



ED-GWL501

一款基于 RASPBERRY PI ZERO 2 W 设计的室内轻网关

上海晶珩电子科技有限公司
2023-02-03

版权声明

ED-GWL501 及其相关知识产权为上海晶珩电子科技有限公司所有。
上海晶珩电子科技有限公司拥有本文件的版权并保留所有权利。未经上海晶珩电子科技有限公司的书面许可，不得以任何方式和形式修改、分发或复制本文件的任何部分。

免责声明

上海晶珩电子科技有限公司不保证本硬件手册中的信息是最新的、正确的、完整的或高质量的。上海晶珩电子科技有限公司也不对这些信息的进一步使用作出保证。如果由于使用或不使用本硬件手册中的信息，或由于使用错误或不完整的信息而造成的物质或非物质相关损失，只要没有证明是上海晶珩电子科技有限公司的故意或过失，就可以免除对上海晶珩电子科技有限公司的责任索赔。上海晶珩电子科技有限公司明确保留对本硬件手册的内容或部分内容进行修改或补充的权利，无需特别通知。

目 录

1	产品概述.....	4
1.1	目标应用.....	4
1.2	规格参数.....	4
1.3	系统框图.....	5
1.4	功能布局.....	5
1.5	包装清单.....	6
1.6	订购编码.....	7
2	快速启动.....	7
2.1	设备清单.....	7
2.2	硬件连接.....	7
2.3	首次启动.....	8
2.3.1	Raspberry Pi OS (Lite).....	8
2.3.2	使能 SSH 功能.....	9
2.3.3	查找设备 IP.....	9
3	接线指南.....	10
3.1	Internal I/O.....	10
3.1.1	micro-SD Card.....	10
4	软件操作指引.....	10
4.1	按键.....	10
4.2	LED 指示.....	11
4.3	以太网配置.....	11
4.4	WiFi.....	12
4.5	蓝牙.....	12
4.5.1	基本用法.....	12
4.5.2	示例.....	13
4.6	GPS.....	14
4.6.1	修改 config.txt 使能串口.....	14
4.6.2	查看 GNSS 信息.....	14
4.6.3	使用 u-center 工具查看定位信息.....	15
4.7	LoRaWAN.....	17
4.7.1	安装 LoRa 服务和 ChirpStack 客户端.....	17
4.7.2	配置 LoRa 服务.....	18
4.7.3	安装 ChirpStack 服务端.....	20
4.7.4	添加 LoRa 网关和终端.....	22
5	操作系统安装.....	25
5.1	镜像下载.....	25
5.2	系统烧录.....	26
6	FAQ.....	26
6.1.1	默认用户名密码.....	26
7	关于我们.....	26
7.1	关于 EDATEC.....	26
7.2	联系方式.....	26

1 产品概述

ED-GWL501 是一款基于 Raspberry Pi Zero 2 W 设计的室内轻网关。本产品使用新一代的 SX1302、SX1303 基带芯片的 LoRa 网关模组，具有传输距离远，节点容量大，接收灵敏度高等特点，此外，本网关性能强劲，结构轻巧，部署简单，可以大大简化和缩短您的开发门槛和设计时间。

1.1 目标应用

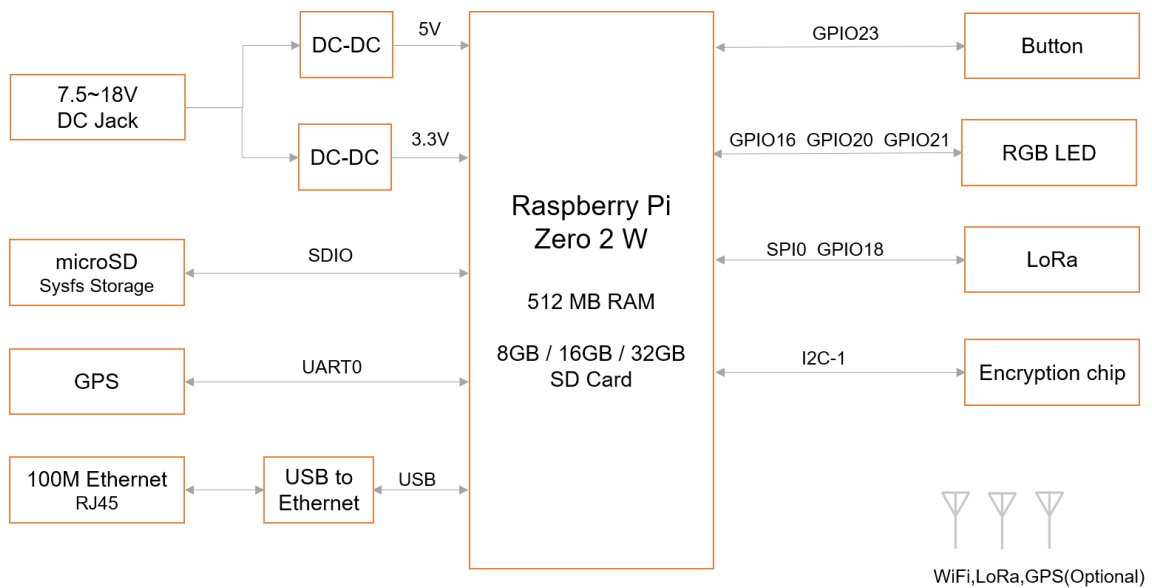
- LoRa 智能网关
- 工业控制
- 智能制造
- 智慧城市
- 智慧交通

1.2 规格参数

功能	参数
CPU	博通 BCM2710A1 四核, ARM Cortex-A53, 1GHz, 64 位 CPU
内存	512MB SDRAM
SD 卡	micro SD 卡
以太网	1x 10/100M 以太网
WiFi / 蓝牙	2.4GHz 802.11 b/g/n WiFi, 蓝牙 4.2, 支持 BLE
LoRa	SX1303+SX1250
	868 ~ 870 MHz (EU868, IN865, RU864)
	902 ~ 928 MHz (US915, AU915, KR920, AS923-1/2/3)
安全	内置 ATECC608A 安全芯片
GPS	支持多卫星系统
	- GPS L1 C/A: 1575.42 ±1.023 MHz
	- BeiDou B1I: 1561.098 ±2.046 MHz
	- GLONASS L1: 1597.78~1605.66 MHz
按键	1x 用户按键
LED 指示灯	1x RGB LED
电源输入	7.5V ~ 18V

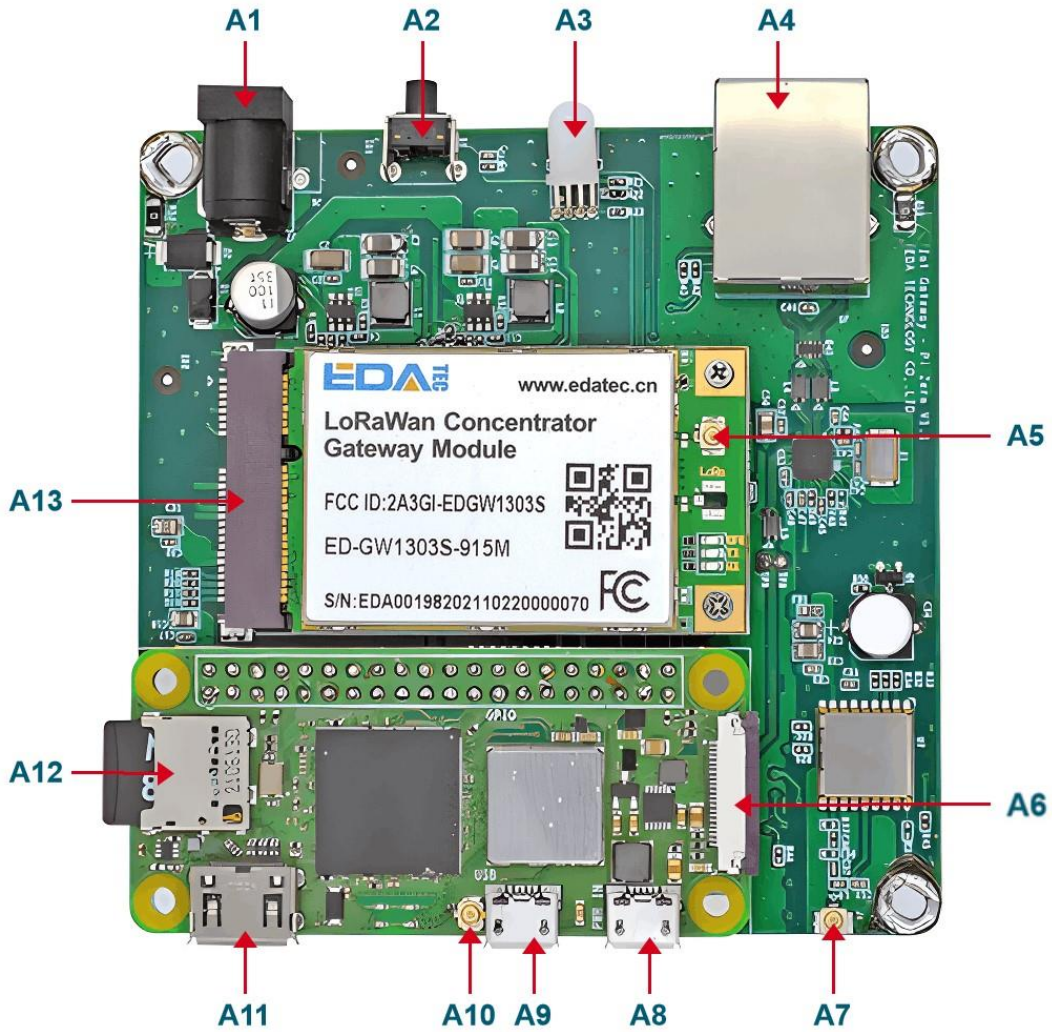
功能	参数
天线配件	WiFi / BT 外置天线, 已与 Raspberry Pi Zero 2 W 一起通过无线认证
	1x 2.5dBi LoRa 天线[可选]
	1x GPS 天线[可选]
工作环境温度	可选-25 ~ 50°C
操作系统	兼容官方 Raspberry Pi OS, 提供 BSP 软件支持包, 并支持 APT 在线安装和更新
软件资源	提供 ChipStack 等 LoRaWAN 网络的示例指引
尺寸	105(长) x 90(宽) x 25(高) mm
认证	CE / FCC
OTA	支持在线更新 BSP

1.3 系统框图



ED-GWL501

1.4 功能布局

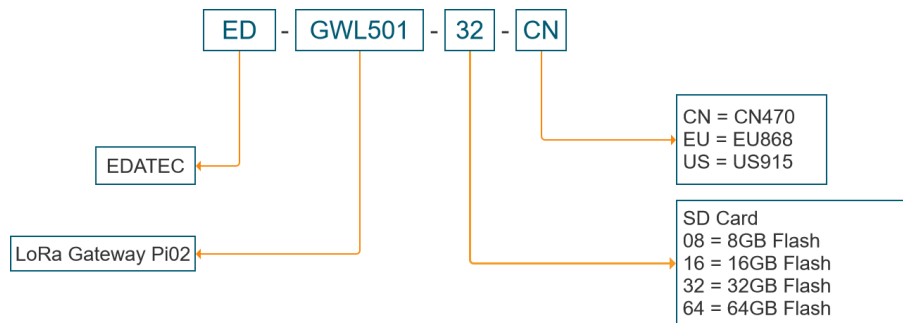


编号	功能描述	编号	功能描述
A1	12V DC 电源插座	A9	USB OTG 接口
A2	按键	A10	WiFi/BT 天线 IPX 接头
A3	RGB LED	A11	Mini HDMI 接口
A4	以太网 RJ45 接口	A12	Micro SD 卡槽
A5	LoRa 天线 IPX 接头	A13	LoRa mini-PCIe 接口
A6	CSI 接口		
A7	GPS 天线 IPX 接头		
A8	Micro USB 供电接口		

1.5 包装清单

- 1x ED-GWL501 主机
- [选配]WiFi / BT 外置天线
- [选配 4G 版本] 1x 4G/LTE 天线

1.6 订购编码



Example

Part# : ED-GWL501-32-CN
Configuration : GWL501 LoRa Gateway
1pcs Raspberry Pi Zero 2 W certified WiFi/Bluetooth Antenna
512MB SDRAM and 32GB SD Card Flash
CN470 LoRa Module

2 快速启动

2.1 设备清单

- 1x ED-GWL501 主机
- 1x WiFi / BT 外置天线
- 1x LoRa 外置天线
- 1x GPS 天线
- 1x 网线
- 1x 12V@2A 电源适配器

2.2 硬件连接

- 1.安装 WiFi 外部天线。
- 2.安装 LoRa 外部天线。
- 3.安装 GPS 外部天线，并将天线置于室外。
- 4.插入网线到以太网网口，网线连接可上网的路由器、交换机等网络设备。
- 5.插入 ED-GWL501 的 DC 电源输入口 (+12V DC) ,并给电源适配器供电。

2.3 首次启动

ED-GWL501 没有电源开关，插入电源线，系统将会开始启动。

2.3.1 Raspberry Pi OS (Lite)

如果您使用我们提供的系统镜像，系统启动后会使用用户名 pi 自动登入，默认密码为 raspberry。

```
[ OK ] Started User Login Management.
[ OK ] Finished Permit User Sessions.
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started OpenSSH Secure Shell server.
[ OK ] Started Modem Manager.
[ OK ] Started Hostname Service.
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Match.
Starting Load/Save RF Kill Switch Status...
[ OK ] Started LSB: Switch to on (unless shift key is pressed).
[ OK ] Started Load/Save RF Kill Switch Status.
Starting Save/Restore Sound Card State...
[ OK ] Finished Save/Restore Sound Card State.
[ OK ] Reached target Sound Card.

Debian GNU/Linux 11 raspberrypi tty1
raspberrypi login: pi (automatic login)

Linux raspberrypi 5.15.32-08+ #1538 SMP PREEMPT Thu Mar 31 19:40:39 BST 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

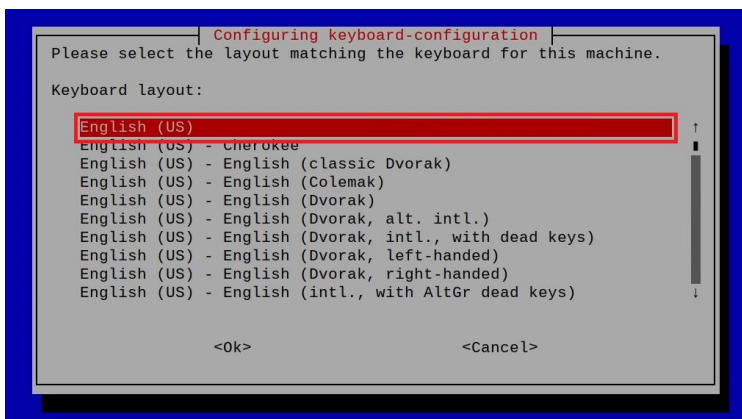
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 31 03:52:21 GMT 2023 from 192.168.168.211 on pts/0

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

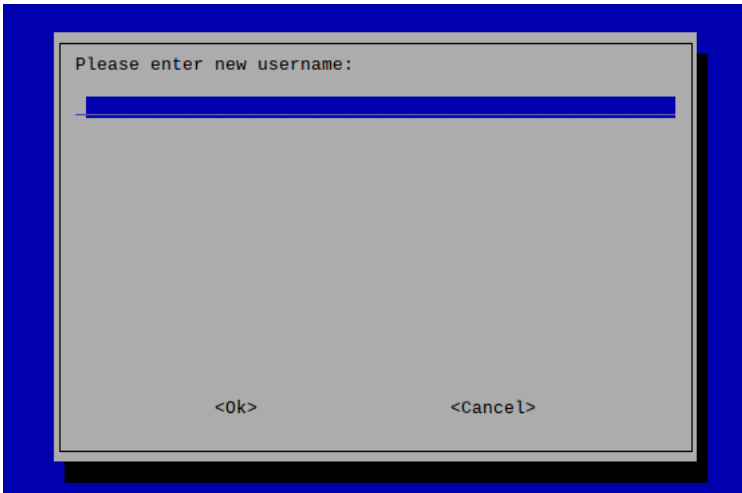
pi@raspberrypi:~$
```

如果您使用官方系统镜像，并且烧录前没有配置镜像，首次启动时，会出现配置窗口，需要依次配置键盘布局，设置用户名及对应密码。

- 设置配置键盘布局



- 创建新用户名



然后按提示设置用户对应的密码，并再次输入密码进行确认。至此您就可以使用刚才设置的用户名及密码进行登入了。

2.3.2 使能 SSH 功能

我们提供的镜像都已经打开了 SSH 功能，如果使用官方镜像则需要使用一下方法打开 SSH 功能。

2.3.2.1 使用 raspi-config 使能 SSH

```
sudo raspi-config
```

1. 选择 3 Interface Options
2. 选择 I2 SSH
3. Would you like the SSH server to be enabled? 选 Yes
4. 选择右下角 Finish

2.3.2.2 添加空文件使能 SSH

在 boot 分区中放入一个名为 ssh 的空文件，设备上电后将会自动使能 SSH 功能。

2.3.3 查找设备 IP

- 设备开启如果接有显示屏可以使用 ifconfig 命令查看当前设备 IP
 - 如果没有显示屏，则可以通过路由器查看分配的 IP
 - 如果没有显示屏，则可以下载 nmap 工具扫描当前网络下的 IP
- nmap 支持 Linux、macOS、Windows 等多个平台。如果希望使用 nmap 扫描 192.168.3.0~255 的网段，则可以使用以下命令：

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

等待一段时间后即会输出结果，类似与下方输出：

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-30 21:19 中国标准时间
Nmap scan report for 192.168.3.1 (192.168.3.1)
Host is up (0.0010s latency).
```

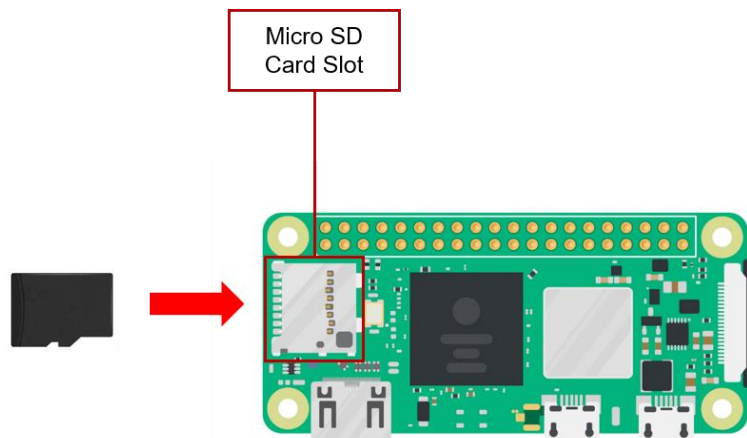
```
MAC Address: XX:XX:XX:XX:XX:XX (Phicomm (Shanghai))
Nmap scan report for DESKTOP-FGEOUUK.lan (192.168.3.33)
Host is up (0.0029s latency).
MAC Address: XX:XX:XX:XX:XX:XX (Dell)
Nmap scan report for 192.168.3.66 (192.168.3.66)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 11.36 seconds
```

3 接线指南

3.1 Internal I/O

3.1.1 micro-SD Card

ED-GWL501 内部集成有 Raspberry Pi Zero 2 W，它具有一个 micro SD 卡插槽，请将烧录好系统的 micro SD 卡正面朝上插入 micro SD 卡插槽。



4 软件操作指引

4.1 按键

ED-GWL501 具有 1 路用户自定义按键，在外壳上的丝印为 **Setup**，连接到 CPU 的 GPIO23 管脚，默认状态下为高电平，当按键按下时，该管脚为低电平。

我们使用 `raspi-gpio` 命令进行测试，按键未按下时查询 GPIO23 管脚

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

level 为 1 表示 GPIO23 管脚为高电平。

按键按下时，查询 GPIO23 管脚

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

level 为 0 表示 GPIO23 管脚为低电平。

4.2 LED 指示

ED-GWL501 具有一个 RGB 三色 LED 指示灯。与 CPU 的 GPIO 管脚对应如下：

RGB LED 管脚	对应 GPIO
蓝色	GPIO16
绿色	GPIO20
红色	GPIO21

GPIO 输出为低时，对应 LED 有效。

我们使用 `raspi-gpio` 命令进行操作，配置参数为 `op` 表示设置为输出，`dl` 设置管脚为低电平，`dh` 设置管脚为高电平。

LED 显示为蓝色

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

LED 显示为绿色

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

LED 显示为红色

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

LED 显示为黄色

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

4.3 以太网配置

ED-GWL501 具有一路自适应 10/100Mbps 以太网接口，丝印标识: Ethernet，是通过内置 USB 接口扩展

而来。网卡接口地址为/sys/class/net/usb0。

树莓派官方系统默认使用的 dhcpcd 作为网络管理工具。

设置静态 IP 通过修改/etc/dhcpcd.conf 来设置，示例设置 usb0，用户可以根据自己的不同需要设置 wlan0 等网络接口。

```
interface usb0
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
```

4.4 WiFi

ED-GWL501 支持 2.4 GHz IEEE 802.11 b/g/n WiFi。

树莓派官方系统默认使用的 dhcpcd 作为网络管理工具。

```
sudo raspi-config
```

1. 选择 1 System Options。
2. 选择 S1 Wireless LAN。
3. 在 Select the country in which the Pi is to be used 窗口中选择您的国家，然后选择 OK，此提示仅在第一次设置 WIFI 时出现。
4. Please enter SSID，输入 WIFI SSID 名称。
5. Please enter passphrase. Leave it empty if none，输入密码，然后重启设备即可。

4.5 蓝牙

ED-GWL501 支持蓝牙 4.2，同时支持蓝牙低功耗（BLE），蓝牙功能默认是开启的。

可使用 bluetoothctl 扫描，配对，连接蓝牙设备，请参考 [ArchLinux-Wiki-Bluetooth](#) 指引配置和使用蓝牙。

4.5.1 基本用法

扫描：

```
bluetoothctl scan on/off
```

发现：

```
bluetoothctl discoverable on/off
```

信任设备：

```
bluetoothctl trust [MAC]
```

连接设备:

```
bluetoothctl connect [MAC]
```

断开设备:

```
bluetoothctl disconnect [MAC]
```

4.5.2 示例

进入 bluetooth shell

```
sudo bluetoothctl
```

使能 Bluetooth

```
power on
```

扫描设备

```
scan on
```

```
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
```

查找开启的蓝牙设备名称，这里测试开启的蓝牙设备名称是 test

```
devices  
  
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
```

配对 test 设备

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

连接蓝牙

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

4.6 GPS

ED-GWL501 网关集成 L76K GPS 模块，它与 CPU 的 UART0 串口相连。模块通过 NMEA 0183 通用协议输出语句上报 GNSS 信息。

4.6.1 修改 config.txt 使能串口

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

在最后面添加

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

4.6.2 查看 GNSS 信息

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

显示 GPS 数据如下：

```
$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 通用语句说明如下：

- \$BDGSV 可视北斗卫星信息
- \$GNRMC 推荐的 GNSS 数据
- \$GNVTG 相对地面航向和速度信息
- \$GNZDA 时间和日期，UTC 格式
- \$GPTXT 文本传送
- \$GNGGA 多星联合定位数据
- \$GNGLL 地理位置纬度和经度
- \$GNGSA 表示 GNSS 精度因子和有效卫星
- \$GPGSV 可视的 GNSS 卫星

4.6.3 使用 u-center 工具查看定位信息

4.6.3.1 安装串口转网络工具 ser2net

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

启用 ser2net 服务

ser2net 配置文件为/etc/ser2net.yaml，默认已配置/dev/ttyS0，波特率为 9600，无校验，对应的 TCP 端口为 2000。

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

4.6.3.2 检查 ser2net 端口转发服务

使用如下指令查询 ser2net 是否已启动 2000 端口转发

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

如果已启动端口转发，将显示如下信息

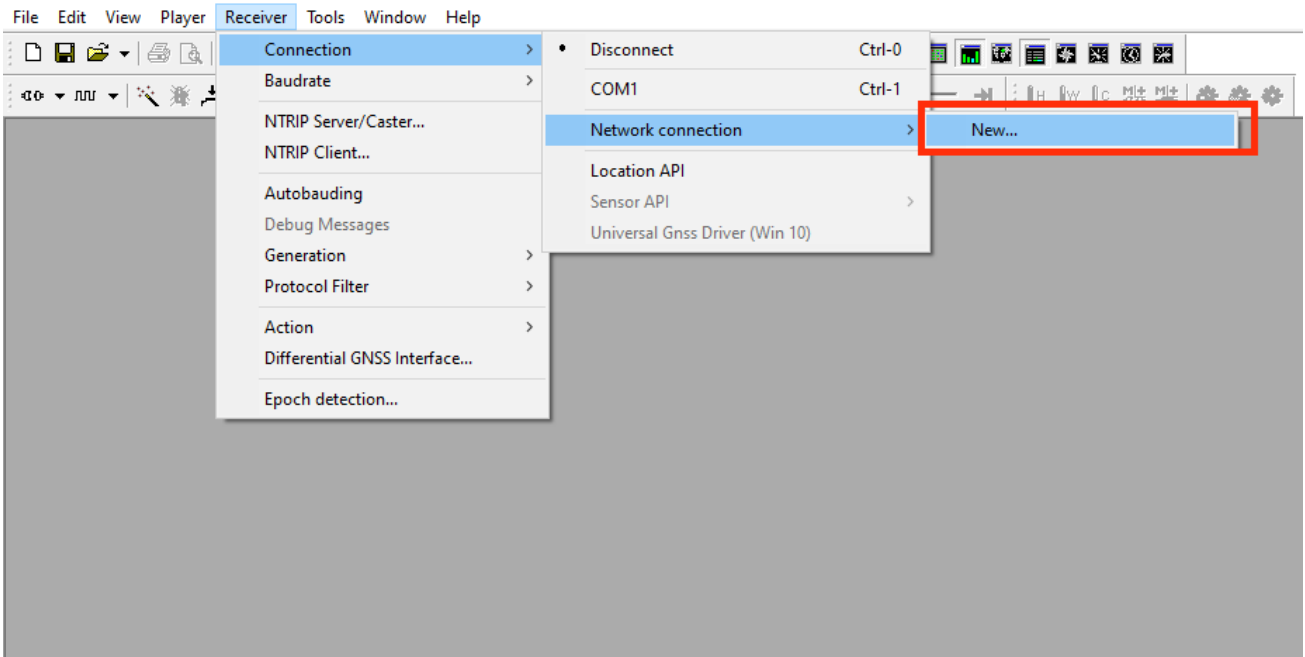
tcp6	0	0 :::2000	:::*	LISTEN	720/ser2net
------	---	-----------	------	--------	-------------

如果无显示，则重启 ser2net 服务

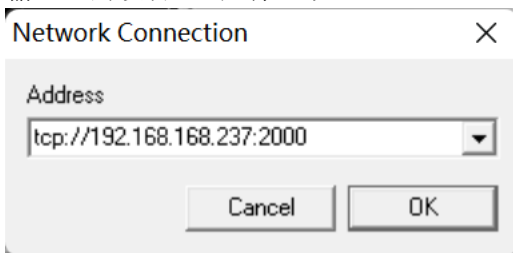
```
sudo systemctl restart ser2net
```

下载 [u-center](#) 工具并安装，如果提示缺少 MSVCR120.dll 文件，请安装 [vcredist_x86.exe](#)。

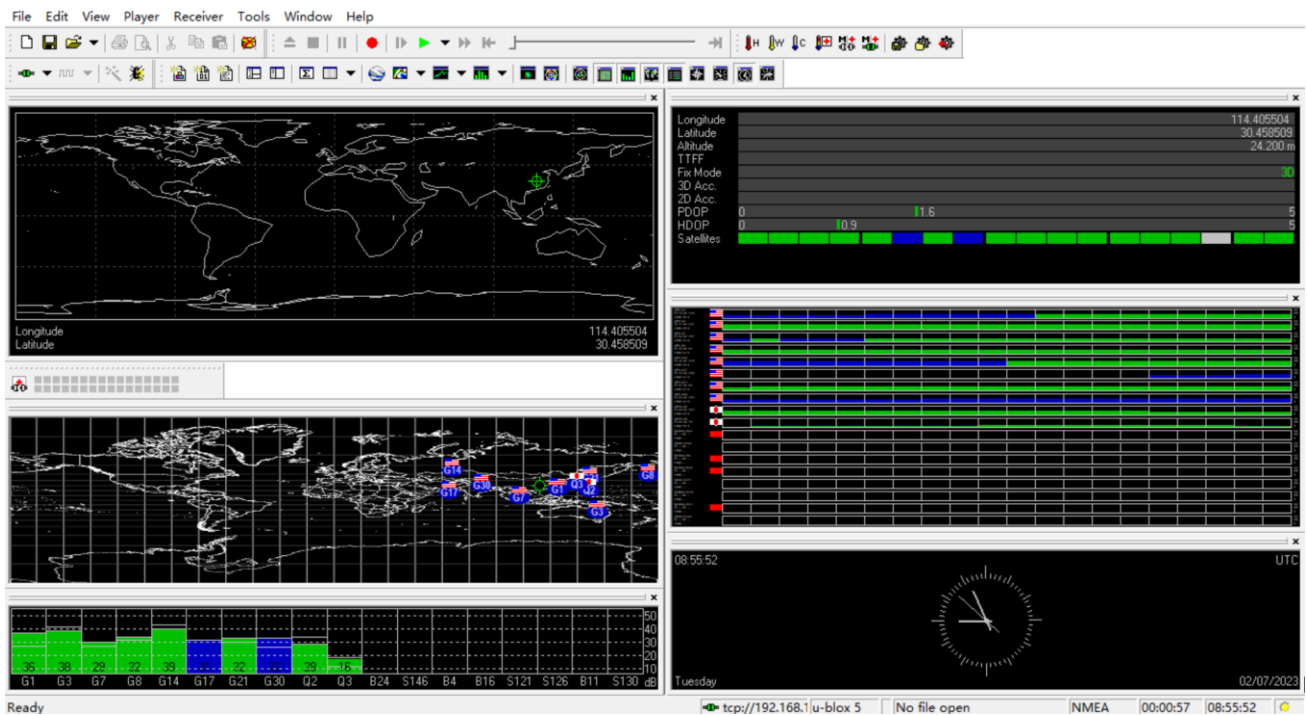
打开 u-center，选择 Receiver->Port->Network connection->New...



输入您的设备 IP 和端口号 2000。



配置完成后会立刻看到 GPS 定位信息



如果 Fix Mode 显示为 No Fix，表示未能成功定位，一般是由于天线在室内的原因造成，请将模块或天线置于户外进行试验。

Longitude	
Latitude	
Altitude	
TTF	
Fix Mode	No Fix
3D Acc.	
2D Acc.	
PDOP	0 4.3 10
HDOP	0 2.0 10
Satellites	

NOTE:模块首次定位，在户外没有大型建筑遮挡的情况下，大概需要 30 秒才能定位成功，如果天气条件恶劣，可能需要更长的定位时间或无法定位。

4.7 LoRaWAN

ED-GWL501 支持 LoRaWAN 开源服务平台 ChipStack，请参考如下步骤进行安装和配置。

4.7.1 安装 LoRa 服务和 ChirpStack 客户端

我们通过 APT 的方式进行安装。

- 添加 edatec APT 仓库

```
$ 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
```

- 安装 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
```

- 修改 config.txt

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

```
dtparam=spi=on
```

```
gpio=16=op,dl
```

```
gpio=20=op,dl
```

```
gpio=21=op,dl
```

修改/etc/modules，在最后面添加 i2c-dev

```
i2c-dev
```

ED-GWL501 使用 i2c-1 和 spidev0.0。

4.7.2 配置 LoRa 服务

4.7.2.1 pktfwd 配置

```
# update region
```

```
$ cat /etc/ed_gwl/region
```

```
EU868 # EU868 / US915
```

pktfwd 使用 1700 作为 UDP 端口

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

4.7.2.2 chirpstack-gateway-bridge 的配置

您可以使用 nano 进行配置文件 chirpstack-gateway-bridge.toml 的编辑

```
$ 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' 需要修改带有网关区域的前缀。

Example:

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

如果您使用 US915 或 CN470 模块请将前缀 eu868 修改为 us915_0 / cn470_10

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

- integration.mqtt 服务器地址需要是您的 chirpstack 服务器

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

修改 chirpstack-gateway-bridge.toml 配置后，重启 chirpstack-gateway-bridge 服务生效。

4.7.2.3 Reboot

```
$ sudo reboot
```

4.7.3 安装 ChirpStack 服务端

配置云端服务器，配置之前首先需要在服务器安装 docker。

安装 docker: <https://docs.docker.com/get-docker/>

安装 docker-compose

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

4.7.3.1 配置 chirpstack-docker

我们采用 docker 容器的方式部署 ChirpStack 服务端。

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

需要对 chirpstack-docker 的 docker-compose.yml 进行配置。

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

删除红色字体部分。

```
$ 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
- POSTGRES_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:
```

启动 chirpstack 服务端

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

4.7.3.2 登录 chirpstack 服务管理界面

在 PC 端浏览器输入服务器的 IP 地址和端口号 8080，在网络正常的情况下会出现登录界面
默认管理员用户名和密码如下：

```
user: admin
```

```
psw : admin
```

4.7.4 添加 LoRa 网关和终端

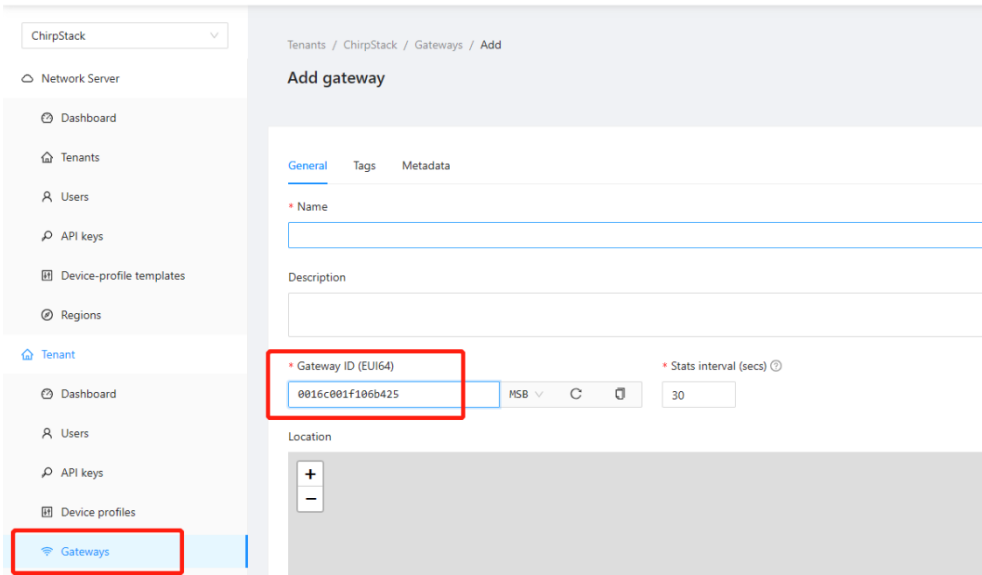
4.7.4.1 获取 LoRa 网关 ID

执行如下指令即可获得 LoRa 网关的 ID，在向 chirpstack 服务端添加 LoRa 网关时，需要添加对应的网关 ID。

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

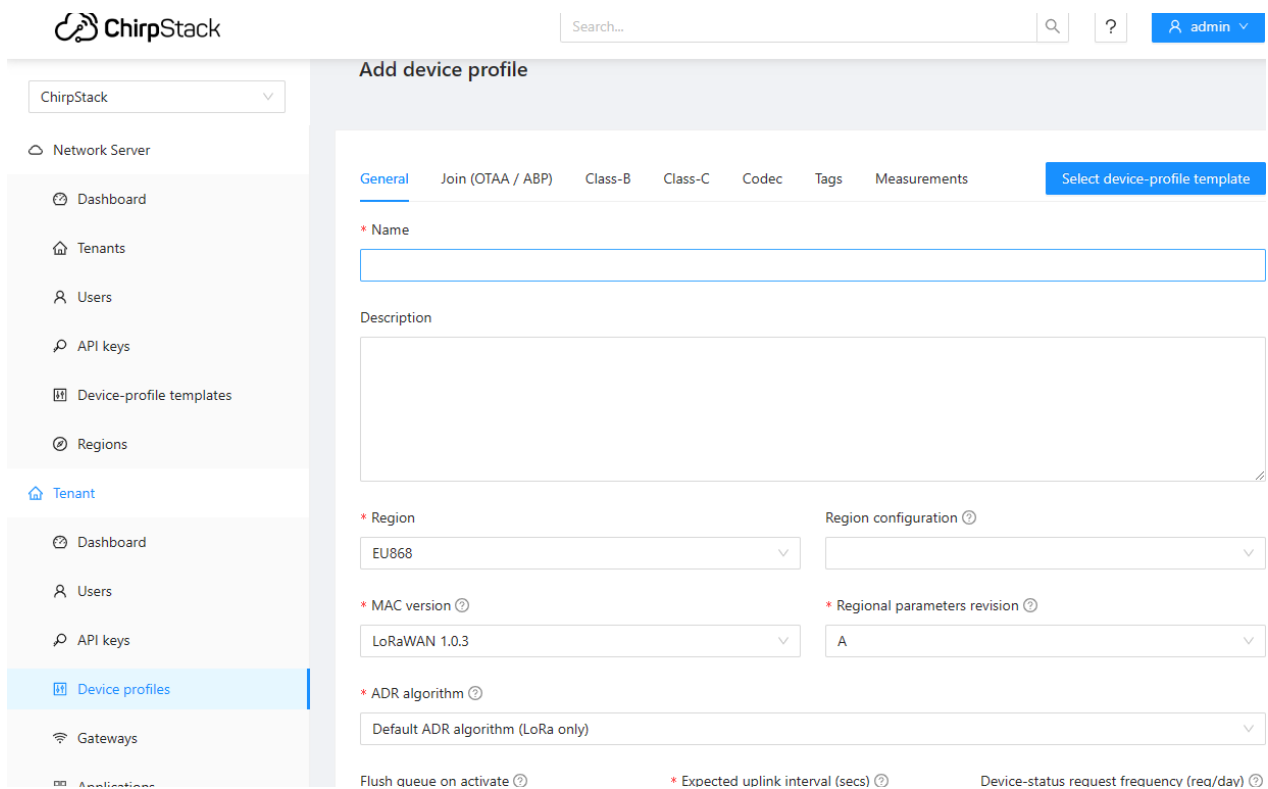
4.7.4.2 添加 LoRa 网关

在 PC 端浏览器打开 chirpstack 管理界面，点击 Gateway -> Add gateway，填入设备对应的 Gateway ID，并设置 Name，然后点击 Submit，如果网络连接正确，稍等片刻即可看到添加的网关变为 Online 状态。



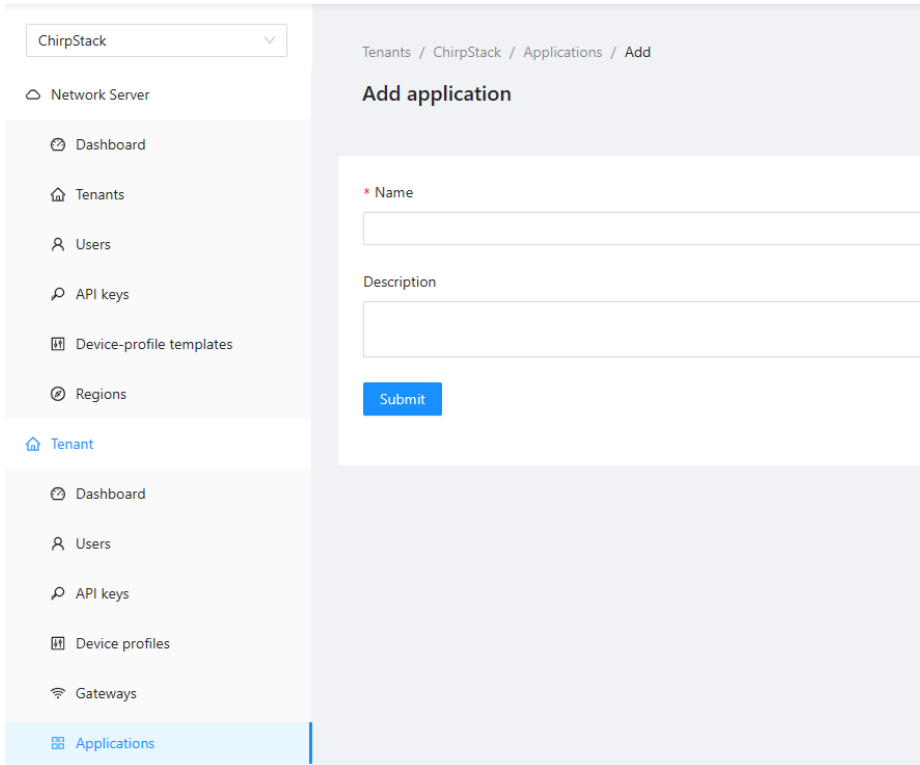
4.7.4.3 Add Device Profile

点击 Device Profile -> Add device profile 可以进一步完善设备信息。



4.7.4.4 Add Application

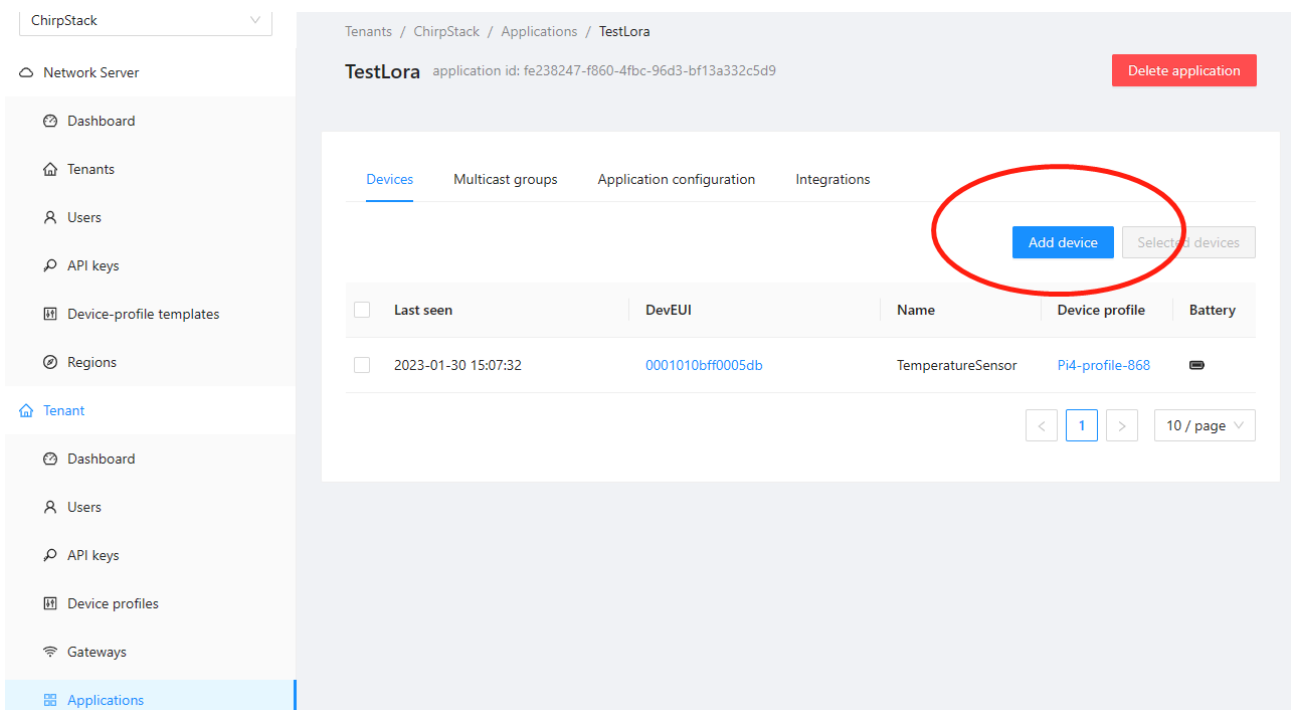
点击 Applications -> Add application



4.7.4.5 Add Device

应该知道 LoRa 终端产品的 DevEUI 和 AppKey，它们由 LoRa 终端设备制造商提供。

点击 Application -> your application -> Add device 进行 LoRa 终端设备的添加。



ChirpStack

Tenants / ChirpStack / Applications / TestLora / Add device

Add device

Device Tags Variables

* Name

Description

* Device EUI (EUI64)

MSB C

* Device profile

Device is disabled ? Disable frame-counter validation ?

Submit

Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

* Application key ?

MSB C

Submit

等待几分钟即可看到设备变成 online 状态。

5 操作系统安装

5.1 镜像下载

我们提供了出厂镜像，如果系统恢复出厂设置，请点击以下链接下载出厂镜像。

Raspberry Pi OS Lite, 32-bit

- Release date: 2023-02-09
- System: 32-bit
- Kernel version: 5.15

- Debian version: 11 (bullseye)

- Downloads: 下载地址 <https://www.123pan.com/s/vjW7Vv-k597A.html> 提取码:DeKv

5.2 系统烧录

ED-GWL501 的系统烧录即对 SD 卡进行烧录。

- 下载安装 [Raspberry Pi Imager](#) 或 [balenaEtcher](#) 镜像烧写工具。
- 将 micro SD 卡插入读卡器，然后将读卡器插入电脑 USB 口。
- 打开镜像烧写工具，选择您要烧录的镜像，路径选择为识别出的大容量存储设备的路径。
- 点击烧录，等待烧录和校验完成，弹出读卡器设备。
- 打开 ED-GWL501 设备上盖，将烧录好镜像的 micro SD 卡插入卡槽。
- 设备重新上电即可。

6 FAQ

6.1.1 默认用户名密码

我们出厂镜像默认用户名是 pi，默认密码是 raspberry。

7 关于我们

7.1 关于 EDATEC

EDATEC 位于上海，是 Raspberry Pi 的全球设计合作伙伴之一。我们的愿景是提供基于 Raspberry Pi 技术平台的物联网、工业控制、自动化、绿色能源和人工智能的硬件解决方案。

我们提供标准的硬件解决方案，定制设计和制造服务，以加快电子产品的开发和上市时间。

7.2 联系方式

邮箱 - sales@edatec.cn / support@edatec.cn

手机 - +86-18621560183

网站 - <https://www.edatec.cn>

地址 - 上海市嘉定区嘉罗公路 1661 号 24 栋 301 室