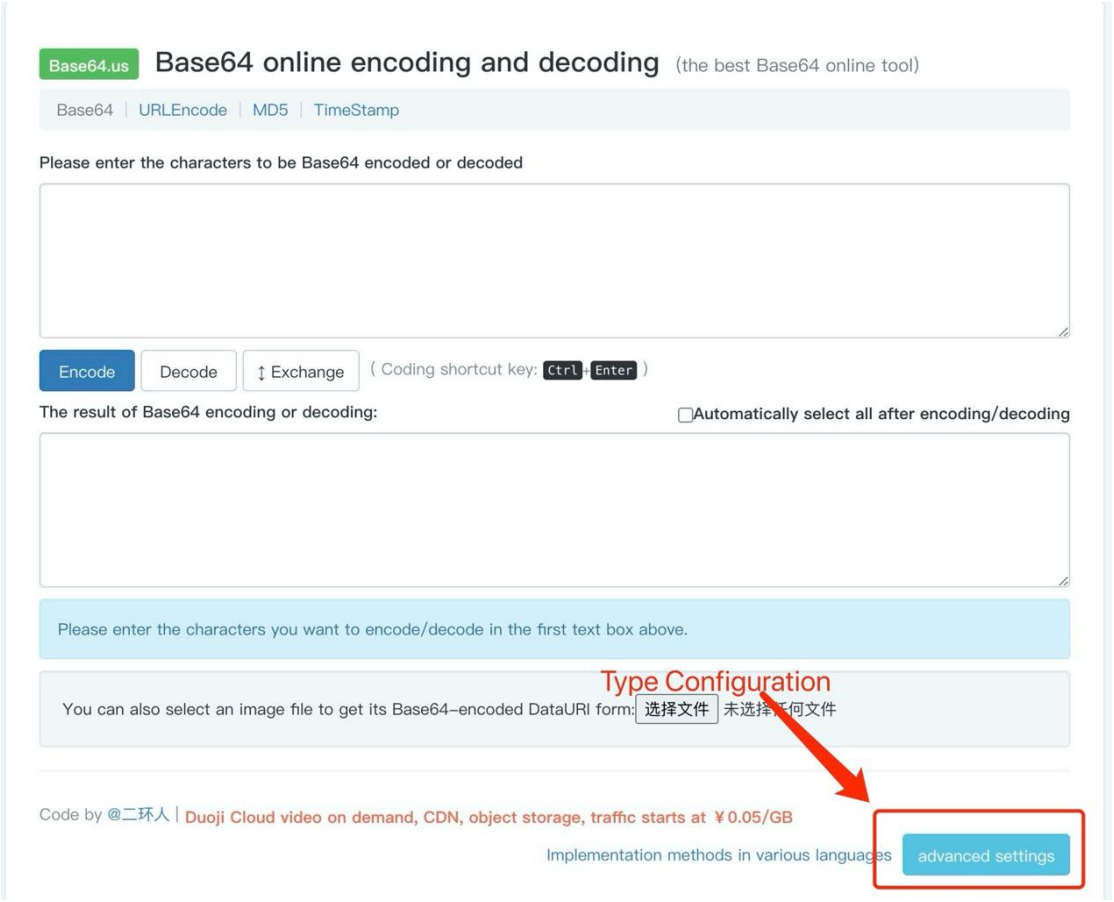
- You can use the gateway's network diagnostic tool to ping the server's IP and check if the gateway network is functioning properly (Path: Network → Network Diagnostics → Ping).

- For MQTT type (Path: LoRa Network Server → Interface → Protocol Configuration):
  - Check if the MQTT switch is turned on.
  - Confirm the server's IP and port.
  - Verify the MQTT connection status.

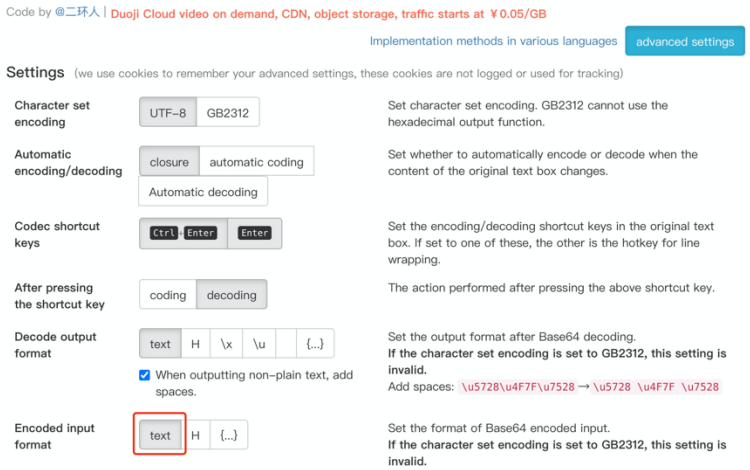
- For TCP type:
  - Check if the corresponding TCP connection switch is turned on.
  - Verify the server's address and port.
  - Check the status of the corresponding connection.

4.1.6.5 Base64 encoding and decoding

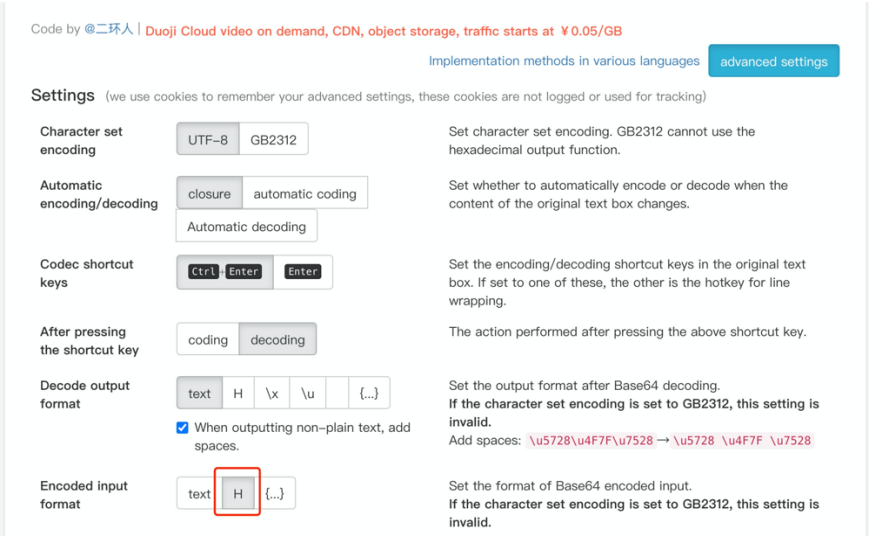
- Online tool address: https://base64.us/



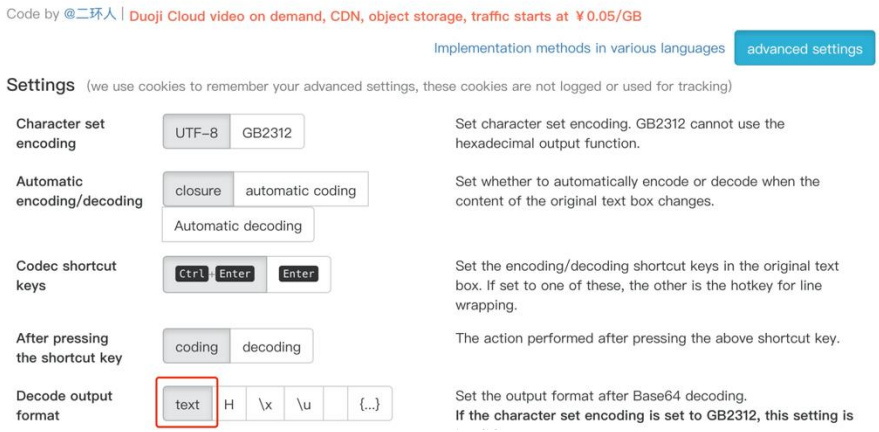
- Mainly involves different data types with different encoding and decoding results, such as text (strings) or Hex (hexadecimal). The encoding and decoding configurations are in the advanced settings shown above.
- Encoding: (When sending downstream data, the data needs to be encoded in base64 format)
  - Text type (1234 → MTIzNA==)



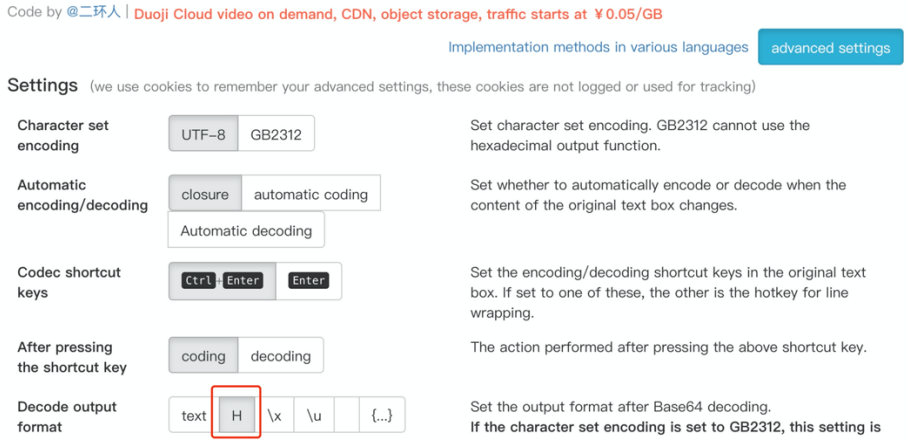
  - Hex type (0x1234 → EjQ=)



- Decoding: (The 'data' field content of the upstream push data needs to be decoded from base64 to actual content)
  - Text type (MTIzNA== → 1234)



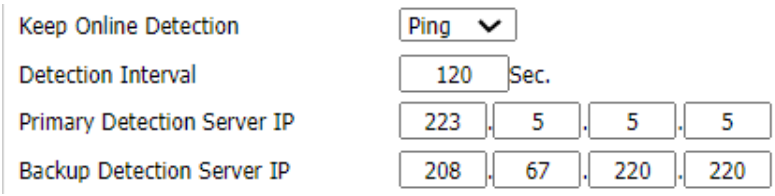
  - Hex tpye (EjQ= → 0x1234)



4.1.7 DHCP-4G

The WAN port's IP address is obtained through DHCP-4G.

Online persistence



"Online persistence" function is used to detect whether the Internet link is in a valid state. If this option is set, F8L10GW-02 will automatically check the Internet link. Once a link disconnection or invalid status is detected, the system will automatically reconnect and establish a valid link. If the network environment is poor or in a private network scenario, it is recommended to use Router mode.

Online persistence modes:

None: Do not use the online persistence function.

Ping: Send ping packets to check the link. If set to this mode, you must also correctly configure the "Online Persistence Check Interval," "Main Server IP for Online Persistence Check," and "Backup Server IP for Online Persistence Check" settings.

Route: Use the route method to check the link. If set to this mode, you must also correctly configure the "Online Persistence Check Interval," "Main Server IP for Online Persistence Check," and "Backup Server IP for Online Persistence Check" settings.

PPP: Use the PPP method to check the link. If set to this mode, you must also correctly configure the "Online Persistence Check Interval" setting.

Online persistence check interval:
The time interval between two online persistence checks, in seconds.

Main server IP for online persistence check:
The IP address of the main server that responds to F8L10GW-02 online check packets. This configuration item is only valid when the "Online Persistence Mode" is set to "Ping" or "Route."

Backup server IP for online persistence check:
The IP address of the backup server that responds to F8L10GW-02 online check packets. This configuration item is only valid when the "Online Persistence Mode" is set to "Ping" or

"Route."

Force Reconnect: This function allows you to specify a time for F8L10GW-02 to reconnect to the Internet.
Time: Enter the correct reconnect time.

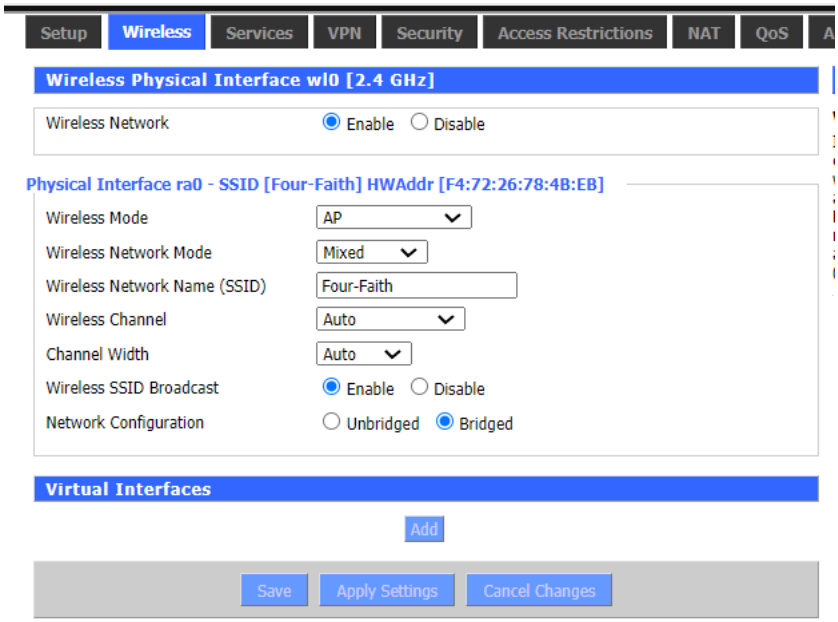
STP

STP（Spanning Tree Protocol）is an acronym for the Spanning Tree Protocol. This protocol can be applied to loop networks, using a specific algorithm to achieve path redundancy while pruning the loop network into a tree-like structure, thereby avoiding the proliferation of packets and infinite loops in the loop network.

4.1.8 Wireless WiFi

The main functions of the wireless Wi-Fi on the F8L10GW-02 base station/gateway are to provide parameter configuration and online upgrade capabilities.

4.1.8.1 Basic Configuration



Enable: Activate the Wi-Fi.

Disable: Deactivate the Wi-Fi.

Wireless Mode: Choose from AP, Client, Ad-hoc, Repeater, and Repeater Bridge.

Wireless Network Mode:

Mixed: Supports wireless devices following 802.11b, 802.11g, and 802.11n standards simultaneously.

BG-Mixed: Supports wireless devices following 802.11b and 802.11g standards simultaneously.

B-Only: Supports only wireless devices following the 802.11b standard.

G-Only: Supports only wireless devices following the 802.11g standard.

NG-Mixed: Supports wireless devices following 802.11g and 802.11n standards simultaneously.

N-Only: Supports only wireless devices following the 802.11n standard.

802.11n Transmission Mode: When the wireless network mode is set to "N-Only," choose its transmission mode:

Greenfield: Use this mode when you are certain that no other 802.11a/b/g devices in the surrounding environment use the same channel. This mode can improve throughput. If there are other 802.11a/b/g devices using the same channel in the environment, your transmitted information may experience errors and retransmissions.

Mixed: This mode is the opposite of the Greenfield mode but may reduce throughput.

Wireless Network Name (SSID): The shared network name for all devices in the wireless network. The SSID is consistent across all devices. The SSID consists of alphanumeric characters, is case-sensitive, and should not exceed 32 characters.

Wireless Channel: There are channels 1-13 to choose from. In environments with

multiple wireless devices, try to avoid using the same channel as other devices.

Channel Width: Options for 20MHz and 40MHz are available.

Wide Channel: When the channel is set to 40MHz, you can choose either upper or lower.

Wireless SSID Broadcast:

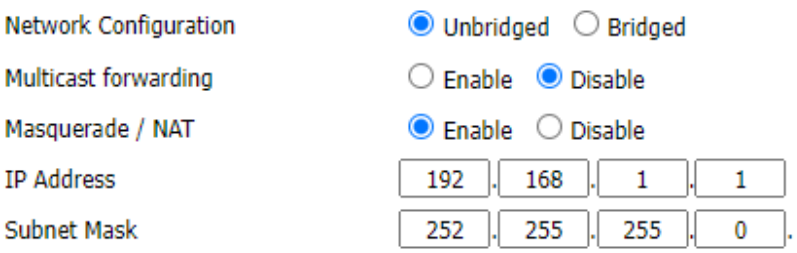
Enable: Broadcast SSID.

Disable: Hide SSID.

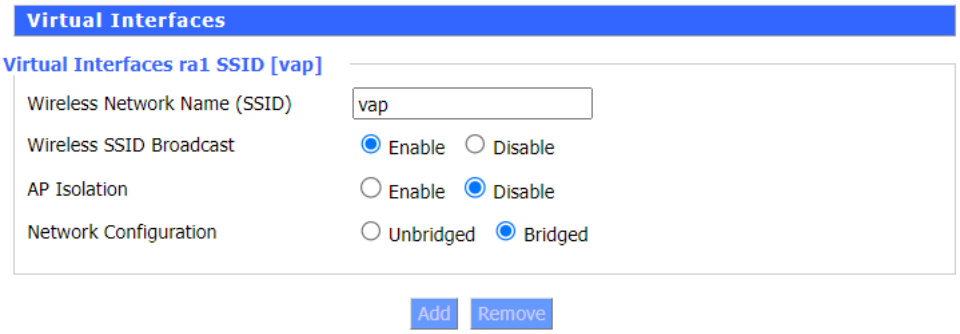
Network Configuration:

Bridged: Bridged to F8L10GW-02. In most cases, please choose Bridged.

Unbridged: Not bridged to F8L10GW-02. IP address needs to be configured manually.



Virtual Interface: Click "Add" to create a virtual interface. After successful addition, click "Remove" to delete the virtual interface.。



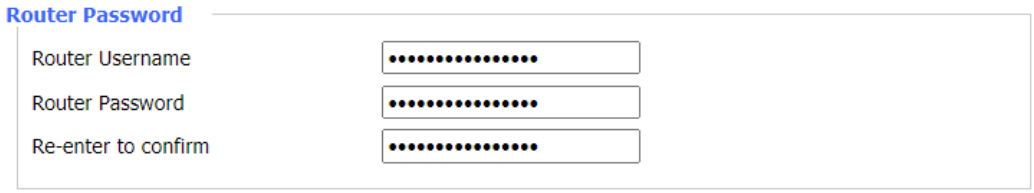
AP Isolation: Isolates all wireless client devices, allowing them to access only the fixed network connected to the AP.

Note: Save Settings: Save changes. After modifying "Wireless Mode," "Wireless Network Mode," "Wireless Width," or "Wide Frequency" options, click this button first before configuring other options.

4.1.9 Management

4.1.9.1 Management

This page allows network administrators to manage specific functionalities of F8L10GW-02, ensuring access and security.



The new password length must not exceed 32 characters and should not contain any spaces. The confirmation password should match the new password you set; otherwise, the configuration will not be successful.

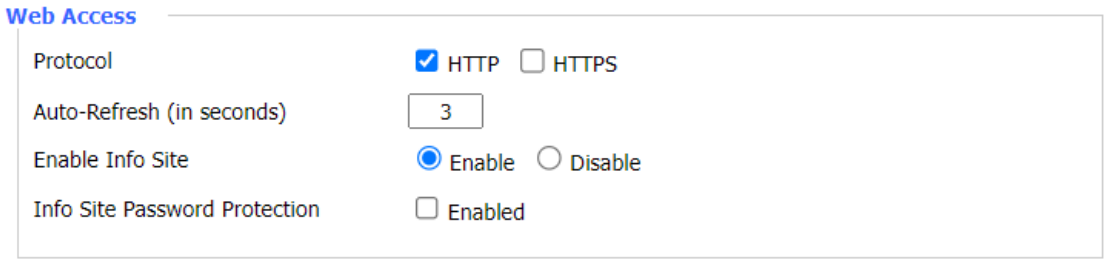
Warning

The default username is: admin.

We strongly recommend changing the factory default password, admin. This ensures that all users attempting to access and modify F8L10GW-02 must provide the correct password for access and usage.

Web Access

This feature allows you to manage F8L10GW-02 using either the HTTP or HTTPS protocol. If you choose to disable this feature, a manual restart will be required. You can also enable or disable the F8L10GW-02 information webpage, allowing password protection for this page (requiring the correct username and password input).

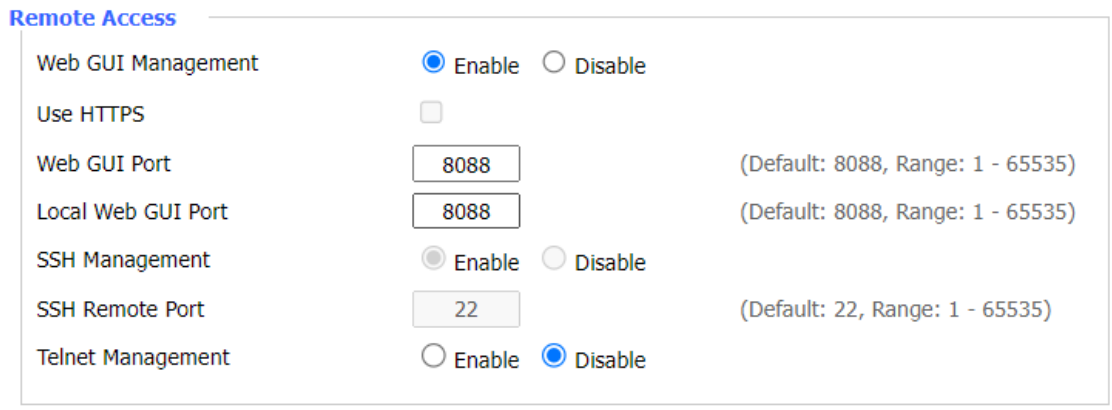


Protocol: The web page supports protocols including HTTP and HTTPS.

Auto Refresh (seconds): Adjust the time interval for the web interface to automatically refresh. 0 indicates that this feature is disabled.

Display system information webpage before login: Enable or disable displaying the system information webpage before login.

System information webpage password protection: Enable or disable the password protection feature for the system information webpage.



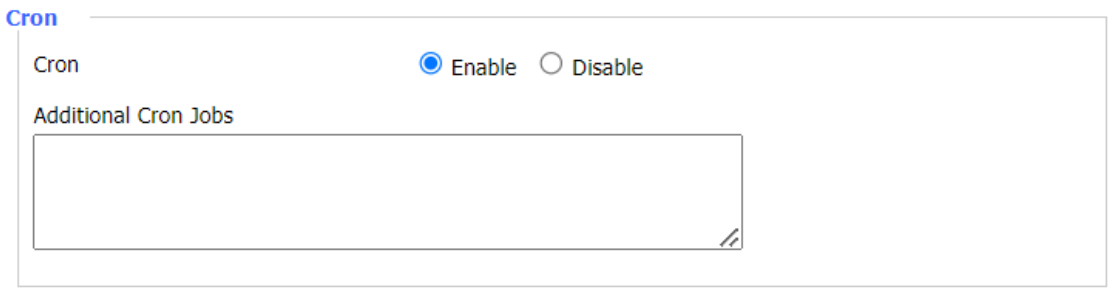
Web Interface Management: This feature allows you to manage F8L10GW-02 remotely over the Internet. To disable this feature, keep the default settings, which is disabled. To enable this feature, select enable and use the specified port on your computer (default is 8080) to remotely manage F8L10GW-02. If you haven't set a password yet, you must also set the default password for your F8L10GW-02. To remotely manage F8L10GW-02, go to http://xxx.xxx.xxx.xxx:8080 (replace 'x' with the Internet IP address of F8L10GW-02, and 8080 represents the specified port) in your web browser's address bar. You will be prompted to enter the password for F8L10GW-02. If you use HTTPS, you need to specify the URL as https://xxx.xxx.xxx.xxx:8080 (not all firmware supports SSL rebuilding).

SSH Management: You can enable SSH to remotely access F8L10GW-02 securely. Please note that to learn about the settings of the SSH daemon, you can access more information on the Services page.

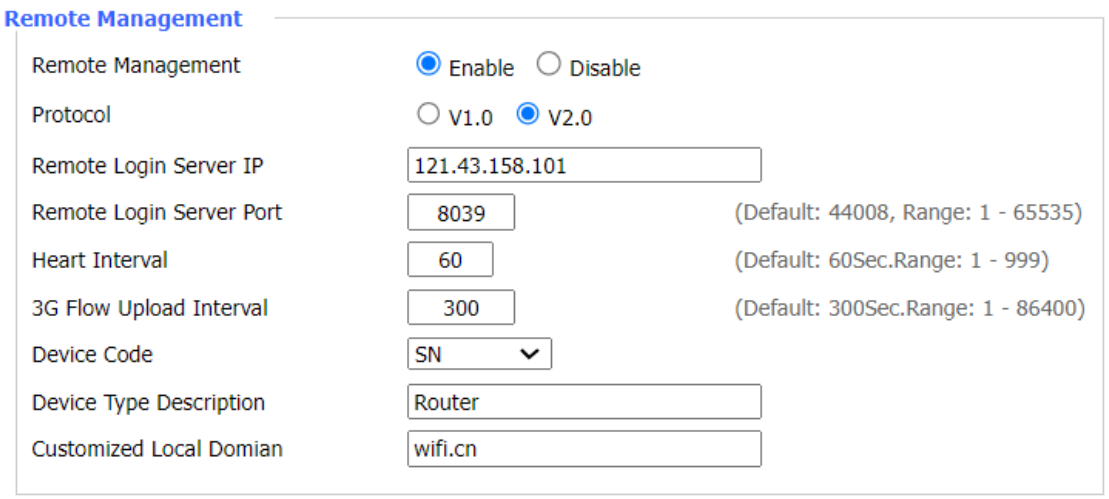
Warning:

If the remote access feature of F8L10GW-02 is enabled, anyone who knows the Internet IP address and password of F8L10GW-02 will be able to change the settings of F8L10GW-02.

Telnet Management: Enable or disable remote Telnet functionality.

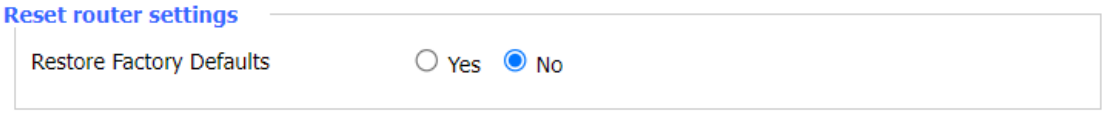


Cron: The cron subsystem is for scheduling Linux commands that you plan to execute. In practice, you may need to use the command line or startup scripts.



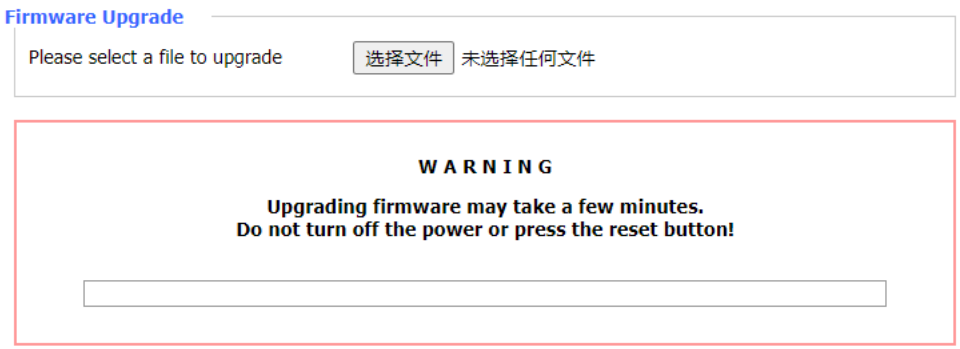
Device Management: Monitor and manage this F8L10GW-02 unit through a custom-developed remote management server, including parameter configuration, Wi-Fi advertising updates, and more.

4.1.9.2 Factory Default



Restore Factory Defaults: Clicking the "**Yes**" button and saving the settings will clear all configurations and restore them to the factory values. When restoring to default settings, all the changes you made will be lost. The default configuration for this function is set to "**No**". For more information, please click "**More**".

4.1.9.3 Firmware Upgrade



Firmware Upgrade: This feature allows you to load new firmware onto the F8L10GW-02. New firmware versions will be released on www.four-faith.com and can be downloaded for free. If the F8L10GW-02 is functioning properly, there is no need to download the updated firmware version unless it includes new features you want to use.

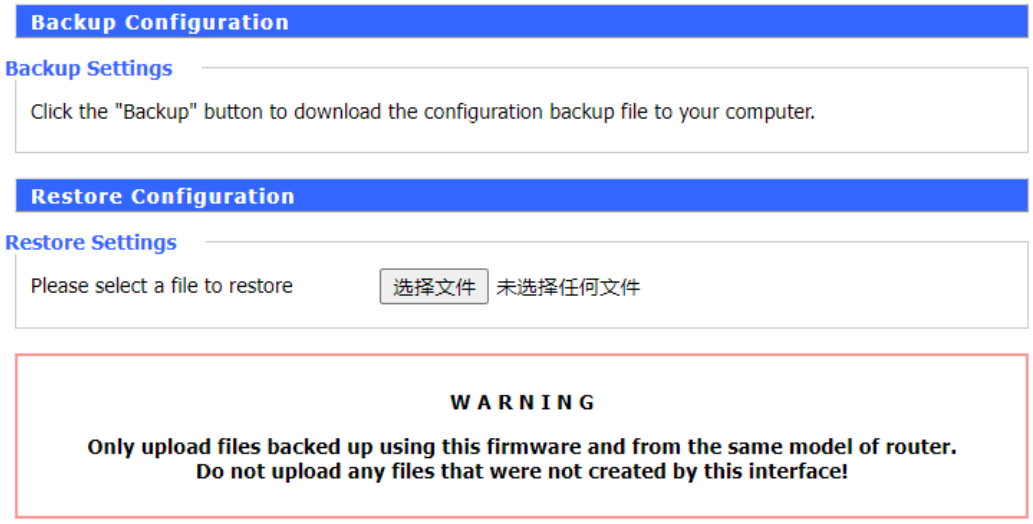
Note: When upgrading the firmware of the F8L10GW-02, configuration settings may be lost, so be sure to backup the settings of the F8L10GW-02 before upgrading the firmware.

After the upgrade, reset to: If you want to reset the firmware version of the F8L10GW-02 to default settings after the upgrade, click the "Default Settings" option.

Click "Browse," select the firmware file to be upgraded, and then click the "Upgrade" button to start the firmware upgrade. The firmware upgrade may take a few minutes, so do not power off or press the reset button during the process.

4.1.9.4 Backup

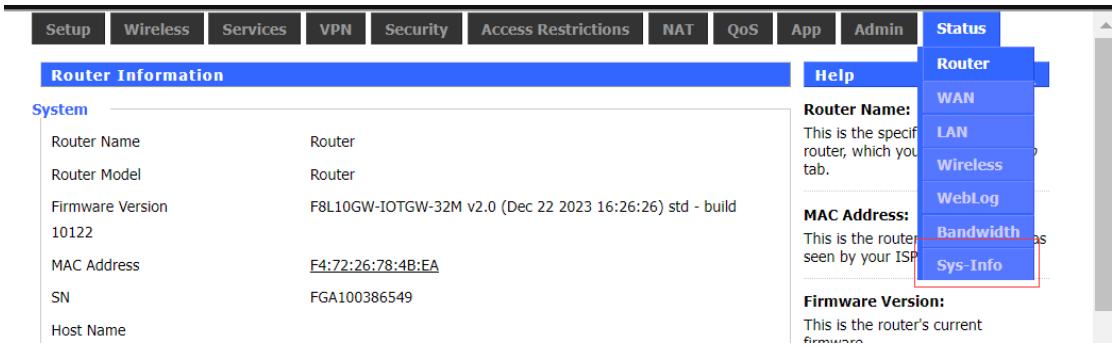
This page is used to backup or restore the configuration file of the F8L10GW-02.

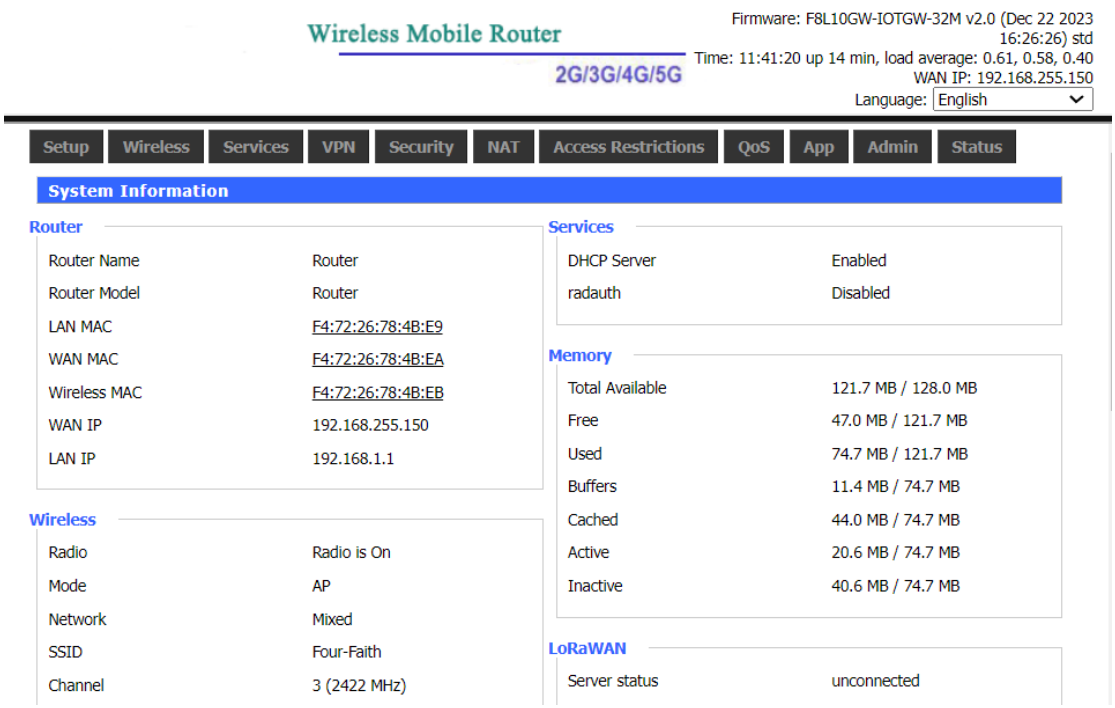


If you want to backup the configuration file of the F8L10GW-02, click the "**Backup**" button. Follow the on-screen instructions.

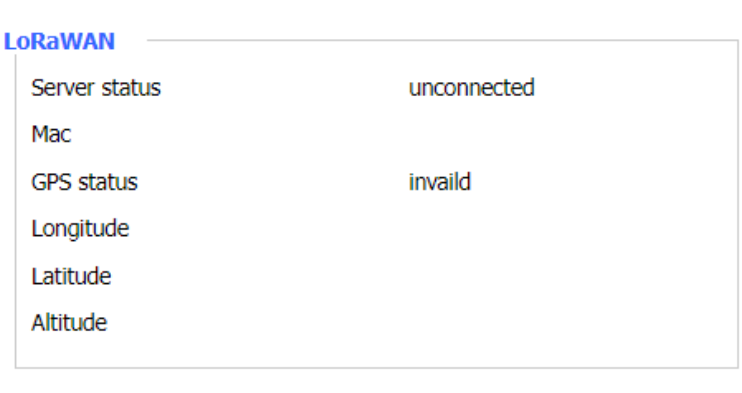
If you want to restore the configuration file of the F8L10GW-02, click the "**Browse**" button, locate the backup file, and follow the on-screen instructions. After selecting the backup file, click the "**Restore**" button.

4.1.10 Status





4.1.10.1 F8L10GW-02

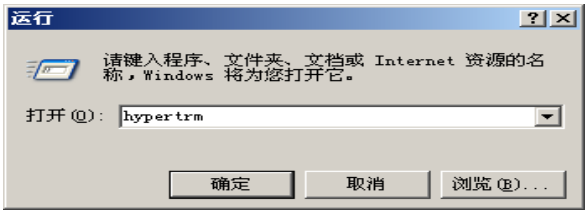


Server Status: The connection status with the specified LoRaWAN server.
Mac: The MAC address of the F8L10GW-02, serving as an identification code for different F8L10GW-02 devices recognized by the LoRaWAN server.
GPS Status: Indicates whether there is GPS signal.
Longitude, Latitude, Altitude: Information obtained from GPS.

Appendix

Capturing debug information through Console using HyperTerminal: Step-by-step guide and configuration methods (WINDOWS XP)

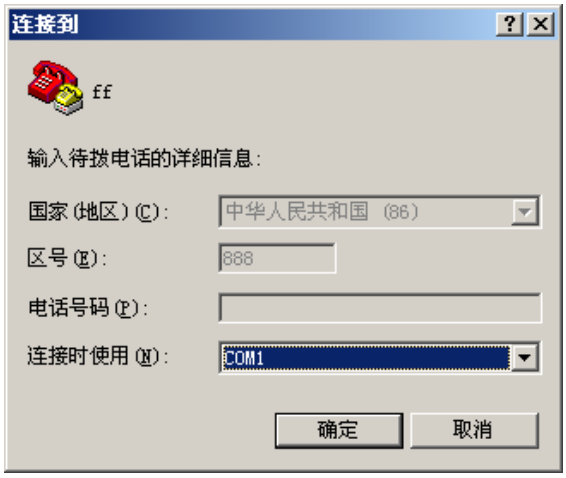
1. Click on "Start" -> "Programs" -> "Accessories" -> "Communications" -> "HyperTerminal" (or directly click "Start" -> "Run" and type "hypertrm" to launch HyperTerminal).



The interface after launching HyperTerminal is as follows:

2. Enter the connection name and select "OK."

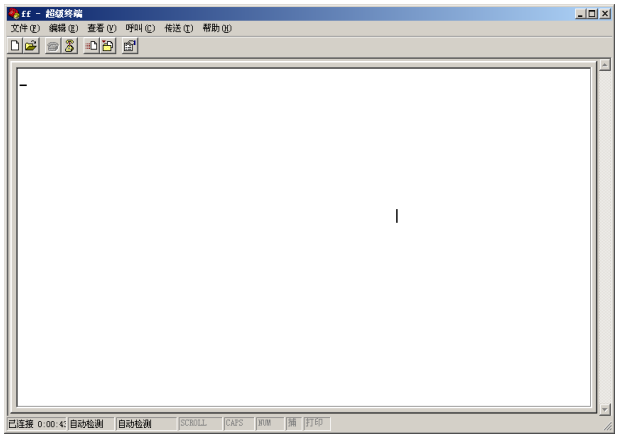
3. Select the PC's physical serial port used to connect to the F8L10GW-02 Console port and choose “Confirm."



4. Select the PC's physical serial port used to connect to the F8L10GW-02 Console port and choose “Confirm."

   Baud rate: 115200
   Data bits: 8
   Parity: None
   Stop bits: 1
   Flow control: None

At this point, HyperTerminal is running normally.



If the user is using the Windows 7 system, they can download a HyperTerminal for Windows 7 online. Alternatively, they can use other commonly used serial interaction software with similar usage.

CE RED Technical specification
Frequency Bands:
GSM 900: 880 MHz to 915 MHz
GSM1800: 1710 MHz to 1785 MHz
WCDMA Band I: 1920 MHz to 19780 MHz
WCDMA Band VIII: 880 MHz to 915 MHz
LTE Band 1: 1920 MHz to 1980 MHz
LTE Band 3: 1710 MHz to 1785 MHz
LTE Band 7: 2500 MHz to 2570 MHz
LTE Band 8: 880 MHz to 915 MHz
LTE Band 20: 832 MHz to 862 MHz
LTE Band 28: 703 MHz to 748 MHz
LTE Band 38: 2570 MHz to 2620 MHz
LTE Band 40: 2305 MHz to 2400MHz
WLAN 802.11b/g/n20: 2412 MHz to 2472MHz
WLAN 802.11n40: 2422 MHz to 2462MHz
LoRa: 863.2-869.8MHz
GPS:1575.42 MHz
BDS: 1561.098 MHz

Max power:
GSM 900: 32.23dBm
GSM1800: 31.21dBm
WCDMA Band I: 23.06dBm
WCDMA Band VIII: 23.36dBm
LTE Band 1: 22.65dBm
LTE Band 3: 22.52dBm
LTE Band 7: 22.65dBm
LTE Band 8: 22.60dBm
LTE Band 20: 22.56dBm
LTE Band 28: 22.59dBm
LTE Band 38: 22.56dBm
LTE Band 40: 22.70dBm
2.4GHz WLAN: 15.67dBm
LoRa: 9.68dBm45BGTR

Modulation Mode:
GSM: GMSK for GSM/GPRS; GMSK and 8PSK for EDGE
WCDMA: QPSK; HSDPA: QPSK/16QAM; HSUPA: BPSK
LTE: QPSK/16QAM/64QAM
2.4G WLAN:
802.11b(DSSS): CCK, DQPSK, DBPSK
802.11g(OFDM): BPSK, QPSK,16-QAM,64-QAM

802.11n(OFDM): BPSK, QPSK,16-QAM,64-QAM
GPS: BPSK

2.4G WIFI Channel Spacing: 5MHz
WCDMA Channel Spacing: 200KHz
GSM Channel Spacing: 200KHz

GSM/WCDMA/LTE: External Antenna, Gain(s): GSM 900: 1.6dBi, GSM1800: 1.6dBi;
WCDMA: B1: 4dBi, B8: 4dBi; LTE: B1: 4dBi; B3: 4dBi; B7: 4dBi; B8: 4dBi; B20: 4dBi;
B28: 4dBi; B38: 4dBi; B40: 4dBi;
WLAN: External Antenna, Gain(s): 2.4GHz: 2.8dBi;
LoRa: External Antenna, Gain(s): 2.5dBi.
GNSS: FPC Antenna.

CE Maintenance

1.Use careful with the earphone maybe excessive sound pressure from earphones and headphones can cause hearing loss.

2.Risk of explosion if battery is replaced by an incorrect type. Dispose of used batteries according to the instructions.
3.The product shall only be connected to a USB interface of version USB3.0.
4.Adapter shall be installed near the equipment and shall be easily accessible.
5.EUT Operating temperature range: 0° C to 50° C .
6.The device complies with RF specifications when the device used at 5mm you're your body.
7.To prevent possible hearing damage. Do not listen at high volume levels for long periods.

Declaration of Conformity

Xiamen Four-Faith Communication Technology Co., Ltd. hereby declares that this LoRaWAN is in compliance with the essential requirements and other relevant provisions of Directive 2014/53/EU. In accordance with Article 10(2) and Article 10(10),This product is allowed to be used in all EU member states.

CE

FCC Caution.

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

(1) This device may not cause harmful interference, and

(2) this device must accept any interference received, including interference that may cause undesired operation.

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Note: This equipment 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 equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-Reorient or relocate the receiving antenna.

-Increase the separation between the equipment and receiver.

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

-Consult the dealer or an experienced radio/TV technician for help.

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20 cm between the radiator & your body.