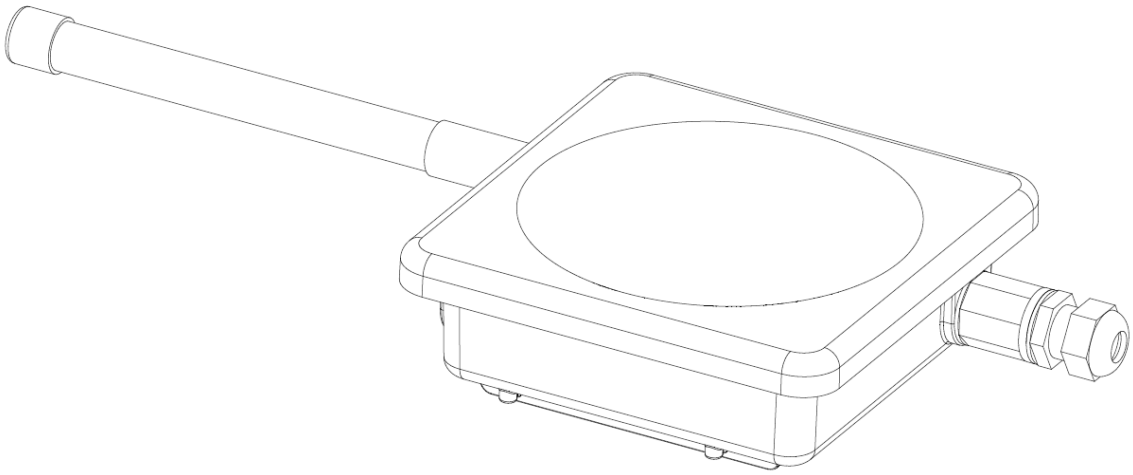


ED-GWL2110

An outdoor gateway based on Raspberry Pi CM4

EDA Technology Co.,Ltd
July 2024

Copyright Statement

Disclaimers

EDA Technology Co., Ltd does not guarantee that the information in this manual is up to date, correct, complete or of high quality. EDA Technology Co., Ltd also does not guarantee the further use of this information. If the material or non-material related losses are caused by using or not using the information in this manual, or by using incorrect or incomplete information, as long as it is not proved that it is the intention or negligence of EDA Technology Co., Ltd, the liability claim for EDA Technology Co., Ltd can be exempted. EDA Technology Co., Ltd expressly reserves the right to modify or supplement the contents or part of this manual without special notice.

Contents

1 Product Overview

ED-GWL2110 is an outdoor gateway based on Raspberry Pi CM4. The whole machine is sealed with all-aluminum alloy outer box, which has good waterproof, moisture-proof, insect-proof and lightning-proof performance. It supports LoRa modules with different frequency bands (external antennas with different frequency bands are required); It supports optional 4G module to ensure that outdoor equipment can upload and download data normally. The device has on-board GNSS module, which can easily meet the positioning requirements. The watchdog module is provided, which can effectively prevent the device from being stuck and greatly increase the stability of the equipment operation. Moreover, this device is equipped with a special encryption chip, which is mounted on the I2C bus to ensure the information security of the device. The device is also equipped with RTC module to ensure the reliability.

1.1 Target Application

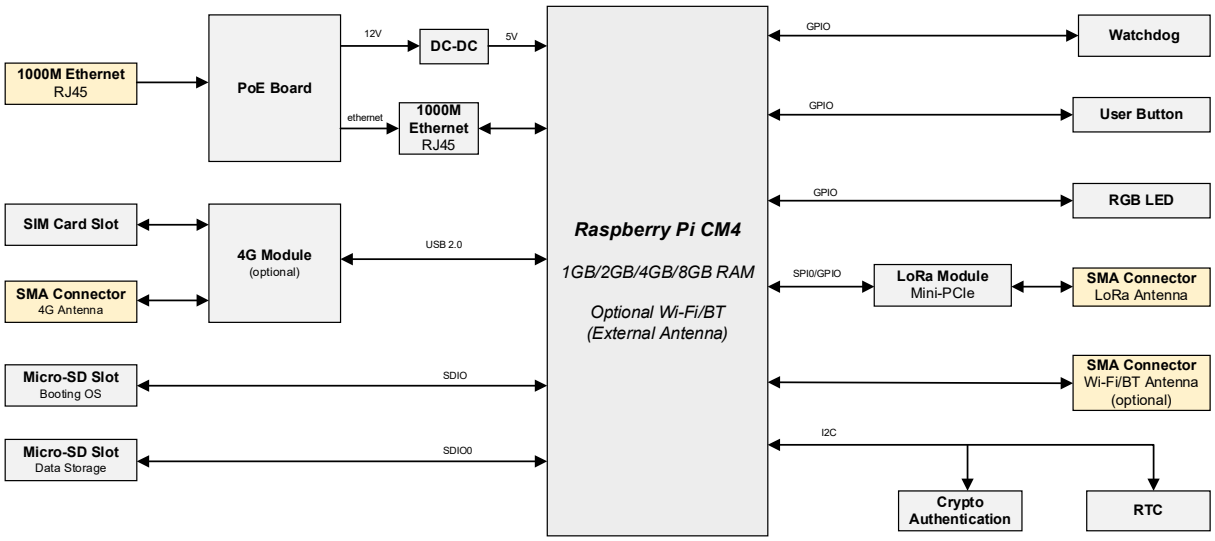
- LoRa intelligent gateway
- Smart manufacturing
- Smart city
- Smart transportation

1.2 Specifications and Parameters

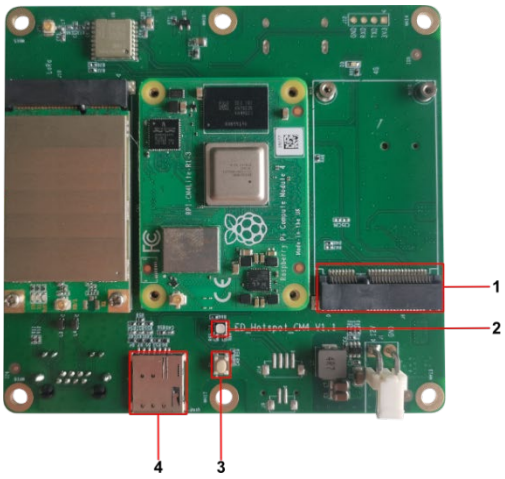
Function	Description
CPU	Broadcom BCM2711, quad core Arm Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	Options for 1GB, 2GB, 4GB, 8GB LPDDR4-3200 SDRAM
SD Card	Options for 32GB and 64GB SD card. It supports booting the OS from SD card.
Ethernet	1x Gigabit Ethernet
Wi-Fi/BT (optional)	● 2.4GHz&5GHz dual Wi-Fi, compatible with IEEE 802.11 b/g/n/ac ● Bluetooth 5.0, support BLE
4G (optional)	Support various 4G LTE modules
LoRa	Compatible with LoRaWAN protocol, support 3 frequency bands ● 868MHz (EU868) ● 915MHz (US915) ● 470MHz (CN470)
GNSS	Built-in GNSS, support multi-satellite system ● GPS L1 C/A: 1575.42 ±1.023 MHz ● BeiDou B1I: 1561.098 ±2.046 MHz ● GLONASS L1: 1597.78~1605.66 MHz
Internal IO	1x Serial (TTL)，available for the system default console ● 1x User-defined button ● 1x RGB 3-color LED

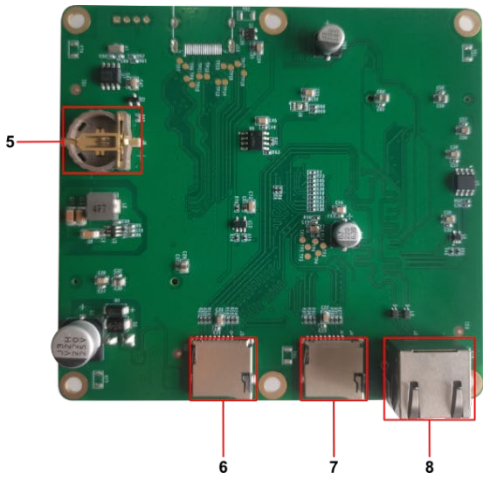
Function	Description
	● 1x RTC battery base, using for installing CR1220 battery ● 1x Nano SIM card slot ● 2x Micro-SD card slot
Expansion Performance	● Support watchdog function to prevent the system from being stuck ● On-board encryption chip
Power Input	PoE power supply, support 802.3af standard
Dimensions	194.2mm(W) x 194.2mm(D) x 65 mm (H)
Case	Cast aluminum waterproof shell, IP65 waterproof grade
Working Temperature	-25°C ~ 60°C

1.3 System Diagram



1.4 Internal IO



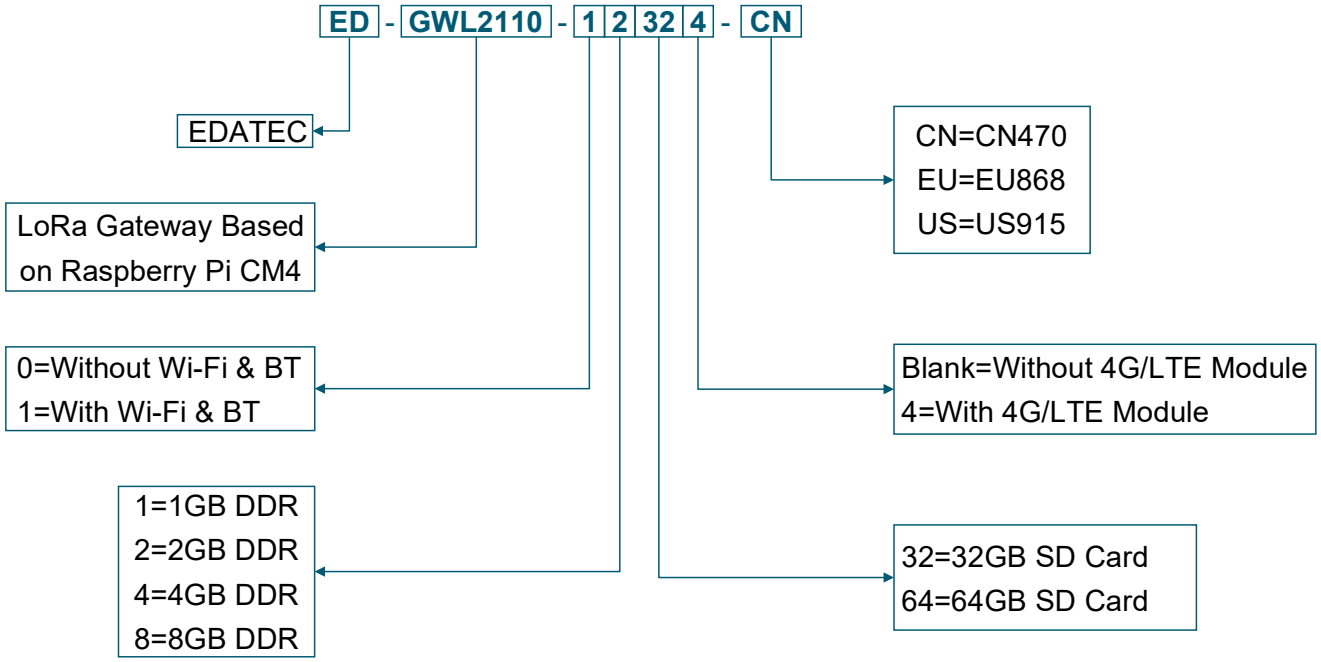


NO.	Function Description
1	4G mini-PCIe Connector
2	RGB LED
3	User Button
4	Nano SIM Card Slot
5	RTC Battery Base
6	Micro-SD Card Slot (User Data Storage)
7	Micro-SD Card Slot (booting OS)
8	Gigabit Ethernet

1.5 Packing List

- 1x ED-GWL2110 Unit
- 1x LoRa Antenna
- [optional Wi-Fi/BT Version]1x 2.4GHz/5GHz Wi-Fi/BT Antenna
- [optional 4G Version]1x 4G/LTE Antenna

1.6 Order Code



Example

P/N: **ED-GWL2110-12324-CN**

Configuration: An outdoor light gateway based on Raspberry Pi CM4, with Wi-Fi & Bluetooth, 2GB DDR, 32GB SD card, 4G and CN470 LoRa frequency.

2 Quick Start

2.1 Equipment List

- 1x ED-GWL2110 Unit
- 1x Wi-Fi/BT External Antenna (optional)
- 1x LoRa External Antenna
- 1x 4G Antenna (optional)
- 1x Network Cable

2.2 Hardware Connection

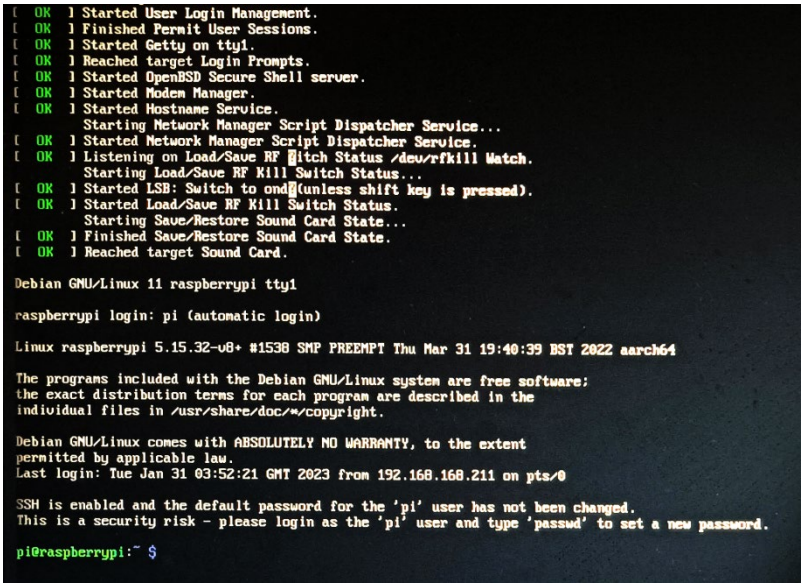
1. Install the Wi-Fi/LoRa/4G external antenna.
2. Insert the network cable into the Ethernet port, and connect the network cable to network devices such as routers/switches with PoE function that can access the Internet.

2.3 First Start

ED-GWL2110 has no power switch. After connecting to PoE power supply, the system will start.

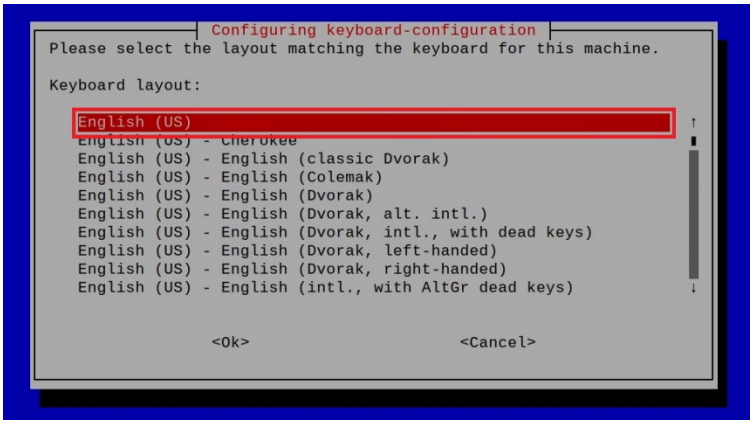
2.3.1 Raspberry Pi OS (Lite)

If you use the OS provided by us, after the system starts, you will automatically log in with the user name pi, and the default password is raspberry.

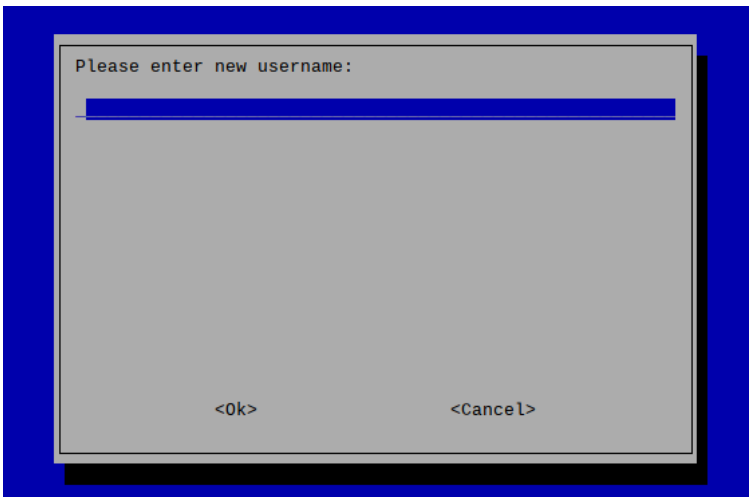


If you use the official OS, and OS is not configured before flashing to SD card, the configuration window will appear when you start it for the first time. You need to configure the keyboard layout, set the user name and the corresponding password in turn.

1. Set the configuration keyboard layout



2. Create a new user name



Then set the password corresponding to the user according to the prompt, and enter the password again for confirmation. At this point, you can log in with the user name and password you just set.

2.3.2 Enable SSH

All the OS we provide have been enabled the SSH. If you use the official OS, you need to use a method to enable the SSH function.

2.3.2.1 Enable SSH via raspi-config command

1. Execute "**sudo raspi-config**" command
2. Choose **3 Interface Options**
3. Choose **I2 SSH**
4. Would you like the SSH server to be enabled? Select **Yes**
5. Choose **Finish**

2.3.2.2 Add Empty File to Enable SSH

Creat an empty file named ssh in the boot partition, and the SSH will be automatically enabled after the device is powered on.

2.3.3 Get The Device IP

- After the device is started, if the display screen is connected, you can use the **ifconfig** command to view the current device IP.
- After the device is started, if there is no display connected, you can check the IP assigned to the device through the router.
- After the device is started, if there is no display screen connected, you can download the Nmap tool to scan the IP under the current network.
  Nmap supports Linux, macOS, Windows and other platforms. If you want to use nmap to scan the network segments from 192.168.3.0 to 255, you can use the following command：

```
nmap -sn 192.168.3.0/24
```

After waiting for a period of time, the result will be output:

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-30 21:19
Nmap scan report for 192.168.3.1 (192.168.3.1)
Host is up (0.0010s latency).
MAC Address: XX:XX:XX:XX:XX:XX (Phicomm (Shanghai))
Nmap scan report for DESKTOP-FGEOUUK.lan (192.168.3.33)
Host is up (0.0029s latency).
MAC Address: XX:XX:XX:XX:XX:XX (Dell)
Nmap scan report for 192.168.3.66 (192.168.3.66)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 11.36 seconds
```

3 Software Operation Guide

3.1 Button

ED-GWL2110 contains a user-defined button inside the device, which is connected to the GPIO23 pin of CPU. It is at a high level by default. When the button is pressed, the pin is at a low level.

You can use the **raspi-gpio** command to test.

- Query the GPIO23 pin when the button is not pressed.

```
raspi-gpio get 23
GPIO 23: level=1 fsel=0 func=INPUT
```

Level of 1 indicates that the GPIO23 pin is high.

- Query the GPIO23 pin When the button is pressed.

```
raspi-gpio get 23
GPIO 23: level=0 fsel=0 func=INPUT
```

Level of 0 indicates that the GPIO23 pin is low.

3.2 LED Indicator

ED-GWL2110 includes an RGB 3-color LED indicator, and the corresponding GPIO pins are as follows:

RGB LED PIN	GPIO
Blue	GPIO16
Green	GPIO20
Red	GPIO21

When the GPIO output is low, the corresponding LED is valid.
You can use the **raspi-gpio** command to operate, and the configuration parameter is **op**, which means output setting, **dl** setting pin is low level, and **dh** setting pin is high level.

The LED is displayed in blue

```
sudo raspi-gpio set 16 op dl
sudo raspi-gpio set 20 op dh
sudo raspi-gpio set 21 op dh
```

The LED is displayed in green

```
sudo raspi-gpio set 16 op dh
sudo raspi-gpio set 20 op dl
sudo raspi-gpio set 21 op dh
```

The LED is displayed in red

```
sudo raspi-gpio set 16 op dh
sudo raspi-gpio set 20 op dh
sudo raspi-gpio set 21 op dl
```

The LED is displayed in yellow

```
sudo raspi-gpio set 16 op dh
sudo raspi-gpio set 20 op dl
sudo raspi-gpio set 21 op dl
```

3.3 Ethernet Configuration

ED-GWL2110 includes one adaptive 10/100/1000M Ethernet interface.

The official OS of Raspberry Pi uses **dhcpcd** as the network management tool by default.

Static IP can be set by modifying "**/etc/dhcpcd.conf**". For example, eth0 can be set, and users can set wlan0 and other network interfaces according to their different needs.

```
interface eth0
static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

3.4 LTE 4G (optional)

Before using 4G, we need to add our APT library first. All the OS given by our company have been added with this library, so users don't need to add it manually.

```
sudo apt update
sudo apt install ed-ec20-qmi
```

Automatic dialing is not started by default. If users want to start automatic dialing, they need to enable LTE "**lte-reconnect.service**" service.

```
sudo systemctl enable lte-reconnect.service
sudo systemctl start lte-reconnect.service
```

After successful dialing, you can use the **ifconfig** command to see the wwan0 network port.

If you need to set APN additionally, you need to modify the dialing command in "**/usr/share/ed-ec20-qmi/LTE-reconnect.sh**".

```
$BSP_HOME_PATH/quectel-CM -4 -f $LOGFILE &
```

The "quectel-CM" dialing configuration information is as follows:

```
$BSP_HOME_PATH/quectel-CM -4 -f $LOGFILE -s <APN> &
```

Restart "**lte-reconnect.service**" after setting.

```
sudo systemctl restart lte-reconnect.service
```

Reset 4G module

```
raspi-gpio set 10 pd
raspi-gpio set 10 op dl
sleep 0.5
raspi-gpio set 10 dh
sleep 0.5
raspi-gpio set 10 dl
```

3.5 Wi-Fi

ED-GWL2110 supports 2.4GHz&5GHz IEEE 802.11 b/g/n dual-band Wi-Fi.

The official OS of Raspberry Pi uses dhcpcd as the network management tool by default.

1. Execute "**sudo raspi-config**" command.
2. Choose **1 System Options**
3. Choose **S1 Wireless LAN**
4. Select your country in the "Select the country in which the pi is to be used" window, and then select OK. This prompt only appears when setting up Wi-Fi for the first time.
5. Please enter SSID, input Wi-Fi SSID name.
6. Please enter passphrase. Leave it empty if none, input password and then restart the device。

3.6 Bluetooth

ED-GWL2110 supports Bluetooth 5.0 and Bluetooth Low Energy (BLE). The Bluetooth function is enabled by default.

Bluetoothctl can be used to scan, pair and connect Bluetooth devices. Please refer to the ArchLinux-Wiki-Bluetooth guide to configure and use Bluetooth.

3.6.1 Basic Configuration Commands

Command	Function Description
bluetoothctl scan on	Enable Bluetooth scanning
bluetoothctl scan off	Disable Bluetooth scanning
bluetoothctl discoverable on	Enable Bluetooth discovery (which can be discovered by the other party)
bluetoothctl discoverable off	Disable Bluetooth discovery
bluetoothctl trust *device_MAC*	Trust device
bluetoothctl connect *device_MAC*	Connect device
bluetoothctl disconnect *device_MAC*	Disconnect device

3.6.2 Configuration Example

This chapter introduces how to configure Bluetooth through a configuration example.

Preparation:

The Bluetooth to be paired has been enabled and its name has been determined.

Steps:

1. Enter the Bluetooth view.

   sudo bluetoothctl

2. Enable bluetooth.

   power on

3. Scan Bluetooth device.

   scan on

   Returned display information:
   Discovery started
   [CHG] Controller B8:27:EB:85:04:8B Discovering: yes
   [NEW] Device 4A:39:CF:30:B3:11 4A-39-CF-30-B3-11

4. Find the name of the turned-on Bluetooth device.

   devices

   Returned display information:
   Device 6A:7F:60:69:8B:79 6A-7F-60-69-8B-79
   Device 67:64:5A:A3:2C:A2 67-64-5A-A3-2C-A2
   Device 56:6A:59:B0:1C:D1 Lefun
   Device 34:12:F9:91:FF:68 test

5. Pairing target devices.

   pair *34:12:F9:91:FF:68*

   34:12:F9:91:FF:68 is target device's device_MAC

   Returned display information:
   Attempting to pair with 34:12:F9:91:FF:68
   [CHG] Device 34:12:F9:91:FF:68 ServicesResolved: yes
   [CHG] Device 34:12:F9:91:FF:68 Paired: yes
   Pairing successful

 **TIP**:

The Bluetooth device to be connected also needs to confirm the pairing request, otherwise the pairing will fail.

6. Add as trusted device.

   trust *34:12:F9:91:FF:68*

   34:12:F9:91:FF:68 is target device's device_MAC

   Returned display information:
   [CHG] Device 34:12:F9:91:FF:68 Trusted: yes
   Changing 34:12:F9:91:FF:68 trust succeeded

3.7 SD Card Extended Storage

3.7.1 Mount

NOTE:

Only Lite version system need to manually unmount the storage device.

You can install the storage device in a specific folder location. It is usually done in the “/mnt” folder, such as “/mnt/mydisk”. Please note that the folder must be empty.

1. Insert the SD card into the SD card slot on your device.
2. Execute the following command to list all disk partitions:
   ```
   sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
   ```
   Using mount points “/” and “/boot”. Your storage device will appear in this list, along with any other connected storage devices.
3. Use the Size, Label and Model columns to identify the name of the disk partition that points to your storage device. For example, “sda1”.
4. Run the following command to get the location of the disk partition:
   ```
   sudo blkid
   ```
   For example, display “/dev/sda1”
5. Create a target folder as the mount point of the storage device. The mount point name used in this example is mydisk. You can specify a name of your choice:
   ```
   sudo mkdir /mnt/mydisk
   ```
6. Mount the storage device at the mount point you created:
   ```
   sudo mount /dev/sda1 /mnt/mydisk
   ```
7. Verify that the storage device has been successfully mounted by listing the following:
   ```
   ls /mnt/mydisk
   ```

3.7.2 Unmount

When the device is turned off, the system will unmount the storage device so that it can be pulled out safely. If you want to uninstall the device manually, you can use the following command:

```
sudo umount /mnt/mydisk
```

If you receive a "destination busy" error, it means that the storage device has not been unmounted. If no error is displayed, you can safely unplug the device now.

3.7.3 Set Automatic Mount in The Command Line.

You can modify the **fstab** setting to mount automatically.

1. First, you need to get the disk UUID.
   ```
   sudo blkid
   ```
2. Find the UUID of the mounted device, such as 5C24-1453.
3. Open the fstab file
   ```
   sudo nano /etc/fstab
   ```
4. Add the following to the fstab file
   ```
   UUID=5C24-1453 /mnt/mydisk fstype defaults,auto,users,rw,nofail 0 0
   ```
   Replace fstype with the type of your file system, which you can find in step 2 of "Mounting storage devices" above, for example, ntfs.
5. If the file system type is FAT or NTFS, add umask = 000 immediately after nofail, which will allow all users to have full read/write access to every file on the storage device.

Information about more fstab commands can be viewed using **man fstab** command.

3.8 RTC

The default shipping system image will integrate the RTC automatic synchronization service we wrote, so guests can automatically synchronize the clock without setting it, and can use RTC without feeling. The general principle is:

1. When the system is turned on, the service automatically reads the saved time from RTC and synchronizes it to the system time.
2. If there is an Internet connection, the system will automatically synchronize the time from the NTP server and update the local system time with Internet time.
3. When the system is shut down, the service automatically writes the system time into RTC and updates the RTC time.
4. Because button cell is installed, although the equipment is powered off, RTC is still working to time.

In this way, we can ensure that our time is accurate and reliable.

WARN：If it is the first time to boot, because there is no effective time in RTC, synchronization may fail, so just restart it directly. When rebooting, the system time will be written into RTC for normal use.

If you don't want to use this service, you can turn it off manually:

```
sudo systemctl disable rtc
sudo reboot
```

Re-enable this service:

```
sudo systemctl enable rtc
sudo reboot
```

Time to read RTC manually:

```
sudo hwclock -r
2022-11-09 07:07:30.478488+00:00
```

Manually synchronize RTC time to the system:

```
sudo hwclock -s
```

Write the system time into RTC:

```
sudo hwclock -w
```

Trouble Shooting

Please first check whether there is an rtc device (/dev/rtc0) mounted:

```
ls /dev/rtc0
```

If not, maybe you used the official standard system, but didn't install our BSP package. Please install ed-rtc package to enable RTC automatic synchronization.

Other possible checkpoints:

- Is CR1220 installed in button cell?
- NTP network time protocol, you need to connect to the Internet to synchronize the time automatically, and you need to open the port (UDP, 123), otherwise the synchronization will fail.

3.9 Watch Dog

ED-GWL2110 is equipped with watchdog module to prevent the system from being stuck.

Watchdog logic table

GPIOx	pin	H/L	H/L	H/L
GPIO17	OE	H	H	L
GPIO16	A	H	L	X
output	Y	H	L	Z

You need to install ed-gwl2100-wdt.dtbo to use watchdog module. File link: todo.
Users need to put it on the device after downloading it. You can use scp command to copy the file to the device directory:

```
scp /path/ed-gwl2100-wdt.dtbo pi@ip-address:/home/pi
sudo cp /home/pi/ed-gwl2100-wdt.dtbo /boot/overlays
sudo chmod +x /boot/overlays/ed-gwl2100-wdt.dtbo
```

And add the following at the end of /boot/config.txt:

```
sudo nano /boot/config.txt

dtoverlay=ed-gwl2100-wdt
```

3.10 GNSS

ED-GWL2110 gateway integrates L76K GPS module, which is connected with UART0 serial port of CPU. The module reports GNSS information through NMEA 0183 general protocol output statement.

3.10.1 Pin Configuration

The WakeUp signal of L76K GPS module is connected to GPIO4. If the pin module is pulled down, it will enter standby mode, and if it is pulled up or suspended, it will return to continuous mode. The Reset signal is connected to GPIO5. Pulling this pin low for 100ms will reset the module. SET signal is connected with GPIO6, which is used to configure the satellite combination. When the pin is suspended or high level, the satellite combination is GPS and Beidou, and when the pin is low level, the satellite combination is GPS and GLONASS.

#	Signal	CM4 Pinout
1	GPS_WakeUp	GPIO4
2	GPS_Reset	GPIO5
3	GPS_Set	GPIO6

3.10.2 Modify config.txt to Enable Serial Port.

```
sudo nano /boot/config.txt
```

Add at the end

```
[all]
enable_uart=1
```

3.10.3 Check GNSS information

```
sudo cat /dev/ttyS0
```

Display GPS data as follows:

```
$BDGSV,3,1,11,04,29,117,20,10,,,19,16,75,160,,24,51,328,,0*4C
$BDGSV,3,2,11,25,,,27,26,,,21,34,12,198,,35,45,063,,0*76
$BDGSV,3,3,11,39,62,159,17,41,,,25,59,44,137,,0*7A
$GNRMC,053557.000,A,3027.47401,N,11424.34027,E,1.17,186.64,070223,,,A,V*05
$GNVTG,186.64,T,,M,1.17,N,2.17,K,A*2D
$GNZDA,053557.000,07,02,2023,00,00*4F
$GPTXT,01,01,01,ANTENNA OPEN*25
$GNGGA,053558.000,3027.47438,N,11424.34119,E,1,07,1.5,75.0,M,-14.1,M,,*52
$GNGLL,3027.47438,N,11424.34119,E,053558.000,A,A*4F
$GNGSA,A,3,07,08,16,31,195,,,,,,,2.1,1.5,1.5,1*05
$GNGSA,A,3,04,39,,,,,,,,,,,2.1,1.5,1.5,4*39
```

```
$GPGSV,3,1,12,04,54,241,16,07,19,314,15,08,63,208,15,09,38,291,,0*67
$GPGSV,3,2,12,16,51,029,17,18,07,046,,21,08,175,,26,24,063,,0*6A
$GPGSV,3,3,12,27,77,065,,31,09,122,22,194,61,058,,195,46,125,21,0*66
```

NMEA 0183 general statement is described as follows:

$BDGSV Visible Beidou satellite information
$GNRMC Recommended GNSS data
$GNVTG Relative ground heading and speed information
$GNZDA Time and date, UTC format
$GPTXT Text transmission
$GNGGA Multi-satellite joint positioning data
$GNGLL Geographical location, latitude and longitude
$GNGSA Represents GNSS accuracy factor and effective satellite.
$GPGSV Visible GNSS satellite

3.10.4 Use the u-center tool to view positioning information.

3.10.4.1 Installs serial port to network tool ser2net.

```
sudo apt-get update
sudo apt-get install ser2net
```

Enable ser2net service

Ser2net configuration file is/etc/ser2net.yaml. By default, /dev/ttyS0 is configured, baud rate is 9600, and there is no check, and the corresponding TCP port is 2000.

```
connection: &con0096
      accepter: tcp,2000
      enable: on
      options:
         banner: *banner
         kickolduser: true
         telnet-brk-on-sync: true
      connector: serialdev,
                     /dev/ttyS0,
                     9600n81,local
```

3.10.4.2 Checks ser2net Port Forwarding Service.

Use the following instructions to query whether ser2net has started 2000 port forwarding.

```
sudo netstat -ltnp | grep 2000
```

If port forwarding has been started, the following message will be displayed

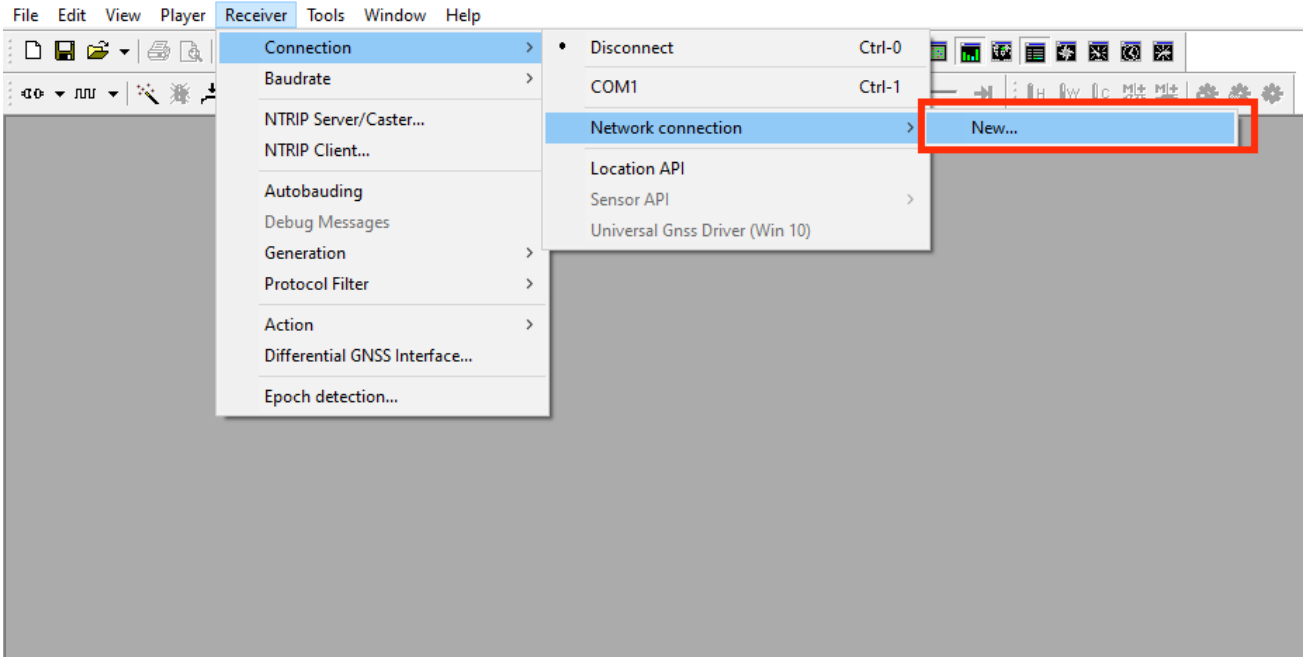
```
tcp6       0      0 :::2000                 :::*                    LISTEN      720/ser2net
```

If not, restart the ser2net service.

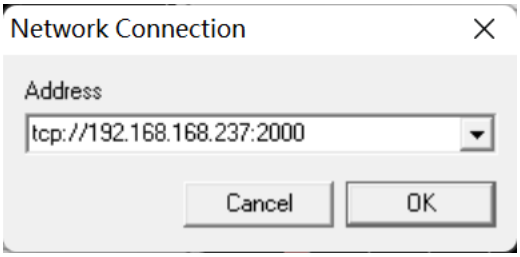
```
sudo systemctl restart ser2net
```

Download and install the u-center tool. If you are prompted that the MSVCR120.dll file is missing, please install vcredist_x86.exe.

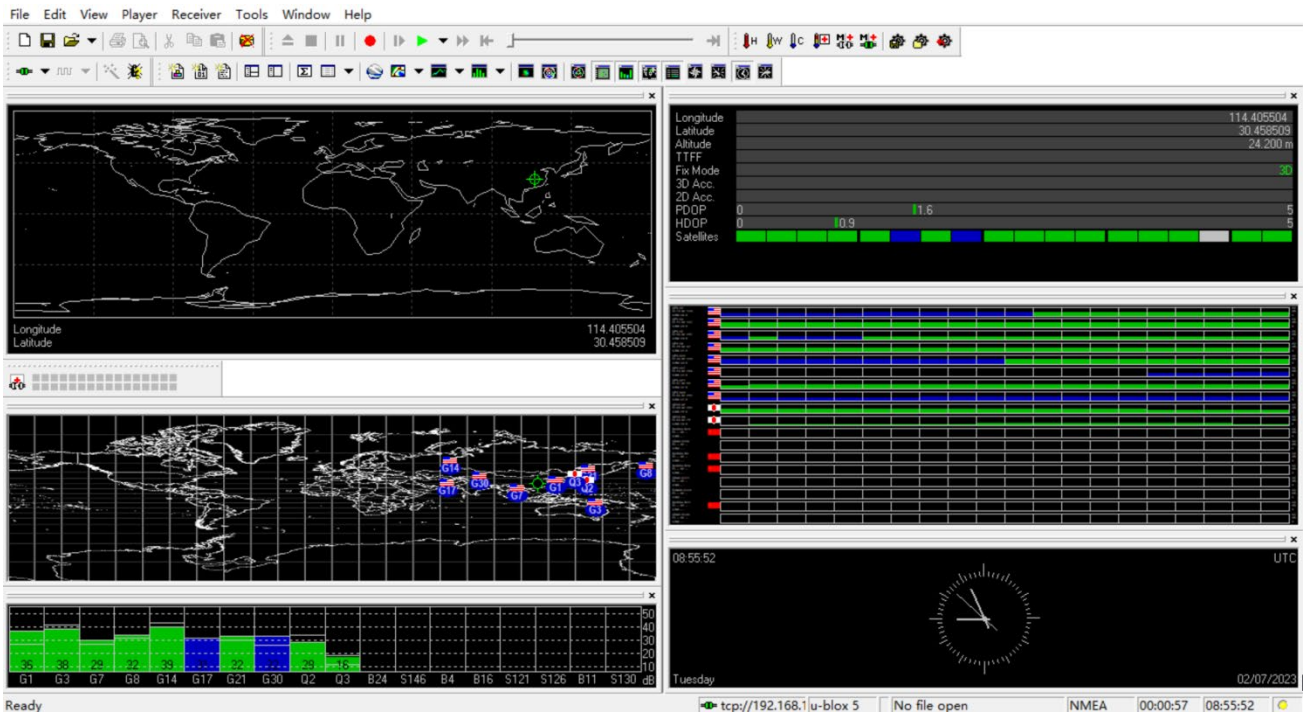
Open u-center，Choose Receiver->Port->Network connection->New…



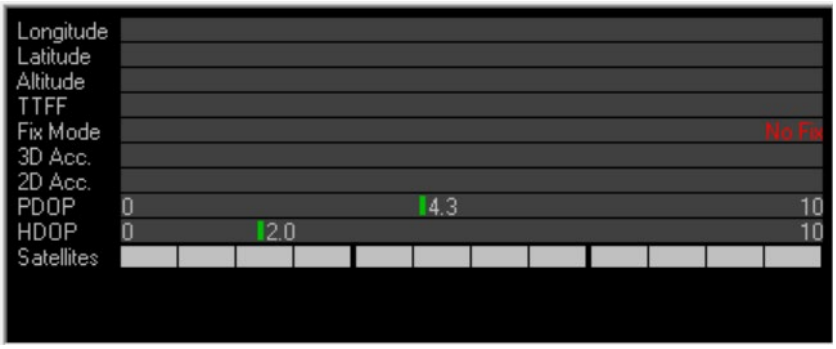
Enter your device IP and port number 2000.



You will see the GPS positioning information immediately after the configuration is completed.



If the Fix Mode is displayed as No Fix, it means that the positioning failed, which is usually caused by the antenna being indoors. Please put the module or antenna outdoors for testing.



NOTE: For the first time, it takes about 30 seconds to locate the module successfully when there is no large building outside. If the weather conditions are bad, it may take longer or it may not be possible to locate it.

3.11 LoRaWAN

ED-GWL2110 supports LoRaWAN open source service platform ChirpStack. Please refer to the following steps for installation and configuration.

3.11.1 Install LoRa Service and ChirpStack Client.

We install it by APT.

- Add edatec APT warehouse

```
$ curl -sS https://apt.edatec.cn/pubkey.gpg | sudo apt-key add -
$ echo "deb https://apt.edatec.cn/raspbian stable main" | sudo tee /etc/apt/sources.list.d/edatec.list
$ sudo apt update
$ sudo apt install -y ed-gwl-pktfwd
```

- Install ChirpStack

```
$ sudo apt install -y apt-transport-https dirmngr
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00
$ echo "deb https://artifacts.chirpstack.io/packages/4.x/deb stable main" | sudo tee
/etc/apt/sources.list.d/chirpstack.list
$ sudo apt update

$ sudo apt install -y chirpstack-gateway-bridge
```

- Modify config.txt

```
[all]
dtparam=i2c_arm=on
dtparam=spi=on

gpio=16=op,dl
gpio=20=op,dl
gpio=21=op,dl
```

Modify /etc/modules，add i2c-dev at the end

```
i2c-dev
```

ED-GWL2110 use i2c-1 and spidev0.0。

3.11.2 Configuring LoRa Service

3.11.2.1 Pktfwd Config

```
# update region
$ cat /etc/ed_gwl/region
EU868     # EU868 / US915
```

pktfwd use 1700 as UDP port

```
$ sudo systemctl restart ed-pktfwd.service
```

3.11.2.2 chirpstack-gateway-bridge Configuration

You can use nano to edit the configuration file chirpstack-gateway-bridge.toml.

```
$ sudo nano /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml
```

```
# This configuration provides a Semtech UDP packet-forwarder backend and
 # integrates with a MQTT broker. Many options and defaults have been omitted
 # for simplicity.
 #
 # See https://www.chirpstack.io/gateway-bridge/install/config/ for a full
 # configuration example and documentation.


 # Gateway backend configuration.
 [backend]
 # Backend type.
 type="semtech_udp"

   # Semtech UDP packet-forwarder backend.
   [backend.semtech_udp]

   # ip:port to bind the UDP listener to
   #
   # Example: 0.0.0.0:1700 to listen on port 1700 for all network interfaces.
   # This is the listener to which the packet-forwarder forwards its data
   # so make sure the 'serv_port_up' and 'serv_port_down' from your
   # packet-forwarder matches this port.
   udp_bind = "0.0.0.0:1700"

 # Integration configuration.
 [integration]
 # Payload marshaler.
 #
 # This defines how the MQTT payloads are encoded. Valid options are:
 # * protobuf:    Protobuf encoding
 # * json:        JSON encoding (easier for debugging, but less compact than 'protobuf')
 marshaler="protobuf"

   # MQTT integration configuration.
   [integration.mqtt]
   # Event topic template.
   event_topic_template="eu868/gateway/{{ .GatewayID }}/event/{{ .EventType }}"

   # Command topic template.
   command_topic_template="eu868/gateway/{{ .GatewayID }}/command/#"

   # MQTT authentication.
```

```
[integration.mqtt.auth]
# Type defines the MQTT authentication type to use.
#
# Set this to the name of one of the sections below.
type="generic"

  # Generic MQTT authentication.
  [integration.mqtt.auth.generic]
  # MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or ws)
  server="tcp://127.0.0.1:1883"

  # Connect with the given username (optional)
  username=""

  # Connect with the given password (optional)
  password=""
```

- 'event_topic_template / command_topic_template' needs to modify the prefix with gateway zone.
  Example:

  ```
  event_topic_template="eu868/gateway/{{ .GatewayID }}/event/{{ .EventType }}"
  ```

  If you use the US915 or CN470 module, please change the prefix eu868 to us915_0/cn470_10.

  ```
  event_topic_template="us915_0/gateway/{{ .GatewayID }}/event/{{ .EventType }}"
  ```

- The server address of integration.mqtt needs to be your chirpstack server.

  ```
  $ sudo systemctl restart chirpstack-gateway-bridge.service
  ```

After modify chirpstack-gateway-bridge.toml config，need restart chirpstack-gateway-bridge service.

3.11.2.3 Reboot

```
$ sudo reboot
```

3.11.3 Install ChirpStack Server

To configure a cloud server, docker needs to be installed on the server before configuration.
Install docker：https://docs.docker.com/get-docker/
Install docker-compose

```
sudo apt install docker-compose
```

3.11.3.1 Config chirpstack-docker

We use docker container to deploy ChirpStack server.

```
$ git clone https://github.com/chirpstack/chirpstack-docker.git
```

The dock-combination. yml of chirpstack dock needs to be configured.

```
$ cd chirpstack-docker
$ nano docker-compose.yml
# Remove the chirpstack-gateway-bridge, because we run the bridge on gateway.
```

Delete the red font part.

```
$ nano docker-compose.yml

version: "3"

services:
  chirpstack:
    image: chirpstack/chirpstack:4
    command: -c /etc/chirpstack
    restart: unless-stopped
    volumes:
      - ./configuration/chirpstack:/etc/chirpstack
      - ./lorawan-devices:/opt/lorawan-devices
    depends_on:
      - postgres
      - mosquitto
      - redis
    environment:
      - MQTT_BROKER_HOST=mosquitto
      - REDIS_HOST=redis
      - POSTGRESQL_HOST=postgres
    ports:
      - 8080:8080

  chirpstack-gateway-bridge-eu868:
    image: chirpstack/chirpstack-gateway-bridge:4
    restart: unless-stopped
    ports:
      - 1700:1700/udp
    volumes:
      - ./configuration/chirpstack-gateway-bridge:/etc/chirpstack-gateway-bridge
    depends_on:
      - mosquitto

  chirpstack-rest-api:
```

```
    image: chirpstack/chirpstack-rest-api:4
    restart: unless-stopped
    command: --server chirpstack:8080 --bind 0.0.0.0:8090 --insecure
    ports:
      - 8090:8090
    depends_on:
      - chirpstack

  postgres:
    image: postgres:14-alpine
    restart: unless-stopped
    volumes:
      - ./configuration/postgresql/initdb:/docker-entrypoint-initdb.d
      - postgresqldata:/var/lib/postgresql/data
    environment:
      - POSTGRES_PASSWORD=root

  redis:
    image: redis:7-alpine
    restart: unless-stopped
    volumes:
      - redisdata:/data

  mosquitto:
    image: eclipse-mosquitto:2
    restart: unless-stopped
    ports:
      - 1883:1883
    volumes:
      - ./configuration/mosquitto/mosquitto.conf:/mosquitto/config/mosquitto.conf

volumes:
  postgresqldata:
  redisdata:
```

Start chirpstack service

```
$ docker-compose up -d
```

3.11.3.2 Logs Into chirpstack Service Management Interface.

Enter the server's IP address and port number 8080 in the PC browser, and the login interface will appear when the network is normal.

The default administrator user name and password are as follows:

```
user: admin
psw : admin
```

3.11.4 Adding LoRa Gateway and Terminal

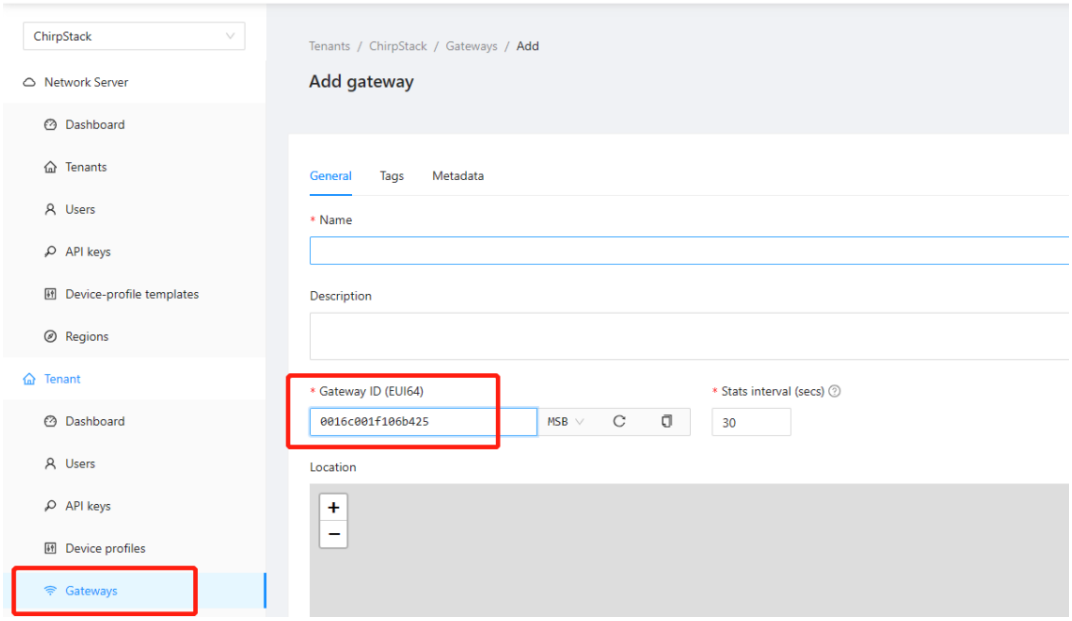
3.11.4.1 Gets LoRa Gateway ID

Execute the following command to get the ID of LoRa gateway. When adding LoRa gateway to chirpstack server, you need to add the corresponding gateway ID.

```
$ /opt/ed-gwl-pktfwd/ed-gateway_id
```

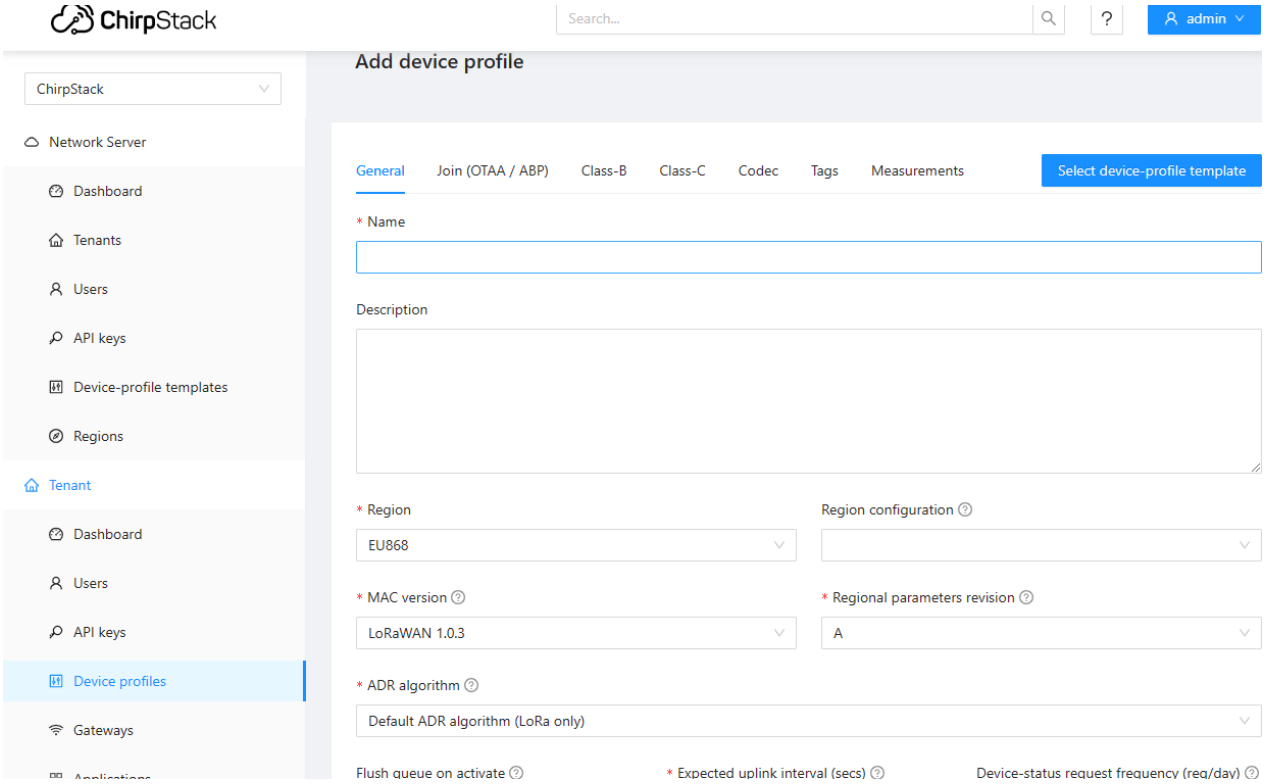
3.11.4.2 Add LoRa Gateway

Open chirpstack management interface in PC browser, click Gateway -> Add gateway, fill in the Gateway ID corresponding to the device, set the Name, and then click Submit. If the network connection is correct, wait a moment to see that the added gateway becomes Online.



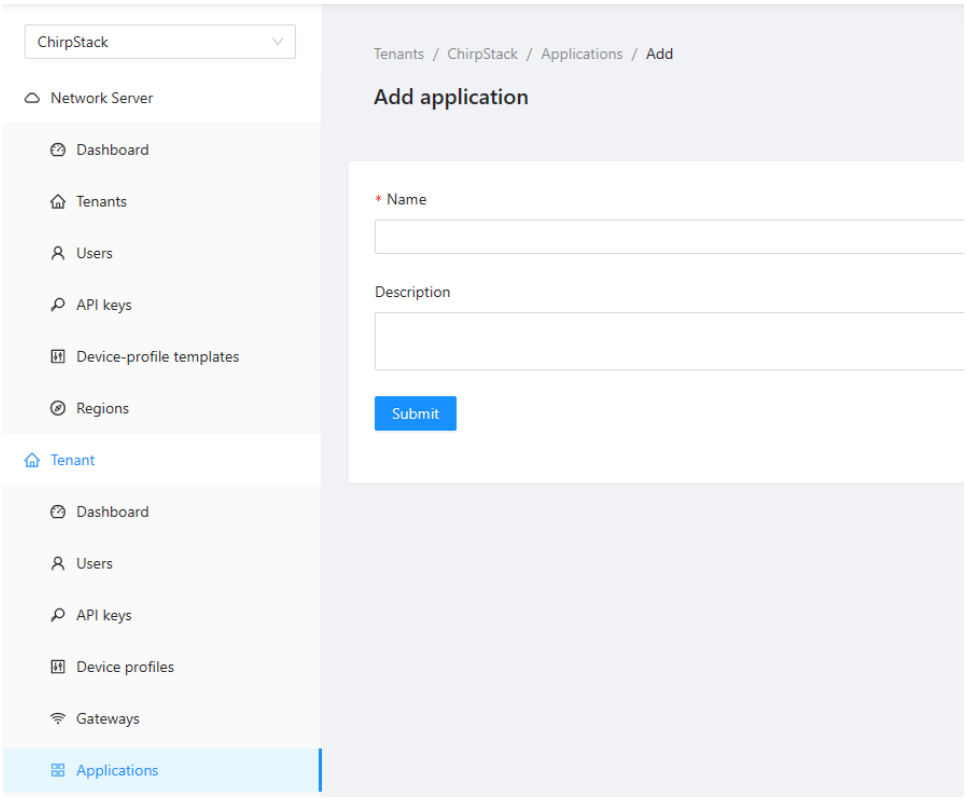
3.11.4.3 Add Device Profile

Click device profile-> add device profile to further improve the device information.



3.11.4.4 Add Application

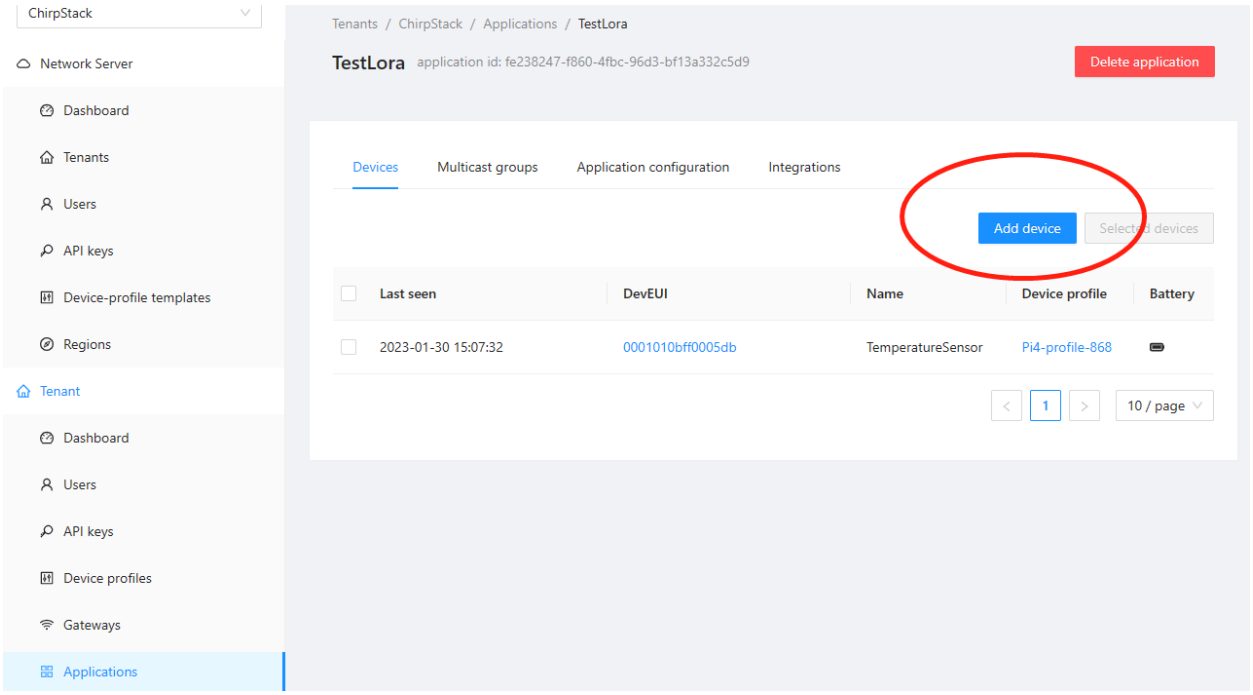
Click Applications -> Add application

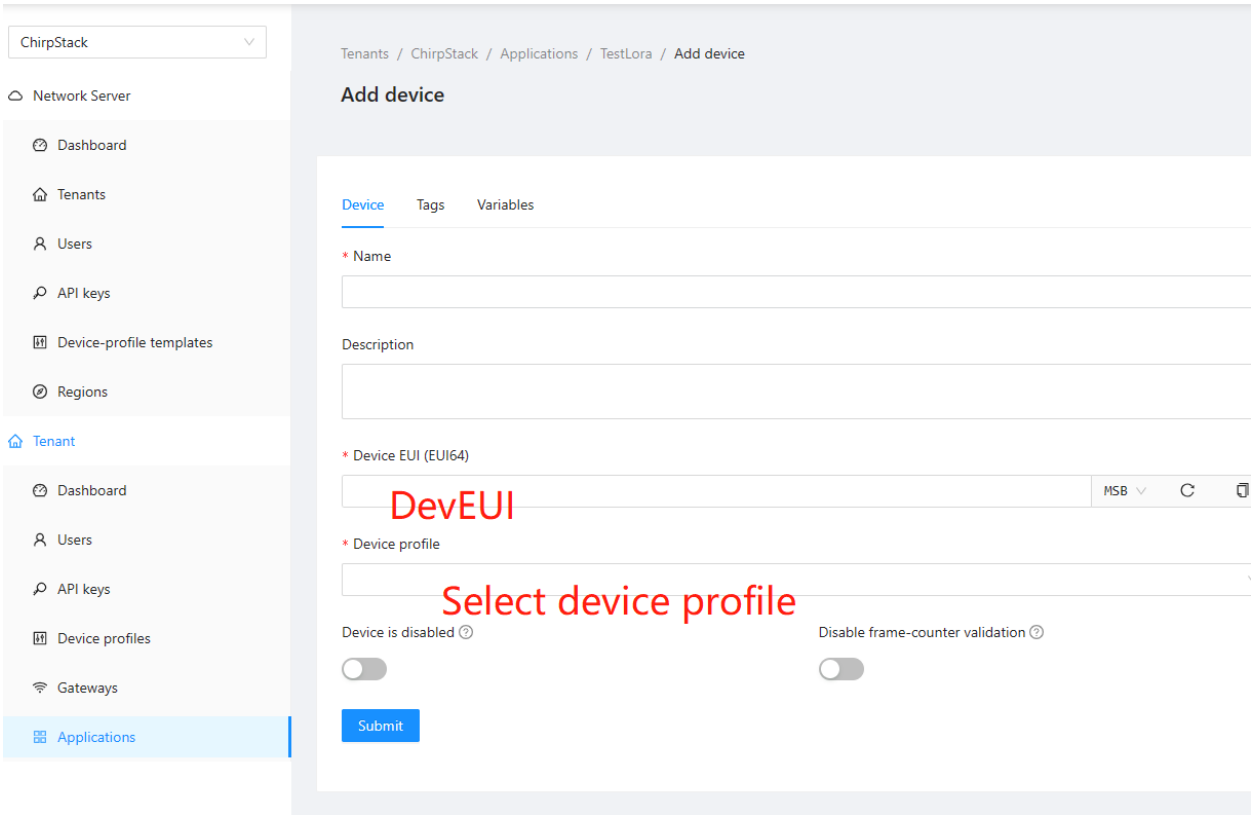


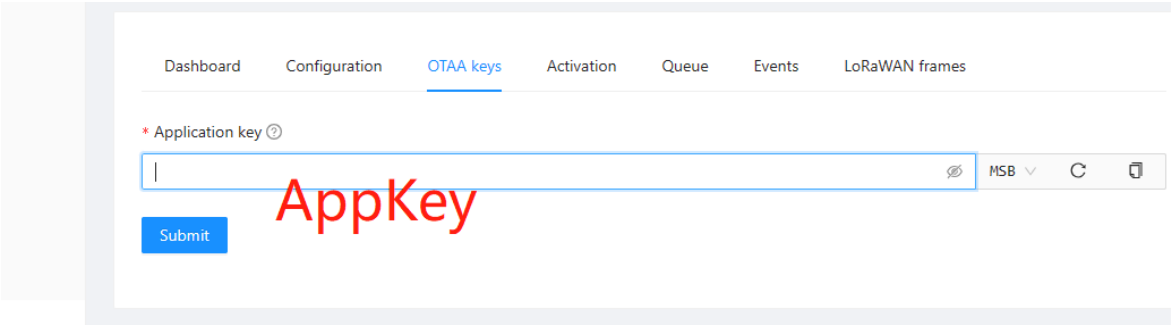
3.11.4.5 Add Device

You should know the DevEUI and AppKey of LoRa terminal products, which are provided by LoRa terminal equipment manufacturers.

Click Application -> your application -> Add device to add LoRa terminal device







Wait a few minutes to see the device become online.

3.12 Encryption chip

ED-GWL2110 is equipped with ATECC608 encryption chip, which is connected to i2c-1 bus, and the default address of the device is 0x60.

atecc: https://github.com/wirenboard/atecc-util

```
atecc -b 1 -c 'serial'
```

4 OS Installation

4.1 Image Download

We have provided the factory image. If the system is restored to factory settings, please click the following link to download the factory image.

Raspberry Pi OS Lite, 32-bit

- Release date: 2023-02-10
- System: 32-bit
- Kernel version: 5.15
- Debian version: 11 (bullseye)
- Downloads: https://1drv.ms/u/s!Au060HUAtEYBgQG59MKsXapwhB5B?e=mGLFx5

4.2 System Flash

ED-GWL2110 uses CM4 Lite version by default, and starts the system through SD card. The system burning of ED-GWL501 means burning SD card.

- Download and install Raspberry Pi Imager or balenaEtcher mirror writing tool.

- Insert the micro SD card into the card reader, and then insert the card reader into the USB port of the computer.
- Open the mirror burning tool and select the mirror you want to burn. The path is the path of the identified mass storage device.
- Click Burn, wait for burn and verification to be completed, and pop up the card reader device.
- Open the top cover of the ED-GWL2110 device and insert the micro SD card with burned image into the corresponding card slot.
- Just re-power the device.

5 FAQ

5.1.1 Default Username and Password

User name：pi
Password： raspberry

6 About Us

6.1 About EDATEC

EDATEC, located in Shanghai, is one of Raspberry Pi's global design partners. Our vision is to provide hardware solutions for Internet of Things, industrial control, automation, green energy and artificial intelligence based on Raspberry Pi technology platform.

We provide standard hardware solutions, customized design and manufacturing services to speed up the development and time to market of electronic products.

6.2 Contact Us

Mail – sales@edatec.cn / support@edatec.cn
Phone – +86-18621560183
Website – https://www.edatec.cn
Address – Building 29, No.1661 Jialuo Highway, Jiading District, Shanghai