LoRAWAN Indoor Gateway
F8926-GW-02

User Manual

V2.0.0

Xiamen Four-Faith Communication Technology Co., Ltd.
https://www.fourfaith.com

Document Revision History

Date	Version	Specification	Author
2022-08-29	V1.0.0	Initial Version	SGK/HGL/YSL/WSC
2022-11-01	V1.0.1	Change the communication field "devEui" to "devEui"	SGK
2023-08-15	V1.0.2	English Version Update	YYL
2023-12-11	V.2.0.0	New version	Jonas

Copyright Statement

Trademark Statement

FCC Caution:

Part 15.21

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

Part 15.19

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.

Part 15.105

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 RF Radiation Exposure Statement:

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

RED Caution:

Hereby, Xiamen Four-Faith Communication Technology Co., Ltd. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

RF Function	Frequency bands	Maximum output power
2.4G WLAN	2412-2472MHz	18.28dBm(EIRP)
LORA	863-870MHz	7.62dBm(ERP)

GSM900	880-915MHz(TX)
	925-960MHz(RX)
		34.50dBm

DSC1800	1710-1785MHz(TX)
	18051880MHz(RX)
		31.50dBm

WCDMA Band I	1920-1980MHz(TX)
	2110-2170MHz(RX)
		25.50dBm

WCDMA Band VIII	880-915MHz(TX)
	925-960MHz(RX)
		25.50dBm

LTE Band 1	1920-1980MHz(TX)
	2110-2170MHz(RX)
		25.00dBm

LTE Band 3	1710-1785MHz(TX)
	1805-1880MHz(RX)
		25.00dBm

LTE Band 7	2500-2570MHz(TX)
	2620-2690MHz(RX)
		25.00dBm

LTE Band 8	880-915MHz(TX)
	925-960MHz(RX)
		25.00dBm

LTE Band 20	832-862MHz(TX)
	791-821MHz(RX)
		25.00dBm

LTE Band 28	703-748MHz(TX)
	758-803MHz(RX)
		25.00dBm

LTE Band 38	2570-2620MHz	25.00dBm
LTE Band 40	2300-2400MHz	25.00dBm

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

Contents

Chapter 1 Product Introduction

1.1 Product Overview

The F8926-GW-02 series gateway is a wireless communication gateway based on the LoRaWAN standard protocol. It connects to various types of standard LoRaWAN protocol application nodes, collects information, and transmits it to the cloud server through wired Ethernet/4G/WIFI methods. This product utilizes a high-performance industrial-grade 32-bit communication processor, supported by an embedded real-time operating system as its software platform. It provides 1 Ethernet WAN (POE) port, 1 LAN port, and 1 WIFI interface, supporting WIFI wireless configuration management and online upgrades, with DC and POE+ power inputs.

The F8926-GW-02 gateway complies with the standard LoRaWAN protocol and supports multiple modes, including the embedded Network Server mode (Network Server deployed within the gateway), Basicstation mode (connecting to an external server corresponding to the Basicstation protocol), and Semtech UDP GWMP Protocol mode (connecting to an external NS server via GWMP UDP protocol).

This product has been widely applied in the IoT industry chain, including sectors such as M2M, smart meters, disaster monitoring, smart sensing, smart photovoltaics, smart grids, intelligent transportation, industrial automation, smart buildings, fire protection, public safety, environmental protection, meteorology, digital healthcare, remote sensing surveying, military, space exploration, agriculture, forestry, water management, coal mining, petrochemicals, and more.

1.2 Product Features

Industrial-Grade Application Design

- ◆ Utilizes High-Performance Industrial-Grade LoRa Module (SX1302)
- ◆ Utilizes High-Performance Industrial-Grade Wireless Module
- ◆ Utilizes High-Performance Industrial-Grade 32-bit Communication Processor
- ◆ Adopts a metal aluminum casing with an IP30 protection rating, metal casing, and system security isolation, making it particularly suitable for industrial field applications.
- ◆ Wide power supply input (DC9~36V), standard: 12V/1.5A
- ◆ Supports POE+（802.3af/at）input

Stable and Reliable

- ◆ WDT Watchdog Design, Ensuring System Stability
- ◆ Utilizes a Comprehensive Anti-Drop Mechanism to Ensure Data Terminals Stay Online Permanently
- ◆ Ethernet Interface with Built-in 1.5KV Electromagnetic Isolation Protection
- ◆ SIM/UIM Card Interface with Built-in 15KV ESD Protection
- ◆ Power Interface with Built-in Reverse Polarity Protection and Overvoltage Protection

- Antenna Interface Lightning Protection (optional)

Standard and User-Friendly

- Provides Standard TYPE-C, 4G, Ethernet, and WiFi Interfaces, Allowing Direct Connection to Serial Devices, Ethernet Devices, and WiFi Devices
- Provides Standard Wired WAN Port (Supports Standard PPPOE Protocol), Allowing Direct Connection to ADSL Devices
- Smart Data Terminal, Enters Data Transmission State Upon Power On
- Provides Powerful Central Management Software for Convenient Device Management (optional)
- Easy to Use, Flexible, Multiple Working Mode Options
- Convenient System Configuration and Maintenance Interfaces (Including Local and Remote WEB and CLI Methods)

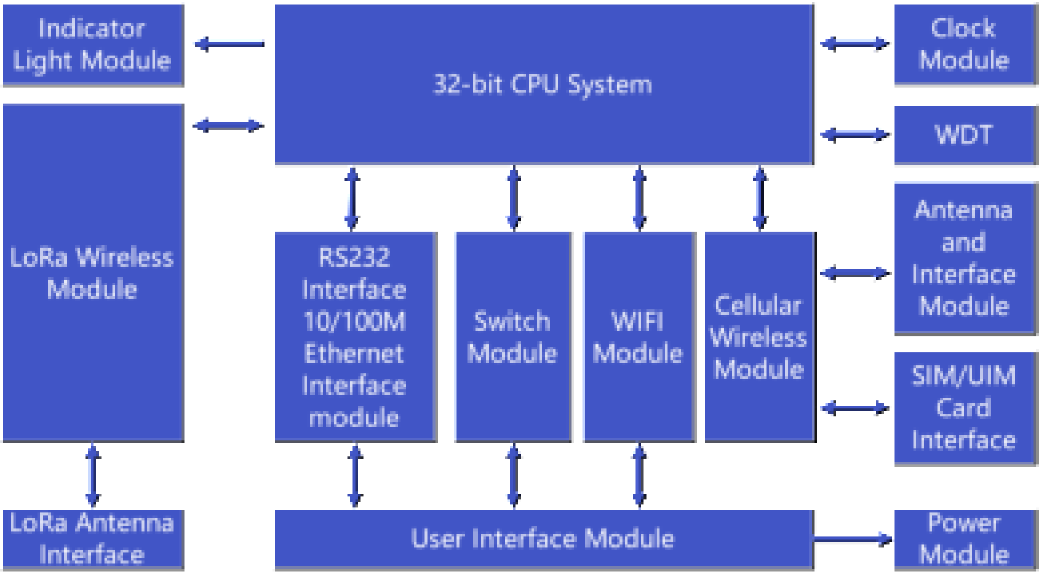
Powerful Functionality

- Provides Wired Ethernet, 4G, WiFi, and Other Data Connection Methods
- LoRaWAN Protocol Versions: 1.0.2 and 1.0.3
- LoRaWAN Protocol：ClassA、ClassC
- WIFI supports 802.11b/g/n
- WiFi supports various encryption methods such as WEP, WPA, WPA2, as well as features like MAC address filtering.
- Supports Semtech UDP GWMP Protocol mode
- Supports embedded Network Server mode, reducing operation and maintenance costs as well as NS deployment costs, for simple and user-friendly management.
- Supports Basicstation mode, with various data encryption methods to ensure data transmission security.
- Supports Platform Connection: LinkWAN, ChirpStack, Tencent Cloud, TTN (The Things Network), AWS, etc.
- Provides HTTP Push, MQTT Subscribe and Publish, and TCP Connection Methods to the Outside
- Supports configuration of MQTT topics for interfacing with the client, and allows data content to be transformed using embedded JavaScript functions.
- Supports multiple WAN connection methods, including static IP, DHCP, L2TP, PPTP, PPPOE, 2.5G/3G/4G.
- Supports intelligent dual-link switching and backup function for wireless cellular and wired WAN (optional).
- Supports VPN client (PPTP, L2TP, OPENVPN, IPSEC, and GRE) (Note: Supported only in the VPN version)
- Supports VPN server (PPTP, L2TP, OPENVPN, IPSEC, and GRE) (Note: Supported only in the VPN version)
- Supports remote management, SYSLOG, SNMP, Telnet, SSHD, HTTPS, and other functions.
- Supports local and remote online upgrades, as well as importing and exporting configuration files.
- Supports NTP and has a built-in RTC.

- Supports various domestic and international DDNS
- Supports MAC address cloning and PPPOE server functionality.
- WiFi supports 802.11b/g/n, and offers various working modes including WiFi AP, AP Client, Repeater, Bridge, and WDS (Wireless Distribution System) (optional)
- WiFi supports various encryption methods including WEP, WPA, WPA2, and offers features like RADIUS authentication and MAC address filtering.
- Supports various online and offline trigger modes, including SMS, ringing, serial data, and network data-triggered online/offline modes.
- Supports APN/VPDN
- Supports multiple DHCP servers and DHCP clients, DHCP binding with MAC addresses, DDNS, firewall, NAT, DMZ host, QoS, traffic statistics, real-time display of data transmission rate, and other functions.
- Supports multiple network protocols including TCP/IP, UDP, FTP (optional), HTTP, and more.
- Supports SPI firewall, VPN passthrough, access control, URL filtering, and other functions.

1.3 Block Diagram of Operation

The block diagram of the router's operation is as follows:



1.4 Product Specifications

Wireless Parameters

Items	Contents
F8926-GW-02	**LoRaWAN Gateway**
Standards and Frequency Bands	Supports Full Network：LTE FDD、LTE TDD、EVDO、WCDMA、TD-SCDMA、CDMA1X、 GPRS/EDGE
Theoretical Bandwidth	LTE FDD: Downlink Speed 100Mbps, Uplink Speed 50Mbps LTE TDD: Downlink Speed 61Mbps, Uplink Speed 18Mbps DC-HSPA+: Downlink Speed 42Mpbs, Uplink Speed 5.76 Mbps TD-HSPA+: Downlink Speed 4.2Mbps, Uplink Speed 2.2Mbps EVDO Rev. A: Downlink Speed 3.1Mbps, Uplink Speed 1.8Mbps
Receiver Sensitivity	<-97dBm

WIFI Wireless Parameters

Items	Contents
Standards and Frequency Bands	Support IEEE802.11b/g/n Standard
Theoretical Bandwidth	IEEE802.11b/g：Maximum Speed of 54Mbps IEEE802.11n：Maximum Speed of 150Mbps
Security Encryption	Supports various encryption methods including WEP, WPA, WPA2, and optional WPS functionality.
Receiver Sensitivity	<-72dBm@54Mpbs

LoRa Parameters

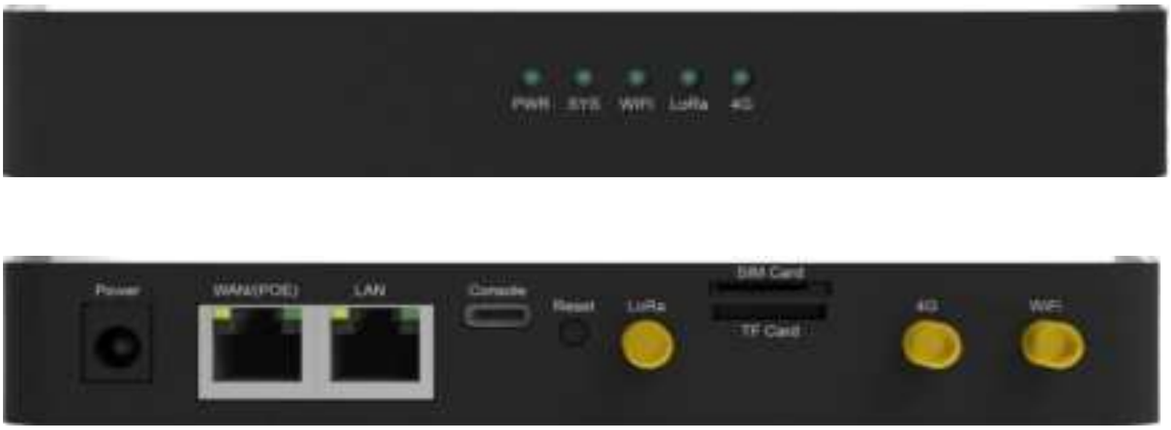
Items	Contents
Operational Channels	Uses a simple star topology network and supports blind repeaters.
LoRaWAN Protocol	ClassA、ClassC
Urban Communication Reference Distance	9km
Reference Distance for Floor Penetration	16 Floors@SF12
Operating Frequency	EU433、CN470-510、CN779-787、EU863-870、US902-928、AU915-928、AS923、KR920-923
Maximum Antenna Receive Sensitivity	-140dbm @LoRa
Communication Bandwidth	125kHz、250kHz、500kHz
Communication Channel	8 Uplink Channels, 1 Downlink Channel
Communication Rate	Adaptive Link Rate
Communication Mode	Half-Duplex
Operating Mode	Supports Transceiving on Different Frequencies and Transceiving on the Same Frequency
Reporting Server Mode	4G、Wire Ethernet
Wireless Management	WiFi Wireless Management and Upgrades

Hardware System

Items	Contents
CPU	Industrial-Grade 32-bit Communication Processor
FLASH	32MB (Expandable up to 64MB)
DDR2	128MB

Interface Type

Items	Contents
Power Interface	Standard 3-pin power socket, with built-in reverse polarity protection and overvoltage protection.
WAN（POE）	WAN/LAN configurable, with 1 10/100M Ethernet port (RJ45 socket), adaptive MDI/MDIX, and built-in 1.5KV electromagnetic isolation protection.
LAN	1 10/100M Ethernet port (RJ45 socket), adaptive MDI/MDIX, and built-in 1.5KV electromagnetic isolation protection.

Console	Type-C USB
Reset Button	By pressing this button, you can restore the parameter configuration of the ROUTER to its factory settings.
TF Card	8GB/32GB，Customizable Support
SIM Card	Supports SIM Cards from the Three Major Carriers (3FF Cards)
Antenna	LoRa、WIFI、4G，3 Antenna Interfaces
Indicator Lights	"PWR", "SYS", "WiFi","LoRa ","4G"　5 Indicator Lights
 PWR SYS WiFi LoRa 4G Power WAN(POE) LAN Console Reset LoRa SIM Card TF Card 4G WiFi Note: There may be differences in accessories and interfaces for different models. Please refer to the actual product.	

Power Supply

Items	Contents
Power Supply	DC 12V/1.5A (Recommended), supports power supply voltage range DC 9~36V
	POE+（802.3af/at）Power Consumption 25W max

Power Consumption

Operational Status	Power Consumption
Standby	Average Current≤120mA@12V
Communication	Transmit Current≤460mA@12V(with 4G) Transmit Current≤143mA@12V(without 4G） Receive Current≤120mA@12V

Physical Characteristics

Items	Contents
Casing	Aluminum Casing, IP30 Protection Level
Dimensions	160X105X24 mm (Excluding Antenna and Mounting Accessories)
Weight	450g(Excluding Accessories)

Other Parameters

Items	Contents
Operating	-35～+75°C（-31～+167℉）

Temperature	
Storage Temperature	-40～+85ºC（-40～+185℉）
Relative Humidity	95%（no condensation）

Chapter 2 Installation

2.1 Overview

The router must be installed correctly in order to achieve its designed functionality. Usually, the installation of the equipment must be carried out under the guidance of authorized and qualified engineers from our company.

➢ *Precautions：*
Please do not install the router while it is powered on.

2.2 Packing List

Please keep the packaging materials when you unpack the box, so that they can be used for operating if needed in the future. The list is as follows:

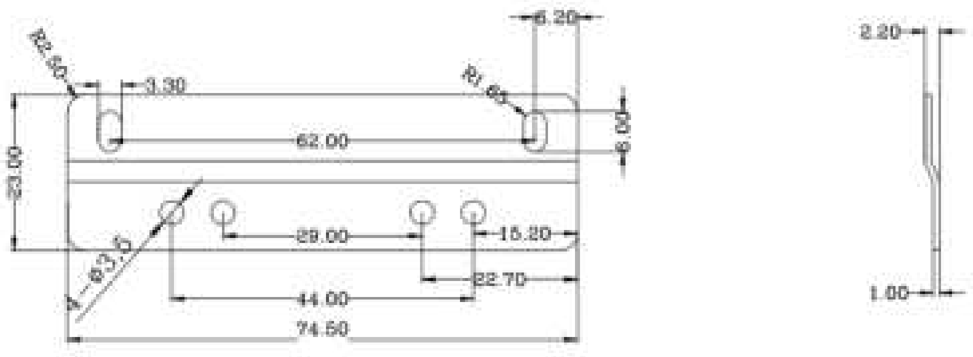
✧ 1 Router Host
✧ 1 Wireless Cellular Stick Antenna (SMA Male Connector)
✧ 1 WiFi Stick Antenna (SMA Female Connector)
✧ 1 LoRa Stick Antenna (SMA Male Connector)
✧ 1 Power Adapter
✧ 1 Ethernet Cable

Note: LoRa suction cup antenna is optional.

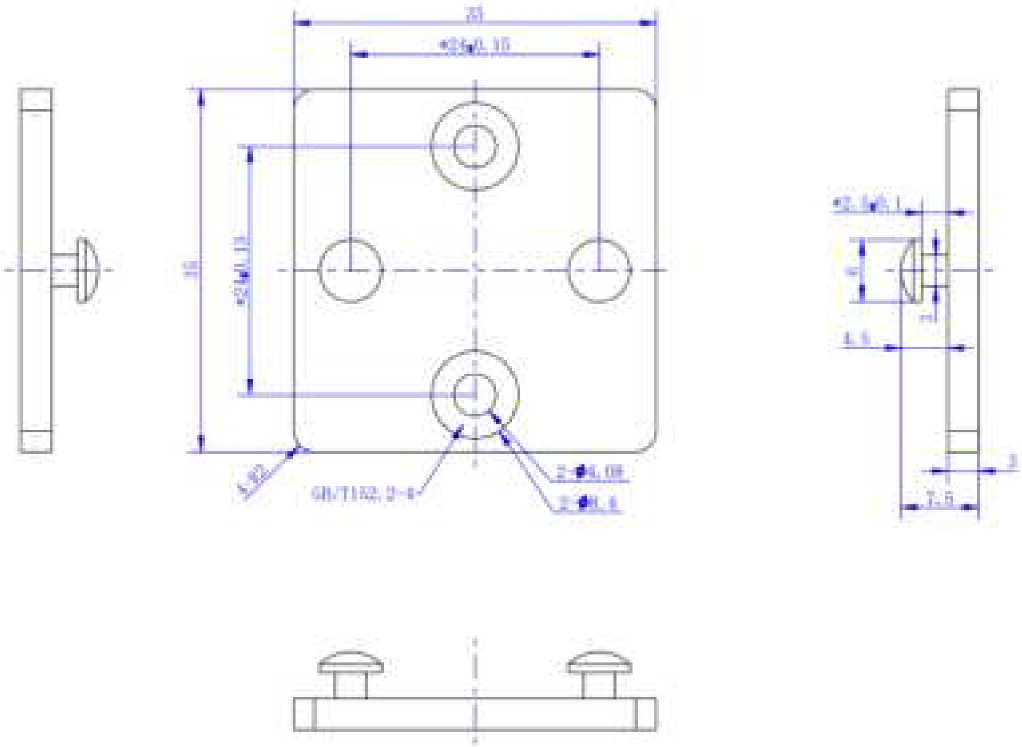
2.3 Installation and Cable Connection

Physical Dimensions：

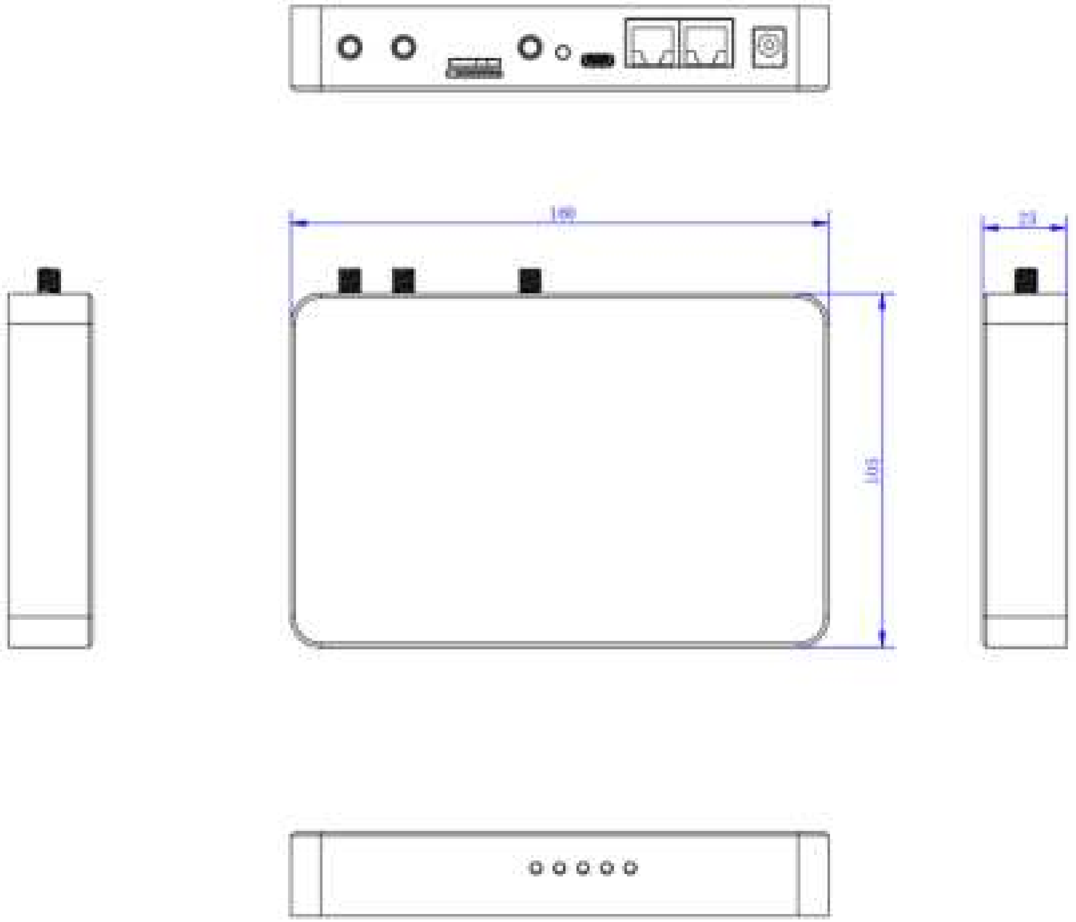
The physical dimensions are shown in the following diagram. (Unit: mm) The specifications for the mounting bracket and router device screws are: M3*5mm countersunk screws.

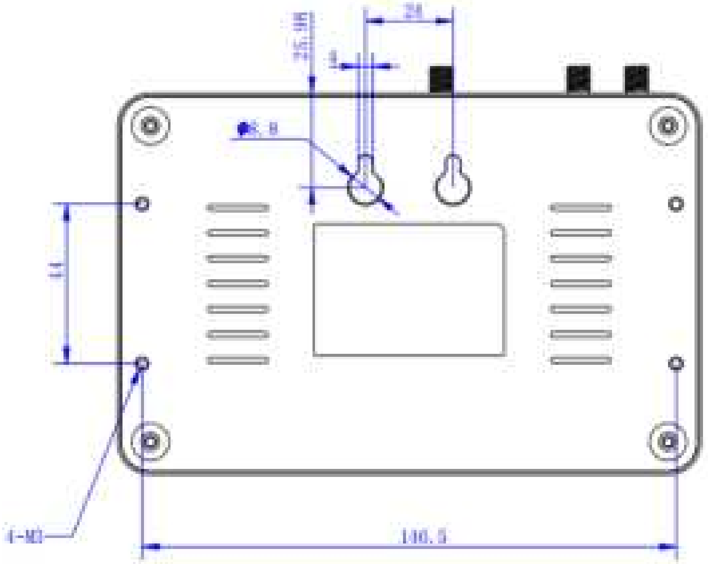


Bracket Dimensions



Wall Mount Bracket Dimensions





Router Dimensions

Note: When using the mounting bracket to install the router, use M3 screws with a depth of 3-4mm screwed into the router. The mounting bracket and wall mount bracket are optional accessories.

Antenna Installation:

Wireless Wide Area Network (WWAN) antenna interface is an SMA female socket (labeled as "4G"). Screw the provided wireless cellular stick antenna with an SMA male connector into this antenna interface and ensure it is tightened securely to maintain signal quality.

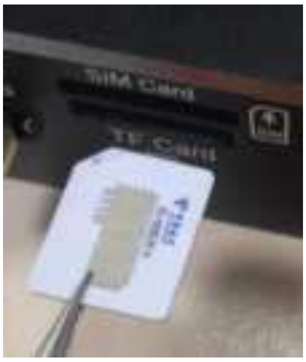
The Wireless Local Area Network (WLAN) antenna interface is an SMA female socket (labeled as "WIFI"). Screw the provided WIFI stick antenna with an SMA male connector into this antenna interface and ensure it is tightened securely to maintain signal quality.

The Long-Range (LoRa) antenna interface is an SMA female socket (labeled as "LoRa"). Screw the provided LoRa stick antenna with an SMA male connector into this antenna interface and ensure it is tightened securely to maintain signal quality.

Note: The wireless cellular 4G antenna, WiFi antenna, and LoRa antenna must not be connected in reverse, otherwise the device will not function properly.

SIM/UIM Card Installation:

When installing the SIM/UIM card, please pay attention to the card's orientation. The golden contacts should face downward, and the cut corner should be positioned at the top-left corner. Gently push the card into the slot until you feel a slight resistance, indicating that the card is secured. To remove the card, simply press the "PUSH" area to release it.



Note: Align the card's cut corner with the printed cut corner, and ensure that the golden contacts are facing downward.

Connect Ethernet Cable:

Insert one end of the Ethernet cable into the LAN port of the Router, and the other end into the Ethernet interface of the user's device. The Ethernet cable connection should be as follows:

RJ45-1	RJ45-2	Wire Color
1	1	White/Orange
2	2	Orange
3	3	White/Green
4	4	Blue
5	5	White/Blue
6	6	Green
7	7	White/Brown
8	8	Brown

Connect Console Wire（TYPE-C）：

Simply use a standard Type-C cable, connect one end to the Router device and the other end to a PC, then install the corresponding drivers.

Driver download address：https://www.wch.cn/search?t=all&q=CH340C

Installation package：

CH341SER.ZIP

2.4 Power Instructions

Routers are commonly used in complex external environments. In order to adapt to these challenging application scenarios and enhance the system's operational stability, advanced power supply technology is employed in the Router. Users can power the Router using the standard configuration of a 12VDC/1.5A power adapter or directly supply it with DC power in the range of 9-36V. When using an external power supply for the Router, it is essential to ensure the stability of the power source (with ripple less than 300mV) and guarantee that momentary voltage doesn't exceed 36V. Additionally, the power supply should provide a power output greater than 8W.

It is recommended to use the standard configuration of a 12VDC/1.5A power adapter or POE+ (802.3af/at) input.

2.5 Indicator Lights Explanation:

The Router provides the following indicator lights: "PWR", "SYS", "WiFi", "LoRa", and "4G". The status explanations for each indicator light are as follows:

Indicator Lights	Status	Specification
Power	ON	Device power normal
	OFF	The device is not powered on / in the shutdown period of the scheduled power on/off function.
SYS	Blinking	The system is running normally.
	OFF	The system is not functioning properly.
WIFI	OFF	The WiFi is not active.
	ON	The WiFi is active.
LoRa	ON	LoRa has been detected.
	OFF	LoRa not detected.
4G	ON	Device has logged into the network.
	OFF	The device is not logged into the network.

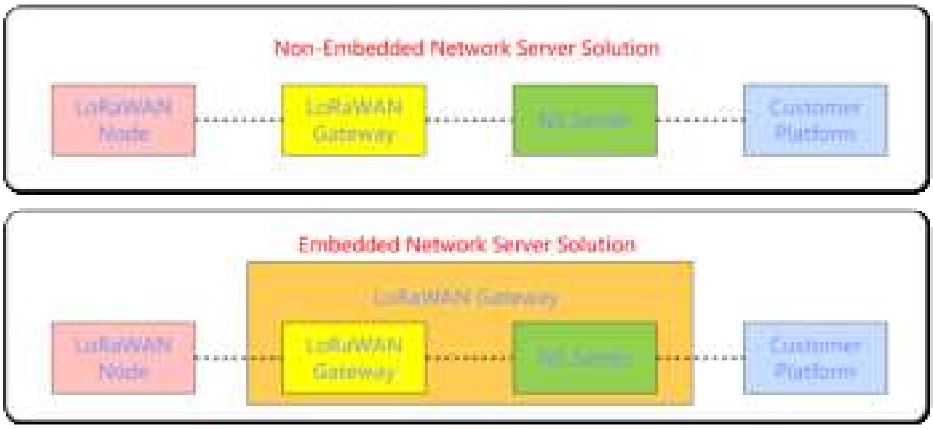
2.6 Reset Button Instructions

The router is equipped with a reset button labeled "Reset." The function of this button is to restore the router's settings to the factory defaults. The procedure is as follows: Insert a pointed object into the "Reset" hole and gently hold down the reset button for about 15 seconds, then release it. At this point, the router will automatically restore the parameter settings to the factory defaults. After approximately 5 seconds, the router will automatically restart (the automatic restart phenomenon is as follows: the "SYS" indicator light will go off for about 10 seconds and then resume normal operation).

Chapter 3 Quick Start Guide

3.1 Introduction to Solution Architecture

3.1.1 Difference Between Embedded and Non-Embedded

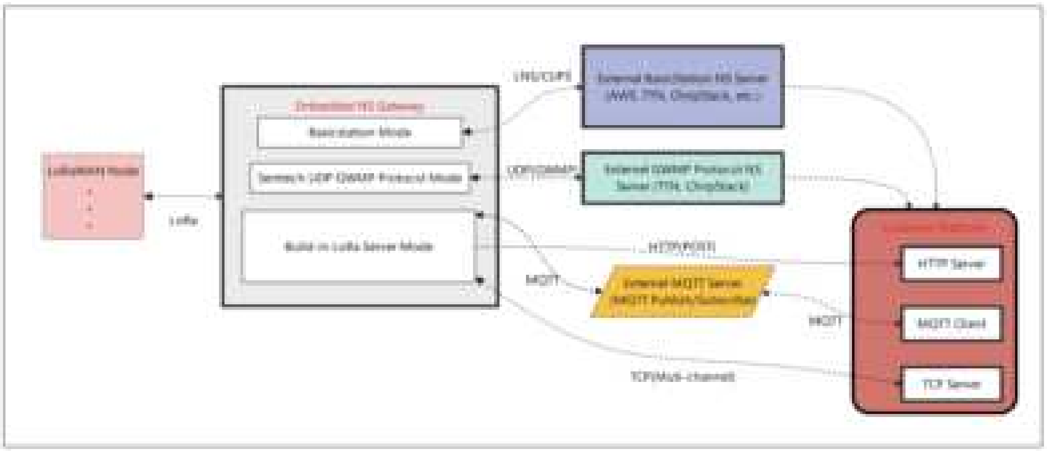


As shown in the diagram above, the main difference between the embedded and non-embedded solutions lies in the position of the Network Server (NS). In the non-embedded solution, the NS is typically deployed on a separate server, while in the embedded solution, the NS is deployed within the gateway itself.

❖ The advantage of the **embedded solution** (embedded mode) is that there is no need to deploy the Network Server (NS) on an external server, which reduces operational costs and allows for a quick and convenient setup of the entire LoRaWAN system. However, the drawback is that the performance and storage capacity of the gateway system are relatively lower compared to a dedicated server. This limitation affects the number of nodes that can be supported and the ability to cache large amounts of information.

❖ The advantage of the **non-embedded solution** (external mode) is that servers have stronger performance and larger storage capacity, enabling them to manage a large number of gateways and nodes. This solution can be deployed through clustering to significantly enhance system performance and availability. However, the drawback is the need for additional server deployment to host the Network Server (NS), which requires maintenance and increases project costs. Setting up the system and troubleshooting may also require more time and effort.

3.1.2 System Framework


LoRaWAN Node
LoRa
Embedded NS Gateway
Basicstation Mode
Semtech UDP GWMP Protocol Mode
Built-in LoRa Server Mode
LNS/CUPS
External BasicStation NS Server (AWS, TTN, ChirpStack, etc.)
UDP/GWMP
External GWMP Protocol NS Server (TTN, ChirpStack)
HTTP(POST)
MQTT
External MQTT Server (MQTT Publish/Subscribe)
MQTT
TCP(Multi-channel)
HTTP Server
MQTT Client
TCP Server

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

❖ In the **Basics Station** (BasicStation mode), data will be exchanged bidirectionally with the corresponding connected server. The gateway only functions as a data forwarding unit. In this scenario, device management, data encryption/decryption, and integration with customer platforms are all performed on the server side.

❖ In the **Semtech UDP GWMP Protocol** (external NS mode), data will be communicated with the external Network Server (NS) using the standard UDP protocol. In this scenario, device management, data encryption/decryption, and integration with customer platforms will be handled within the external NS server. For instance, commonly used external NS servers include those provided by Four-Faith Cloud.

❖ In the **Built-in LoRa Server** (internal NS mode), data will be routed to the NS server that is integrated within the gateway. In this scenario, device management, data encryption/decryption, and integration with customer platforms will be handled within the built-in NS server, also known as the LoRa Network Server. Clients can achieve data push functionality through configuration of an HTTP server (HTTP POST only supports uplink push and doesn't support downlink data), or through MQTT and TCP methods for both uplink and downlink data. The embedded NS serves as the core network for LoRaWAN. This product theoretically supports a large number of gateways and device connections. It manages tasks such as device provisioning, data encryption/decryption, uplink and downlink data transmission, and data pushing. Uplink data from devices, after being decrypted by LoRaWAN, establishes a connection with the customer platform via an interface. Customers can use MQTT for data publishing or TCP for downlink data, which is encrypted by LoRaWAN and sent to the specified device.

3.2 Accessing the Configuration Interface

3.2.1 Accessing the Web Management Platform

1) Method 1: After powering on the gateway, the default WiFi SSID is "Four-Faith," and the default password is blank. Once successfully connected to the WiFi, the LAN IP address of the gateway is 192.168.1.1. You can then access the web management platform by entering http://192.168.1.1 (or simply 192.168.1.1) into your browser's address bar.

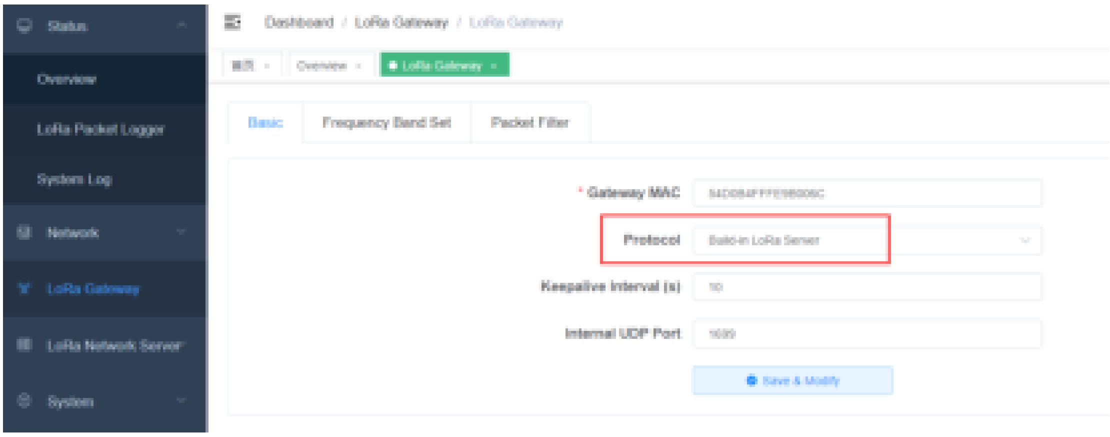
2) Method 2: If you already know the WAN address of the gateway (e.g., set to static IP 192.168.1.88), you can directly access the web management platform by visiting http://192.168.1.88 in your web browser.

3) Login using the default credentials: Username: admin, Password: admin. Click "Login" to access the Web management platform.

Note: Please use Google Chrome browser as other browsers might have compatibility issues.

3.2.2 To add a device in the embedded mode

1. **Identify the frequency band and corresponding frequencies for your device** (e.g., a standard EU868 terminal, frequencies: 868.1MHz, 868.3MHz, 868.5MHz)
2. **Confirm if the embedded mode is enabled** (default is embedded mode). If not, change it to embedded mode.
   Path: LoRa Gateway → Basic Settings

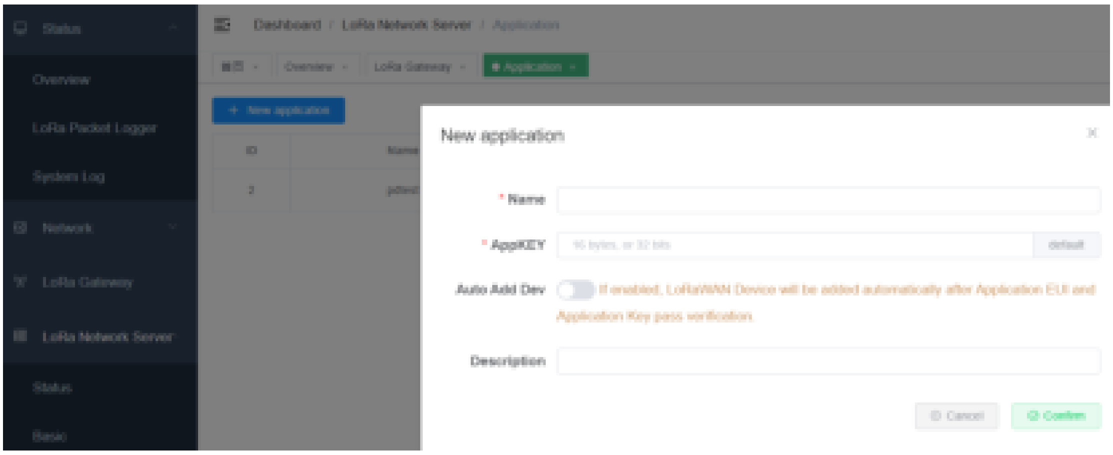


* Gateway MAC: 54D084FFFE9B006C
Protocol: Build-in LoRa Server
Keepalive Interval (s): 10
Internal UDP Port: 1680

3. **Check if the gateway's frequency band and frequency points match** (default frequency points are determined by regional parameters). If they don't match, modify them to match.
   Path: LoRa Gateway → Frequency Band Configuration

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

Path: LoRa Network Server => Applications => Add Application



New application

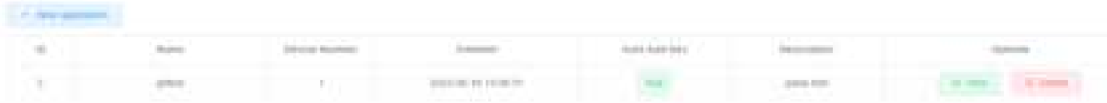
* Name

* AppKEY: 16 bytes, or 32 bits — default

Auto Add Dev: If enabled, LoRaWAN Device will be added automatically after Application EUI and Application Key pass verification.

Description

In the above figure, both AppKEY and AppEUI are generated by clicking on the "default" on the right side (these values are default values provided by Four-Faith; for non-Four-Faith devices, please modify them accordingly). Choose either ClassA or ClassC based on the device type, then click "Confirm" to proceed with the addition. After adding, the following page will appear:



5. **Device Onboarding**

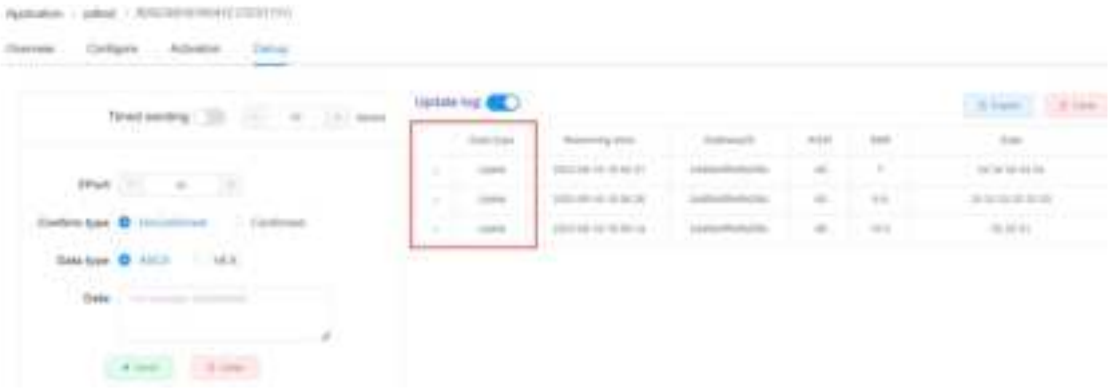
Device Initiation of Network Join Request and Verification of Successful Joining; if Joining Fails, Follow these Troubleshooting Steps:

1) Verify if the gateway can receive the network join request sent by the device (you can use a packet capture tool, path: Status → LoRa Packet Logger).

2) If the gateway receives the network join request but does not see the join accept packet (Join Accept), it's usually due to a mismatch between the AppKey or AppEUI configured in the application and the device.

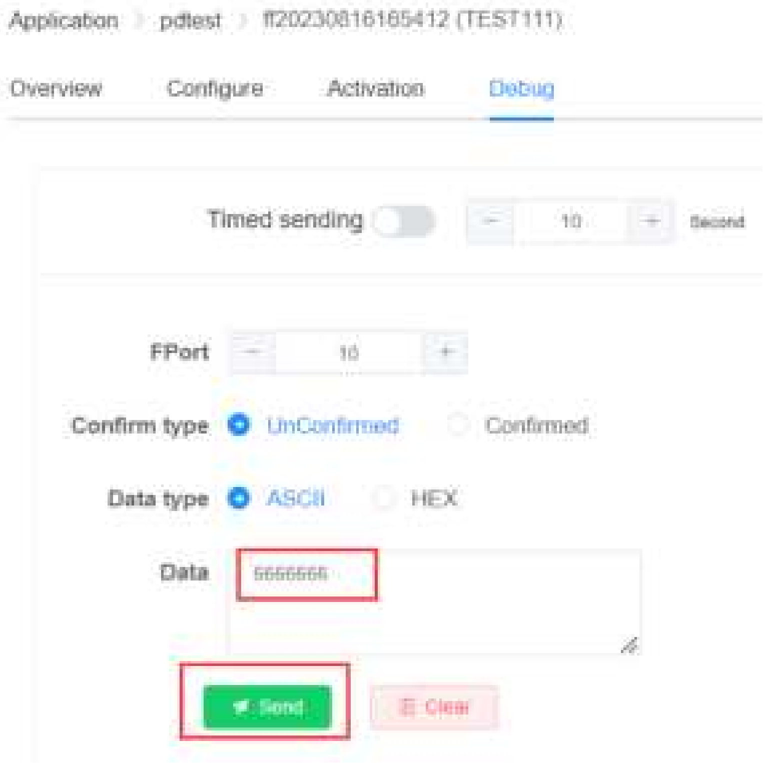
6. Upstream Data from Devices

After the device successfully joins the network, instruct the device to send any data. You can then navigate to the corresponding application in the device list to view the data:
Path: LoRa Network Server → Applications → Select the corresponding application (click to view) → Find the corresponding device (click to view) → Online Debugging



7. Send Data to Device

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



Application > pdtest > ff202308161165412 (TEST111)

Overview Configure Activation Debug

Timed sending ○ − 10 + Second

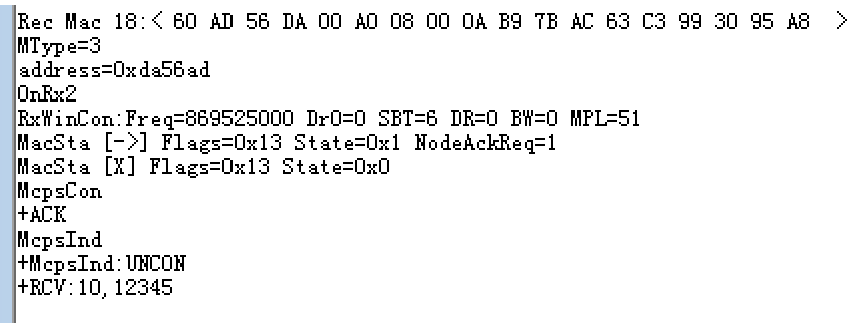
FPort − 10 +

Confirm type ● UnConfirmed ○ Confirmed

Data type ● ASCII ○ HEX

Data 5666666

The Four-Faith module receives data as follows:



```
Rec Mac 18:< 60 AD 56 DA 00 A0 08 00 0A B9 7B AC 63 C3 99 30 95 A8  >
MType=3
address=0xda56ad
OnRx2
RxWinCon:Freq=869525000 Dr0=0 SBT=6 DR=0 BW=0 MPL=51
MacSta [->] Flags=0x13 State=0x1 NodeAckReq=1
MacSta [X] Flags=0x13 State=0x0
McpsCon
+ACK
McpsInd
+McpsInd:UNCON
+RCV:10,12345
```

Precaution：

The device types are divided into ClassA and ClassC, with the following data reception methods:

1. In ClassA mode, after sending data, it won't be directly delivered to the device. The data will be sent to the device only after the device sends an uplink data transmission.

2. In ClassC mode, when sending data, it will be directly delivered to the device. If the device doesn't receive the data, please verify whether the NS configuration type matches the device configuration type. If they don't match, make the necessary changes and rejoin the network before conducting data communication tests.

Chapter 4 Detailed Introduction to Function Pages

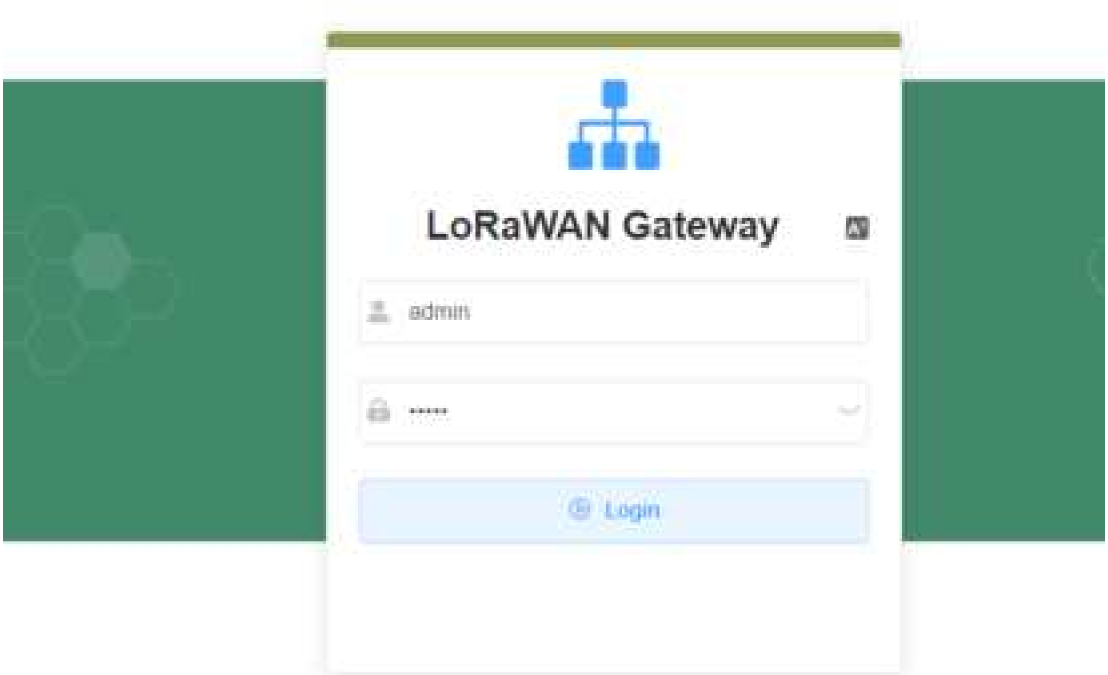
4.1 Interface Management Configuration

4.1.1 Web Management Platform

1) Method 1: After the gateway is powered on, the default WiFi name is "Four-Faith" and the default password is blank. Once the WiFi connection is successful, the LAN address of the gateway will be set to 192.168.1.1. You can then log in by visiting http://192.168.1.1 (or simply entering 192.168.1.1) in your web browser.

2) Method 2: If you already know the WAN address of the gateway (for example, if it's set to a static IP like 192.168.1.88), you can directly access it by visiting http://192.168.1.88 in your web browser.

3) Login using the default credentials: Username - admin, Password - admin. Click on "Login" to access the Web Management Platform.



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

4.1.2 Directory Details

Below, we will introduce the functions of each page in the order of the directory:

- Status
  - Overview: The gateway listens to data statistics and displays system parameter information.
  - LoRa Message Recorder: Display of received data and downstream data on the gateway.
  - System Log: Operational logs during runtime.
- Network
  - WAN Interface: Gateway WAN configuration, you can configure network information here, such as setting up DHCP or static IP.
  - Wi-Fi：wifi Parameters and Security Configuration
  - Network Diagnostics: Includes Ping, Traceroute, and Nslookup commands.
  - Firewall: Basic firewall parameter configuration.
- LoRa Gateway: Gateway mode configuration, frequency channel parameter configuration, packet filtering, etc.
- LoRa Network Server
  - Status: Display of embedded NS statistical information.
  - Basic Settings: Configuration of NS-related parameters, such as ADR switch, RX2 parameter settings, etc.
  - Gateway: Display of gateway information.
  - Application: Display of application information, including device list and more.
  - Multicast: Multicast management.
  - Interfaces: Configuration of protocols for integration with client platforms, data transformation, heartbeat settings, etc.
- System
  - System: Embedded NS version information, system time settings, etc.
  - Change Password: Modify the password for the Web management platform.
  - Reboot: Restart the gateway button.
  - Factory Reset: Factory reset button.

4.1.3 Management Configuration

4.1.3.1 Status

1. Overview

- **Path:** Status -> Overview
- **Function:** Displays communication statistics of the gateway, making it easy to view and analyze the RF environment around the gateway. This helps determine device communication status, identify potential interference, and make assessments regarding device connectivity.
- **Details:**

✧ **Received Packets**: The number of packets received since system startup.

✧ **Sent Packets**: The number of packets sent since system startup.

✧ **Active Nodes:** The number of uplink nodes received by the gateway.

✧ **Busy Nodes:** Nodes that have sent uplink data twice within 10 seconds are considered busy nodes. This statistic reflects the count over the past hour**.**

✧ **LoRa Channel Utilization Statistics:** Channel utilization status in various time intervals over the past 24 hours.

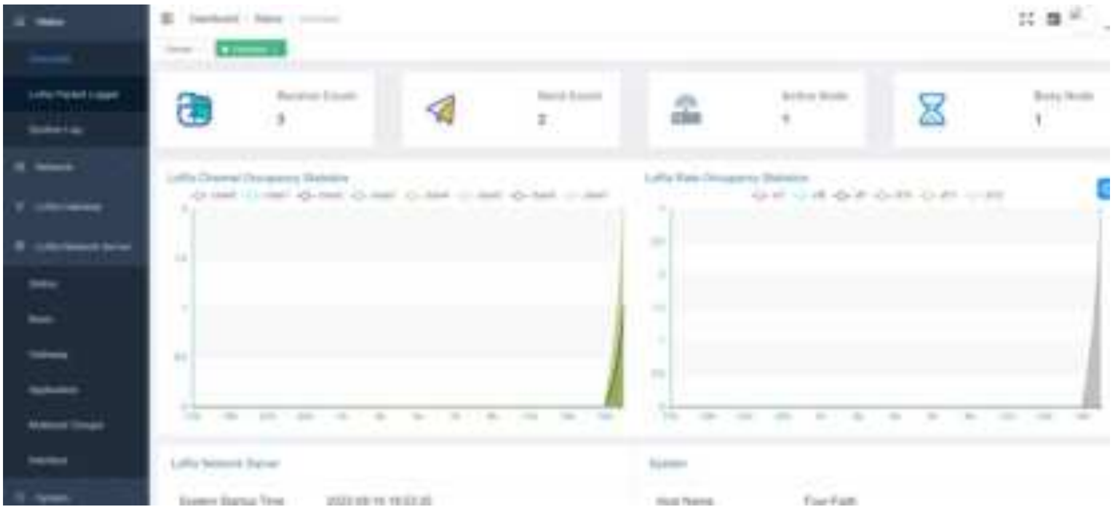
✧ **LoRa Data Rate Utilization Statistics:** Data rate utilization status in various time intervals over the past 24 hours.

✧ **LoRa Network Server:** Includes system startup time, LoRa protocol, device count, NS device uplink count, NS device downlink count, and NS MQTT connection status.

✧ **System:** Includes host name, LAN MAC address, WAN MAC address, wireless MAC address, WAN IP address, LAN IP address, and WAN protocol.

✧ **Wireless**: Includes wireless switch, mode, network mode, name, channel, and transmission power.

➢ Preview:



2. LoRa Message Record

➢ **Path:** Status -> LoRa Message Recorder

➢ **Function:**

■ Display LoRaWAN data received by the gateway and data sent by the gateway.

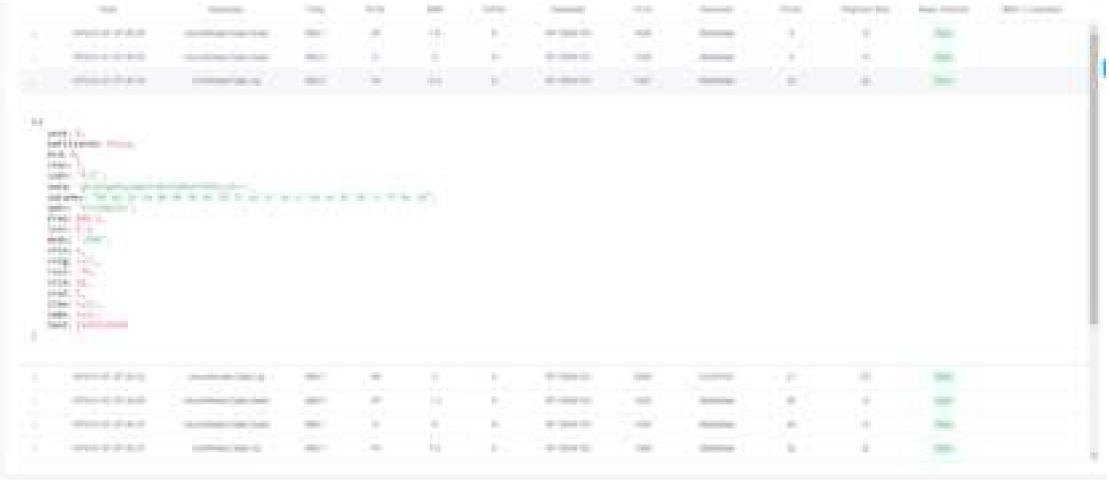
■ It can be used to analyze the communication between the gateway and devices, and allows for analysis of issues based on data types, such as unanswered join requests, missing downlink data, communication quality, and more.

➢ **Details:**

■ Update Log Switch: It is enabled by default. When disabled, it allows for expanding the view of data. While it's disabled, data is still received normally, and once enabled again, it automatically updates the list.

- LoRaWAN Data Type Selection: This option is used to facilitate the analysis of communication issues by selecting different LoRaWAN data types.
- Packet Filter Status: Indicates whether the packet has been filtered. Filtered data will not be reported to the NS server. The filtering configuration can be found in LoRa Gateway → Packet Filter. The options are as follows: All - Displays both filtered and unfiltered data. Unfiltered - Displays only unfiltered data. BeFiltered - Displays only data that has been filtered.
- devAddr：Search by Short Address
- Time：Received Data Time
- DataType：Data Type
  - ALL
  - Join Request
  - Join Accept
  - Unconfirmed Data Up
  - Unconfirmed Data Down
  - Confirmed Data Up
  - Confirmed Data Down
- Freq：Communication Frequency Point
- RSSI：Signal Strength
- SNR：Signal-to-Noise Ratio
- TxPwr：Transmit Power, this value is 0 during uplink
- FCnt：Frame Count, can be used to determine if there are any packet losses or retransmissions.

➢ **Preview：**



3. LoRa Packet Logger

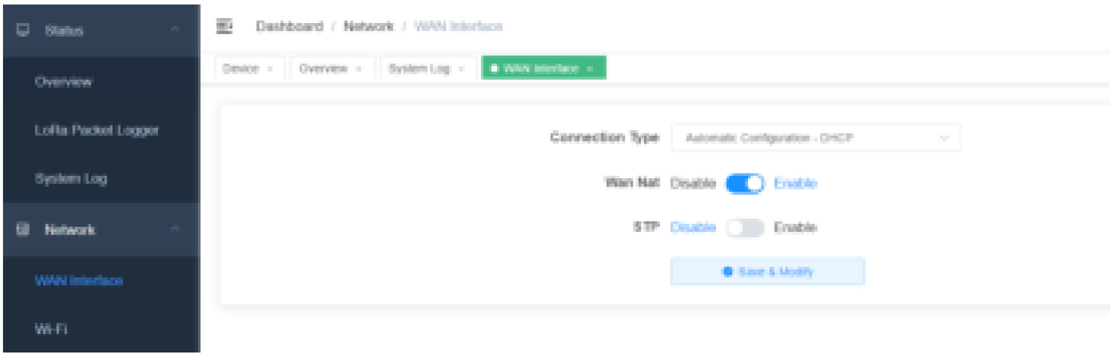
➢ **Path：** Status → LoRa Packet Logger

- ➢ **Function:** The logs can be used to analyze the overall operation of the gateway, abnormal device communication situations, and other anomalies.
- ➢ **Details:**
  - ✧ Switch: Enabled by default, when paused, new data is stored in the browser cache and will be updated when re-enabled.
- ➢ **Preview:**

4.1.3.2 Network

1. Network

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



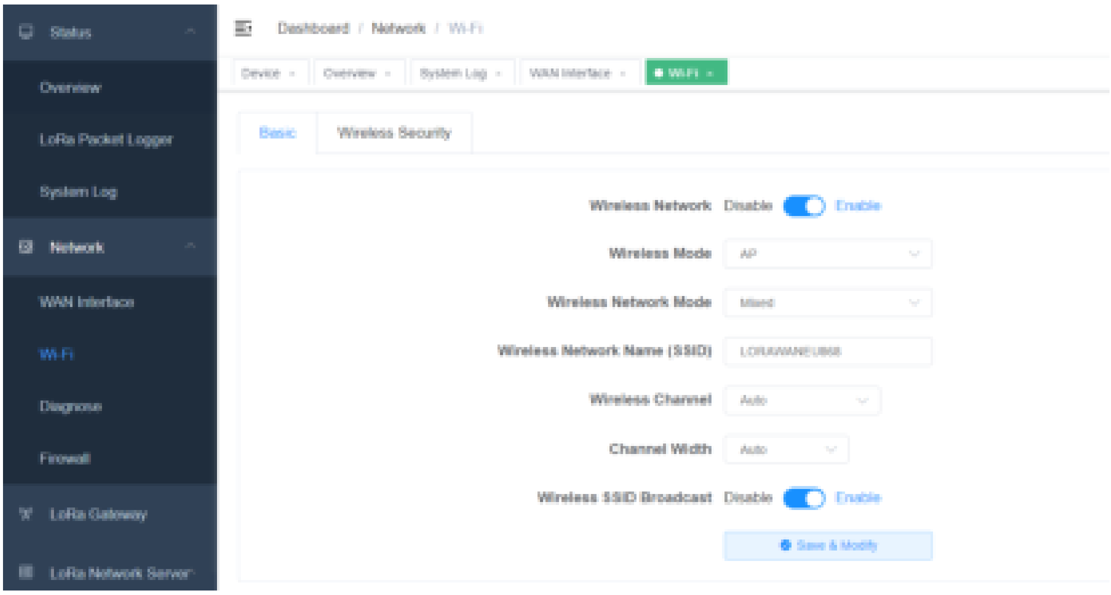
Connection Type: Automatic Configuration - DHCP

Wan Nat: Disable [✓ Enable]

STP: [✓ Disable] Enable

2. WIFI

- ➢ **Path:** Network → WiFi
- ➢ **Function:** wifi Parameter Configuration、Security Configuration
- ➢ **Detail:**
  - ✧ Configure various modes based on the mode parameters.
- ➢ **Preview:**



Wireless Network: Disable ● Enable

Wireless Mode: AP

Wireless Network Mode: Mixed

Wireless Network Name (SSID): LORAWANEU868

Wireless Channel: Auto

Channel Width: Auto

Wireless SSID Broadcast: Disable ● Enable

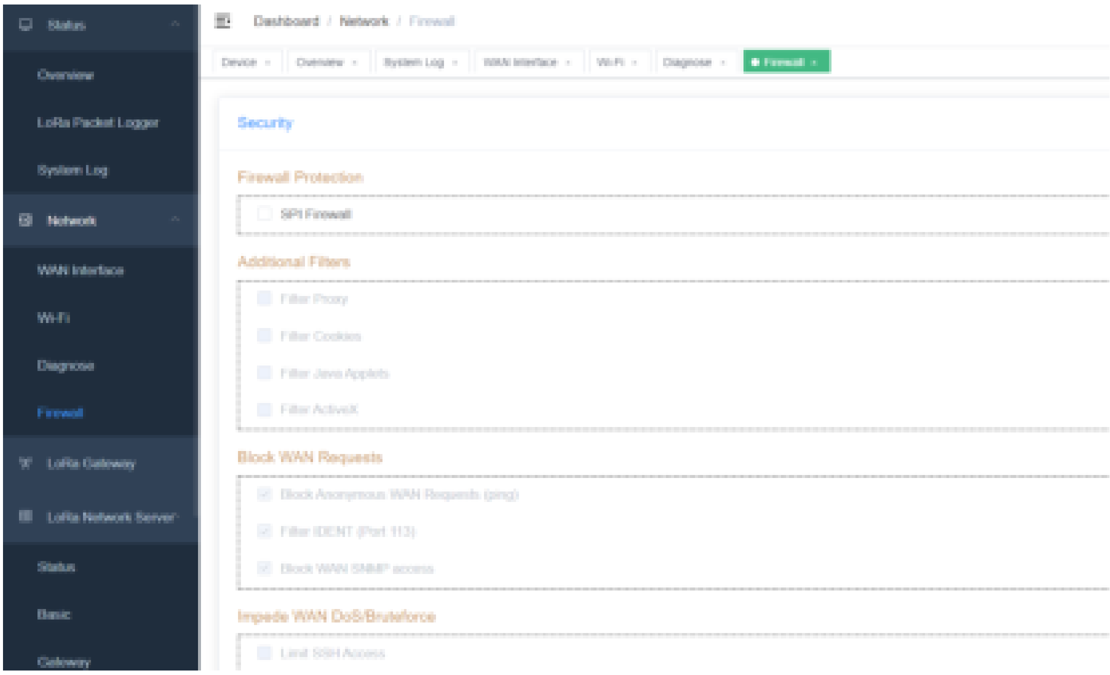
3. Network Diagnosis

- **Path**: Network → Network Diagnosis
- **Function**: Support Ping、Traceroute、NsLookup Commands
- **Details**:
  - Ping: A program used to test network connectivity.
  - Traceroute: The command uses the ICMP protocol to trace all the routers between your computer and the target computer.
  - NsLookup: It is a command-line tool for monitoring whether DNS servers in the network can perform proper domain name resolution.
- **Preview**:

4. Firewall

- **Path**: Network → Firewall
- **Function**: Configuration of Firewall Parameters
- **Details**:
  - Configure parameters according to the page display.

- ➢ **Preview:**



Security

Firewall Protection

□ SPI Firewall

Additional Filters

□ Filter Proxy

□ Filter Cookies

□ Filter Java Applets

□ Filter ActiveX

Block WAN Requests

☑ Block Anonymous WAN Requests (ping)

☑ Filter IDENT (Port 113)

☑ Block WAN SNMP access

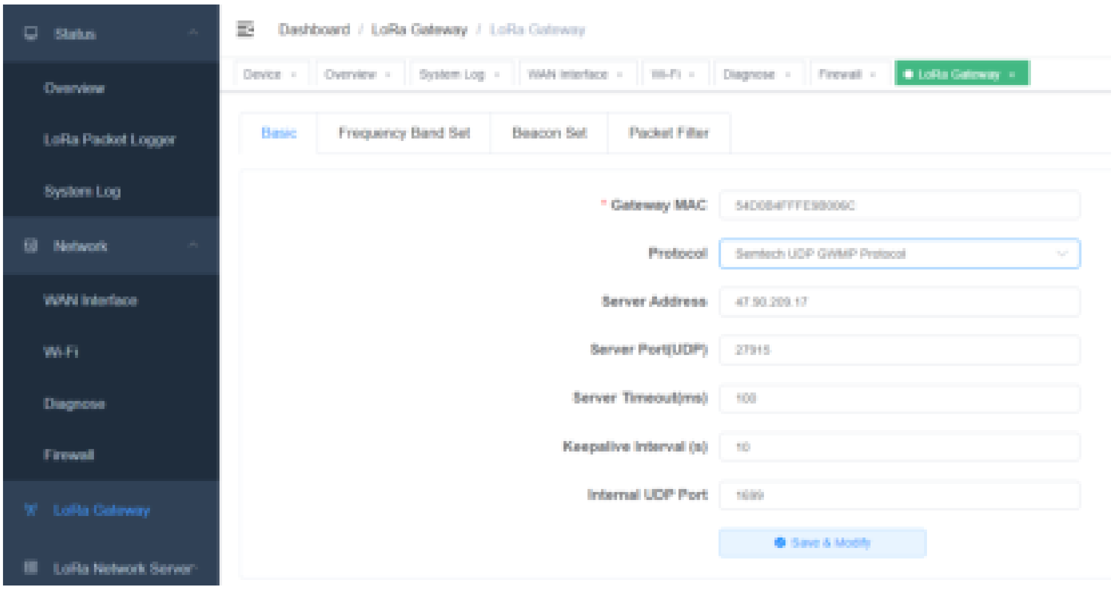
Impede WAN DoS/Bruteforce

□ Limit SSH Access

4.1.3.2 LoRa Gateway

1. Basic Settings

- ➢ **Path:** LoRa Gateway → Basic Settings
- ➢ **Function:** Gateway Protocol Configuration, which can be configured as Build-in LoRa Server, Semtech UDP GWMP Protocol, Basics Station mode.
- ➢ **Details:**
  - ✧ Semtech UDP GWMP Protocol – GWMP Forwarding Mode
    - ■ Gateway MAC: Gateway's unique identifier, with a length of 8 bytes (16 characters), typically not modified.
    - ■ Protocol: UDP GWMP Protocol, which connects 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 Time (milliseconds): The timeout duration for waiting for acknowledgment when sending data reports. Generally, no modifications are needed.
    - ■ Keepalive Interval (seconds): The interval duration for the "pull_data" command in the protocol. Generally, no modifications are needed.
    - ■ Internal UDP Communication Port: In the case of gateway cascading applications, this port number, configured as the server port for the gateway, should match the server port of the sub-gateway.



* Gateway MAC: 54D0B4FFFE98006C

Protocol: Semtech UDP GWMP Protocol

Server Address: 47.90.209.17

Server Port(UDP): 27915

Server Timeout(ms): 100

Keepalive Interval (s): 10

Internal UDP Port: 1699

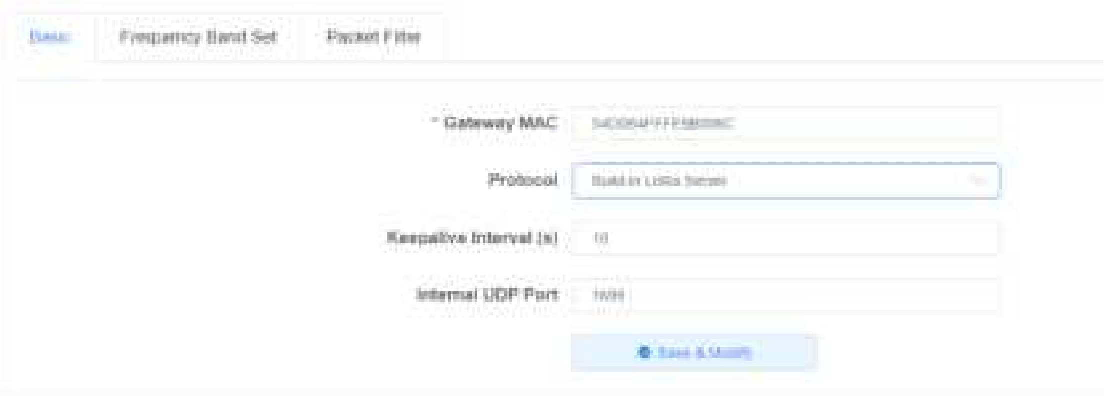
✧ Build-in LoRa Server - Built-in NS Mode

■ Gateway MAC：Gateway's unique identifier, which is 8 bytes in length (16 bits), is usually not modified.

■ Protocol: Built-in NS mode, equivalent to deploying NS inside the gateway.

■ Keepalive Interval(s): The interval time for the pull_data command in the protocol, usually not modified.

■ Internal UDP Communication Port: In gateway cascading applications, configure this port number as the server port of the sub-gateway.



* Gateway MAC: 54D0B4FFFE98006C

Protocol: Build in LoRa Server

Keepalive Interval (s): 10

Internal UDP Port: 1699

✧ Basics Station - More secure and reliable protocols (via WebSocket or HTTP) are used to connect to the Network Server (NS)

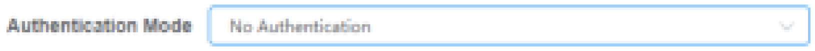
■ Gateway MAC: The unique identifier of the gateway, with a length of 8 bytes (16 characters). Generally, it is not modified.

■ Protocol：Basicstation Mode

■ Server：LNS Protocol (for regular data communication) or CUPS Protocol (for adding gateway upgrade-related protocols).

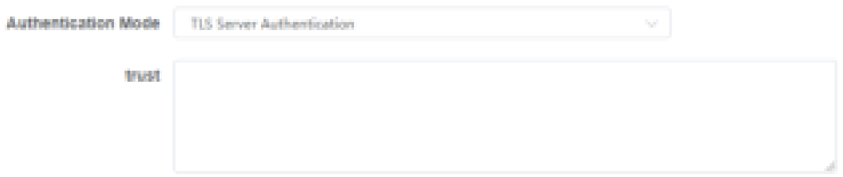
- URI：Server Address (IP or domain name) for connection.
- Port：Server corresponding port.
- Authentication Mode：Security authentication mode (detailed introduction of various modes' application scenarios will be provided later), below is a brief introduction of each mode:

◆ No Authentication：Establish regular WebSocket or HTTP connections without the need for authentication (e.g. ChirpStack configured in this mode and integrated with TTN platform).



Authentication Mode: No Authentication

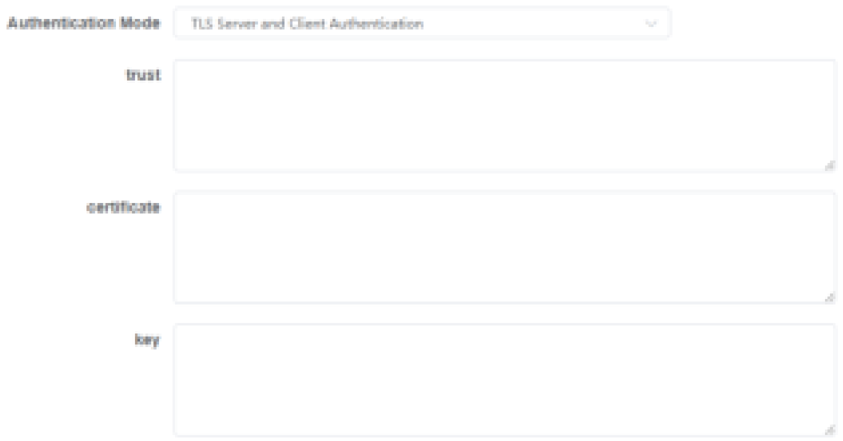
◆ TLS Server Authentication：TLS server identity authentication is achieved by establishing TLS connections (wss, https) to authenticate the server (LNS or CUPS) (e.g. ChirpStack configured in this mode).



Authentication Mode: TLS Server Authentication

trust

◆ TLS Server and Client Authentication：TLS server and client identity authentication is achieved by the gateway establishing TLS connections (wss, https) to authenticate the server (LNS or CUPS), and the server verifies the gateway by requesting its certificate and a signature with a private key (e.g. when integrating with the AWS platform).



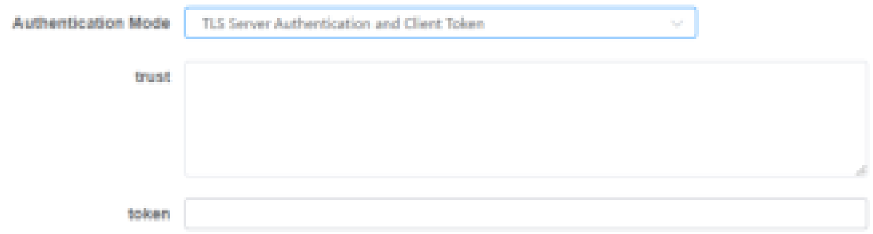
Authentication Mode: TLS Server and Client Authentication

trust

certificate

key

◆ TLS Server Authentication and Client Token：The gateway authenticates the server (LNS or CUPS) by establishing a TLS connection (wss, https), and the server verifies the gateway's identity by inspecting the secure token provided by the gateway (e.g. when integrating with TTN platform).

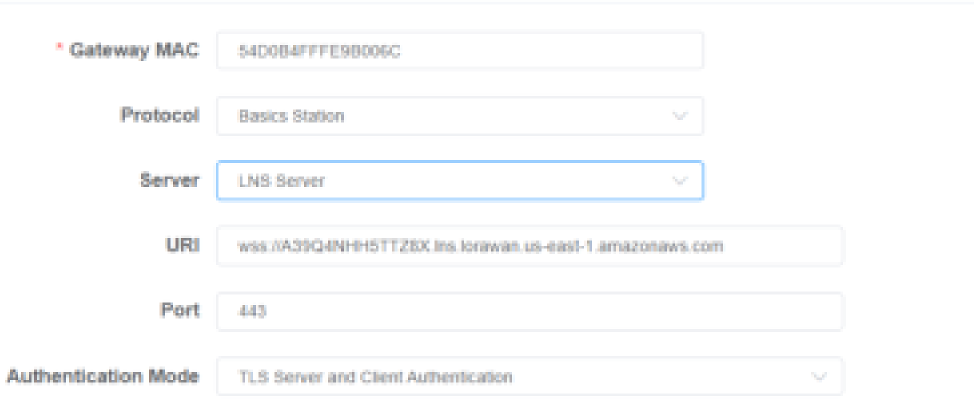


Authentication Mode: TLS Server Authentication and Client Token

trust

token

- **Preview:**



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

Server: LNS Server

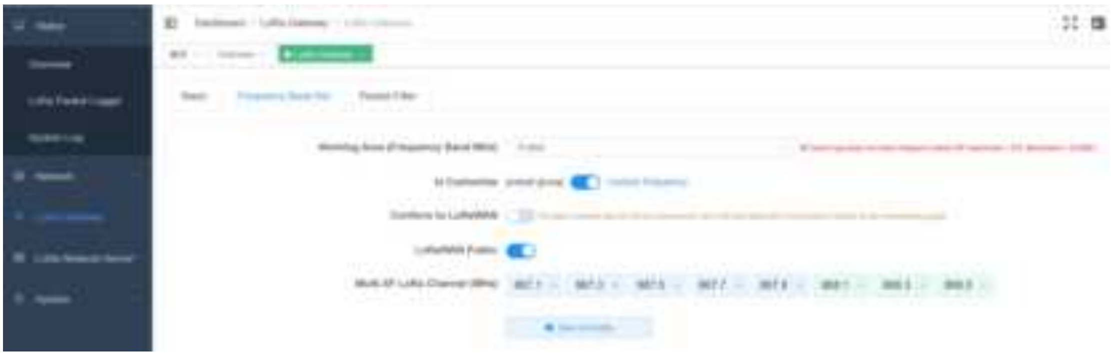
URI: wss://A39Q4NHH5TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

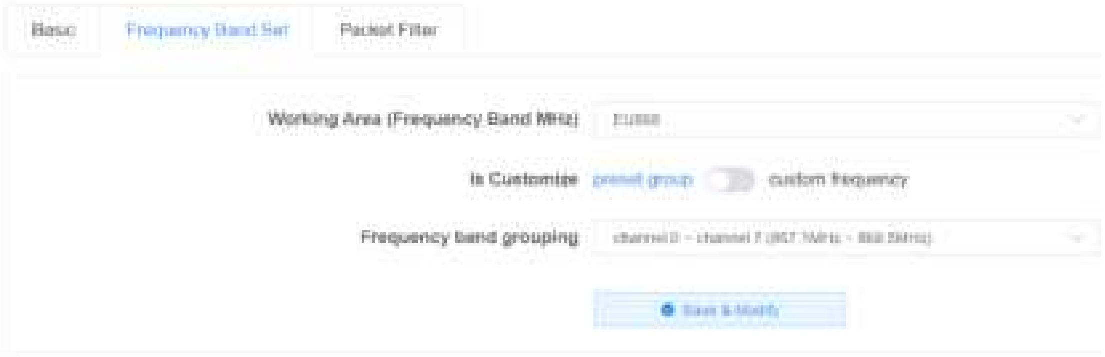
Authentication Mode: TLS Server and Client Authentication

2. Frequency Band Configuration

- **Path:** LoRa Gateway → Frequency Band Settings
- **Function:** Configuration of Gateway Frequencies, applicable to modes: Semtech UDP GWMP Protocol or Build-in LoRa Server. For Basics Station mode, frequency settings are configured in the NS server.
- **Details:**
  - ✧ Frequency configuration is primarily supported through three methods:
    - ■ Custom Frequency Mode: This method provides a straightforward way to visualize the allocated frequency points, as shown in the diagram below. The left-colored frequency points (e.g. 867.1) are deletable, while the right-colored frequency points (e.g. 868.1) are essential and cannot be removed as they represent mandatory fields for the frequency band. To delete a frequency point, simply click the "×" icon next to it. To add a new frequency point, click the "+ Add" button on the far right.



    - ■ Preset Group Mode: This method is the most convenient. Based on your needs, you can select the corresponding preset group. The displayed frequency points represent the starting frequency and ending frequency, with a general interval of 0.2MHz between them. There are a total of 8 frequency points in this preset group.



Basic | Frequency Band Set | Packet Filter

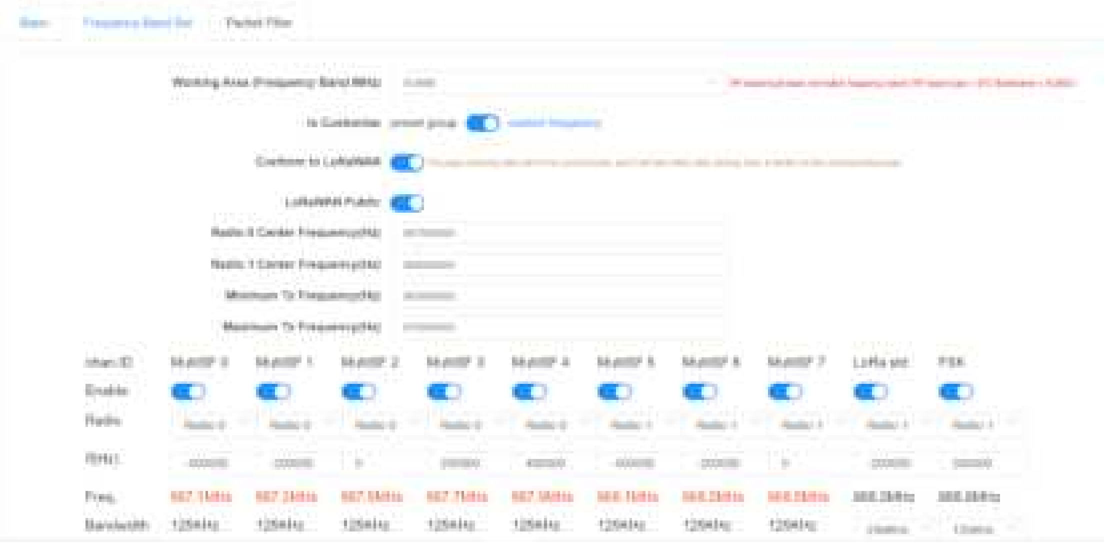
Working Area (Frequency Band MHz): EU868

Is Customize: preset group ○ custom frequency

Frequency band grouping: channel 0 – channel 7 (867.1MHz – 868.5MHz)

Save & Modify

- Custom Frequency Points + Conform to LoRaWAN Mode: This method aligns closely with the gateway's configuration file structure and is the most comprehensive configuration approach. When the other two methods cannot meet the requirements, this method should be used for configuration.



Basic | Frequency Band Set | Packet Filter

Working Area (Frequency Band MHz): EU868 [illegible]

Is Customize: preset group ● custom frequency

Conform to LoRaWAN: ● [illegible]

LoRaWAN Public: ●

Radio 0 Center Frequency(Hz): [illegible]

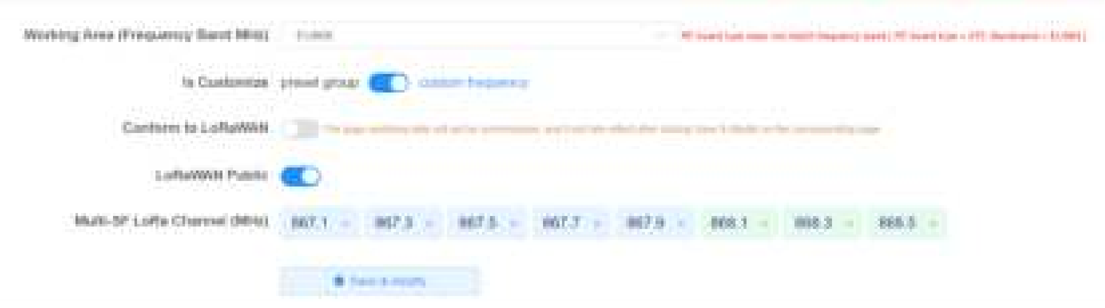
Radio 1 Center Frequency(Hz): [illegible]

Minimum Tx Frequency(Hz): [illegible]

Maximum Tx Frequency(Hz): [illegible]

chan ID	MultiSF 0	MultiSF 1	MultiSF 2	MultiSF 3	MultiSF 4	MultiSF 5	MultiSF 6	MultiSF 7	LoRa std	FSK
Enable	●	●	●	●	●	●	●	●	●	●
Radio	Radio 0	Radio 0	Radio 0	Radio 0	Radio 0	Radio 1	Radio 1	Radio 1	Radio 1	Radio 1
IF(Hz)	-400000	-200000	0	200000	400000	-400000	-200000	0	[illegible]	[illegible]
Freq.	867.1MHz	867.3MHz	867.5MHz	867.7MHz	867.9MHz	868.1MHz	868.3MHz	868.5MHz	[illegible]	[illegible]
Bandwidth	125kHz	125kHz	125kHz	125kHz	125kHz	125kHz	125kHz	125kHz	[illegible]	[illegible]

➢ **Preview:**



Working Area (Frequency Band MHz): EU868 [illegible]

Is Customize: preset group ● custom frequency

Conform to LoRaWAN: ○ [illegible]

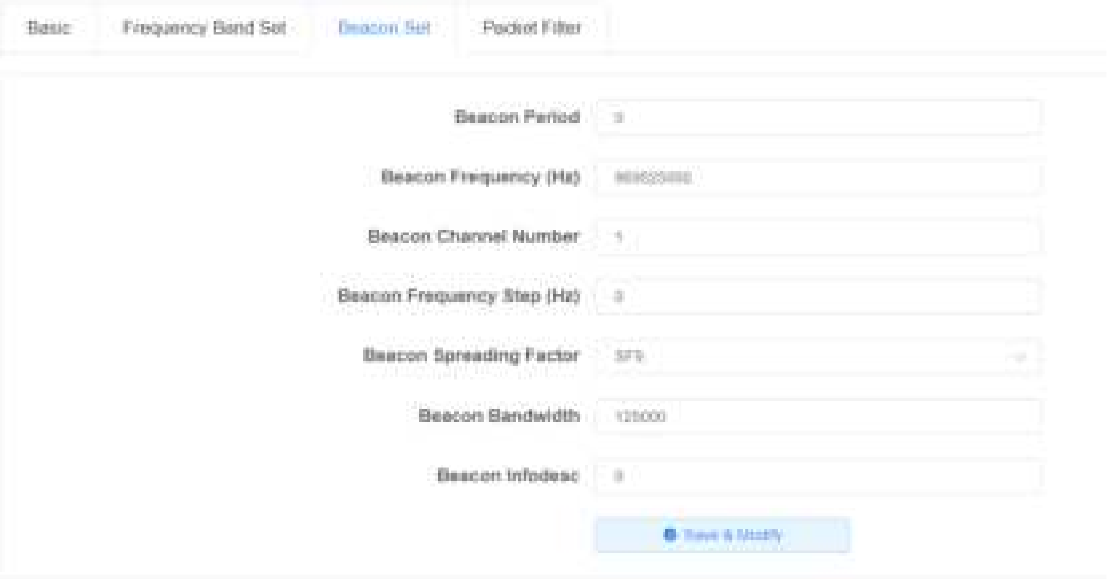
LoRaWAN Public: ●

Multi-SF LoRa Channel (MHz): 867.1 × | 867.3 × | 867.5 × | 867.7 × | 867.9 × | 868.1 × | 868.3 × | 868.5 ×

Save & Modify

3. Beacon Set

- ➢ **Path:** LoRa Gateway → Beacon Set
- ➢ **Function:** Configuring the ClassB parameters of the gateway can be done using the Semtech UDP GWMP Protocol mode.
- ➢ **Details:**
  - ✧ Beacon Period: Period, when set to 0, indicates that it is turned off.
  - ✧ Beacon Frequency (Hz): Frequency Point
  - ✧ Beacon Spreading Factor: Spreading Factor
  - ✧ Beacon Bandwidth: Beacon Packet Bandwidth
  - ✧ Beacon Tx Power: Transmission Power
- ➢ **Preview:**



Beacon Period: 0
Beacon Frequency (Hz): 869525000
Beacon Channel Number: 1
Beacon Frequency Step (Hz): 0
Beacon Spreading Factor: SF9
Beacon Bandwidth: 125000
Beacon Infodesc: 0

4. Packet Filter

- ➢ **Path:** LoRa Gateway → Packet Filter
- ➢ **Function:** At the gateway side, packets can be filtered based on configured rules to reduce the amount of irrelevant data transmitted to the NS server, thus easing the processing load on the NS. This can be configured using the Semtech UDP GWMP Protocol or Build-in LoRa Server modes.
- ➢ **Details:**
  - ✧ Supports configuring NetID and JoinEUI.
  - ✧ NetID: Network ID filtering, the short address portion allocated during device joining is related to the network ID. By configuring this value, non-joining interference data can be effectively filtered out. In the embedded mode, this value can be configured to the network ID configured for this gateway, thereby effectively avoiding interference from other devices' data.
  - ✧ JoinEUI（AppEUI）: JoinEUI filtering, a component of the device's triplet, can be configured with multiple sets of range values here. Once set, JoinEUI values outside the specified ranges will be filtered.
- ➢ **Preview:**

4.1.3.3 LoRa Network Server

1. Status

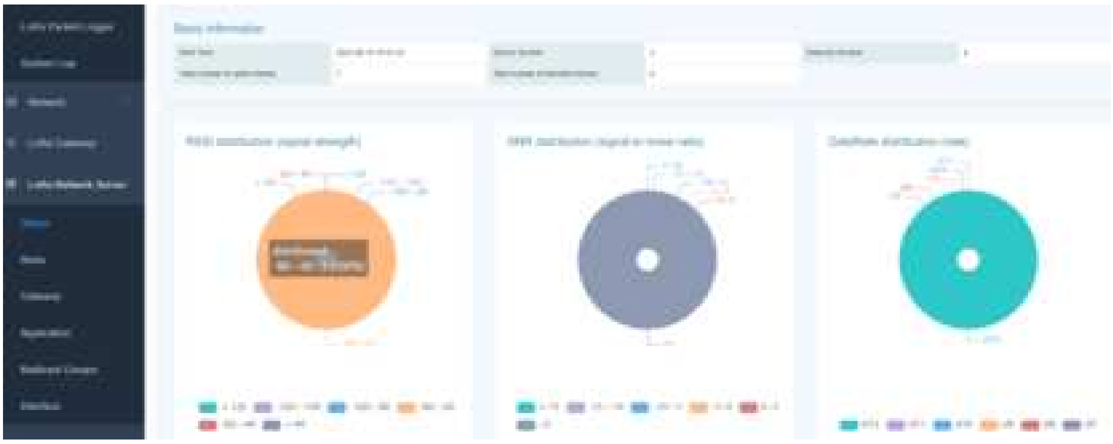
- **Path：** LoRa Network Server → Status
- **Function：** Display statistics of flow to the embedded NS server.
- **Details：**

✧ Basic Information: Mainly includes the number of gateways, the number of devices, and the statistics of device uplink and downlink data counts.

✧ This can be used to analyze the communication quality between gateways and nodes, based on the distribution of RSSI (Received Signal Strength Indicator), SNR (Signal-to-Noise Ratio), and DataRate values.

✧ Communication Distribution: This displays the curve chart of uplink and downlink communication situations, allowing analysis of whether the distribution of uplink and downlink data matches the expected patterns.

- **Preview：**



2. Basic Setting

- **Path**：LoRa Network Server → Basic Setting
- **Function**：Configuring NS Server Parameters
- **Details**：

✧ Operating Region: Corresponds to the frequency band in the regional parameter table. This setting cannot be configured here and should remain consistent with the configuration in LoRa Gateway → Frequency Band Configuration → Operating Region.

✧ Enable Dynamic Data Rate Adjustment (ADR): Determines whether the Adaptive Data Rate (ADR) feature is enabled.

✧ ADR Margin: This value affects the sensitivity of ADR adjustment. A larger value makes the adjustment less aggressive, while a smaller value makes the adjustment more aggressive.

✧ Minimum Data Rate: The lowest data rate used for ADR adjustment.

✧ Maximum Data Rate: The highest data rate used for ADR adjustment.

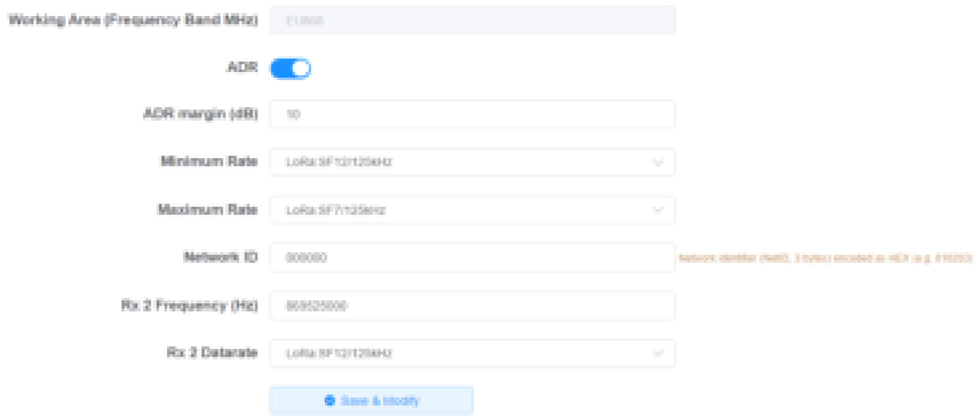
✧ Network ID: A parameter used to generate the device's short address. It can be configured in the filtering parameters to avoid interference.

✧ Rx2 Frequency：Frequency of rx2 Window

✧ Rx2 Datarate：Datarate of rx2 Window

✧ Downlink Transmit Power (dBm): The transmit power configuration for downlink messages. When set to -1, it will follow the transmit power specifications defined in the regional parameters table.

- **Preview**：



Working Area (Frequency Band MHz): EU868

ADR: ●

ADR margin (dB): 10

Minimum Rate: LoRa SF12/125kHz

Maximum Rate: LoRa SF7/125kHz

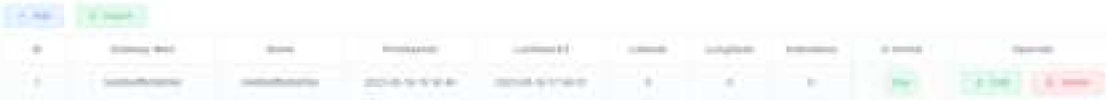
Network ID: 000000 — Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203)

Rx 2 Frequency (Hz): 869525000

Rx 2 Datarate: LoRa SF12/125kHz

3. Gateway

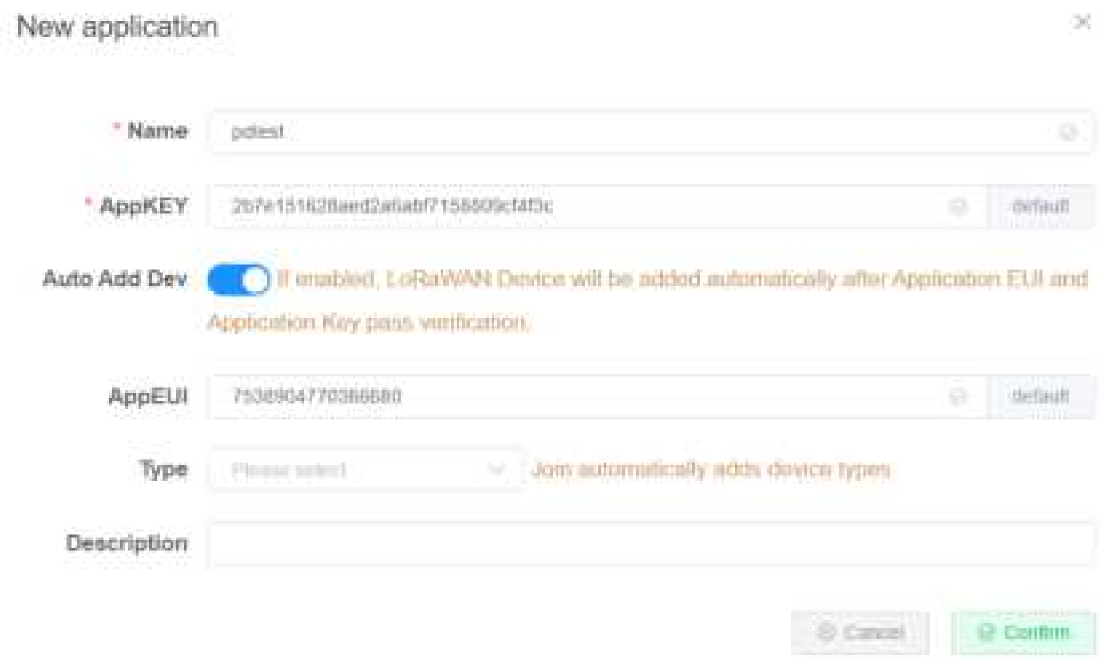
- **Path:** LoRa Network Server → Gateway
- **Function:** Gateway Addition, Modification, and Deletion in Embedded NS: Gateways are usually automatically added when they connect and typically do not require manual addition.
- **Details:**
  - ✧ Displaying the Gateway List: The list shows detailed information for each gateway, including online status and more.
- **Preview:**



4. Application

- **Path:** LoRa Network Server → Application
- **Function:** It is equivalent to the grouping function, where different groups correspond to different application scenarios for easier management.
- **Details:**
  - ✧ Add Application: After clicking the "Add" button, the following page will open.
    - ■ Name: This is equivalent to the group name, just for identification purposes.
    - ■ AppKEY: The AppKEY corresponding to the terminal, which is used to verify the value when adding devices automatically (clicking on the right side of "default" will change it to the default value of Four-Faith).
    - ■ Auto-Add Devices: When selected, devices can be added automatically without the need to add them in advance. Once the AppKEY and AppEUI validation is successful, devices will be added automatically.
    - ■ AppEUI (JoinEUI): One of the triplets required for device configuration. When enabling auto-add devices, this needs to be configured (clicking on the "default" option will change it to the default value provided by Four-Faith).

- Type: The device type corresponding to auto-added devices, either ClassA or ClassC.
- Description: Description information.



New application

* Name: pdtest

* AppKEY: 2b7e151628aed2a6abf7158809cf4f3c

Auto Add Dev: If enabled, LoRaWAN Device will be added automatically after Application EUI and Application Key pass verification.

AppEUI: 7538904770386680

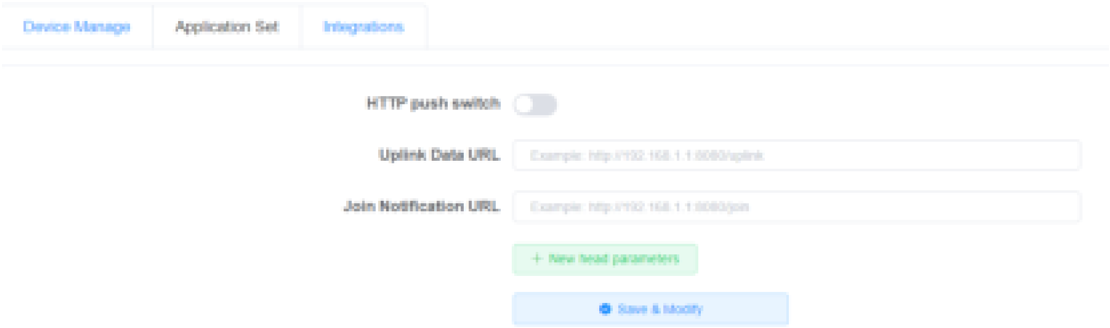
Type: Please select — Join automatically adds device types

Description:

✧ Delete: It is not possible to delete the application if devices are associated with it. You must delete the devices first before deleting the application.

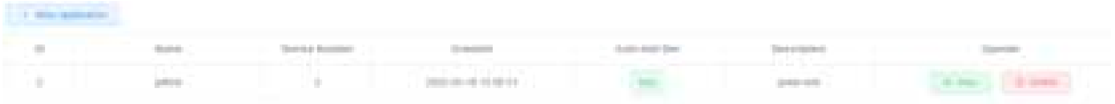
✧ View: Once inside the application, you can access the list of devices and other related information.

- Device Management: Detailed explanation of device addition, deletion, modification, and retrieval.
- Application Configuration: Similar to the creation process, here you can modify existing applications.
- Interface Management: Configure HTTP POST, when this feature is enabled, all data from devices under this application will be pushed to the specified address using the HTTP POST method.



HTTP push switch: off
Uplink Data URL: Example: http://192.168.1.1:8080/uplink
Join Notification URL: Example: http://192.168.1.1:8080/join

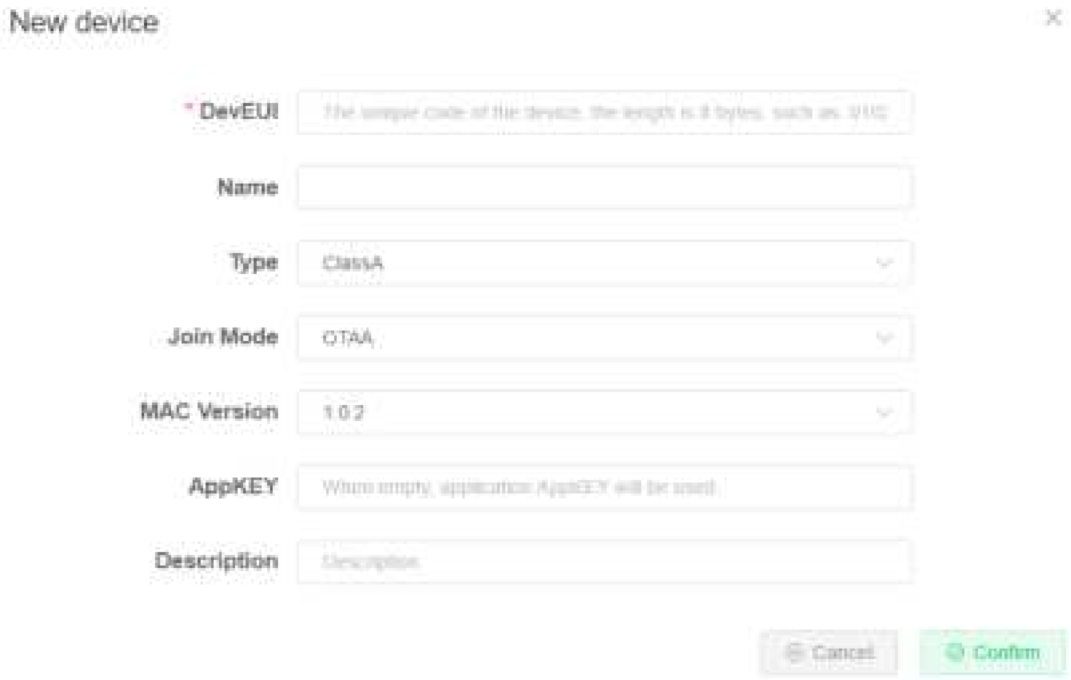
- ➢ **Preview:**



5. Devices

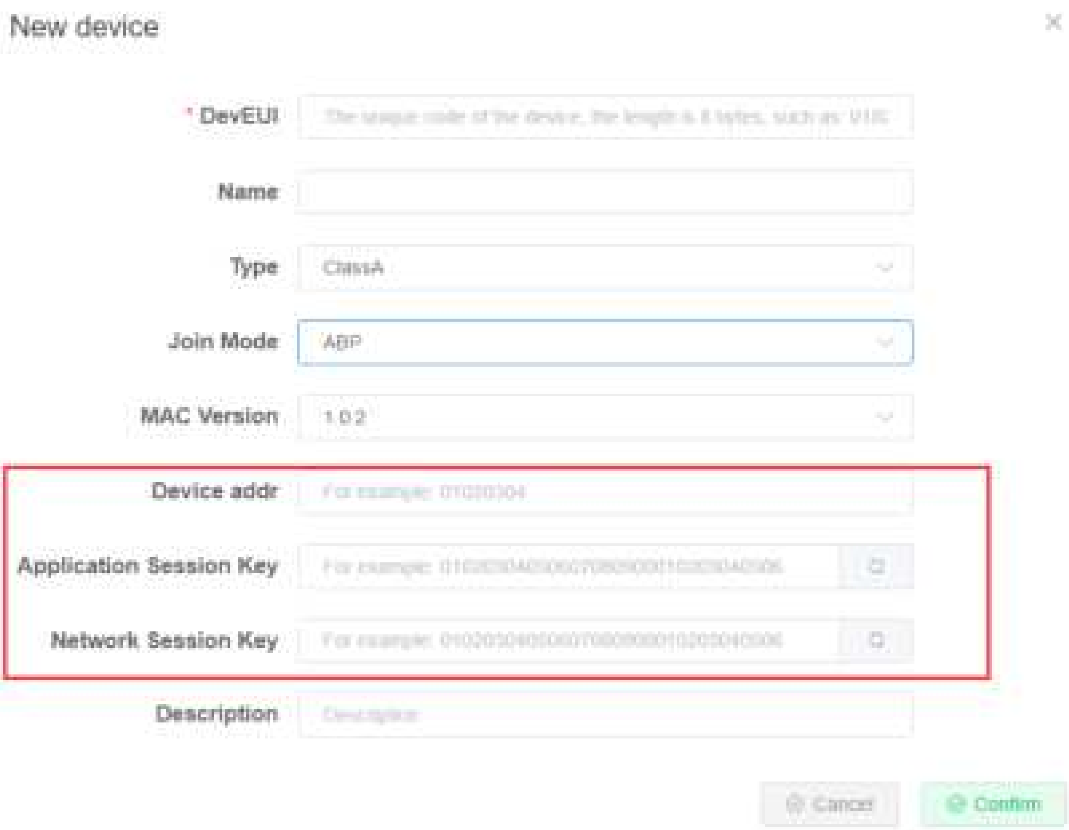
- ➢ **Path:** LoRa Network Server → Devices
- ➢ **Function:** Device Addition, Deletion, Modification, and Query: Web Entry: LoRa Network Server -> Applications -> View -> Device Management
- ➢ **Details:**

  ✧ Add New Device: Setting the Basic Parameters of the Device. Joining methods include OTAA (where the device initiates joining) or ABP (where no joining is required). When the AppKEY of this device is different from the AppKEY of the application, you can specify a specific AppKEY here.



* DevEUI: The unique code of the device, the length is 8 bytes, such as: 0102
Name:
Type: ClassA
Join Mode: OTAA
MAC Version: 1.0.2
AppKEY: When empty, application AppKEY will be used.
Description: Description

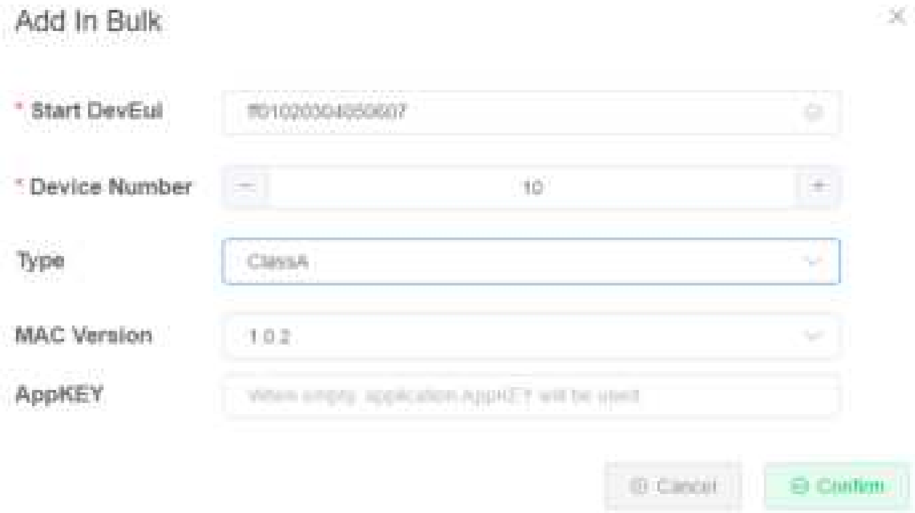
■ ABP Mode: In this mode, you need to input the Short Address and Session Keys information as shown in the boxes.



New device

* DevEUI: The unique code of the device, the length is 8 bytes, such as: V1IE
Name:
Type: ClassA
Join Mode: ABP
MAC Version: 1.0.2
Device addr: For example: 01020304
Application Session Key: For example: 01020304050607080900010203040506
Network Session Key: For example: 01020304050607080900010203040506
Description: Description

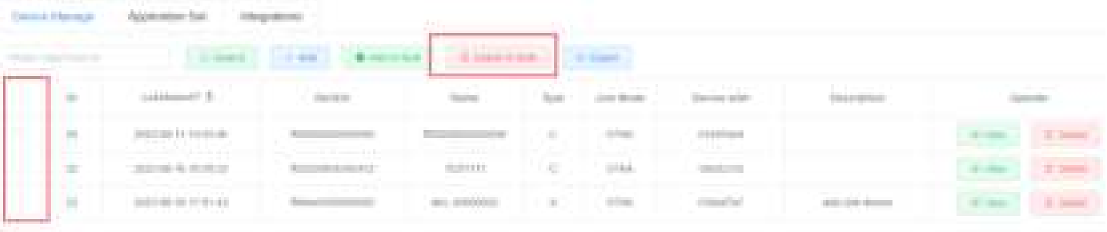
✧ Bulk Add: The parameters for bulk adding devices are similar to adding a single device. However, it's important to note that bulk adding can only be used for adding OTAA devices.



Add In Bulk

* Start DevEui: ff01020304050607
* Device Number: 10
Type: ClassA
MAC Version: 1.0.2
AppKEY: When empty, application AppKEY will be used

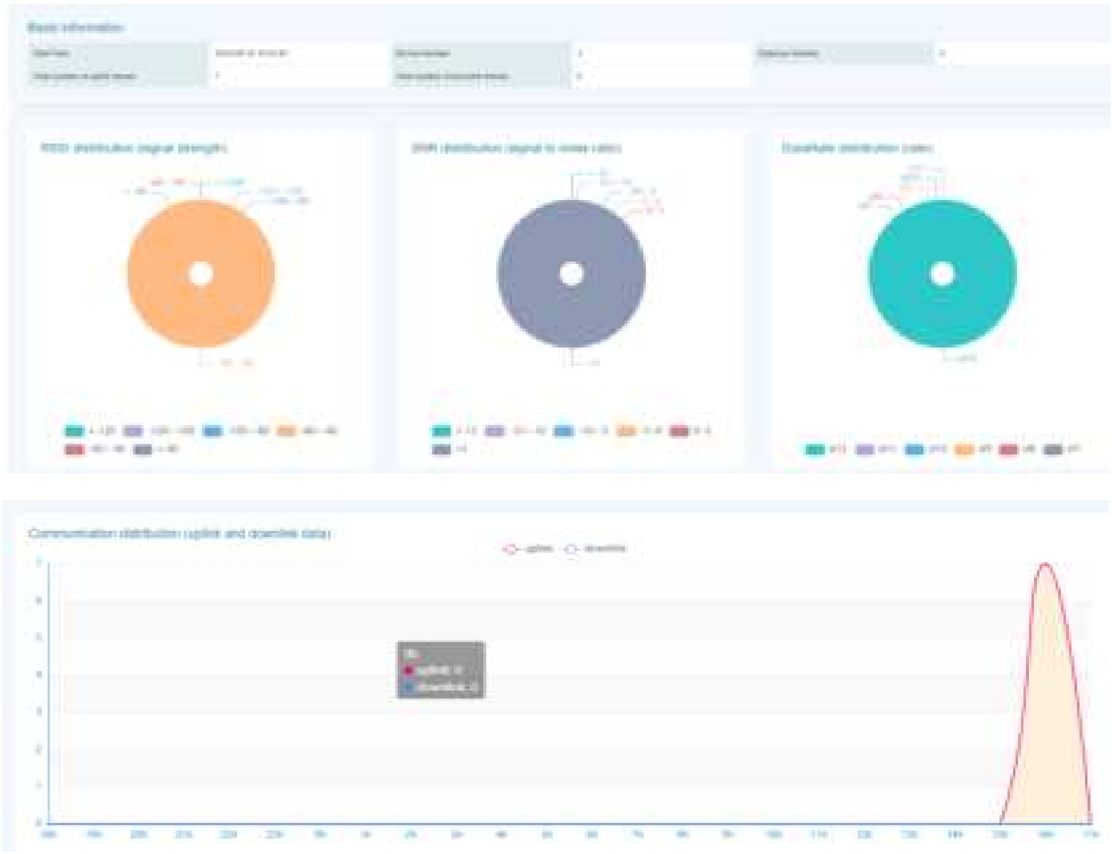
✧ Bulk Delete: To perform a bulk deletion, you need to first check the checkboxes next to the devices you want to delete, and then click on the "Bulk Delete" button.



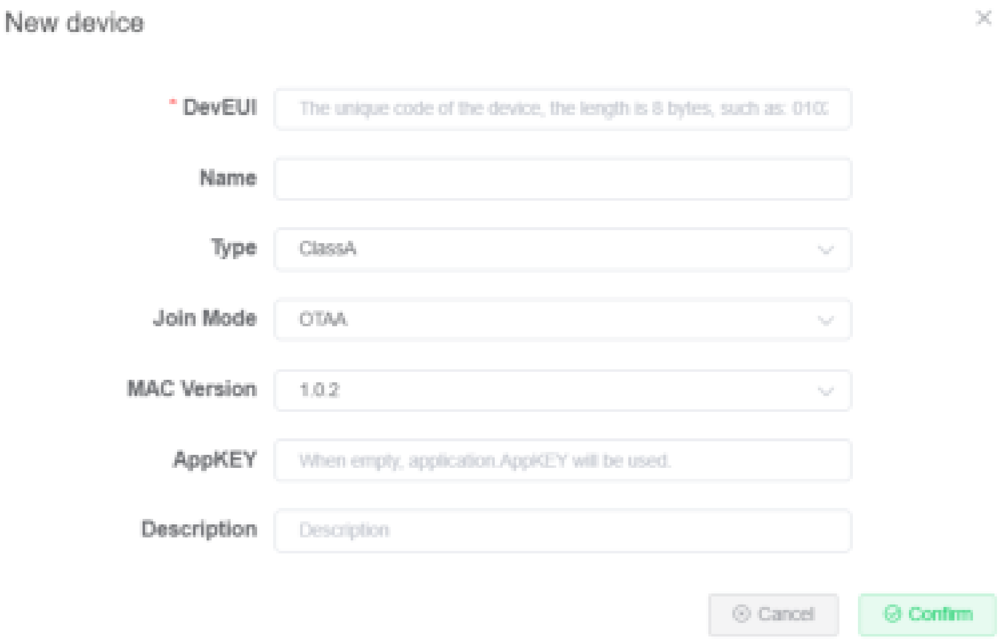
✧ Export: This option allows you to export the device information as an Excel spreadsheet, providing an easy way for backup and management.

✧ Device Details: By clicking on the "View" option next to the corresponding device, you can access the detailed information about that device.

- Overview: This section displays the device's uplink information and relevant statistical data. It can be used to analyze packet loss and other device communication issues.



- Configuration: Adjust parameters for the device.



New device

* DevEUI: The unique code of the device, the length is 8 bytes, such as: 010:

Name:

Type: ClassA

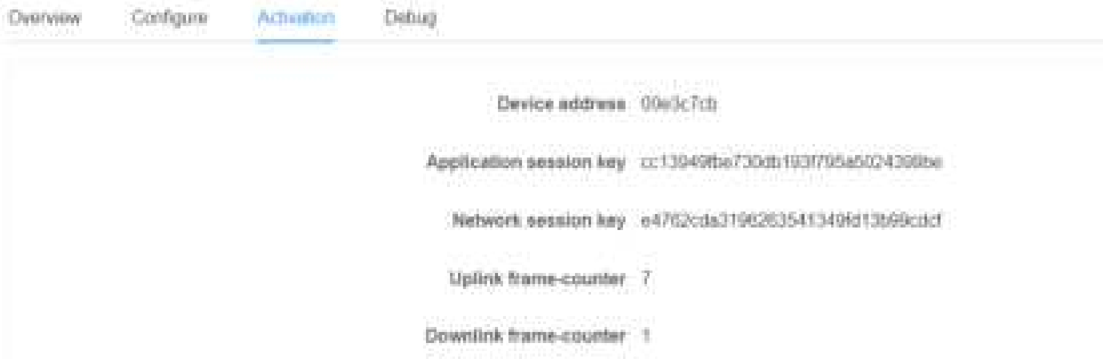
Join Mode: OTAA

MAC Version: 1.0.2

AppKEY: When empty, application AppKEY will be used.

Description: Description

- Activation Information: Display parameters after the device joins the network.



Device address: 00e3c7cb

Application session key: cc13949fbe730db193f795a5024398be

Network session key: e4762cda31962635413496d13b99cdcf

Uplink frame-counter: 7

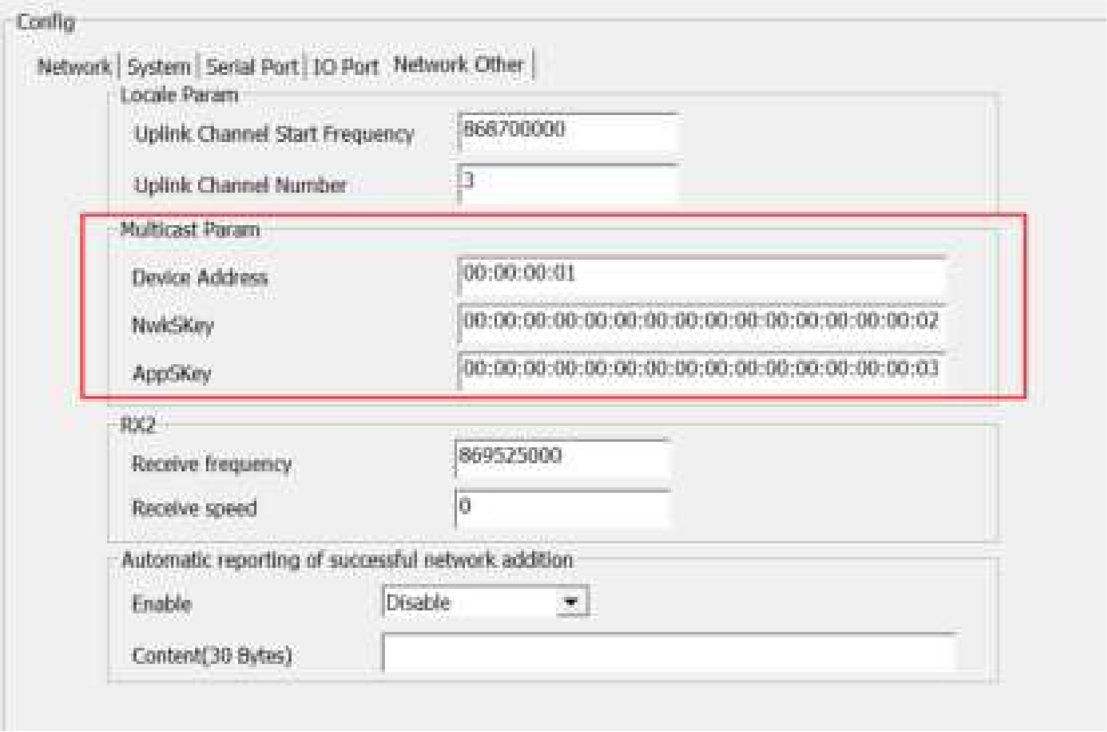
Downlink frame-counter: 1

- Online Debugging: Allows for data downlink (scheduled downlink) and displays uplink data for debugging purposes.

➢ **Preview:**

6. Multicast

- **Path：** LoRa Network Server → Multicast
- **Function：** In this context, multicast refers to the ability to send data to multiple devices with the same configuration parameters within the NS. Multicast data can be sent using MQTT, and you can also test sending multicast data through the web interface.
- **Details：**
  - ✧ Add Multicast: Below are the corresponding values for Four-Faith devices



Locale Param

Uplink Channel Start Frequency: 868700000

Uplink Channel Number: 3

Multicast Param

Device Address: 00:00:00:01

NwkSKey: 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:02

AppSKey: 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:03

RX2

Receive frequency: 869525000

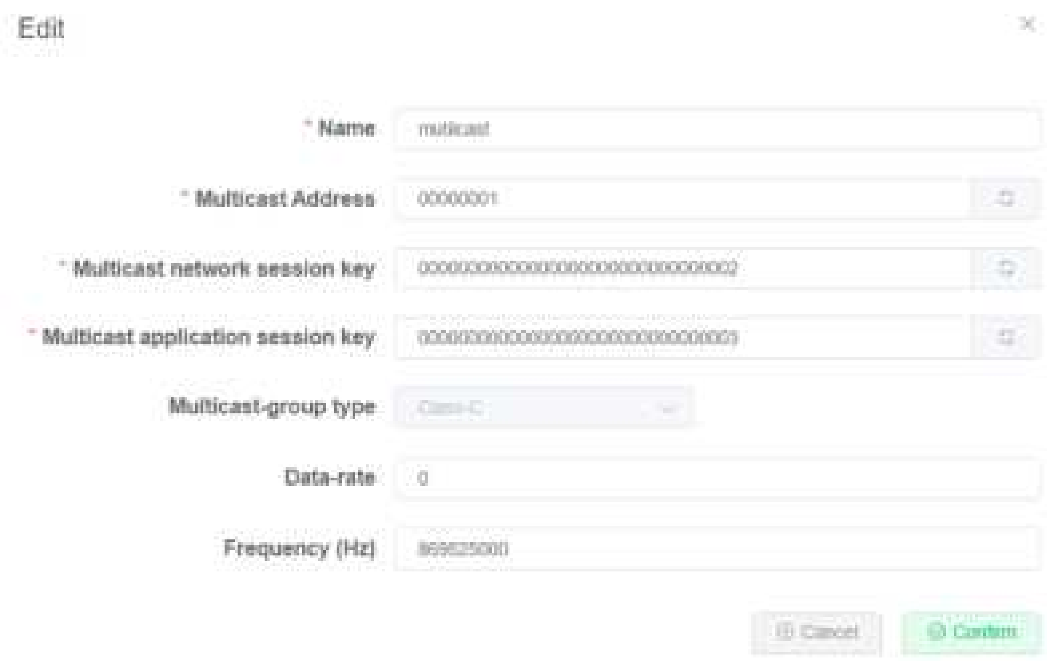
Receive speed: 0

Automatic reporting of successful network addition

Enable: Disable

Content(30 Bytes):

  - ✧ Based on the values provided above, configure the multicast parameters.



Edit

* Name: multicast

* Multicast Address: 00000001

* Multicast network session key: 00000000000000000000000000000002

* Multicast application session key: 00000000000000000000000000000003

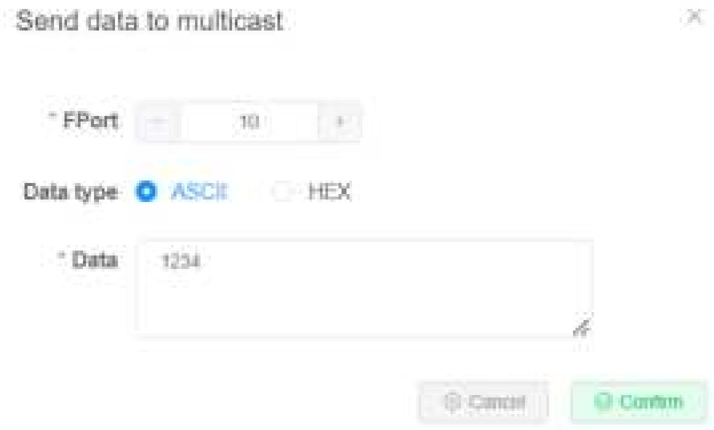
Multicast-group type: Class-C

Data-rate: 0

Frequency (Hz): 869525000

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

✧ Send Data Test: Click on "Downlink" to open the data sending page.

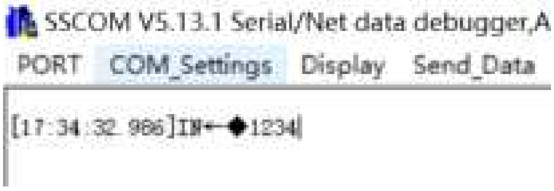


Send data to multicast

* FPort: 10

Data type: ● ASCII ○ HEX

* Data: 1234



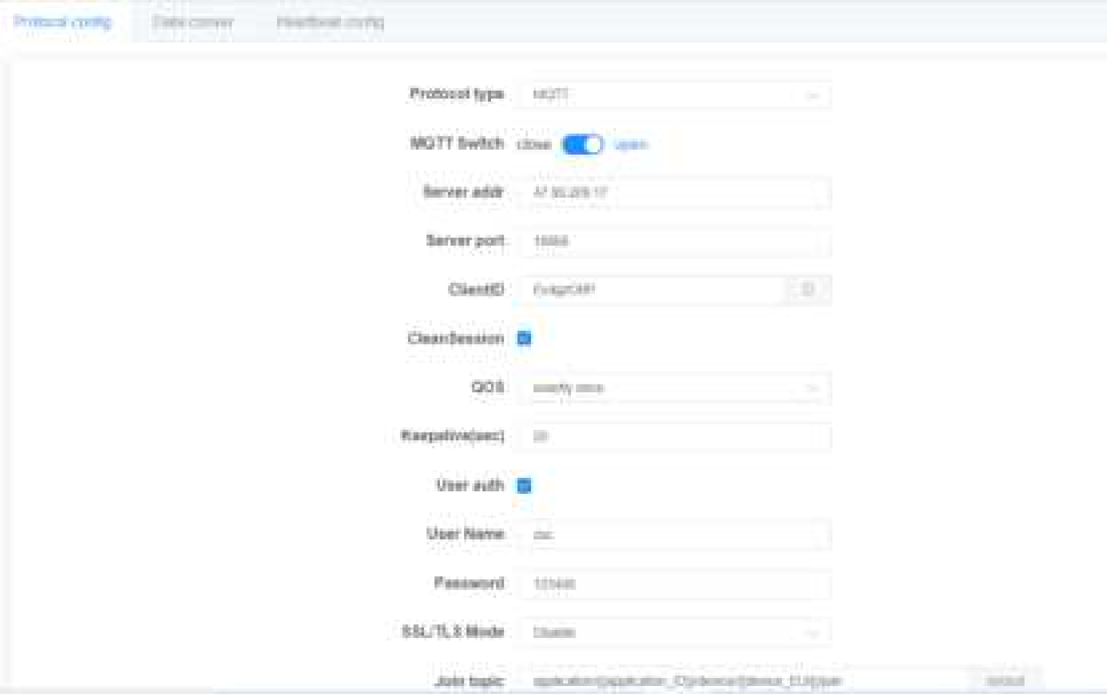
```
[17:34:32.986]IN←◆1234
```

✧ During actual usage, multicast data can be sent using MQTT or TCP. Please refer to the data format for more details.

7. Interface

➢ **Path:** LoRa Network Server → Interface

- ➢ **Function：** Configuration page for integrating the internal NS with a customer platform is available, supporting both MQTT and TCP communication methods. Data can be transformed using JavaScript functions, and heartbeat configurations are also supported.
- ➢ **Details：**
  - ✧ Protocol Configuration
    - ■ NONE： Not enabled.
    - ■ MQTT：MQTT parameter configuration, specific topics and data formats are detailed in the data format section.



Protocol config | Data conver | Heartbeat config

Protocol type: MQTT

MQTT Switch: close ● open

Server addr: [illegible]

Server port: [illegible]

ClientID: [illegible]

CleanSession: ☑

QOS: exactly once

Keepalive(sec): [illegible]

User auth: ☑

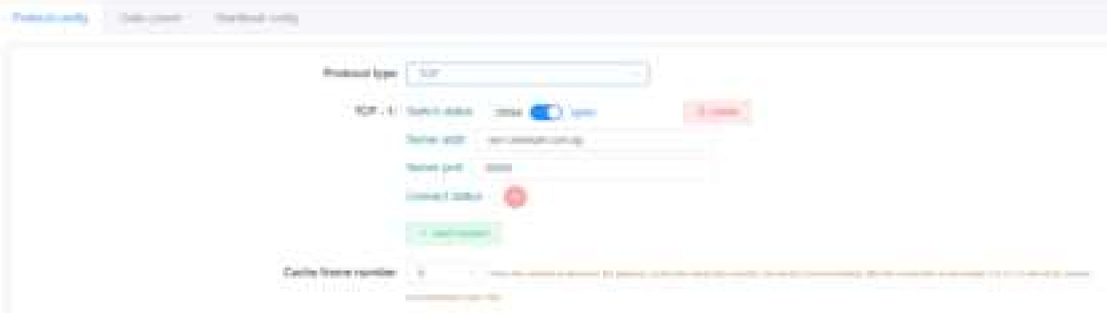
User Name: [illegible]

Password: 123456

SSL/TLS Mode: Disable

Join topic: application/{application_ID}/device/{device_EUI}/join

    - ■ TCP： Integrating with TCP servers allows for simultaneous connections to multiple servers, and the connection status can be used to determine the connection situation.



Protocol type: TCP

TCP - 1: Switch status close ● open

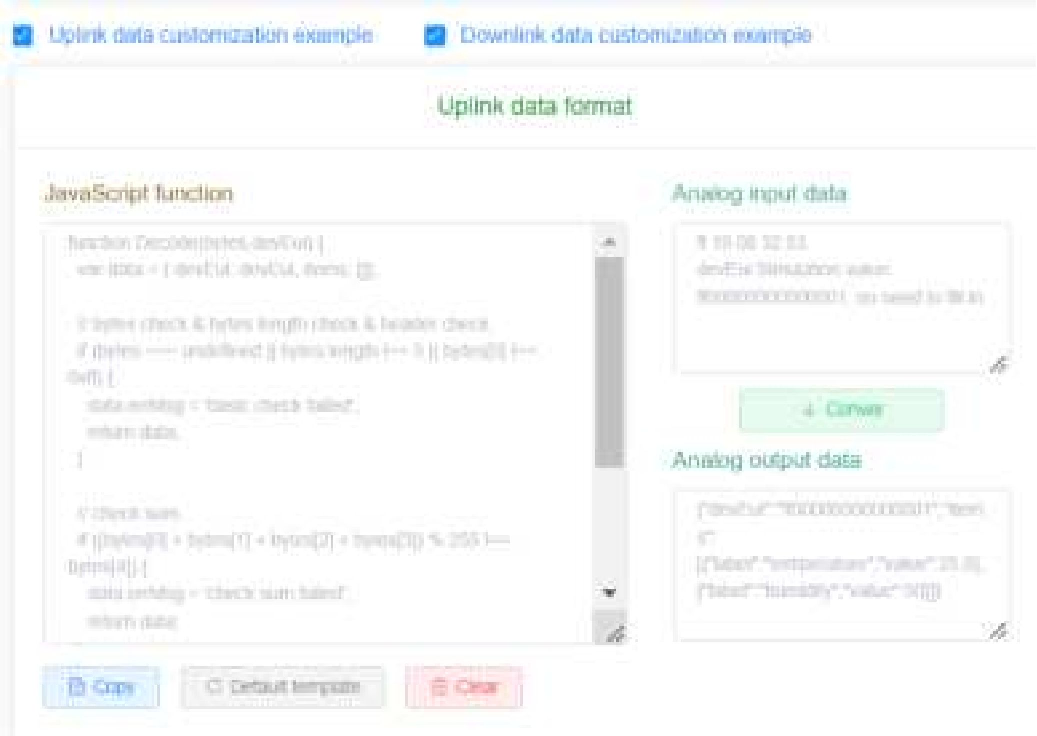
Server addr: [illegible]

Server port: [illegible]

Connect status: ●

Cache frame number: [illegible]

  - ✧ Data transformation: If no configuration is done here, the default data format will be used for communication. If you need to transform data, you can configure functions for the conversion. After uplink and downlink data reaches the gateway, it can be transformed using specified functions before forwarding.
    - ■ Uplink transformation

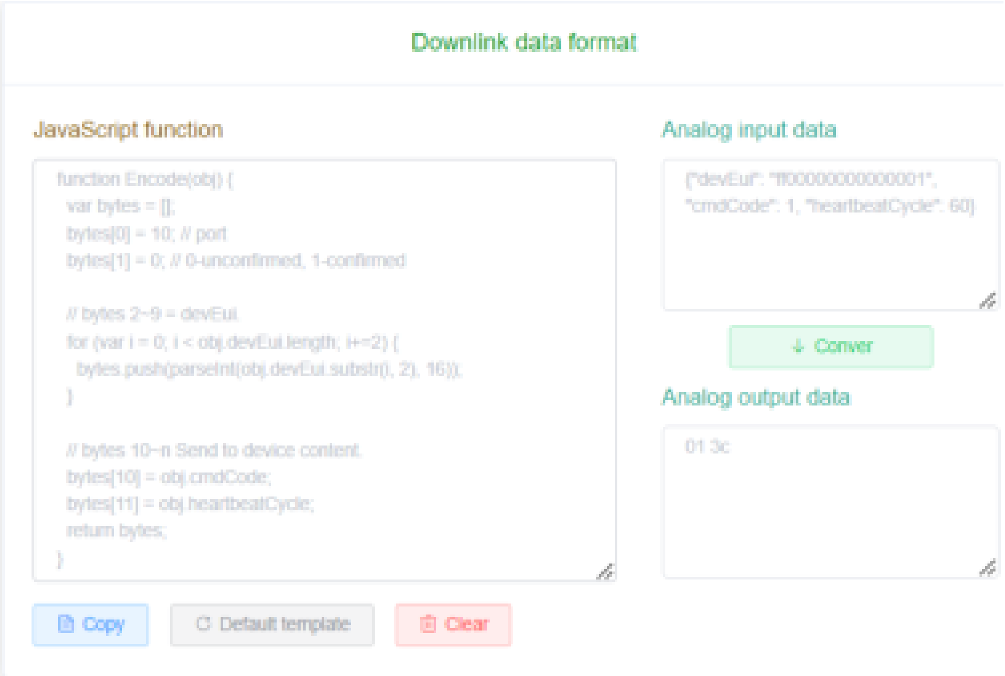


```
function Decode(bytes, devEui) {
  var data = { devEui: devEui, items: [] };

  // bytes check & bytes length check & header check
  if (bytes === undefined || bytes.length !== 5 || bytes[0] !==
0xff) {
    data.errMsg = 'basic check failed';
    return data;
  }

  // check sum
  if ((bytes[0] + bytes[1] + bytes[2] + bytes[3]) % 255 !==
bytes[4]) {
    data.errMsg = 'check sum failed';
    return data;
```

```
ff 19 08 32 53
devEui Simulation value:
ff00000000000001, no need to fill in
```

```
{"devEui":"ff00000000000001","item
s":
[{"label":"temperature","value":25.8},
{"label":"humidity","value":50}]}
```

■ Downlink transformation

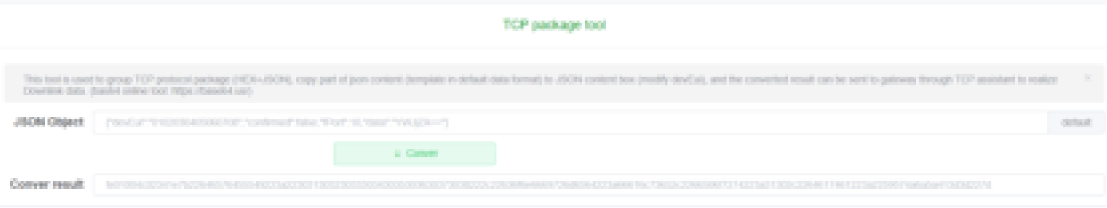


```
function Encode(obj) {
  var bytes = [];
  bytes[0] = 10; // port
  bytes[1] = 0; // 0-unconfirmed, 1-confirmed

  // bytes 2~9 = devEui
  for (var i = 0; i < obj.devEui.length; i+=2) {
    bytes.push(parseInt(obj.devEui.substr(i, 2), 16));
  }

  // bytes 10~n Send to device content
  bytes[10] = obj.cmdCode;
  bytes[11] = obj.heartbeatCycle;
  return bytes;
}
```

```
{"devEui": "ff00000000000001",
"cmdCode": 1, "heartbeatCycle": 60}
```

```
01 3c
```

■ TCP packet generation tool: During the testing phase of using TCP to connect to the server, you can use this tool to generate corresponding data for testing by sending it through the TCP server. In actual projects, you can write a program to generate the data.



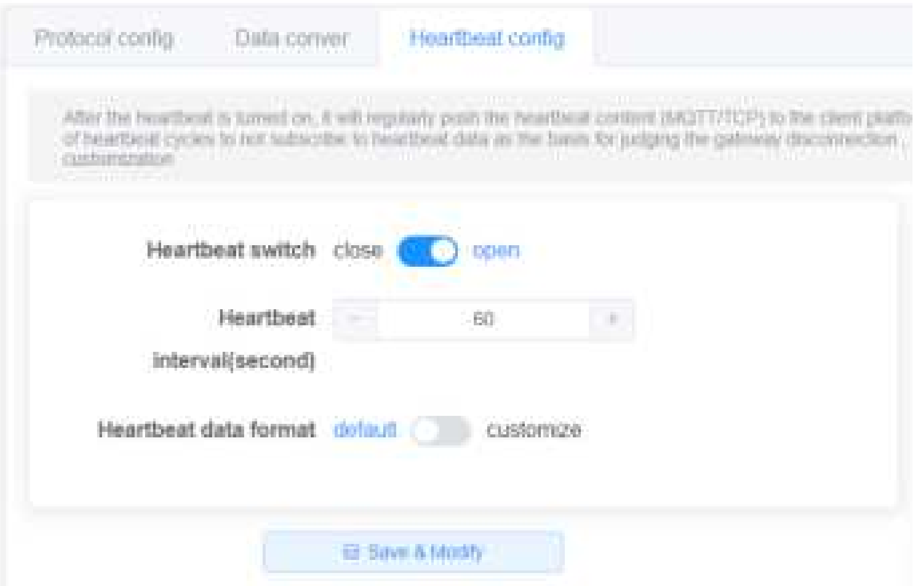
TCP package tool

This tool is used to group TCP protocol package (HEX+JSON), copy part of json content (template in default data format) to JSON content box (modify devEui), and the converted result can be sent to gateway through TCP assistant to realize Downlink data. (base64 online tool: https://base64.us)

JSON Object: [illegible]

Conver result: [illegible]

✧ Heartbeat Configuration: You can configure the heartbeat switch, heartbeat interval time, and heartbeat data format. It supports configuring a custom string as the heartbeat data. Heartbeat is mainly used to periodically report status information. The gateway can also use heartbeats to determine the connection status between itself and the MQTT server.



After the heartbeat is turned on, it will regularly push the heartbeat content (MQTT/TCP) to the client platfo[illegible] of heartbeat cycles to not subscribe to heartbeat data as the basis for judging the gateway disconnection [illegible] customization

Heartbeat switch: close ● open

Heartbeat interval(second): 60

Heartbeat data format: default ○ customize

4.1.3.4 System

1. System

➢ **Path：** System → System

➢ **Function：** View Program Version, Configure Token Duration, Time Settings and Language Switch

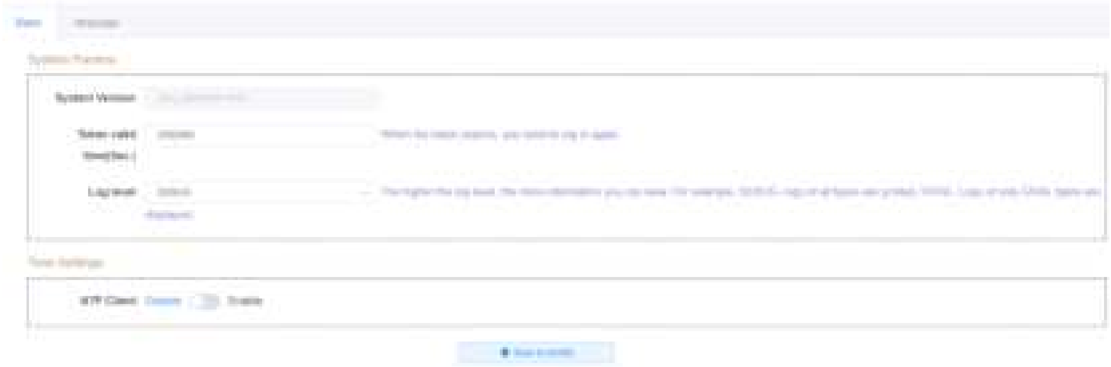
➢ **Details：**

✧ System Program Version: Use this to troubleshoot related issues by checking the version.

✧ Token Expiry Time: The shorter the time, the more frequent the need for webpage login.

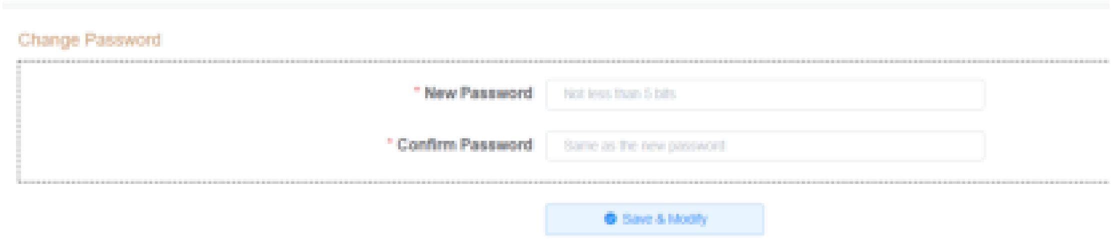
✧ NTP Time Configuration: Configure NTP

➢ **Preview：**



2. Change Password

- **Path:** System → Change Password
- **Function:** Change the gateway system password, length range 5-32.
- **Details:**
  - ✧ Enter the new password and confirm the password. After modification, log out of the system. When logging in again, use the new password.
- **Preview:**

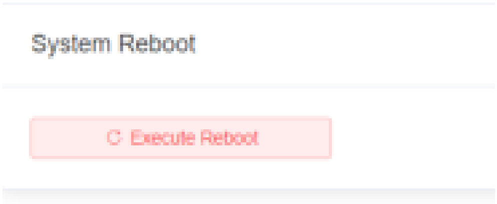

Change Password

* New Password: Not less than 5 bits

* Confirm Password: Same as the new password

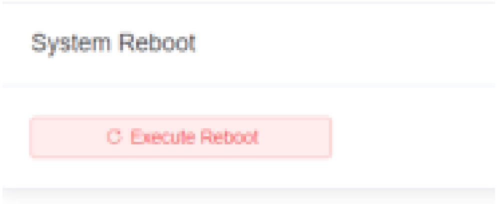
3. Restart

- **Path:** System → Restart
- **Function:** Restart Gateway
- **Details:**
  - ✧ Click to restart the gateway.
- **Preview:**


System Reboot

4. Restore to factory settings.

- **Path:** System → Restore to factory settings.
- **Function:** Clicking on this will restore the gateway to its factory settings, primarily affecting router-related parameters such as network settings (LoRa-related parameters like device lists, join information, etc., will not be deleted).
- **Details:**
  - ✧ Clicking this button will initiate the factory reset process.
- **Preview:**



4.1.4 Data Format

4.1.4.1 Data Explanation

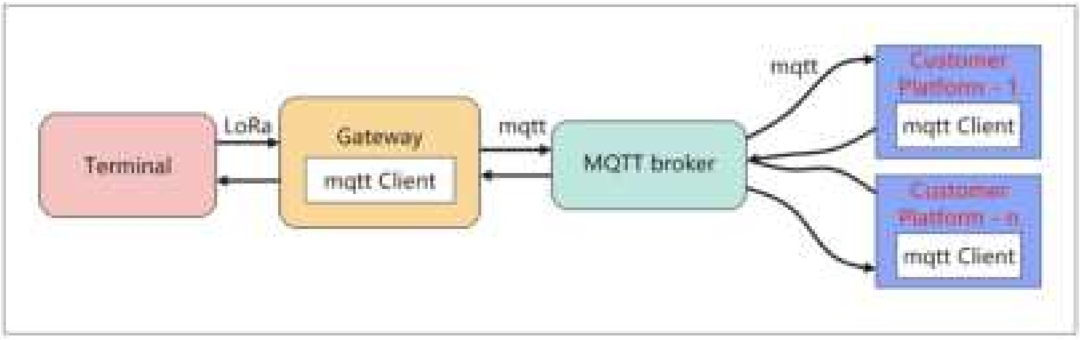
1. Data Format Explanation

Protocol for connecting to the client includes MQTT, TCP, and HTTP. Both MQTT and TCP support two-way communication, while HTTP only supports the gateway pushing data to the client using the POST method and does not support downlink. Below is the data explanation for various protocols:

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

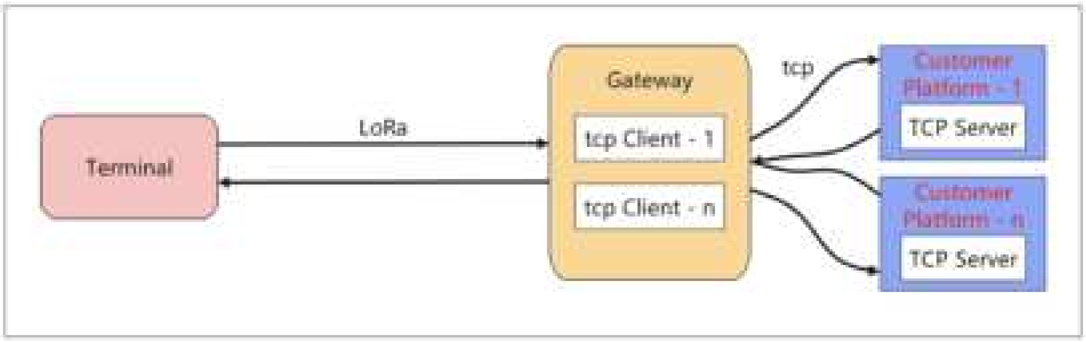
Note: The JSON content data format is consistent within the same type, and if JavaScript function transformation is applied, it will be applied to all data.

2. MQTT Data Flow



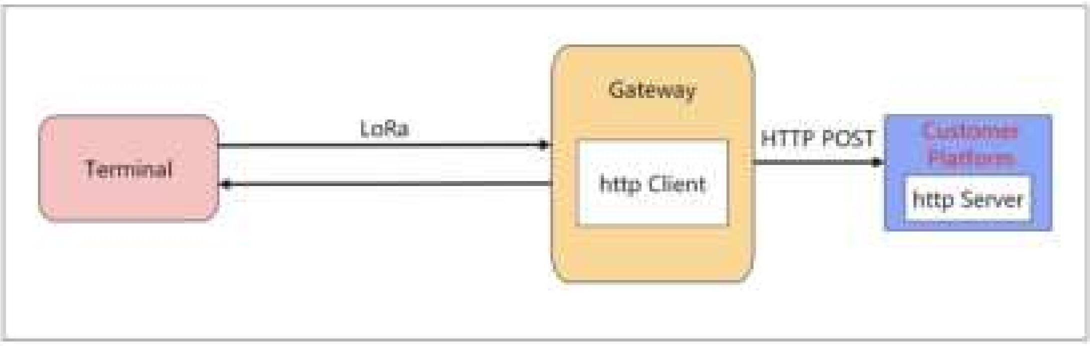
As shown in the diagram, in this mode, you need to deploy an MQTT Broker first. Both the gateway and the client platform establish connections with it and subscribe to relevant topics according to the topic format. If the client needs multiple sets of data, multiple clients can connect and subscribe.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

3. TCP



As depicted in the diagram above, in this mode, the client platform opens a TCP server, while the gateway is configured in TCP mode and points to the corresponding server's IP and port. This configuration allows for the establishment of multiple TCP connections simultaneously.

4. HTTP

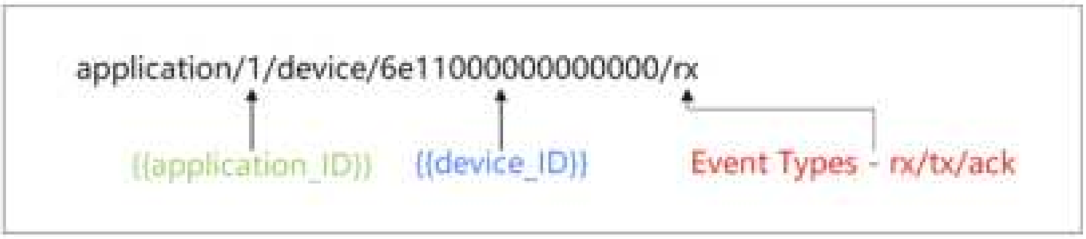


As shown in the diagram above, the configuration for this mode is within the interface settings of each application. This mode only supports data pushing and does not support downstream data.

4.1.4.2 MQTT Data Format

1. MQTT Topic and Data Format

❖ The default MQTT topic format is as follows:



❖ The MQTT approach primarily consists of topics and data content. Topics are displayed and can be modified within the interface.

❖ The default topic includes {{application_ID}} and {{device_EUI}}

■ {{application_ID}}: Application ID, it will be replaced with the corresponding application ID of the device when reporting data (e.g., application/1/device/6e11000000000000/rx). For downlink data, it also needs to be replaced with the actual application ID of the device (e.g., application/1/device/6e11000000000000/tx).

■ {{device_EUI}}: Device unique identifier, it will be replaced with the device's EUI when reporting data (e.g., application/1/device/6e11000000000000/rx). For downlink data, it also needs to be replaced with the actual device EUI (e.g., application/1/device/6e11000000000000/tx). When this field is included in the topic, the JSON content of the downlink data can omit the device's unique identifier.

❖ Modification Description

■ You can modify the topic, for example, change it to "lorawan/uplink" or similar.

■ {{application_ID}}: This can be removed, and after removal, the topic will only lack the application ID.

■ {{device_EUI}}: If removed, the topic won't be able to identify the corresponding device. Therefore, the JSON content of the downlink data must include the device's unique identifier (as explained in the following content).

❖ Example of Subscribed Topics

■ Subscribe to a Single Device for a Single Event: application/1/device/6e11000000000000/rx

■ Subscribe to All Events for a Single Device：application/1/device/6e11000000000000/+

■ Subscribe to a Single Event for All Devices in an Application：application/1/device/+/rx

■ Subscribe to All Events for All Devices in an Application：application/1/device/#

■ Subscribe to a Single Event for All Devices in All Applications：application/+/device/+/rx

■ Subscribe to All Events for All Devices in All Applications：application/+/device/+/+ or application/#

❖ The data content is in JSON format, and the specific format is detailed below. It's important to note that if the {{device_EUI}} placeholder is removed from the downlink topic, the topic won't be able to identify the specific device. In this case, you need to look for the "devEui" field in the data content. If the "devEui" field is also absent, the specific device data will be lost.

2. Uplink Data

❖ Execution Condition: Forward when receiving business data reports from successfully joined devices.

❖ Default Topic Format: application/{{application_ID}}/device/{{device_EUI}}/rx

❖ Default Topic Example：application/1/device/6e11000000000000/rx

❖ Default JSON Data Content Example:

```
{
    "applicationID": "1",
    "applicationName": "temperature",
    "deviceName": "dev_00000000",
    "devEui": "6e11000000000000",
    "rxInfo": [{
        "gatewayID": "ff0000000000000a",
        "name": "ff0000000000000a",
        "time": "", // Only when the gateway can receive GPS signals will there be 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, you can refer to the "Base64 Encoding and Decoding"
section later for more information.
}
```

3. Join Data

❖ Execution condition: Pushed upon receiving a device's join request and responding to the join accept message.

- Default Topic format：application/{{application_ID}}/device/{{device_EUI}}/join
- Default Topic Example： application/1/device/6e11000000000000/join
- Default Data Content Example：

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

4. Downlink Data

- Execution Condition: Sending business data to the device
- Default Topic Format： application/{{application_ID}}/device/{{device_EUI}}/tx
- Default Topic Example： application/1/device/6e11000000000000/tx
- Default JSON Data Content Example：

```
{
   "devEui": "6e11000000000000",
   "confirmed": true,
   "fPort": 12,
   "data": "MTIzNA==" // Base64 encoded, please refer to the "Base64 Encoding and Decoding" section
below. This corresponds to "1234".
}
```

- Convenient test data (the data above contains spaces, which might cause transmission failures):

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

5. Downlink acknowledgment packet response:

- Execution condition: After receiving the downlink acknowledgment packet, push the data when the device responds.
- 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: When multicast information needs to be sent to devices with the same triplets as 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

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

7. Heartbeat data

- ❖ Execution condition: Heartbeat switch is turned on, heartbeat interval > 0, heartbeat content is not 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	Byte count	Function	Identifier	Value example
0	1	Frame header	header	0xFE
1	1	Version number (currently V1)	version	0x01
2	2	JSON data length (big-endian)	length	0x0001
4	1	Data Type	type	0x00-Heartbeat Packet
5	2	Random key Random number (big endian)	random	0x1234
7	n	JSON Content	JSON Object	{...}

❖ The first 7 bytes are the TCP data header, and starting from the 7th byte is the JSON content. This JSON content is the same as that used in MQTT and HTTP.

2. Uplink Data

Offset	Byte count	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	Information about the receiving gateway	Array of structures
- gatewayID	Gateway unique identifier	string
- name	Gateway Name	string
- time	GPS Time	string
- rssi	Signal strength	float64
- loRaSNR	Signal-to-Noise Ratio	float64
- location	GPS Location (When GPS signal is not available, the value is { })	
- longitude	Longitude	float64
- latitude	Latitude	float64
- altitude	Altitude	float64
TxInfo	Device Data Transmission Parameters	
- frequency	Frequency Point	uint32
- dr	Rate	uint8
adr	Whether ADR request is enabled	bool
fCnt	Uplink frame counter	uint32
fPort	Uplink Port	uint8
data	Business data (in base64 encoded format)	string

3. Activation data

Offset	Byte Count	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01

<table>
<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 EUI</td><td>string</td></tr>
<tr><td>devAddr</td><td>Short address assigned to the device during activation</td><td>string</td></tr>
</table>
</td></tr>
</table>

4. Downlink Data

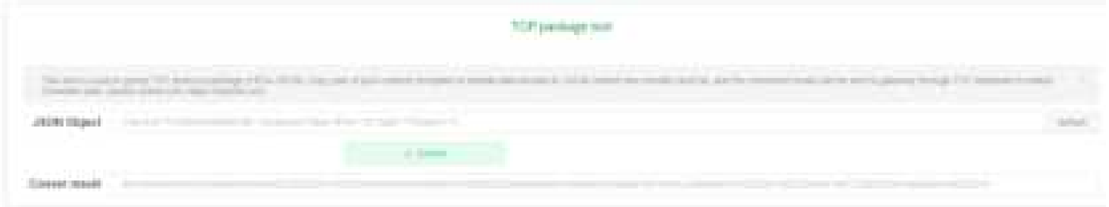
Offset	Byte Count	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x004D
4	1	type	0x02
5	2	random	0x1234
7	77	JSON object	{ "devEui": "ff00000000000001", "confirmed": false, "fPort": 10,

<table>
<tr><td></td><td></td><td></td><td>"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>Confirmation packet flag (default: false)</td><td>bool</td></tr><tr><td>fPort</td><td>Port（Default 10）</td><td>uint8</td></tr><tr><td>data</td><td>Sending business data (base64 encoded)</td><td>string</td></tr></table></td></tr>
</table>

❖ Convenient test data (the data above may contain spaces, which can sometimes cause sending failures)

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

Note: You can use the TCP Packet Generator tool available on the web page (Path: LoRa Network Server -> Interfaces -> Data Conversion -> TCP Packet Generator) to generate the corresponding data for testing, as shown below:



Among them, the data that can be used for testing when sending downlink data to a TCP server is as follows (during testing, you usually need to modify the devEui):

```
fe01004b0204427b22646576455549223a22666630303030303030303030303030303031222c22636f6e6669726d65642
23a747275652c2266506f7274223a31302c2264617461223a224d54497a4e413d3d227d
```

5. Response for a Downlink Confirmation Packet

Offset	Byte Count	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x0000
4	1	type	0x05
5	2	random	0x1234

<table>
<tr><td>7</td><td>77</td><td>JSON object</td><td>{
"applicationID": "1",
"applicationName": "app1",
"deviceName": "dev_00000000",
"devEui": "6e00000000000000",
"acknowledged": true
}
<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 Corresponding Name</td><td>String</td></tr>
<tr><td>deviceName</td><td>Device Name</td><td>String</td></tr>
<tr><td>devEui</td><td>Device EUI</td><td>String</td></tr>
<tr><td>acknowledged</td><td>Response status: Success - true</td><td>Bool</td></tr>
</table></td></tr>
</table>

6. Downlink Multicast Data

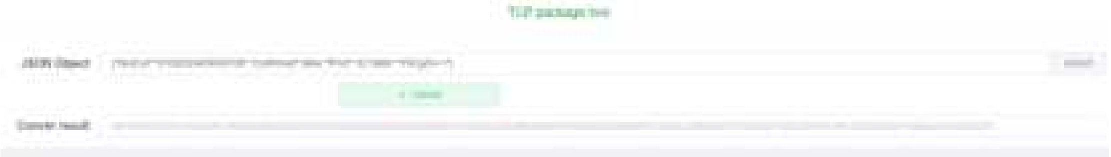
<table>
<tr><th>Offset</th><th>Byte Count</th><th>Function</th><th>Value or Description</th></tr>
<tr><td>0</td><td>1</td><td>header</td><td>0xFE</td></tr>
<tr><td>1</td><td>1</td><td>version</td><td>0x01</td></tr>
<tr><td>2</td><td>2</td><td>length</td><td>0x0000</td></tr>
<tr><td>4</td><td>1</td><td>type</td><td>0x04</td></tr>
<tr><td>5</td><td>2</td><td>random</td><td>0x1234</td></tr>
<tr><td>7</td><td>77</td><td>JSON object</td><td>{
"multicastGroupId": 1,
"fPort": 10,
"data": "YWJjZA=="
}
<table>
<tr><th></th><th>Description</th><th>Type</th></tr>
<tr><td>multicastGroupId</td><td>Multicast ID</td><td>int</td></tr>
</table></td></tr>
</table>

			fPort	Port（Default 10）	uint8
			data	Sending business data (Base64 encoded)	string

❖ Convenient test data

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

Note: You can use the TCP packet tool on the web page (Path: LoRa Network Server → Interfaces → 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 example content above is as follows:

```
fe01003304341e7b226d756c7469636173747726f75704964223a312c2266506f7274223a31302c2264617461223a2259574a6a5a413d3d227d
```

7. Heartbeat Data

Offset	Byte Count	Function	Value or Description
0	1	header	0xFE
1	1	version	0x01
2	2	length	0x01BC
4	1	type	0x00
5	2	random	0x1234

7	n	JSON object	{
			"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
        }]
}
```

	Description	Type
gateways	Array of Gateway Information	
- gatewayID	Gateway Unique Identifier	string
- lastSeenAt	Gateway Last Uplink Time	string
- isOnline	Online status, true: online, false: offline	bool
- longitude	Longitude	float64
- latitude	Latitude	float64
applications	Application information array	
- applicationID	Application ID	int
- name	Application Name	string
- deviceNum	Total number of devices under this application	int
- activatNum	Number of devices that are already activated (joined)	int
- isAutoJoin	Whether this application allows automatic device provisioning	bool

4.1.4.4 HTTP Push Data Format

❖ HTTP is configured for each application, configuration path: LoRa Network Server -> Applications -> View (corresponding to the APP) -> Interface Management.

❖ The data content for HTTP push is in JSON format, and its content is consistent with the JSON content for MQTT and TCP methods (please refer to the previous two sections).

❖ When a JavaScript function is configured for parsing, the JSON data will no longer use the default data format; instead, it will use the transformed data format.

❖ The HTTP method only supports data pushing and does not support downstream data.

4.1.4.5 JavaScript Function Transformation Method

❖ The purpose of the function transformation method

■ The function transformation method allows for converting hexadecimal or string data reported by devices during uplink transmission into corresponding JSON format field data. This enables seamless integration with specific platforms without requiring customizations.

■ When sending downlink data, the function converts the JSON data sent by the client platform into corresponding hexadecimal data, which is then transmitted to the device.

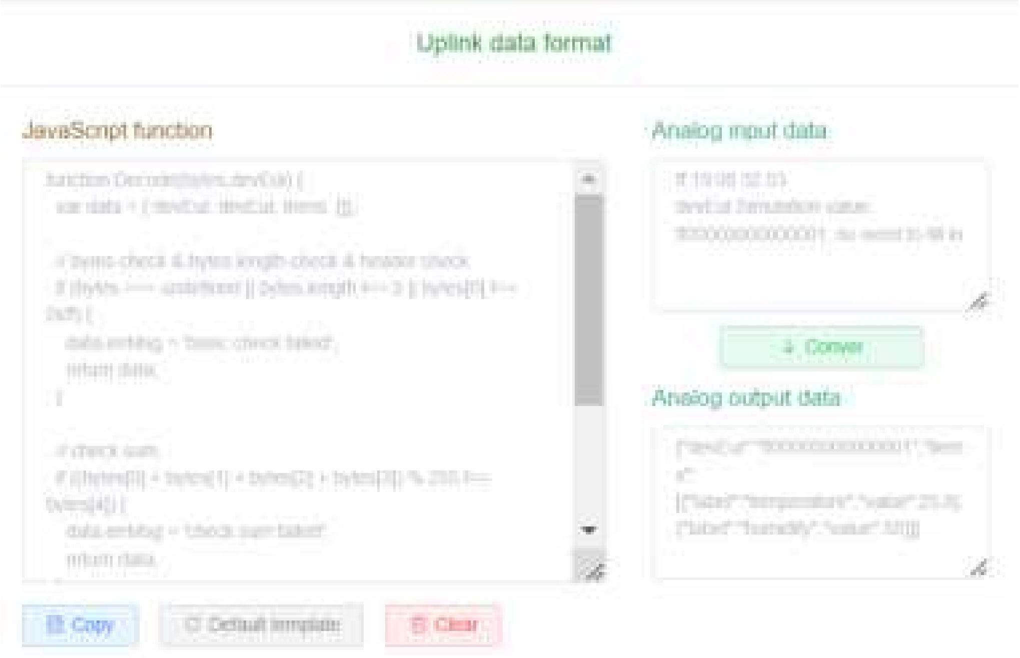
❖ When this conversion function is not configured, the default data format is used.

❖ The gateway supports function conversion for both uplink and downlink data, and by default, both are in a disabled state.

1. Uplink Data Transformation

❖ When the device reports data as hexadecimal values like "ff 19 08 32 53", you can transform it into the following JSON format: {"devEui":"ff00000000000001","items":[{"label":"temperature","value":25.8},{"label":"humidity","value":50}]} (where ff is the protocol fixed header, 19 is the temperature integer part, 08 is the temperature decimal part, 32 is the humidity value, and 53 is the checksum). Once this transformation is successfully configured, the JSON-format data received by the client will be as described.

❖ Configuration Path: LoRa Network Server → Interface → Data Transformation → Upstream Data Format



```
function Decode(bytes, devEui) {
  var data = { devEui: devEui, items: [] };

  // bytes check & bytes length check & header check
  if (bytes === undefined || bytes.length !== 5 || bytes[0] !==
0xff) {
    data.errMsg = 'basic check failed';
    return data;
  }

  // check sum
  if ((bytes[0] + bytes[1] + bytes[2] + bytes[3]) % 256 !==
bytes[4]) {
    data.errMsg = 'check sum failed';
    return data;
```

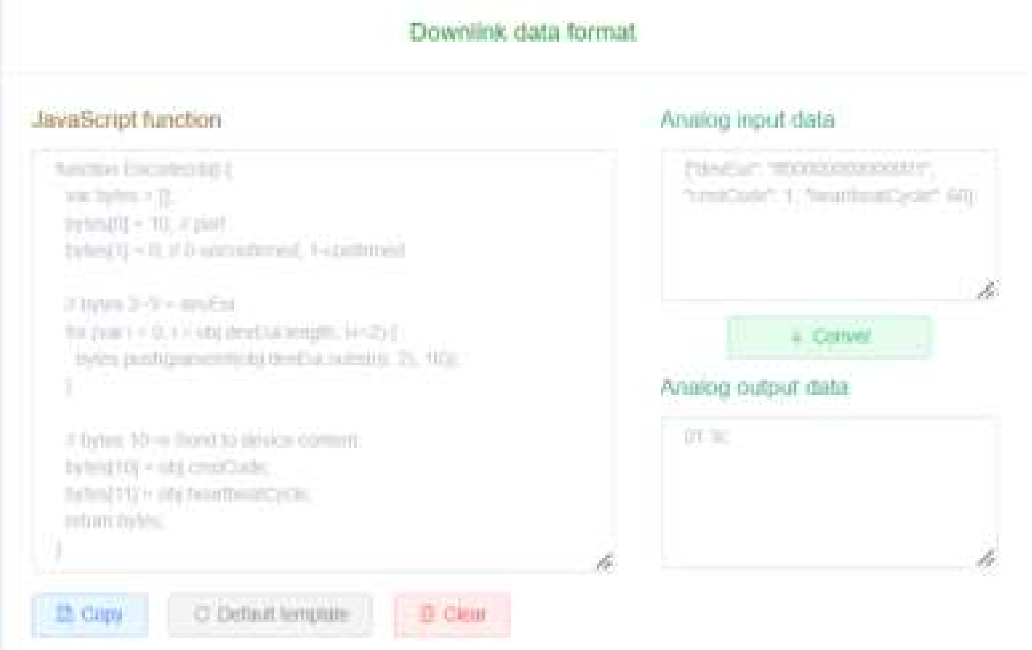
```
ff 19 08 32 53
devEui (simulation value:
ff00000000000001, [illegible]
```

```
{"devEui":"ff00000000000001","items":[{"label":"temperature","value":25.8},{"label":"humidity","value":50}]}
```

2. Downlink Data Transformation

❖ When the device sends data {"devEui": "ff00000000000001", "cmdCode": 1, "heartbeatCycle": 60} and applies a function transformation ---(function transformation)---> 01 3c (01-command code for heartbeat cycle configuration, 3c-heartbeat cycle value), it will be transformed into 013c and sent to the terminal.

❖ Configuration Path: LoRa Network Server → Interfaces → Data Transformation → Downlink Data Format



Downlink data format

JavaScript function

```
function Encode(obj) {
  var bytes = [];
  bytes[0] = 10; // port
  bytes[1] = 0; // 0-unconfirmed, 1-confirmed

  // bytes 2~9 = devEui
  for (var i = 0; i < obj.devEui.length; i+=2) {
   bytes.push(parseInt(obj.devEui.substr(i, 2), 16));
  }

  // bytes 10~ = Send to device content
  bytes[10] = obj.cmdCode;
  bytes[11] = obj.heartbeatCycle;
  return bytes;
}
```

Analog input data

```
{"devEui": "8000000000000011",
"cmdCode": 1, "heartbeatCycle": 60}
```

Analog output data

```
01 3c
```

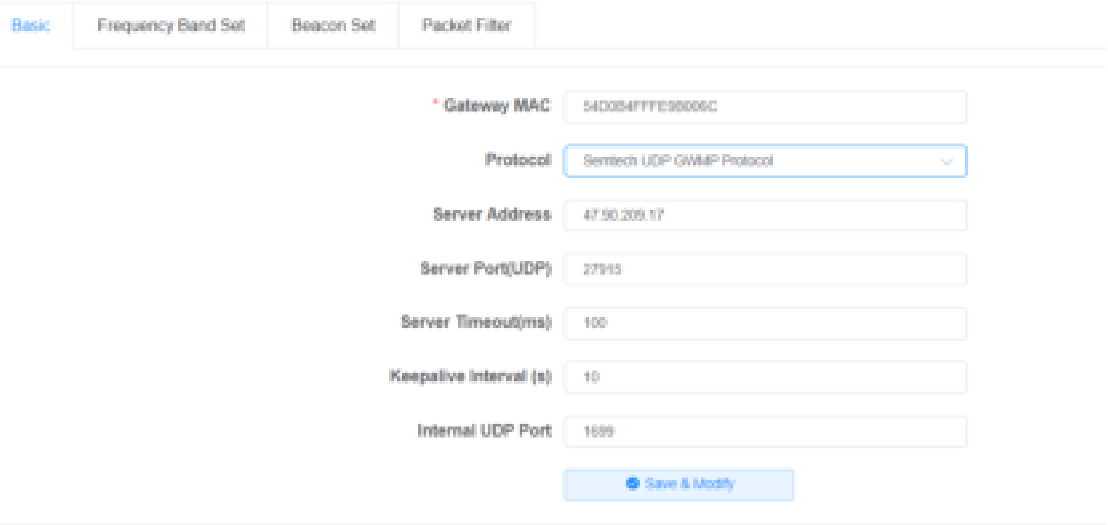
4.1.5 Common Platform Integration

4.1.5.1 Four-Faith Cloud NS

❖ The standard network server (NS) used by Four-Faith Cloud adopts the Semtech UDP GWMP Protocol.

❖ In this mode, the gateway implements the data forwarding function.

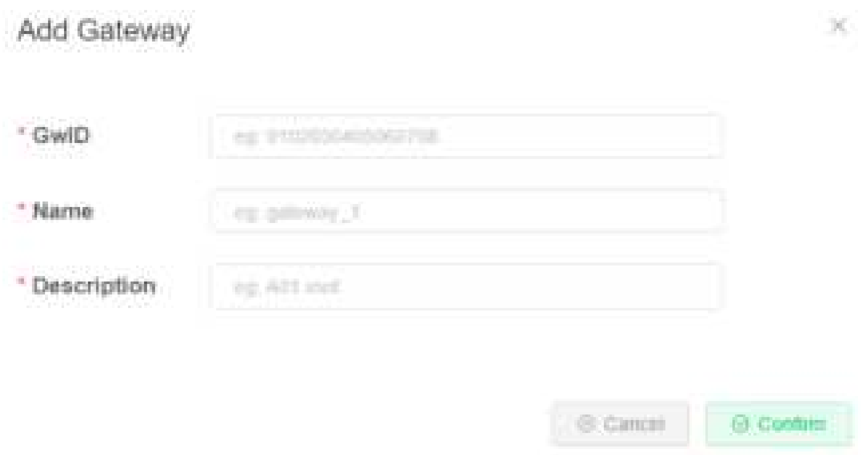
❖ Configuration Path: LoRa Gateway → Basic Settings, Main configurations include protocol selection, server address, and server port (UDP). The specific settings are as follows:



* Gateway MAC	54D0B4FFFE98006C
Protocol	Semtech UDP GWMP Protocol
Server Address	47.90.209.17
Server Port(UDP)	27915
Server Timeout(ms)	100
Keepalive Interval (s)	10
Internal UDP Port	1699

❖ Open CSTool：http://47.90.209.17:51868/#/ns/gateways

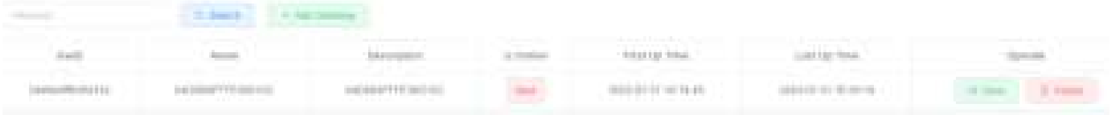
❖ Create Gateway



Add Gateway

* GwID: eg: [illegible]
* Name: eg: gateway_1
* Description: eg: A01 xxx

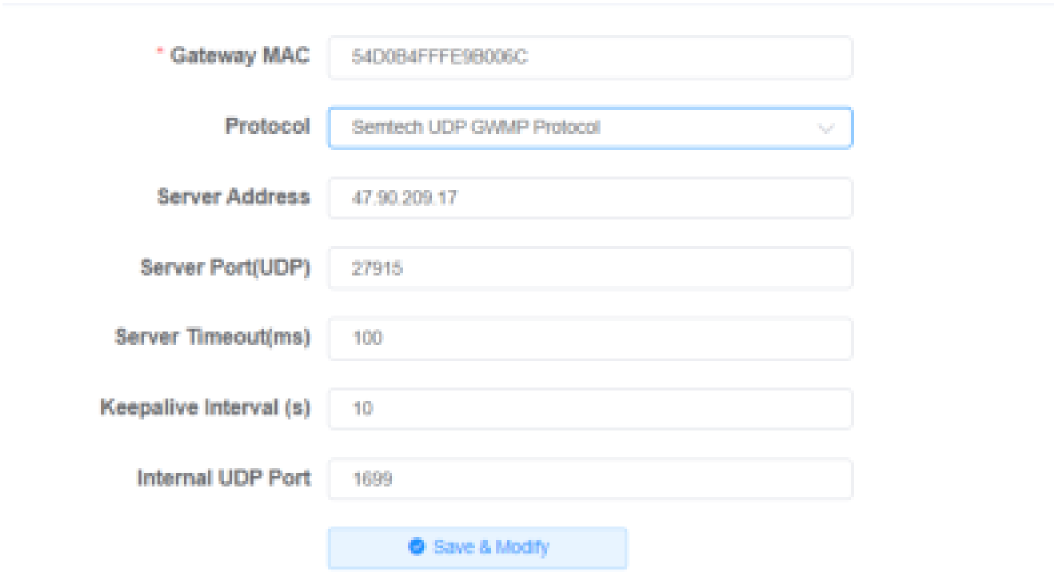
❖ Viewing Gateway Status: As shown in the following image, it is evident that the gateway is currently online.



4.1.5.2 ChirpStack Platform（GWMP）

❖ ChirpStack is a versatile open-source Network Server (NS) that supports multiple connectivity methods. One of the commonly used methods for integration is through the GWMP (Gateway Management Protocol) protocol.

❖ Configuration Path: LoRa Gateway → Basic Settings, Main Configuration Protocol, Server Address, Server Port(UDP), Specific configurations are as follows:



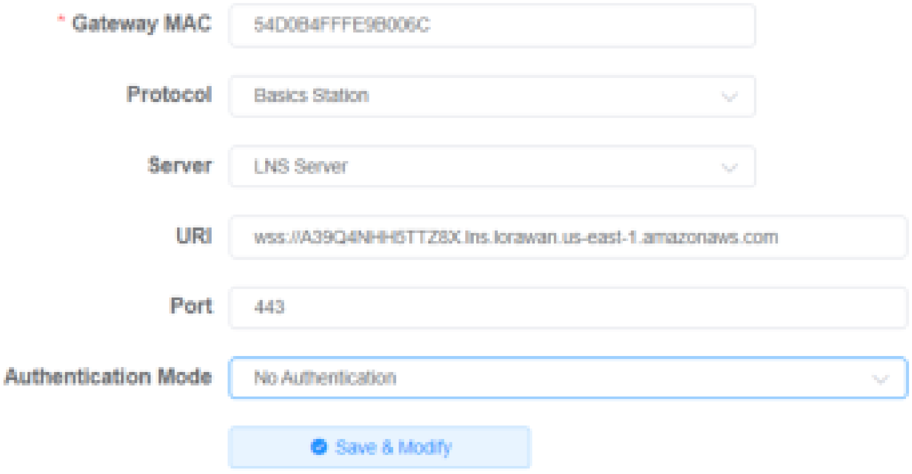
* Gateway MAC: 54D0B4FFFE9B006C
Protocol: Semtech UDP GWMP Protocol
Server Address: 47.90.209.17
Server Port(UDP): 27915
Server Timeout(ms): 100
Keepalive Interval (s): 10
Internal UDP Port: 1699

4.1.5.3 ChirpStack Platform（LNS）

ChirpStack can be configured to use the Basicstation protocol for integration. This mode of integration is generally referred to as LNS (LoRa Network Server). It supports two modes: No Authentication and TLS Server Authentication. The following examples illustrate the configuration for both modes of integration.

1. LNS - No Authentication

❖ By configuring the protocol, server protocol type, URI, port, and mode selection, you can make the necessary changes. Once these modifications are successfully applied, the configuration will be updated.



* Gateway MAC: 54D0B4FFFE9B006C

Protocol: Basics Station

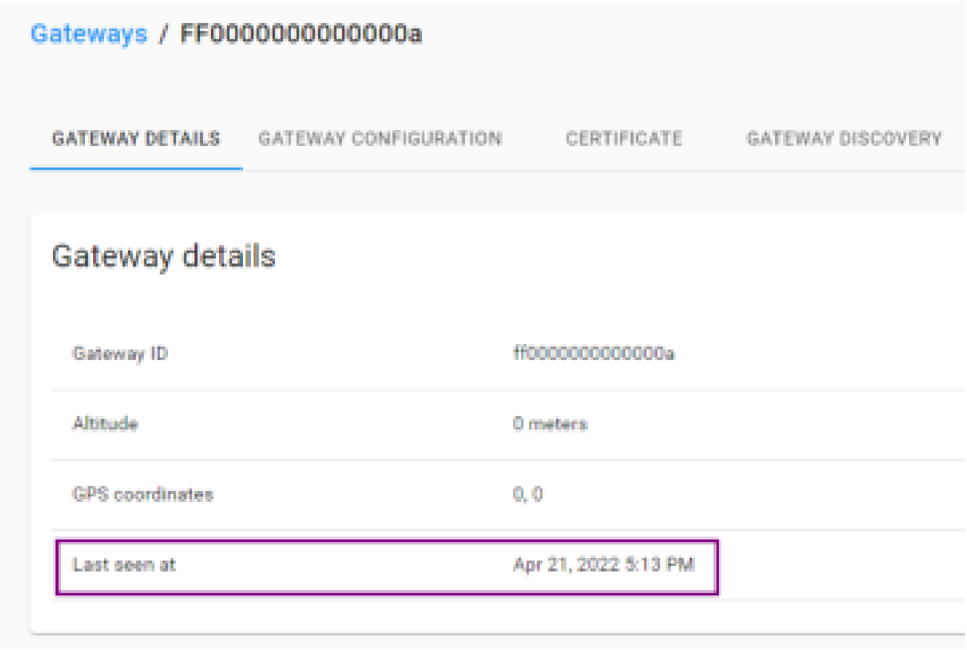
Server: LNS Server

URI: wss://A39Q4NHH6TTZ8X.lns.lorawan.us-east-1.amazonaws.com

Port: 443

Authentication Mode: No Authentication

❖ The "Last seen at" on the platform indicates the gateway's connection status.



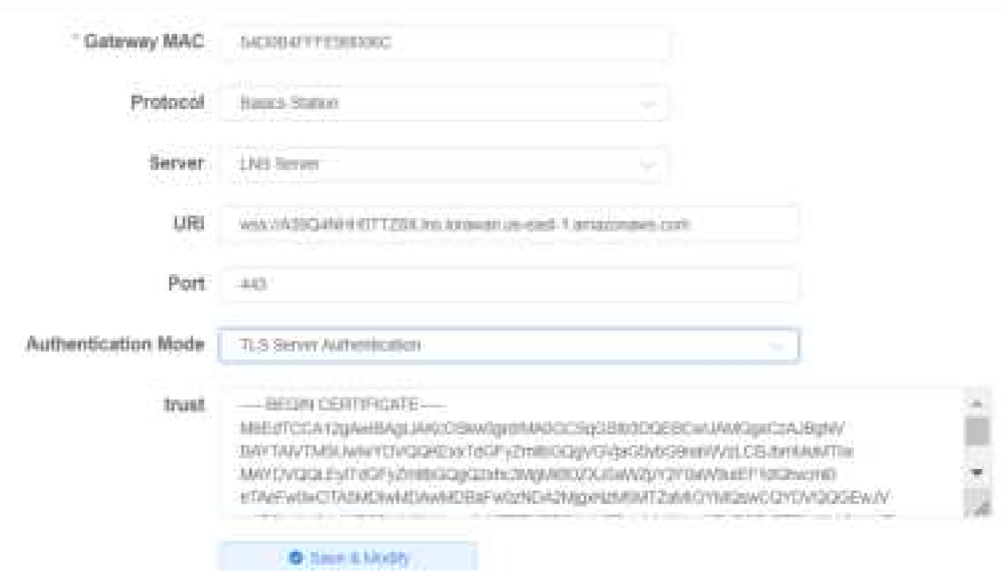
Gateways / FF0000000000000a

Gateway details

Gateway ID	ff0000000000000a
Altitude	0 meters
GPS coordinates	0, 0
Last seen at	Apr 21, 2022 5:13 PM

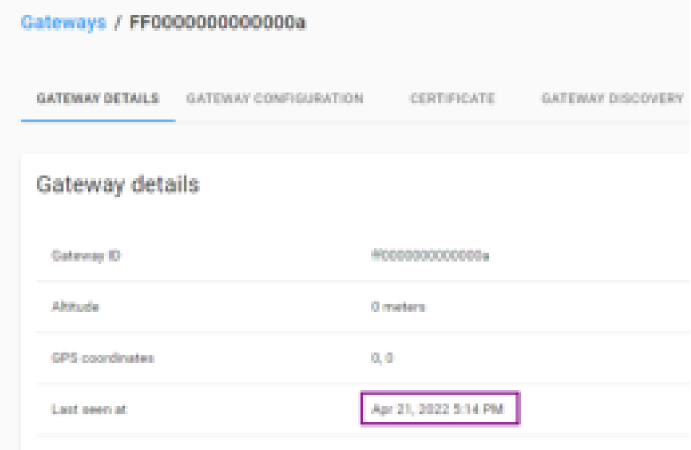
2. LNS - TLS Server Authentication

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



* Gateway MAC	54C0B4FFFE5B50C
Protocol	Basics Station
Server	LNS Server
URI	wss://A39Q4NHH0TTZ5X.lns.lorawan.us-east-1.amazonaws.com
Port	443
Authentication Mode	TLS Server Authentication
trust	-----BEGIN CERTIFICATE----- [illegible] [illegible] [illegible] [illegible]

- ❖ The "Last seen at" on the platform indicates the gateway's connection status.



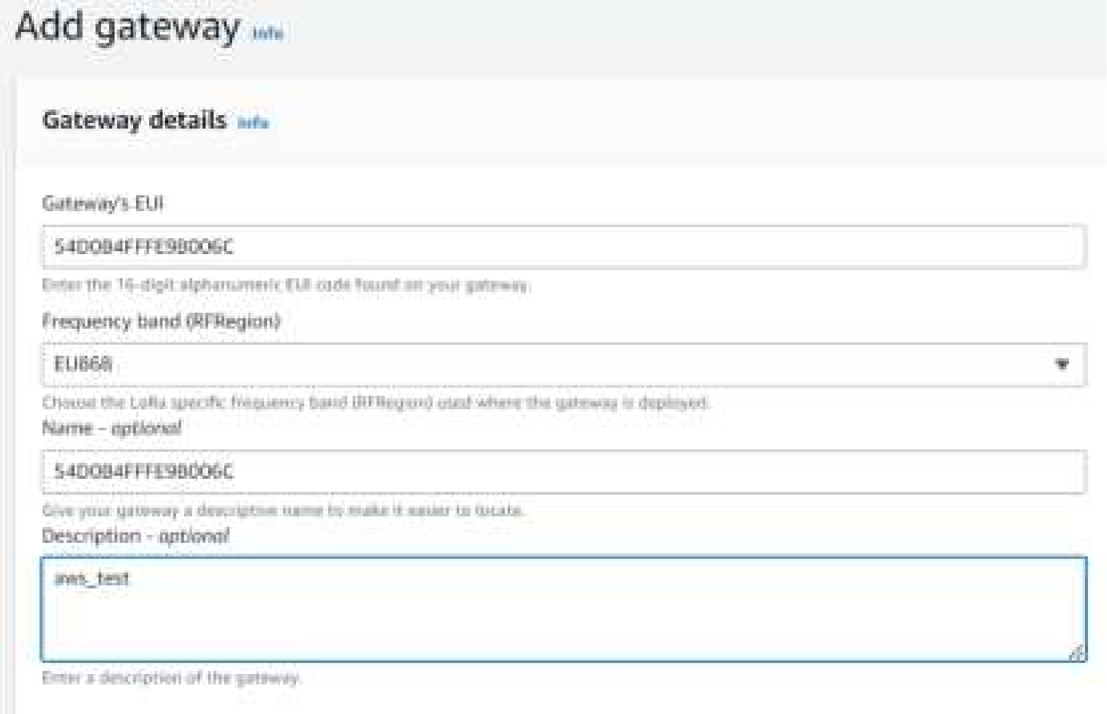
Gateways / FF0000000000000a

Gateway details

Gateway ID	ff0000000000000a
Altitude	0 meters
GPS coordinates	0, 0
Last seen at	Apr 21, 2022 5:14 PM

4.1.5.4 AWS Platform（LNS）

- ❖ Create a gateway on the AWS platform.



Add gateway Info

Gateway details Info

Gateway's EUI

54D0B4FFFE98006C

Enter the 16-digit alphanumeric EUI code found on your gateway.

Frequency band (RFRegion)

EU868

Choose the LoRa specific frequency band (RFRegion) used where the gateway is deployed.

Name - optional

54D0B4FFFE98006C

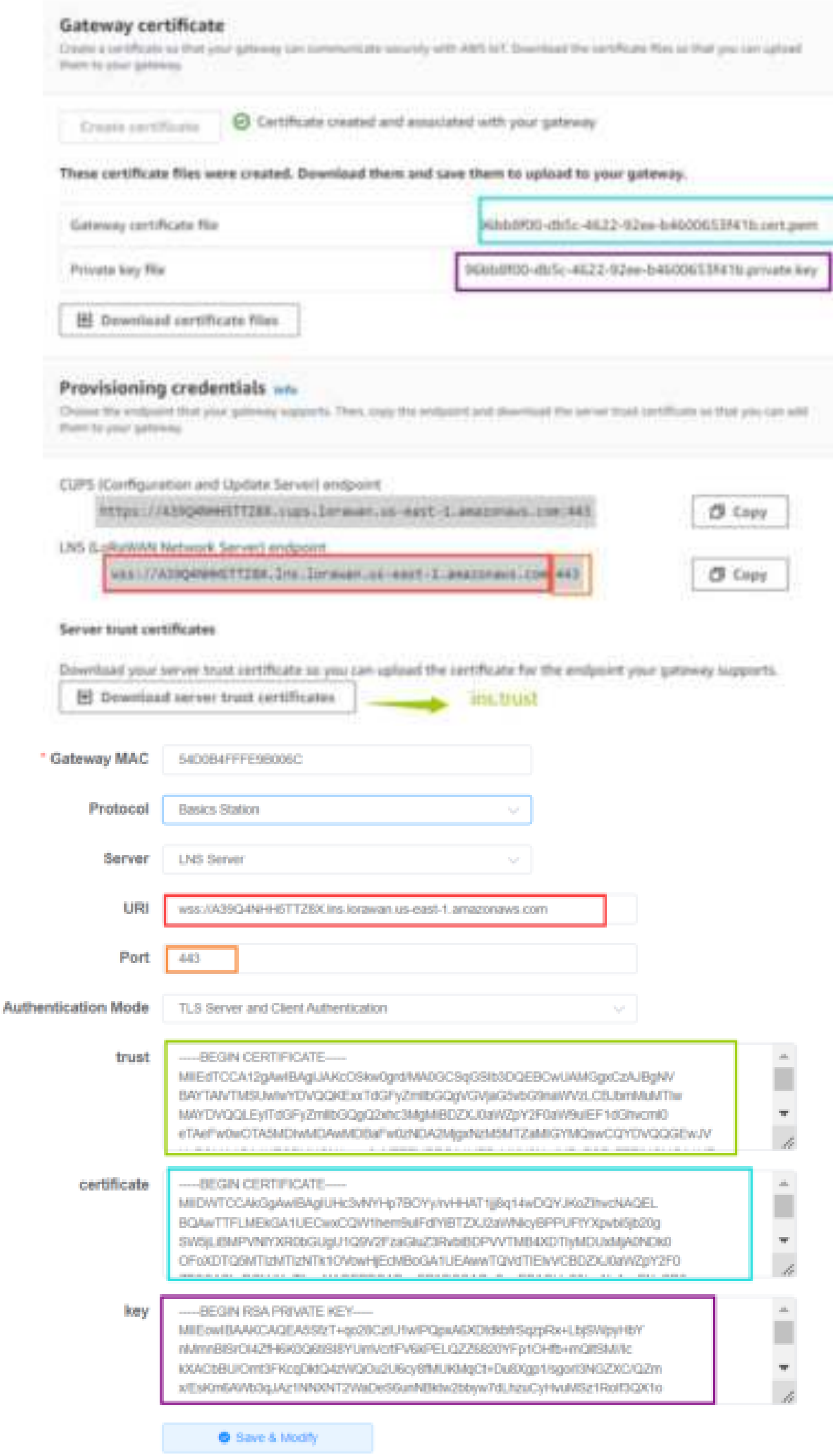
Give your gateway a descriptive name to make it easier to locate.

Description - optional

aws_test

Enter a description of the gateway.

❖ Download the corresponding keys generated by the gateway and configure the corresponding parameters for the gateway. Choose the mode "TLS Server and Client Authentication." In the following image, boxes with the same color represent matching content.



Gateway certificate

Create a certificate so that your gateway can communicate securely with AWS IoT. Download the certificate files so that you can upload them to your gateway.

Certificate created and associated with your gateway

These certificate files were created. Download them and save them to upload to your gateway.

Gateway certificate file	6bb8f00-db5c-4622-92ee-b4600653f41b.cert.pem
Private key file	96bb8f00-db5c-4622-92ee-b4600653f41b.private.key

Provisioning credentials Info

Choose the endpoint that your gateway supports. Then, copy the endpoint and download the server trust certificate so that you can add them to your gateway.

CUPS (Configuration and Update Server) endpoint

`https://A39Q4NHH6TTZ8X.cups.lorawan.us-east-1.amazonaws.com:443`

LNS (LoRaWAN Network Server) endpoint

`wss://A39Q4NHH6TTZ8X.lns.lorawan.us-east-1.amazonaws.com:443`

Server trust certificates

Download your server trust certificate so you can upload the certificate for the endpoint your gateway supports.

* Gateway MAC	54D0B4FFFE9B006C
Protocol	Basics Station
Server	LNS Server
URI	wss://A39Q4NHH6TTZ8X.lns.lorawan.us-east-1.amazonaws.com
Port	443
Authentication Mode	TLS Server and Client Authentication

trust

```
-----BEGIN CERTIFICATE-----
MIIEdTCCA12gAwIBAgIJAKcOSkw0grd/MA0GCSqGSIb3DQEBCwUAMGgxCzAJBgNV
BAYTAlVTMSUwIwYDVQQKExxTdGFyZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTIw
MAYDVQQLEylTdGFyZmllbGQgQ2xhc3MgMiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTAeFw0wOTA5MDIwMDAwMDBaFw0zNDA2MjgxNzM5MTZaMIGYMQswCQYDVQQGEwJV
```

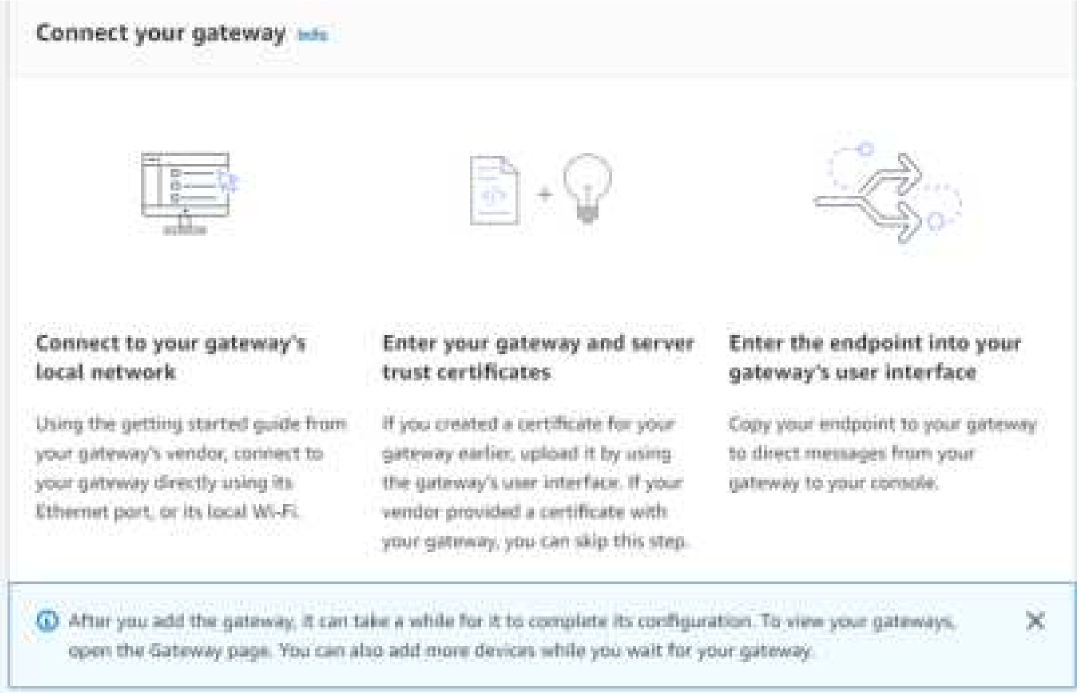
certificate

```
-----BEGIN CERTIFICATE-----
MIIDWTCCAkGgAwIBAgIUHc3vNYHp7BOYy/nHHAT1jj8q14wDQYJKoZIhvcNAQEL
BQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi5jb20g
SW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIyMDUxMjA0NDk0
OFoXDTQ5MTIzMTIzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0
```

key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA5S6zT+qo28CzIU1wlPQpxAGXDIdkbfrSqzpRx+LbjSWpyHbY
nMmnBISrOl4ZfH6K0Q6lIS8YUmVcrtFV6kPELQZZ6820YFp1OHfb+mQlt8M/Ic
kXACbBUlOmt3FKcqDktQ4zWQOu2U6cy8fMUKMqC1+Du8Xgp1/sgorl3NGZXC/Q2m
x/EsKm6AWb3qJAz1NNXNT2WaDeSGunNBktw2bbyw7dLhzuCyHvuMSz1RolfIQX1o
```

❖ Continuing Gateway Creation:



Connect your gateway Info

Connect to your gateway's local network

Using the getting started guide from your gateway's vendor, connect to your gateway directly using its Ethernet port, or its local Wi-Fi.

Enter your gateway and server trust certificates

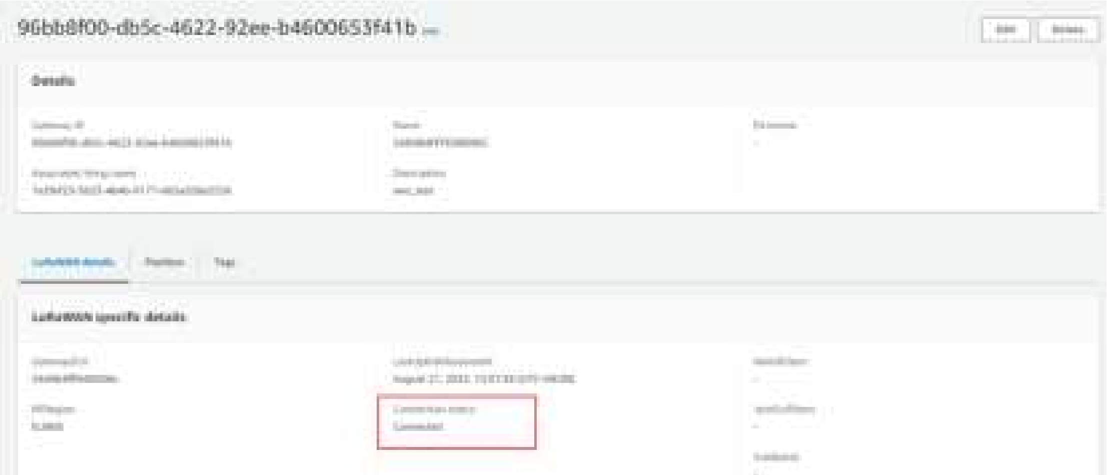
If you created a certificate for your gateway earlier, upload it by using the gateway's user interface. If your vendor provided a certificate with your gateway, you can skip this step.

Enter the endpoint into your gateway's user interface

Copy your endpoint to your gateway to direct messages from your gateway to your console.

After you add the gateway, it can take a while for it to complete its configuration. To view your gateways, open the Gateway page. You can also add more devices while you wait for your gateway.

❖ After successful configuration, you can observe the gateway's connection status on the AWS platform.



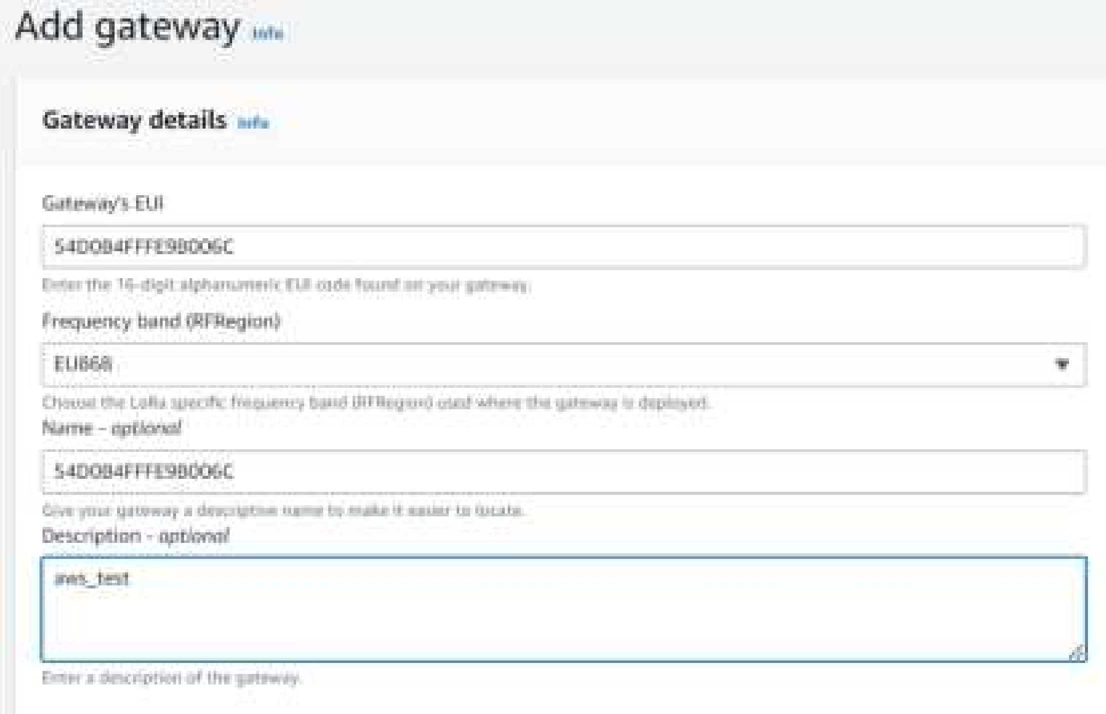
96bb8f00-db5c-4622-92ee-b4600653f41b

Details

LoRaWAN specific details

4.1.5.5 AWS Platform（CPUS）

❖ Create a gateway on the AWS platform.



Add gateway Info

Gateway details Info

Gateway's EUI

54D0B4FFFE9B006C

Enter the 16-digit alphanumeric EUI code found on your gateway.

Frequency band (RFRegion)

EU868

Choose the LoRa specific frequency band (RFRegion) used where the gateway is deployed.

Name - *optional*

54D0B4FFFE9B006C

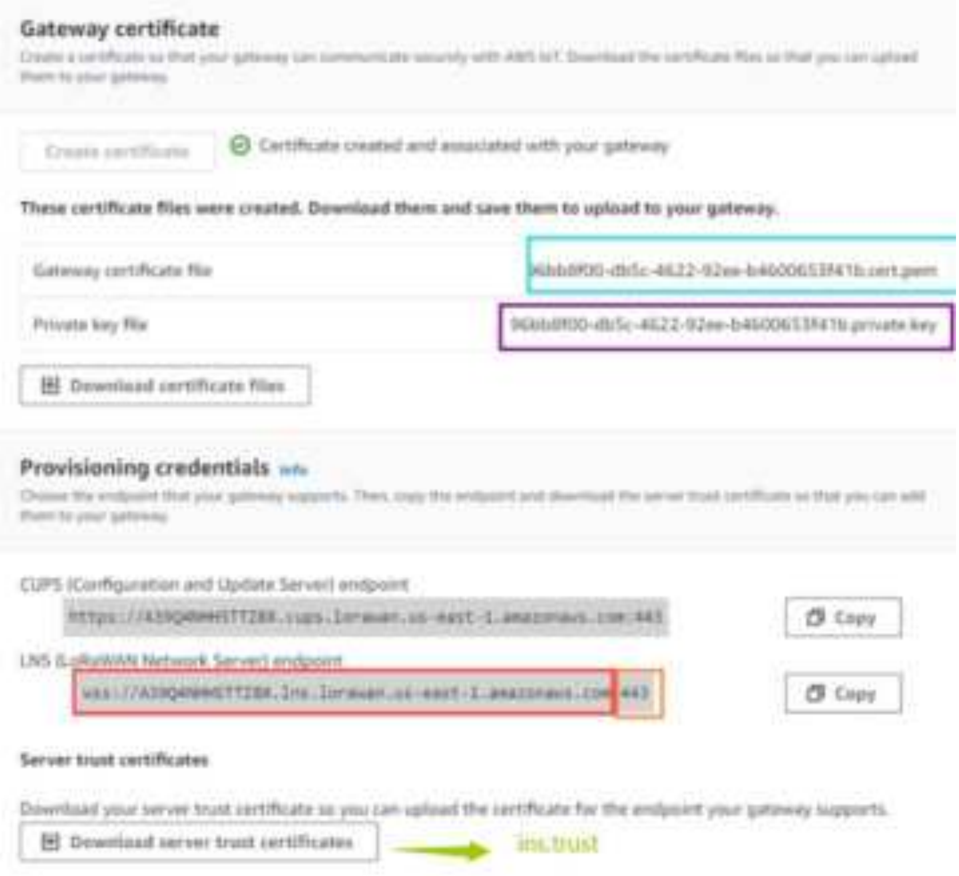
Give your gateway a descriptive name to make it easier to locate.

Description - *optional*

aws_test

Enter a description of the gateway.

❖ Download the corresponding keys generated by the gateway, and configure the corresponding parameters of the gateway. Select the TLS Server and Client Authentication mode. In the diagram below, boxes with the same color indicate the same content.



Gateway certificate

Create a certificate so that your gateway can communicate securely with AWS IoT. Download the certificate files so that you can upload them to your gateway.

Create certificate   Certificate created and associated with your gateway

These certificate files were created. Download them and save them to upload to your gateway.

Gateway certificate file	96bb8f00-db5c-4622-92ee-b4600653f41b.cert.pem
Private key file	96bb8f00-db5c-4622-92ee-b4600653f41b.private.key

Download certificate files

Provisioning credentials Info

Choose the endpoint that your gateway supports. Then, copy the endpoint and download the server trust certificate so that you can add them to your gateway.

CUPS (Configuration and Update Server) endpoint

https://A39Q4NMHSTTZ8X.cups.lorawan.us-east-1.amazonaws.com:443   Copy

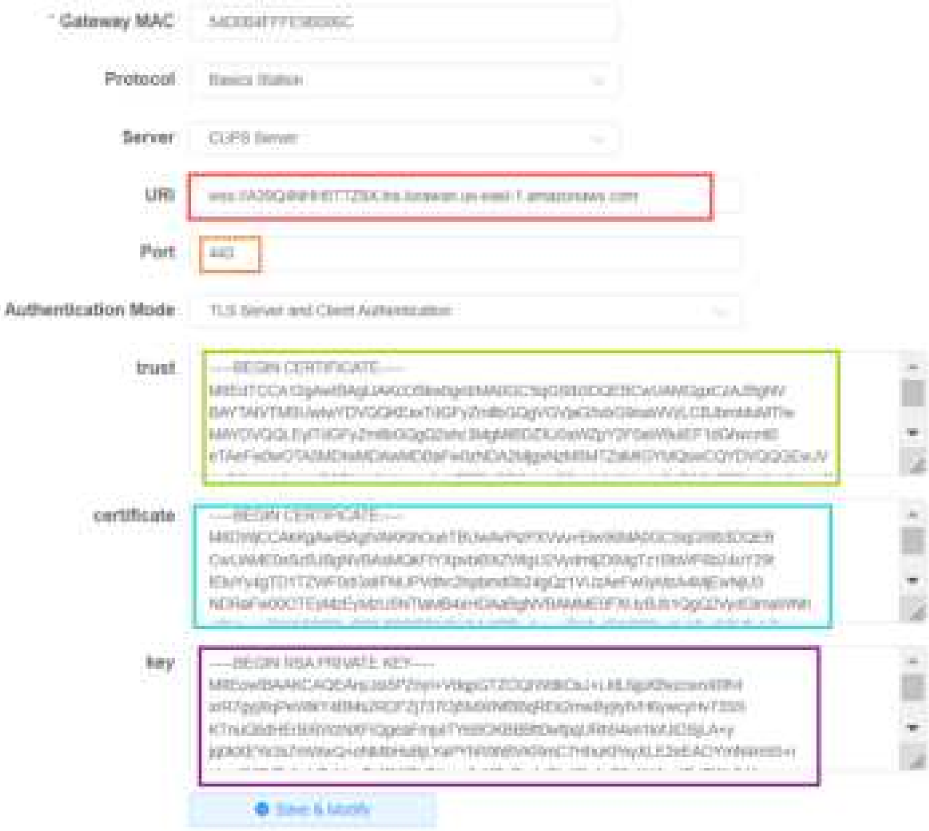
LNS (LoRaWAN Network Server) endpoint

wss://A39Q4NMHSTTZ8X.lns.lorawan.us-east-1.amazonaws.com:443   Copy

Server trust certificates

Download your server trust certificate so you can upload the certificate for the endpoint your gateway supports.

Download server trust certificates



* Gateway MAC: [illegible]

Protocol: Basics Station

Server: CUPS Server

URI: wss://[illegible].lns.lorawan.us-east-1.amazonaws.com

Port: 443

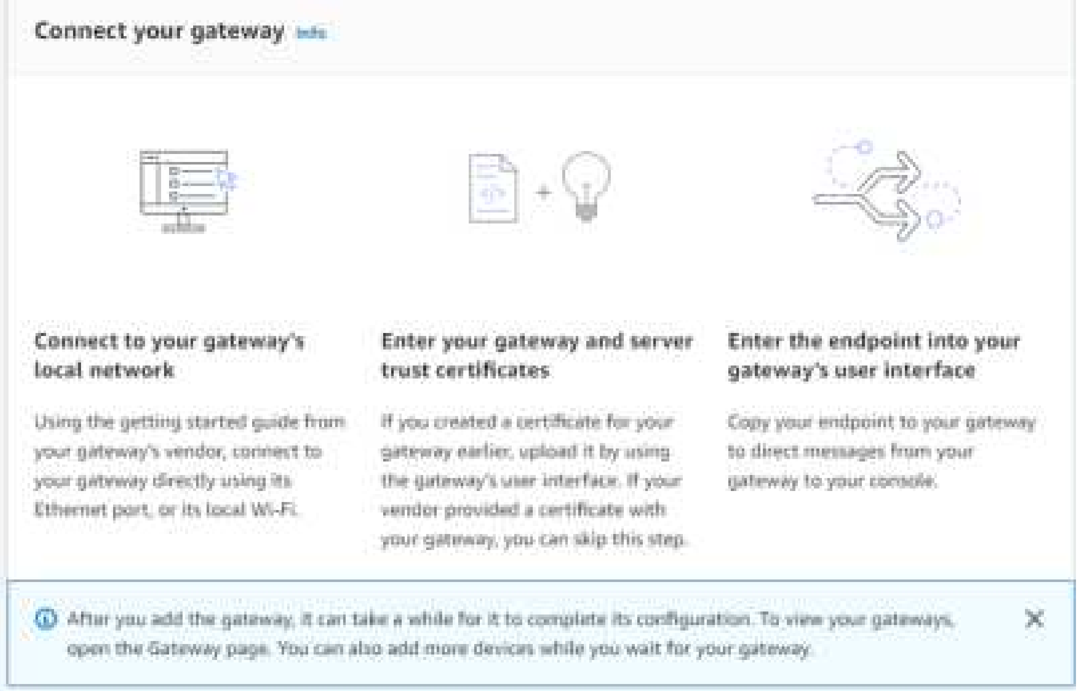
Authentication Mode: TLS Server and Client Authentication

trust:
```
-----BEGIN CERTIFICATE-----
[illegible]
```

certificate:
```
-----BEGIN CERTIFICATE-----
[illegible]
```

key:
```
-----BEGIN RSA PRIVATE KEY-----
[illegible]
```

❖ Continuing the gateway creation:



Connect your gateway

Connect to your gateway's local network

Using the getting started guide from your gateway's vendor, connect to your gateway directly using its Ethernet port, or its local Wi-Fi.

Enter your gateway and server trust certificates

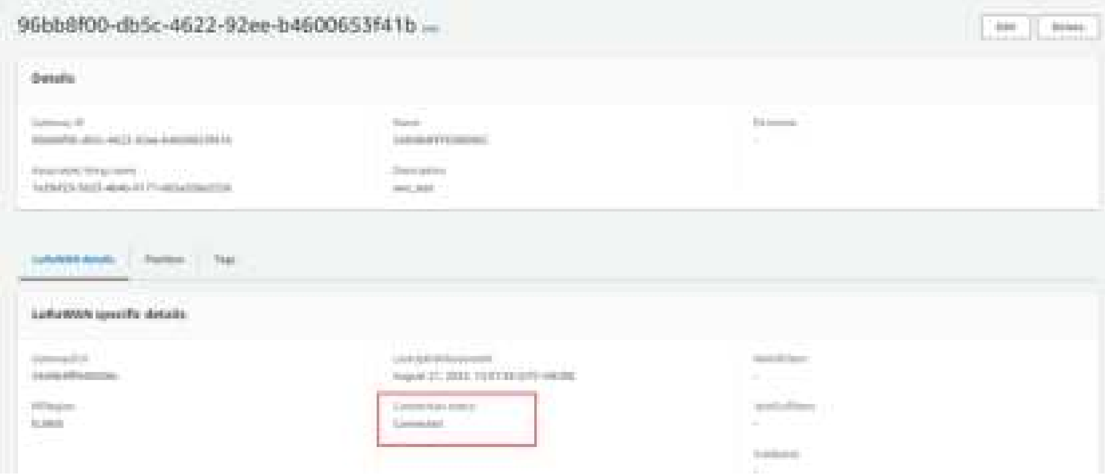
If you created a certificate for your gateway earlier, upload it by using the gateway's user interface. If your vendor provided a certificate with your gateway, you can skip this step.

Enter the endpoint into your gateway's user interface

Copy your endpoint to your gateway to direct messages from your gateway to your console.

After you add the gateway, it can take a while for it to complete its configuration. To view your gateways, open the Gateway page. You can also add more devices while you wait for your gateway.

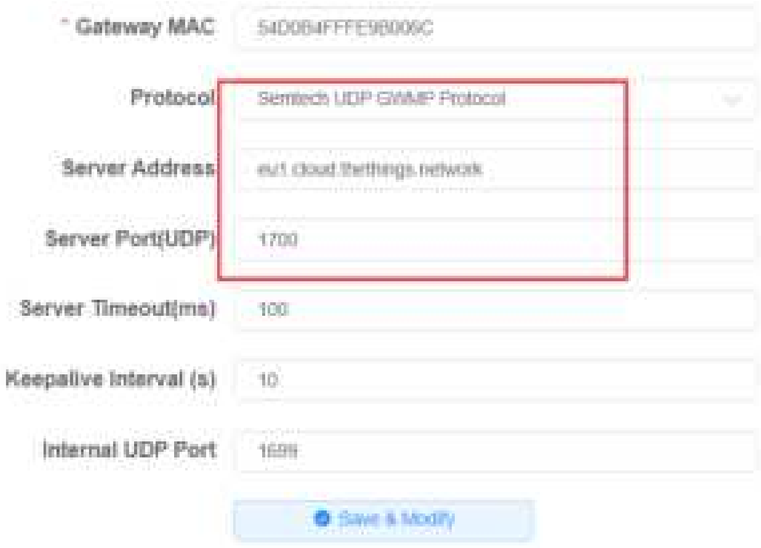
❖ After successfully configuring the gateway, you can view the gateway's connection status on the AWS platform.



96bb8f00-db5c-4622-92ee-b4600653f41b

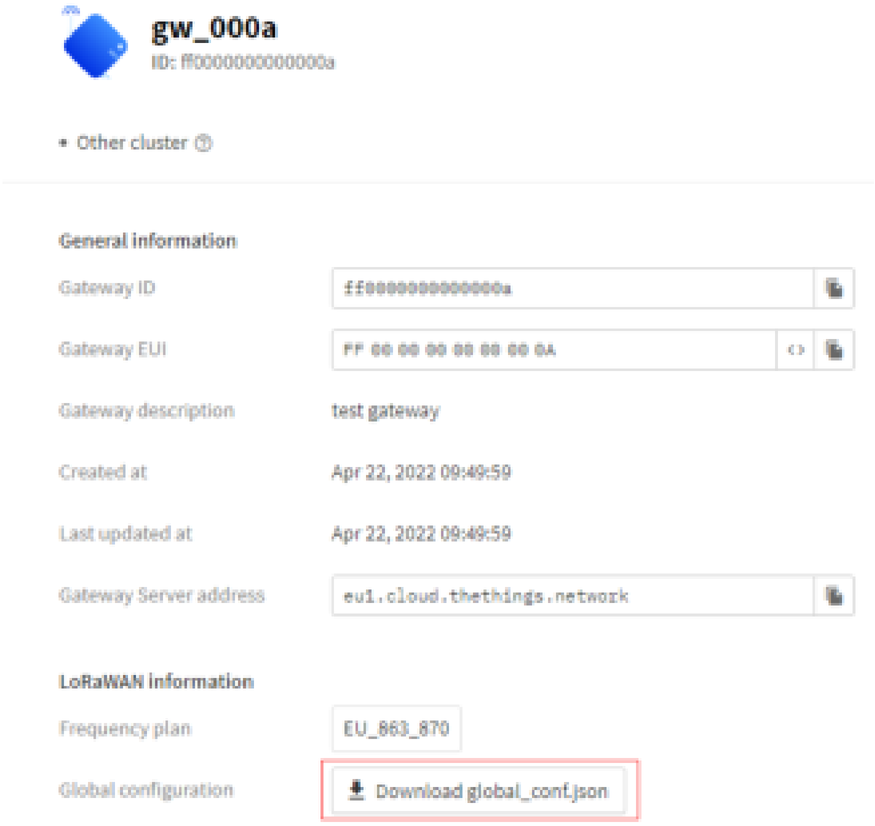
4.1.5.6 TTN Platform（GWMP）

- ❖ The TTN platform supports GWMP and Basicstation modes of access.
- ❖ When using the GWMP protocol, it is consistent with other platforms. You only need to configure the server IP and port.



* Gateway MAC	54D0B4FFFE9B009C
Protocol	Semtech UDP GWMP Protocol
Server Address	eu1.cloud.thethings.network
Server Port(UDP)	1700
Server Timeout(ms)	100
Keepalive Interval (s)	10
Internal UDP Port	1689

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



gw_000a
ID: ff0000000000000a

Other cluster

General information

Gateway ID: ff0000000000000a

Gateway EUI: FF 00 00 00 00 00 00 0A

Gateway description: test gateway

Created at: Apr 22, 2022 09:49:59

Last updated at: Apr 22, 2022 09:49:59

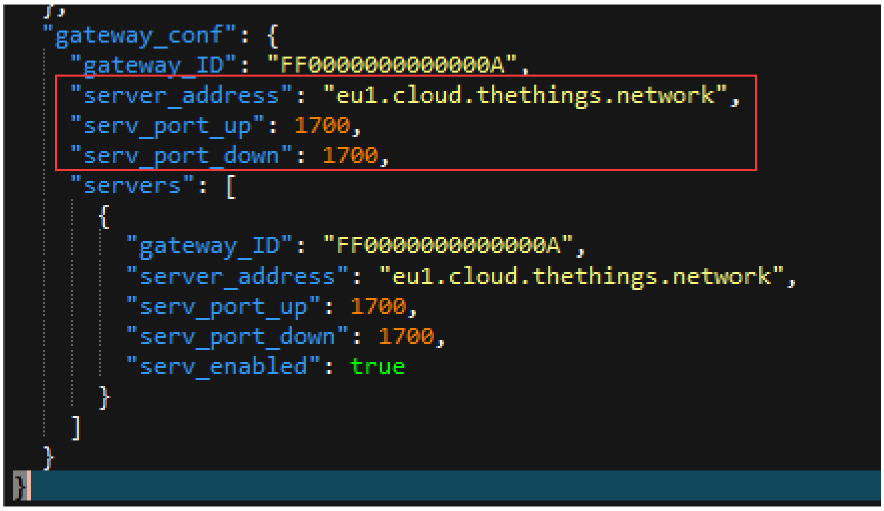
Gateway Server address: eu1.cloud.thethings.network

LoRaWAN information

Frequency plan: EU_863_870

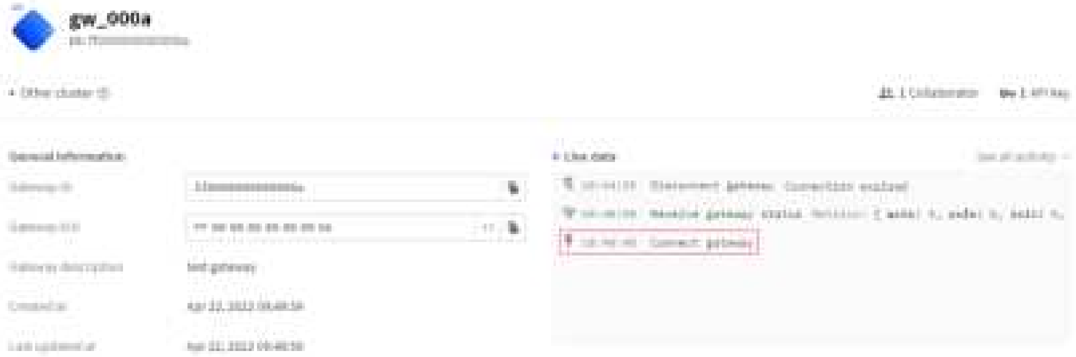
Global configuration: Download global_conf.json

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



```
},
"gateway_conf": {
  "gateway_ID": "FF0000000000000A",
  "server_address": "eu1.cloud.thethings.network",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  "servers": [
    {
      "gateway_ID": "FF0000000000000A",
      "server_address": "eu1.cloud.thethings.network",
      "serv_port_up": 1700,
      "serv_port_down": 1700,
      "serv_enabled": true
    }
  ]
}
}
```

❖ After successfully configuring the gateway, you can see the connection information on the TTN platform.



gw_000a

Other cluster

1 Collaborator 1 API Key

General information

Live data

Gateway description test gateway

Disconnect gateway

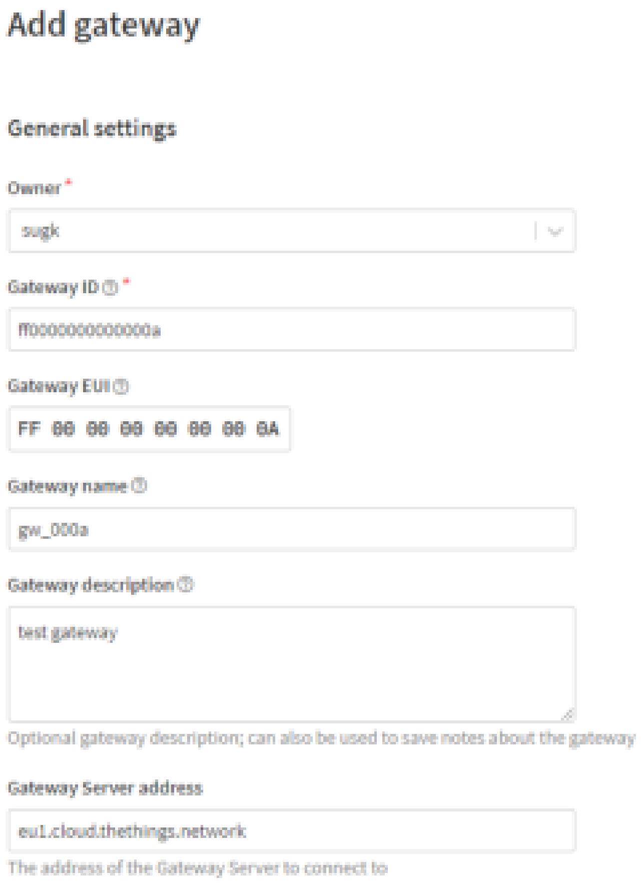
Receive gateway status

Connect gateway

4.1.5.7 TTN Platform（LNS）

❖ The TTN platform also supports connection using the Basicstation LNS protocol, with the mode being TLS Server Authentication and Client Token.

❖ Add 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

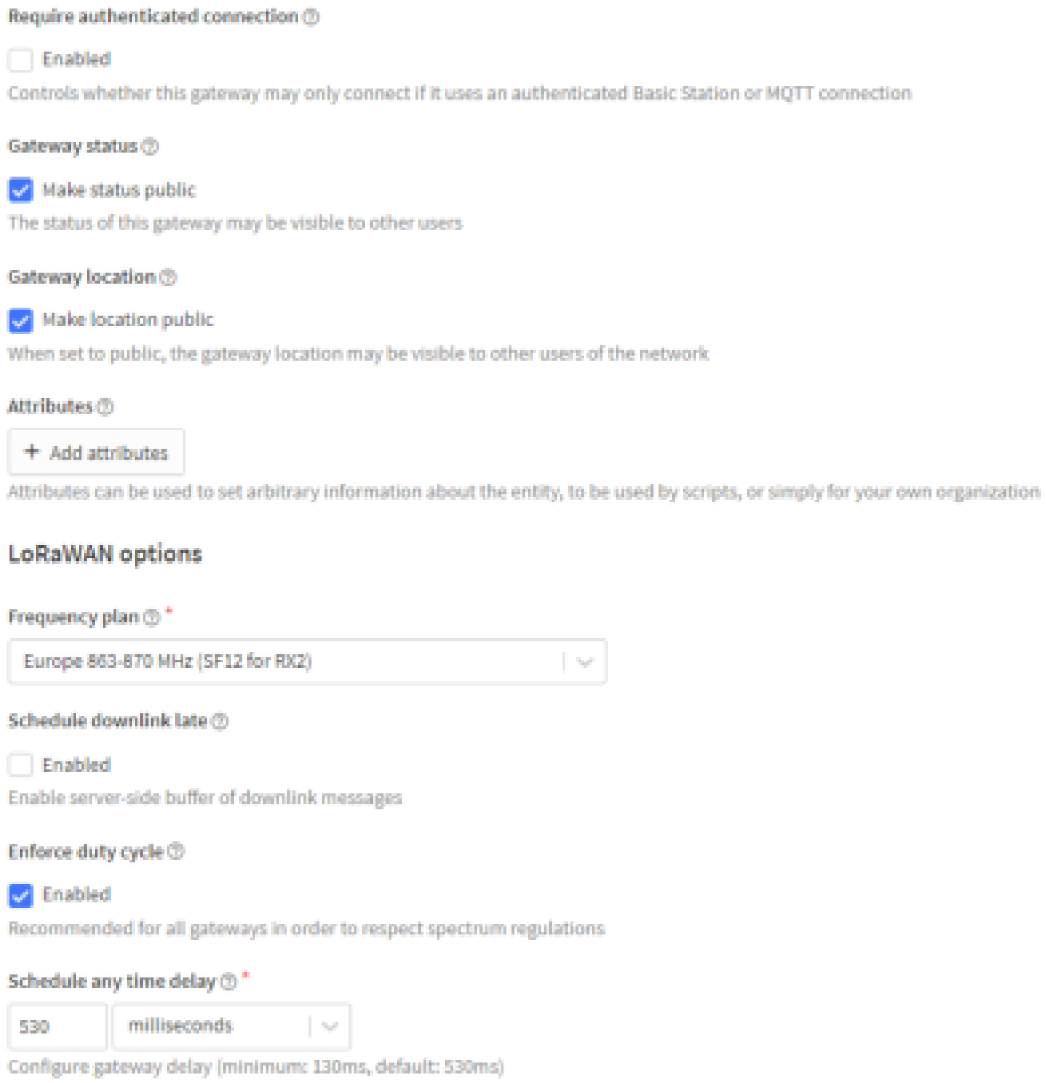
test gateway

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



Require authenticated connection ⓘ

□ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status ⓘ

☑ Make status public

The status of this gateway may be visible to other users

Gateway location ⓘ

☑ Make location public

When set to public, the gateway location may be visible to other users of the network

Attributes ⓘ

+ Add attributes

Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

LoRaWAN options

Frequency plan ⓘ *

Europe 863-870 MHz (SF12 for RX2)

Schedule downlink late ⓘ

□ Enabled

Enable server-side buffer of downlink messages

Enforce duty cycle ⓘ

☑ Enabled

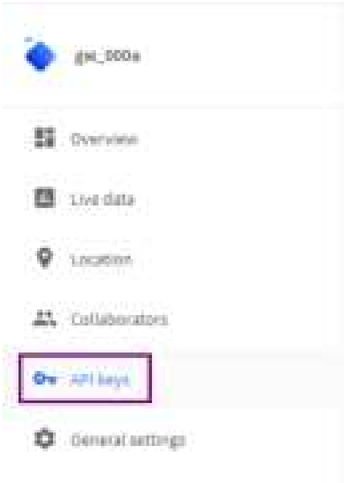
Recommended for all gateways in order to respect spectrum regulations

Schedule any time delay ⓘ *

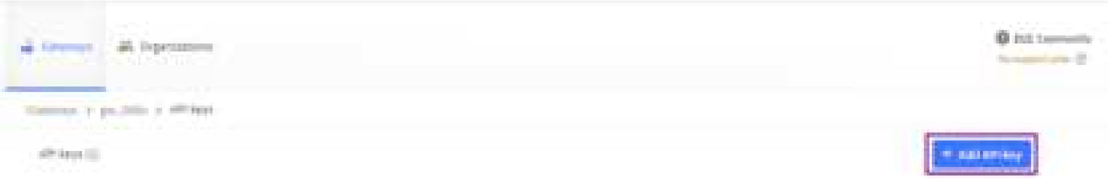
530 milliseconds

Configure gateway delay (minimum: 130ms, default: 530ms)

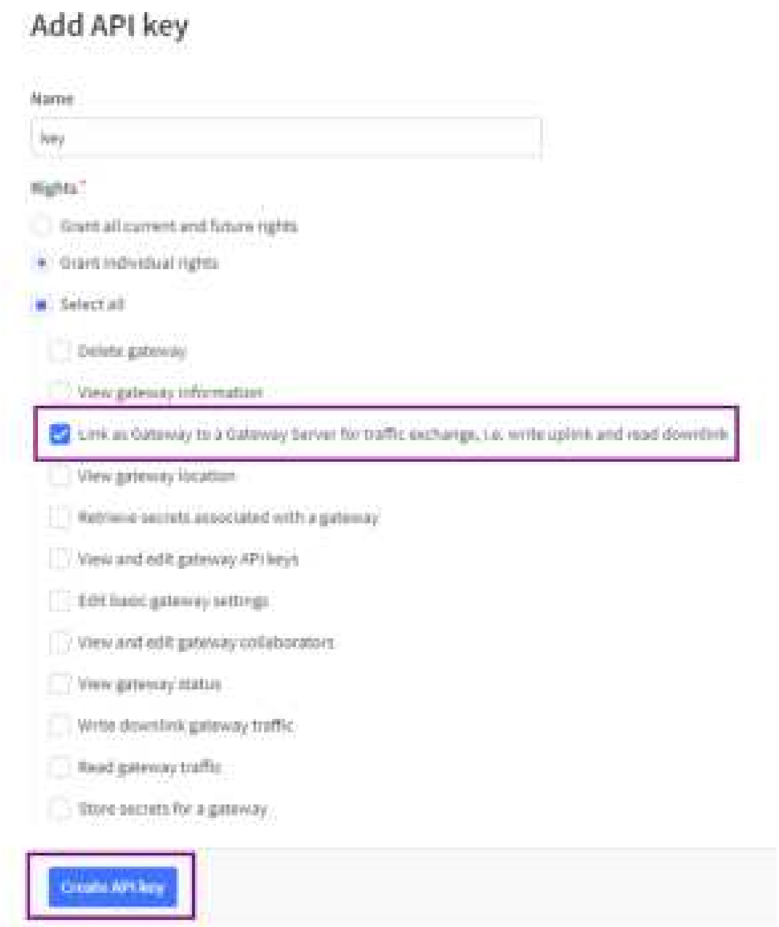
- ❖ Obtaining the Token Value



- ❖ Adding API key



❖ Follow the diagram below, and create an API key.



Add API key

Name

key

Rights*

○ Grant all current and future rights

● Grant individual rights

■ Select all

□ Delete gateway

□ View gateway information

☑ Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink

□ View gateway location

□ Retrieve secrets associated with a gateway

□ View and edit gateway API keys

□ Edit basic gateway settings

□ View and edit gateway collaborators

□ View gateway status

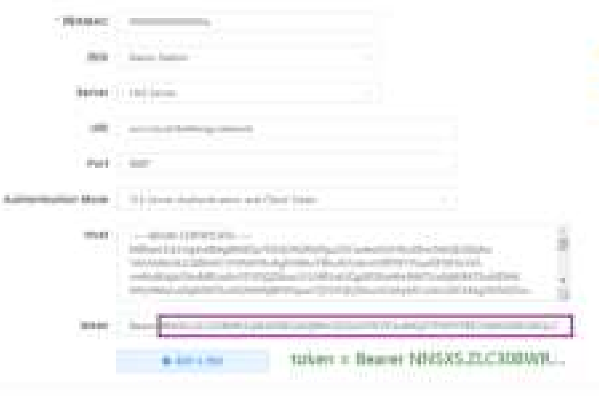
□ Write downlink gateway traffic

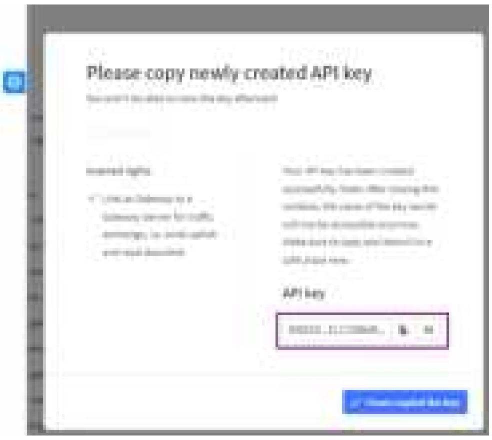
□ Read gateway traffic

□ Store secrets for a gateway

Create API key

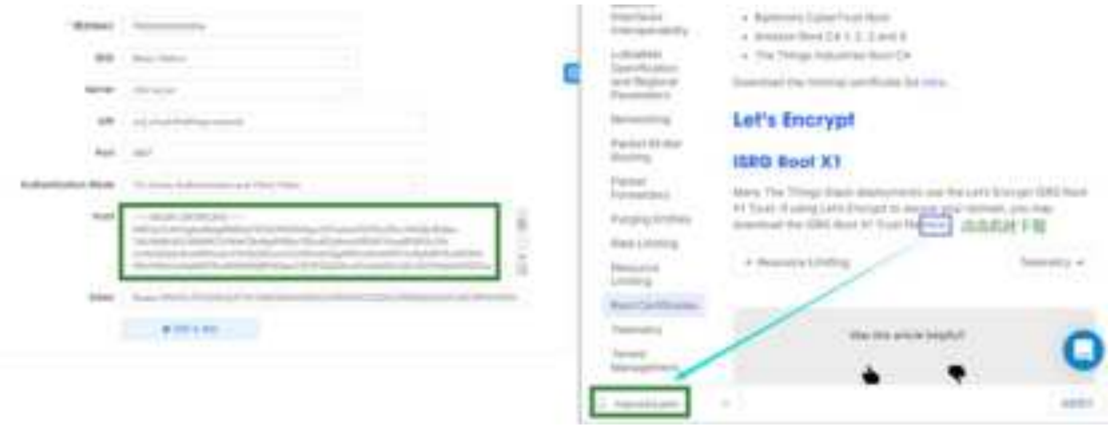
❖ Token Specification： token = Bearer + Space + API key, eg. Bearer NNSXS...



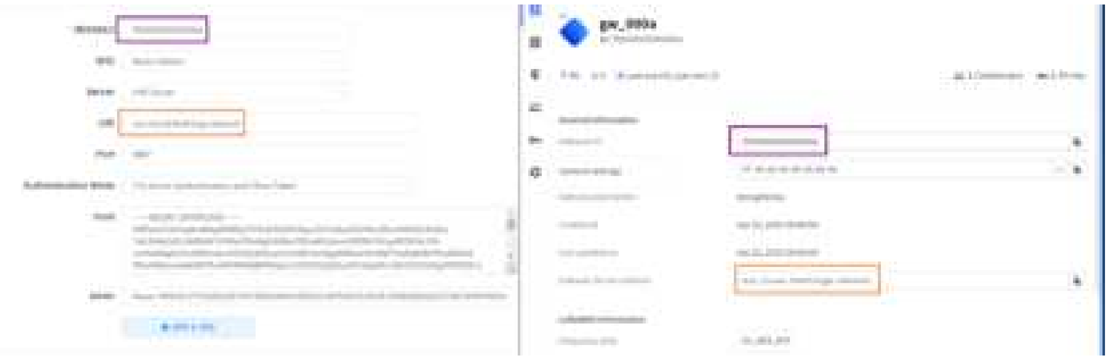


❖ Trust Explanation: This content is from the downloaded file "isrgrootx1.pem" from

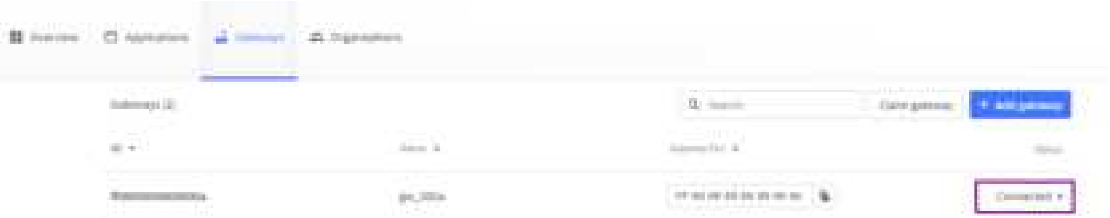
the TTN platform. The file download path is:
https://www.thethingsindustries.com/docs/reference/root-certificates/#lets-encrypt



- URI Configuration



- Port Configuration: Fixed to 8887
- After successfully configuring the gateway, you can check the gateway's connection status to determine whether the connection has been established.

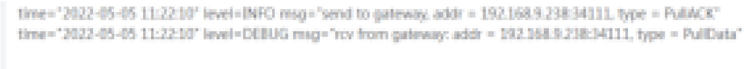


4.1.6 Common Issues

4.1.6.1 Gateway Status

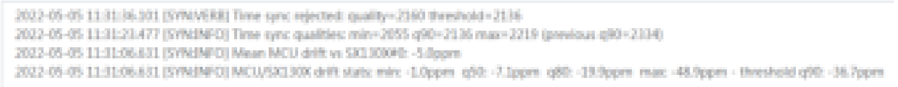
1. Troubleshooting the internal program status of the gateway

- For Semtech UDP GWMP Protocol or Build-in LoRa Server, you can troubleshoot by checking whether the logs show "PullData" and "PullACK" messages. If there is no response within 30 seconds, it indicates a potential issue with the gateway.



```
time="2022-05-05 11:22:10" level=INFO msg="send to gateway. addr = 192.168.9.238:34111, type = PullACK"
time="2022-05-05 11:22:10" level=DEBUG msg="rcv from gateway: addr = 192.168.9.238:34111, type = PullData"
```

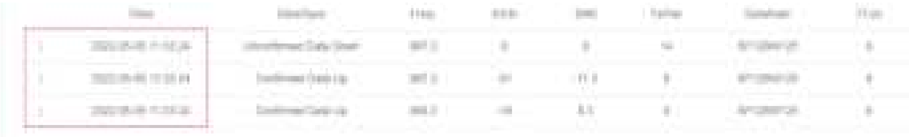
- ❖ When using the Basicstation mode, you can determine by checking whether the logs appear.



```
2022-05-05 11:31:36.101 [SYN:VERB] Time sync rejected: quality=2160 threshold=2136
2022-05-05 11:31:21.477 [SYN:INFO] Time sync qualities: min=2055 q90=2136 max=2219 (previous q90=2134)
2022-05-05 11:31:06.631 [SYN:INFO] Mean MCU drift vs SX130X#0: -5.0ppm
2022-05-05 11:31:06.631 [SYN:INFO] MCU/SX130X drift stats: min: -1.0ppm  q50: -7.1ppm  q80: -19.9ppm  max: -48.9ppm  - threshold q90: -36.7ppm
```

2. Can the gateway receive RF data

- ❖ Open the LoRa Packet Logger and use a device with the same frequency configuration as the gateway to send data or initiate joining. As long as the LoRa module can capture logs, it indicates that the RF module is working properly.



	Time	Data Type	Freq	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	Unconfirmed Data Down	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	Confirmed Data Up	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	Confirmed Data Up	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

4.1.6.2 Communication Device

1. Abnormal reception of uplink data

- ❖ Antenna confirmation, are the antenna frequency bands of the gateway and terminal correct? Is the antenna properly installed? Is the gateway feed line correctly installed?
- ❖ Frequency confirmation, compare the frequency points configured on the device with the frequency points configured on the gateway to see if they match.
- ❖ Open the LoRa packet logger on the gateway, have the terminal send data or initiate the joining process, and check if the gateway can listen to the terminal's data.

2. Cannot receive downlink data

- ❖ Antenna Confirmation: Is the antenna frequency band of the gateway and terminal correct? Is the antenna properly installed? Is the gateway's feeder line correctly installed?
- ❖ Check Packet Logger: Verify if there are logs of downlink data in the packet logger.
  - ■ Class A devices need to wait for an uplink from the device before sending downlink data.
  - ■ Class C devices will send data immediately.
- ❖ Make sure that the frequency and data rate you are sending on match the frequency and data rate the device is listening on. (For the Four-Faith modules, you can set DBL=2 to view this information.)
- ❖ Is the device type consistent?
  - ■ For devices in Class A, if the server is in Class C and sends data immediately, but the device is not in its receive window, this can result in data loss.
  - ■ For devices in Class C, if the server is in Class A and sends data, the device won't immediately receive it. The server needs to wait for the device to send an uplink before it can transmit the downlink. If the device doesn't send an uplink, it won't receive the downlink. After adjusting the device or server's class type, the device needs to be rejoined to synchronize the settings.

4.1.6.3 Device Joining Abnormality

- First, check whether the gateway can receive the joining request packet initiated by the device. If it cannot be received, please refer to the "Communication with Devices Troubleshooting" section.
- Embedded NS
  - Check whether the device has been registered in the embedded NS or if the automatic device addition feature has been enabled.
  - Automatic device addition requires verifying whether the AppEUI and AppKey match.
  - For devices that have already been registered, it's necessary to check if the AppKey matches.
- Non-Embedded NS Server
  - Check if the device has been added to the platform.
  - Verify if the device's AppEUI and AppKey match. The AppKey is a mandatory validation field, while the validation of AppEUI depends on the platform's requirements.
- If the gateway can see the Join Accept downlink packet but the device doesn't receive it, verify whether the device's frequency band matches that of the Network Server (NS). If they don't match, it can result in a mismatch between the listened frequency or data rate and the downlink, causing the data to not be received properly.

Note: The failure of joining is not related to the inconsistency in device types. For example, if the device is classA and the server is classC, the joining can still succeed.

4.1.6.4 Customer Platform Integration

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

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

- TCP Type
  - Check if the corresponding TCP connection switch is turned on.
  - Verification of the server's address and port
  - Check Corresponding Connection Status

4.1.6.5 base64 Encoding and Decoding

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

❖ It mainly involves different data types, resulting in different encoding and decoding outcomes, such as text (strings) or Hex (hexadecimal). The configuration for encoding and decoding can be found in the advanced settings shown in the above image.

❖ Encoding: (When sending downlink data, the data needs to be encoded in base64 format.)

- ■ Text Type （1234 → MTIzNA==）

- ■ Hex Type (0x1234 → EjQ=)

❖ Decoding: (The content of the 'data' field in the upstream push data needs to be decoded from base64 to actual content)

- ■ Text Type（MTIzNA== → 1234）

- ■ Hex Type (EjQ= → 0x1234)