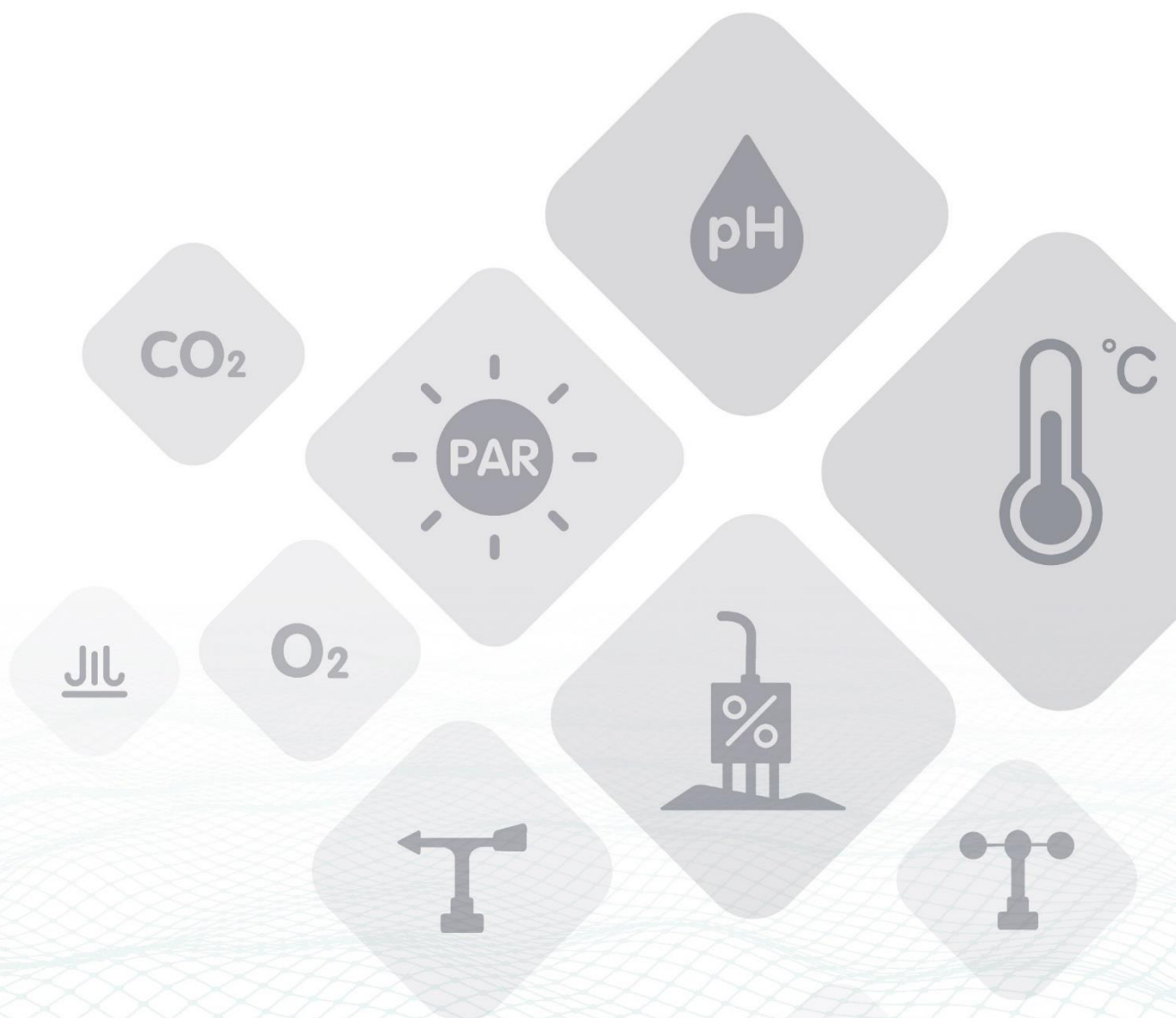


## SenseCAP 传感器终端

## 接入第三方 LoRaWAN 网关和服务商

版本: V1.0



## 目录

1. 概述 .....	4
2. 获取 SenseCAP 终端的参数 .....	5
3. 设备开机 .....	6
4. The Things Network 配置例程 .....	8
4.1 网关联网和在 TTN 中创建网关 .....	9
4.1.1 LPS8 网关联网 .....	9
4.1.2 在 TTN 中创建网关 .....	11
4.2 配置 LPS8 连接到 TTN .....	14
4.2.1 通用设置 .....	14
4.2.2 频率设置 .....	15
4.2.3 通道设置 .....	16
4.3 创建 LoRaWAN 终端设备 .....	18
5. LoRaServer 配置例程 .....	22
5.1 准备工作 .....	23
5.1.1 安装镜像 .....	23
5.1.2 登录系统 .....	23
5.2 EU868 网关配置和设备注册 .....	25
5.2.1 网关配置 .....	25
5.2.2 设备注册 .....	30
5.3 US915 网关配置和设备注册 .....	35
6. 数据包解析 .....	36

代码附录..... 41

深圳矽速科技股份有限公司授权使用

# 1. 概述

SenseCAP 传感器终端最初是基于 The Things Network LoRaWAN 服务器设计的，终端的固件是使用标准的 LoRaWAN 1.0 协议，因此传感器能接入其他第三方 LoRaWAN 网关和服务器。

使用配套的 SenseCAP LoRaWAN 网关无需复杂的配置，开箱即用，同时提供云平台和 API 等服务，可以极大的缩短您项目的开发时间。关于 SenseCAP LoRaWAN Gateway 更多信息，可访问：

<http://solution.seeedstudio.com>

SenseCAP 传感器终端使用 OTAA (Over The Air Activation) 入网方式，因此理论上在使用 OTAA 时涉及三个参数：设备 EUI, App EUI, App Key。通过配置这三个参数，每个 LoRaWAN 网络管理员能将 SenseCAP 终端加入到自己的 LoRaWAN 网络中。

本例程将以 Dragino 的 LPS8 LoRaWAN 网关和基于树莓派的 RHF0M301 LoRaWAN 网关为例，分别将 SenseCAP 终端的数据通过网关上传到 The Things Network 和 LoRaServer 服务器。

网关详情及购买链接：

LPS8 Indoor LoRaWAN Gateway

<https://www.seeedstudio.com/LPS8-Indoor-LoRaWAN-Gateway-Included-SX1308-LoRa-Concentrator-p-4251.html>

LoRa/LoRaWAN Gateway - 915MHz for Raspberry Pi 3

<https://www.seeedstudio.com/LoRa-LoRaWAN-Gateway-915MHz-for-Raspberry-Pi-3-p-2821.html>

LoRa LoRaWAN Gateway - 868MHz Kit with Raspberry Pi 3

<https://www.seeedstudio.com/LoRa-LoRaWAN-Gateway-868MHz-Kit-with-Raspberry-Pi-3-p-2823.html>

## 2. 获取 SenseCAP 终端的参数

设备 EUI (Device EUI) 在 SenseCAP 终端外壳的标签上。



记录如图所示的 Device EUI 和 Device Code，然后使用 HTTP API 的方式获取 App EUI 和 App Key。可以使用浏览器或者任意的 HTTP 工具完成 HTTP GET 请求。

```
curl
https://sensecap-makerapi.seeed.cc/v1/security/device/node/acquirePrivateLorawanDeviceinfo?nodeEui=2cf7f12204400000&deviceCode=E012780D304A1118
```

在 API 中需要将 nodeEui 和 deviceCode 替换为我们设备的 device EUI 和 Device Code。我们将得到如以下格式内容的返回值。

```
{
  "code": "0",
  "data": {
    "nodeEui": "2cf7f12204400000",
    "deviceCode": "E012780D304A1118",
    "lorawanInformation": {
      "dev_eui": "2cf7f12204400000",
      "app_eui": "#####",
      "app_key": "#####"
    }
  }
}
```

记录下 App EUI 和 App Key，在后文配置参数时需要使用。

### 3. 设备开机

1) 传感器的电源开关在设备内部，手握住逆时针转动即可很方便地拧开传感器头。



2) 拧开后，拨动电源开关到 'ON' 一侧，右边的 LED 灯会闪烁一下，表示上电成功。等待 5~10s 左右，LED 灯会连续快闪 2s，表示设备入网成功。



#### LED指示灯状态

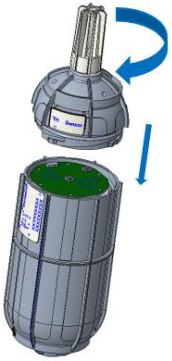
设备上电之后

1. 开机后1s左右会闪烁1次，然后常灭
2. 等待5s左右，会持续快闪2s，表示入网成功
3. 入网成功后，保持常灭以达到省点目的
4. 若开机后15s都没有进入快闪状态，请按复位键重新入网

设备入网成功后 LED 灯会快速闪烁 2 秒，如果超过 15 秒还不进入快闪，建议按设备复位按键进行重新入网。

**注意：**之所以会出现入网失败的情况，原因是单个设备入网完成后，该设备将会在一分钟内持续向云端同步设备信息，这个过程设备会快速发送数据，并占用 LoRa 通信信道，如果此时有多台设备同时开机入网，可能会造成 LoRa 网络拥堵，导致设备入网失败，所以建议每台设备间隔 1 分钟开机入网。

3) 入网成功后，请快速将传感器头插入装回设备主体，顺时针将传感器头拧入，装回传感器头时注意两张标签要对齐，内部的连接器才是可靠连接状态，只有传感器头装回后，设备才能正常上传测量到的环境数据。



- **传感器开机和重启注意**

如果要重启设备，建议使用 RESET 按键复位重启设备，而不是直接快速拨动设备电源。如果要通过电源开关来重启设备，请在关机后等待 3s 后再开机，因为设备在低功耗模式下工作，设备内含极小的电容所带电量都可以让设备持续工作一段时间，从而导致复位失败。

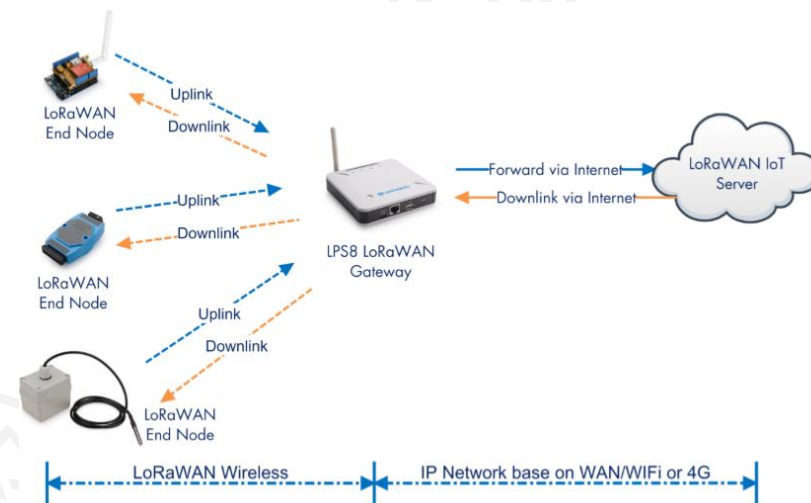


## 4. The Things Network 配置例程

使用不同的 LoRa 网络服务器可能会有不同的配置操作，对于某些网络服务器，网关可以从 Web 服务配置（如使用相应的 LoRa 应用服务），而有的网络服务器，是在网关内部配置（如使用 SSH 命令行）。通常，我们需要配置网关使用的通道（频率），设备注册会因为网络服务器的不同而有所差异，但是都会提供 OTAA 的设备注册接口。

LoRaWAN 在全球不同地区有不同的频率使用规范，SenseCAP 传感器终端遵循 The Things Network 的全球频率计划。为了加快连接过程，SenseCAP 传感器终端减少了 US902-908 和 CN470-510 的通道数量。

本例程使用 LPS8 LoRaWAN 网关，它完全适配 LoRaWAN 协议，基本架构如下图所示。



了解网关更多详情请访问：

<https://www.seeedstudio.com/LPS8-Indoor-LoRaWAN-Gateway-Included-SX1308-LoRa-Concentrator-p-4251.html>



## 4.1 网关联网和在 TTN 中创建网关

### 4.1.1 LPS8 网关联网

LPS8 网关支持多种方式入网，为快速连接，这里使用 WAN port 入网。

其他入网方式以及 LPS8 的使用可参考 Documents/User Manual：

<https://www.seeedstudio.com/LPS8-Indoor-LoRaWAN-Gateway-Included-SX1308-LoRa-Concentrator-p-4251.html>

1) 启动 LPS8，并将 LPS8 连接到路由器。

PC 搜索网络，会出现一个开放的 Wi-Fi 网络：**dragino-xxxxxxx**

使用 PC 连接到此 Wi-Fi，PC 会获得一个地址 10.130.1.xxx，LPS8 获得默认地址 **10.130.1.1**



2) 进入 Web 后台

在此 PC 上打开一个浏览器，访问 <http://10.130.1.1/>，将出现登录页面。

User Name: root

Password: dragino

dragino-1d1694

## Authorization Required

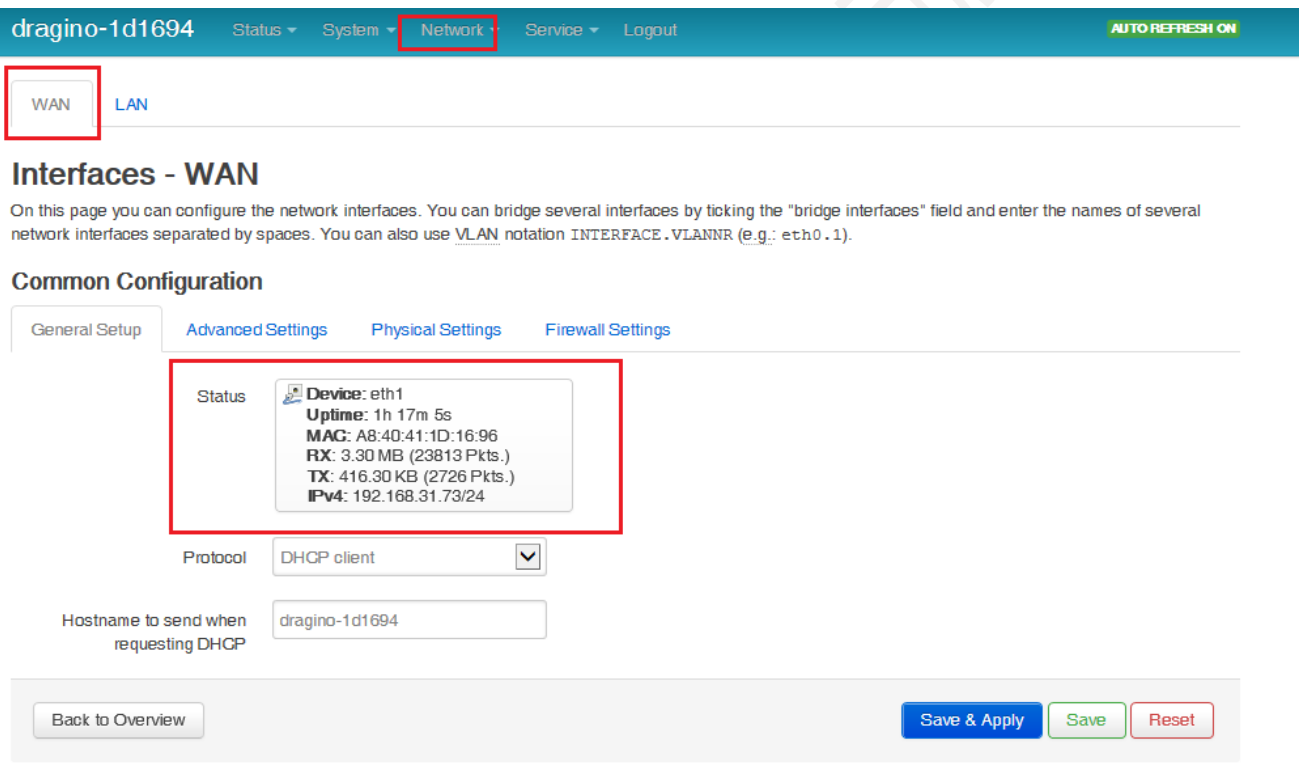
Please enter your username and password.

Username	<input type="text" value="root"/>
Password	<input type="password" value="*****"/>

DRAGINO TECHNOLOGY CO., LIMITED

### 3) 检查网络连接

若网关联网成功，可以进入“Network” - “Interfaces” - “WAN”看到连接状态。



The screenshot shows the web interface for dragino-1d1694. The top navigation bar includes 'dragino-1d1694', 'Status', 'System', 'Network', 'Service', and 'Logout'. The 'Network' menu is highlighted. Below the navigation bar, there are two tabs: 'WAN' (selected) and 'LAN'. The main content area is titled 'Interfaces - WAN'. It contains a description of the page's purpose and a 'Common Configuration' section. This section has four sub-tabs: 'General Setup', 'Advanced Settings', 'Physical Settings', and 'Firewall Settings'. The 'Advanced Settings' tab is active, showing a 'Status' box with the following information: Device: eth1, Uptime: 1h 17m 5s, MAG: A8:40:41:1D:16:96, RX: 3.30 MB (23813 Pkts.), TX: 416.30 KB (2726 Pkts.), and IPv4: 192.168.31.73/24. Below the status box, there is a 'Protocol' dropdown menu set to 'DHCP client' and a 'Hostname to send when requesting DHCP' field containing 'dragino-1d1694'. At the bottom of the configuration area, there are buttons for 'Back to Overview', 'Save & Apply', 'Save', and 'Reset'.

DRAGINO TECHNOLOGY CO., LIMITED

检查网络是否可用：“Network” - “Diagnostics” - “Ping”。

如下图所示，代表网络可用。

## Diagnostics

### Network Utilities

openwrt.org

openwrt.org

IPv4 **Ping**

Traceroute

openwrt.org

Nslookup

Install iputils-traceroute6 for IPv6 traceroute

```
PING openwrt.org (139.59.209.225): 56 data bytes
64 bytes from 139.59.209.225: seq=0 ttl=50 time=205.846 ms
64 bytes from 139.59.209.225: seq=1 ttl=50 time=205.560 ms
64 bytes from 139.59.209.225: seq=2 ttl=50 time=205.589 ms
64 bytes from 139.59.209.225: seq=3 ttl=50 time=205.218 ms
64 bytes from 139.59.209.225: seq=4 ttl=50 time=205.655 ms

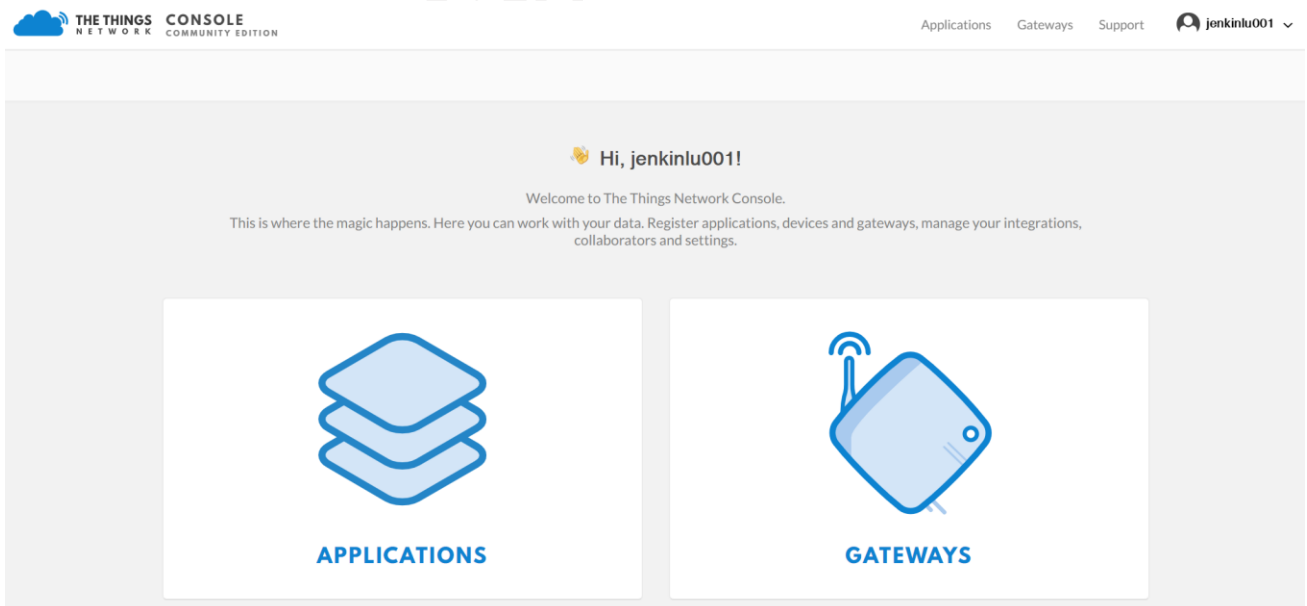
--- openwrt.org ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 205.218/205.573/205.846 ms
```

DRAGINO TECHNOLOGY CO., LIMITED

## 4.1.2 在 TTN 中创建网关

TTN 官网: <https://www.thethingsnetwork.org>

1) 根据提示创建账号, 进入 “Console”。



THE THINGS NETWORK CONSULE COMMUNITY EDITION

Applications Gateways Support jenkins001

Hi, jenkins001!

Welcome to The Things Network Console.

This is where the magic happens. Here you can work with your data, Register applications, devices and gateways, manage your integrations, collaborators and settings.

APPLICATIONS

GATEWAYS

2) 获取 gateway id

回到网关 Web 后台, “Service” - “LoRaWan Gateway” - “Gateway ID”

## LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

General Settings Radio Settings Channels Settings

IoT Service	LoRaWan/RAW forwarder
Debug Level	Little message output
Service Provider	The Things Network
Server Address	ttn-router-us-west
Server port for upstream	1700
Server port for downstream	1700
Gateway ID	a840411d16944150
Status keepalive in seconds	30
Frequency Plan	Customized Bands

[See logread --> FreqINFO for detail](#)

Save & Apply

Save

Reset

DRAGINO TECHNOLOGY CO., LIMITED

示例网关的 ID 为: **a840411d16944150**

### 3) 创建网关

“GATEWAYS” - “register gateway”。

**Gateway EUI** 填写 Gateway ID: **a840411d16944150**, 勾选 “I’m using the....” 后, 会变为十六进制码。

**Description** 自定义名称。

**Frequency Plan** 根据所使用的网关频段选择, 常用的选项为 “China 470-510MHz”, “Europe 868MHz”,

“United States 915MHz” 等, 这里示例网关为 915MHz, 因此选择 “United States 915MHz”。

**Router** 会自动选择 “ttn-router-us-west”。

## REGISTER GATEWAY

## Gateway EUI

The EUI of the gateway as read from the LoRa module

A8 40 41 1D 16 94 41 50

8 bytes

 I'm using the legacy packet forwarder

Select this if you are using the legacy [Semtech packet forwarder](#).

## Description

A human-readable description of the gateway

LPS8-GW

## Frequency Plan

The [frequency plan](#) this gateway will use

United States 915MHz

## Router

The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.

ttn-router-us-west

## Location

The exact location of you gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.

## 4.2 配置 LPS8 连接到 TTN

### 4.2.1 通用设置

首先确保 LPS8 已经成功连接到 Internet。

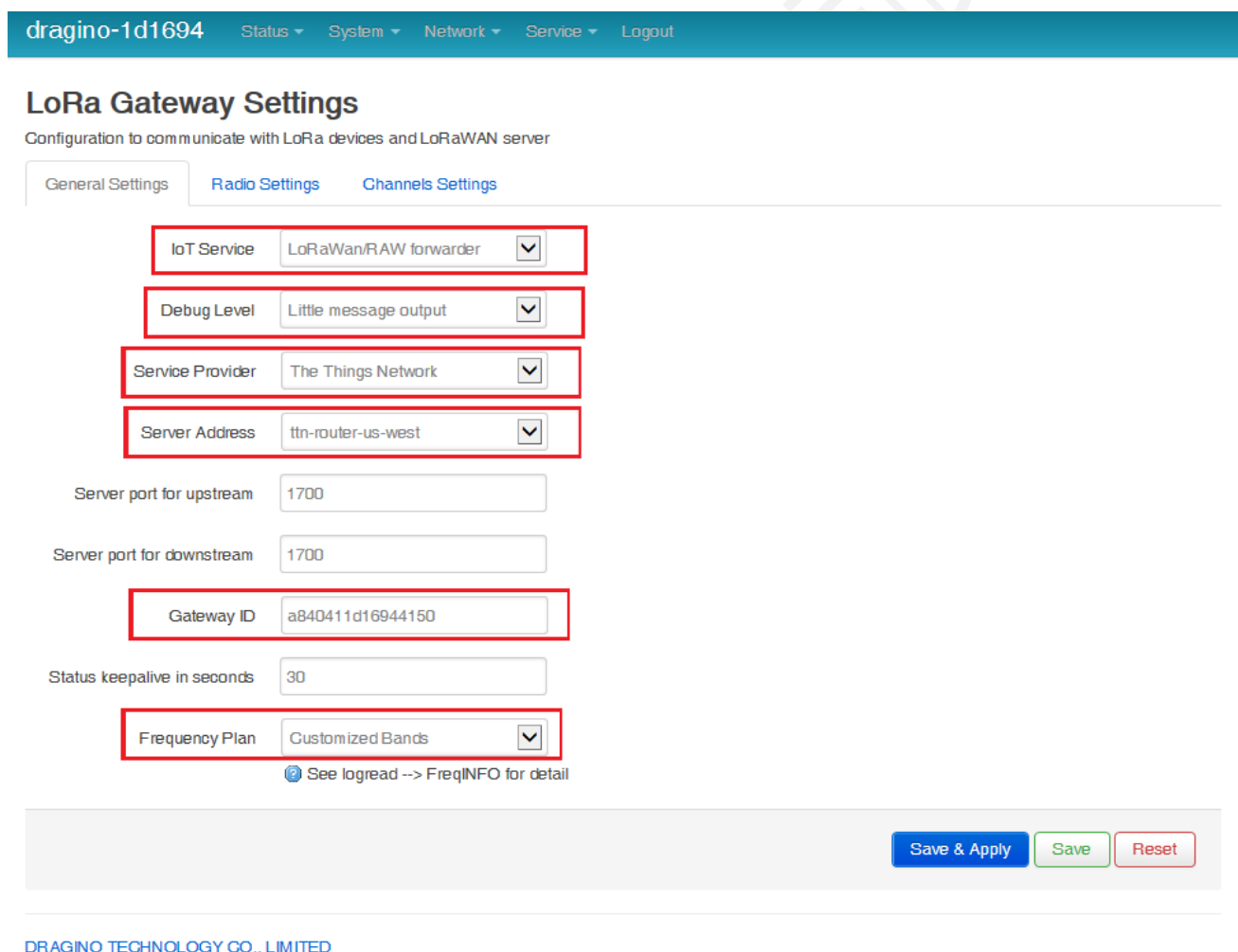
IoT Service: LoRaWan/Raw forwarder

Service Provider: The Things Network

Server Address: ttn-router-us-west (匹配 TTN 上的设置)

Frequency Plan: Customized Bands

其他配置选择默认或根据需要自行配置。



dragino-1d1694 Status System Network Service Logout

### LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

General Settings Radio Settings Channels Settings

IoT Service LoRaWan/Raw forwarder

Debug Level Little message output

Service Provider The Things Network

Server Address ttn-router-us-west

Server port for upstream 1700

Server port for downstream 1700

Gateway ID a840411d16944150

Status keepalive in seconds 30

Frequency Plan Customized Bands

[See logread --> FreqINFO for detail](#)

Save & Apply Save Reset

DRAGINO TECHNOLOGY CO., LIMITED

## 4.2.2 频率设置

具体参数可参考文末[附录](#)。

radio 0 enable√

Radio\_0 frequency: 904300000

Radio\_0 for tx√

Radio\_0 tx min frequency: 923000000

Radio\_0 tx max frequency: 928000000

radio 1 enable√

Radio\_1 frequency: 905000000

dragino-1d1694 Status System Network Service Logout

### LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

General Settings Radio Settings Channels Settings

radio 0 enable

Radio\_0 frequency

Radio\_0 for tx

Radio\_0 tx min frequency

Radio\_0 tx max frequency

radio 1 enable

Radio\_1 frequency

Radio\_1 for tx

Save & Apply

Save

Reset



### 4.2.3 通道设置

## LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

General Settings

Radio Settings

Channels Settings

multiSF channel 0 enable <input checked="" type="checkbox"/>	multiSF channel 4 enable <input checked="" type="checkbox"/>
multiSF channel 0 radio <input type="text" value="radio0"/>	multiSF channel 4 radio <input type="text" value="radio1"/>
multiSF channel 0 IF <input type="text" value="-400000"/>	multiSF channel 4 IF <input type="text" value="-300000"/>
multiSF channel 1 enable <input checked="" type="checkbox"/>	multiSF channel 5 enable <input checked="" type="checkbox"/>
multiSF channel 1 radio <input type="text" value="radio0"/>	multiSF channel 5 radio <input type="text" value="radio1"/>
multiSF channel 1 IF <input type="text" value="-200000"/>	multiSF channel 5 IF <input type="text" value="-100000"/>
multiSF channel 2 enable <input checked="" type="checkbox"/>	multiSF channel 6 enable <input checked="" type="checkbox"/>
multiSF channel 2 radio <input type="text" value="radio0"/>	multiSF channel 6 radio <input type="text" value="radio1"/>
multiSF channel 2 IF <input type="text" value="0"/>	multiSF channel 6 IF <input type="text" value="100000"/>
multiSF channel 3 enable <input checked="" type="checkbox"/>	multiSF channel 7 enable <input checked="" type="checkbox"/>
multiSF channel 3 radio <input type="text" value="radio0"/>	multiSF channel 7 radio <input type="text" value="radio1"/>
multiSF channel 3 IF <input type="text" value="200000"/>	multiSF channel 7 IF <input type="text" value="300000"/>
lorastd channel enable <input checked="" type="checkbox"/>	
LoRa channel radio <input type="text" value="radio0"/>	
LoRa channel IF <input type="text" value="300000"/>	
LoRa channel SF <input type="text" value="8"/>	
LoRa channel BW <input type="text" value="500k"/>	

Save & Apply

Save

Reset


最后选择 “Save & Apply” 。

回到 TTN，可见 Gateway 的状态为 “connected” 。

### GATEWAY OVERVIEW ⚙ settings

**Gateway ID** eui-a840411d16944150


**Description** LPS8-GW

**Owner**  [jenkinlu001](#) [Transfer ownership](#)

**Status** ● connected

**Frequency Plan** United States 915MHz

**Router** ttn-router-us-west

**Gateway Key**  base64 

**Last Seen** 23 seconds ago

**Received Messages** 492

**Transmitted Messages** 41

深圳矽速科技股份有限公司

## 4.3 创建 LoRaWAN 终端设备

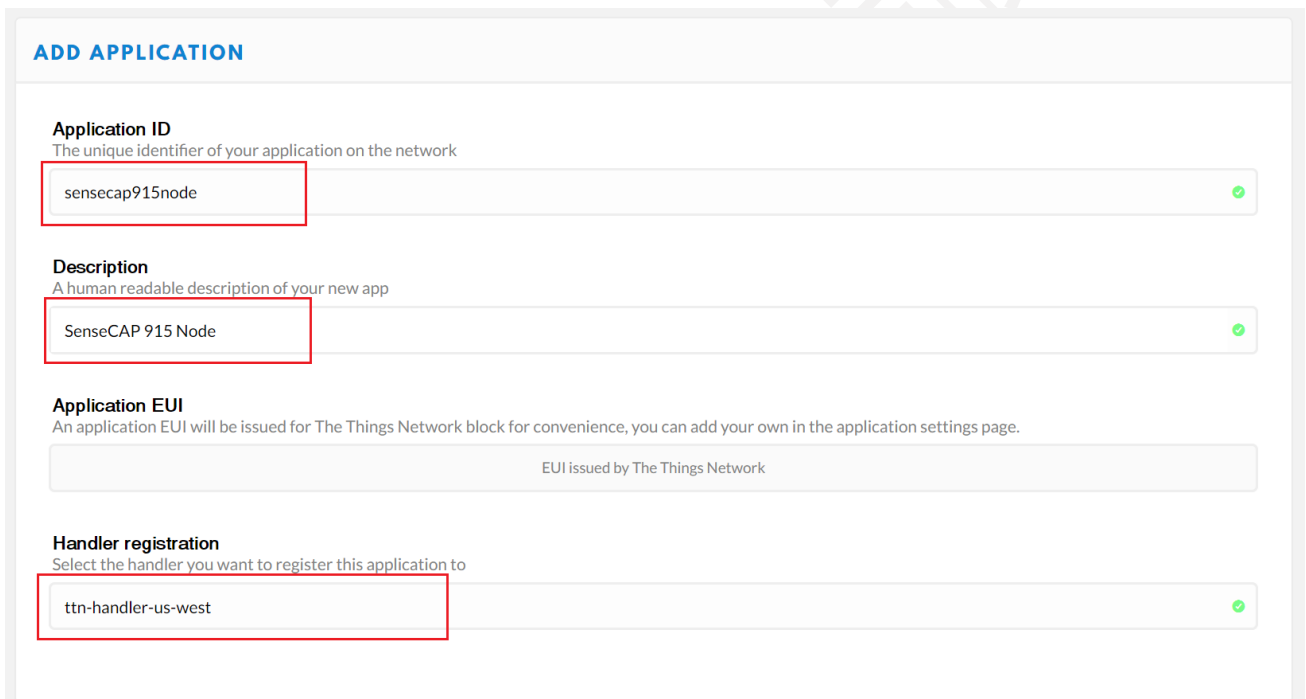
### 1) 增加应用

点击页面上方的“Applications”。



选择“add application”。

Application ID 自定义；Description 自定义；Handler registration 根据频段选择相应的，这里使用 915MHz 的终端节点，因此选择“ttn-handler-us-west”。



**ADD APPLICATION**

**Application ID**  
The unique identifier of your application on the network

sensecap915node

**Description**  
A human readable description of your new app

SenseCAP 915 Node

**Application EUI**  
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.

EUI issued by The Things Network

**Handler registration**  
Select the handler you want to register this application to

ttn-handler-us-west

### 2) 获取 Device EUI、App EUI 和 App Key

参考[第二章](#)得到终端设备的 Device EUI、App EUI 和 App Key。

示例所用终端节点的 Device EUI、App EUI 和 App Key 如下：

```

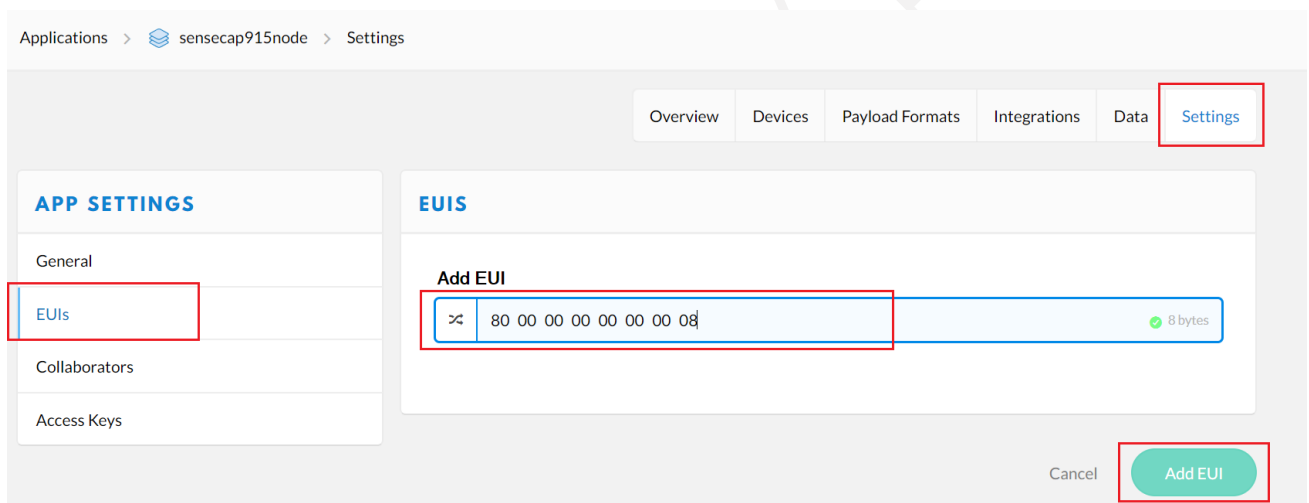
"code": "0",
"data": {
  "nodeEui": "2cf7f1211070004c",
  "deviceCode": "9688E6DD3EF249E2",
  "lorawanInformation": {
    "dev_eui": "2CF7F1211070004C",
    "app_eui": "8000000000000008",
    "app_key": "161583FCE23A44EC8104303B37DD89E3"
  }
},
    
```

### 3) 增加 App EUI

在之前创建的 Application 中进入 “Setting” 。

“Settings” - “EUIs” - “Add EUI”

填入获取到的 “app\_eui” 并保存。



### 4) 增加 Device

选择 “Devices” - “add device” 。

Device ID 自定义； Device EUI 填入获取的 “dev\_eui” ； App Key 填入获取的 “app\_key” ； App EUI 选择获取的 “app\_eui” 。

选择 “Add Device” 。

**REGISTER DEVICE**
bulk import devices

---

**Device ID**  
This is the unique identifier for the device in this app. The device ID will be immutable.

✔

**Device EUI**  
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

✔ 8 bytes

**App Key**  
The App Key will be used to secure the communication between you device and the network.

✔ 16 bytes

**App EUI**

⌵

### 5) 传感器上电开机。

传感器的电源开关在设备内部，手握住逆时针转动即可很方便地拧开传感器头。



拧开后，拨动电源开关到 ‘ON’ 一侧，右边的 LED 灯会闪烁一下，表示上电成功。等待 10s 左右，LED 灯会连续快闪 2s，表示设备入网成功。

进入 “Data” 页面，在页面中会出现入网数据（若未出现数据，可按传感器板面正中处的 RESET 按键，重新入网）。

Applications > sensecap915node > Data

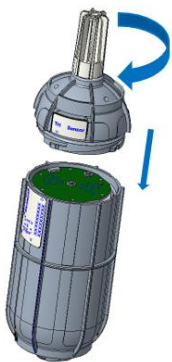
Overview Devices Payload Formats Integrations **Data** Settings

**APPLICATION DATA** || pause 🗑 clear

Filters: uplink downlink activation ack error

time	counter	port		dev id:	
15:25:22	0			th-sensor	
15:25:21	4	2	confirmed	th-sensor	payload: 00 19 00 A7 F0 43 00 00 00 B4 C7
15:25:11	0			th-sensor	
15:25:10	3	2	confirmed	th-sensor	payload: 00 00 00 01 01 00 01 00 07 00 64 00 3C 00 01 20 01 00 00 00 00 28 90
15:24:54	0			th-sensor	
15:24:52	2	2	confirmed	th-sensor	payload: 00 00 00 00 00 00 00 00 00
15:24:41	0			th-sensor	
15:24:39	1	2	confirmed	th-sensor	payload: 00 00 00 00 00 00 00 00 00
15:24:22	0			th-sensor	
15:24:21	0	2	confirmed	th-sensor	payload: 00 00 00 00 00 00 00 00 00

入网成功后，请快速将传感器头插入装回设备主体，顺时针将传感器头拧入，装回传感器头时注意两张标签要对齐，内部的连接器才是可靠连接状态，只有传感器头装回后，设备才能正常上传测量到的环境数据。



#### 6) 添加更多设备

重复 2) ~5) 步骤即可。

## 5. LoRaServer 配置例程

LoRaServer 的配置中会涉及 channel, 在 SenseCAP 终端在不同频段开启的 channel 如下:

- EU863-870: 0,1,2
- US902-908: 8, 9, 10, 11, 12, 13, 14, 15
- CN470-510: 80, 81, 82, 83, 84, 85, 86, 87

本例程将使用基于树莓派 3B 的 LoRaWAN 网关作为示例, 若想要了解更多内容, 请访问:

LoRa/LoRaWAN Gateway - 915MHz for Raspberry Pi 3

<https://www.seeedstudio.com/LoRa-LoRaWAN-Gateway-915MHz-for-Raspberry-Pi-3-p-2821.html>

LoRa LoRaWAN Gateway - 868MHz Kit with Raspberry Pi 3

<https://www.seeedstudio.com/LoRa-LoRaWAN-Gateway-868MHz-Kit-with-Raspberry-Pi-3-p-2823.html>



## 5.1 准备工作

在本例程中，我们使用 LoRaServer 连接 ([www.loraserver.io](http://www.loraserver.io))。



### 5.1.1 安装镜像

设备使用基于树莓派 3B 的 RHF0M301 LoRaWAN 网关，首先下载镜像文件，并烧录到 SD 卡。

镜像下载：<https://artifacts.loraserver.io/downloads/lora-gateway-os/raspberrypi/raspberrypi3/3.0.0test2/>

#### Index of /downloads/lora-gateway-os/raspberrypi/raspberrypi3/3.0.0test2/

../			
licenses/	10-Aug-2019 12:57	-	
<a href="#">lora-gateway-os-base-raspberrypi3-201908100919..&gt;</a>	10-Aug-2019 12:56	59054080	
<a href="#">lora-gateway-os-base-raspberrypi3-201908100919..&gt;</a>	10-Aug-2019 12:56	130464960	
<a href="#">lora-gateway-os-full-raspberrypi3-201908100923..&gt;</a>	10-Aug-2019 12:56	87034368	
<a href="#">lora-gateway-os-full-raspberrypi3-201908100923..&gt;</a>	10-Aug-2019 12:58	184444450	

选择 lora-gateway-os-full 版本。

安装及烧录方法参考：<https://www.loraserver.io/lora-gateway-os/install/raspberrypi/>

### 5.1.2 登录系统

设备上电，开机后输入用户名和密码。（用户名和密码均默认为：admin）

可使用路由器查看 IP 地址，也可以进入系统后，输入命令：ifconfig

```

MiWiFi-R1CM-srv:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr B8:27:EB:91:AD:B8
          inet addr:192.168.31.39  Bcast:192.168.31.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe91:adb8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:276016 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12022 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13519096 (12.8 MiB)  TX bytes:2569037 (2.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5281730 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5281730 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1174146767 (1.0 GiB)  TX bytes:1174146767 (1.0 GiB)
    
```

获得 IP 地址后，为方便操作，使用 SSH 方式登录到系统。

```
LoRaServer.io
documentation and copyright information: www.loraserver.io
Commands:
> sudo gateway-config - configure the gateway
> sudo monit status - display service monitor
```

输入：sudo gateway-config

输入密码，进入配置页，选择“2 Setup LoRa concentrator shield”；

```
LoRa Gateway OS
Configuration options:
1 Set admin password
2 Setup LoRa concentrator shield
3 Edit packet-forwarder config
4 Edit LoRa Gateway Bridge config
5 Restart packet-forwarder
6 Restart LoRa Gateway Bridge
7 Configure WIFI
< OK > < Quit >
```

选择“6 RisingHF -RHFOM301”；

```
Setup LoRa concentrator shield
Select shield:
1 IMST - iC880A
2 IMST - iC980A
3 Pi Supply - LoRa Gateway HAT
4 RAK - RAK2245
5 RAK - RAK831
6 RisingHF - RHFOM301
7 Sandbox - LoRaGo PORT
< OK > < Cancel >
```

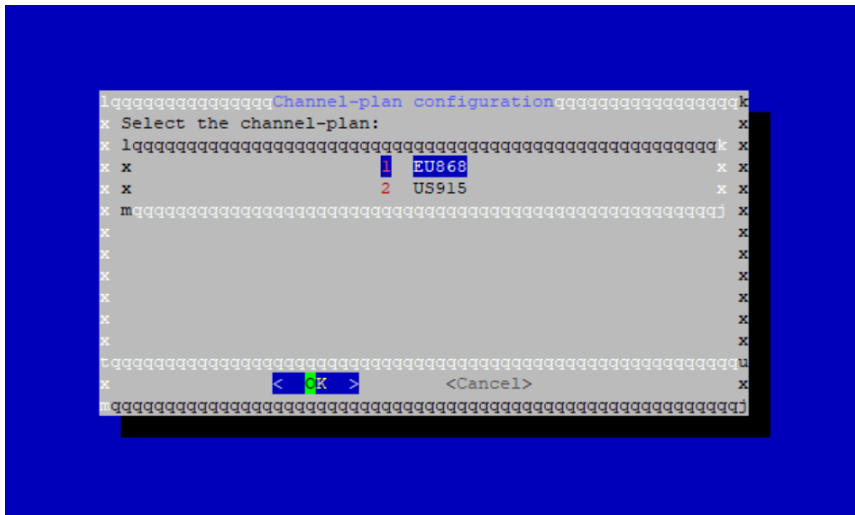
然后可以根据使用频段选择“EU868”和“US915”。

## 5.2 EU868 网关配置和设备注册

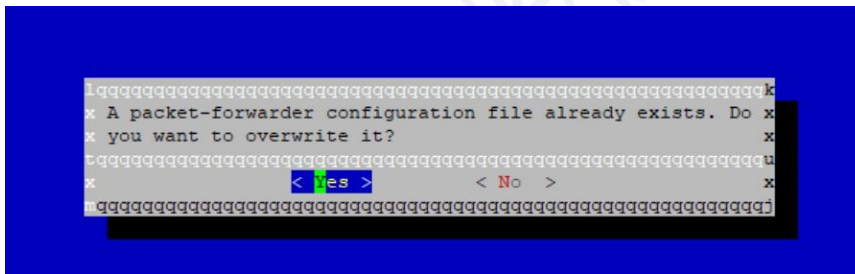
### 5.2.1 网关配置

#### 1) 网关频段选择

根据上述步骤，选择“1 EU868”。



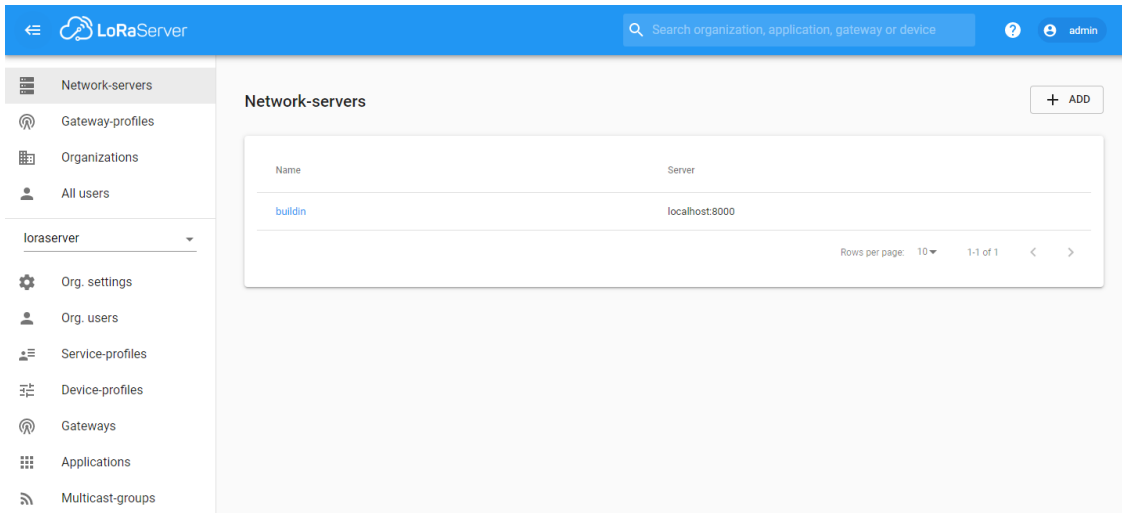
连续选择“Yes”或“OK”，直到返回“LoRa Gateway OS”界面。



#### 2) 网页中的网关参数配置

在浏览器中输入 <http://192.168.31.39:8080> (192.168.31.39 替换为网关正在使用的 IP)

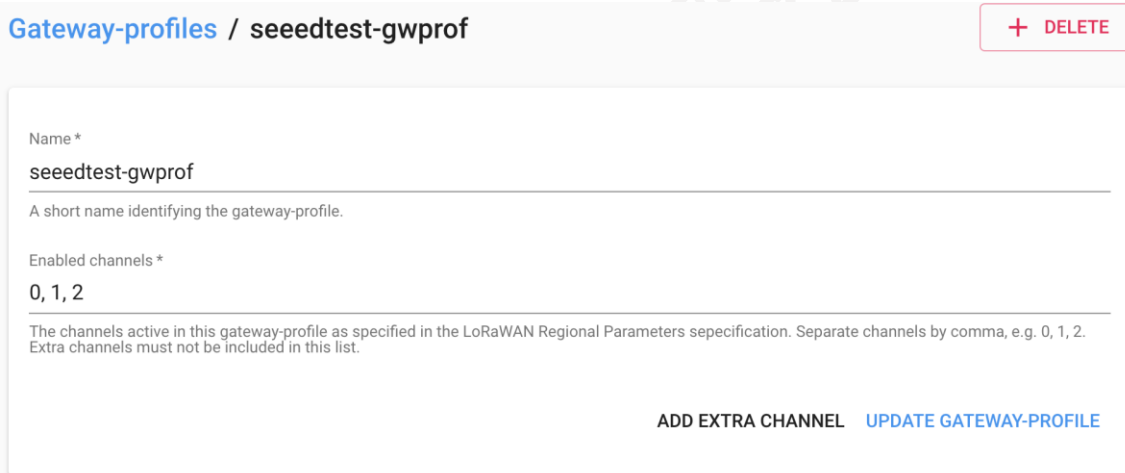
注意：请将电脑和网关的网络配置在同一路由器下，若仍然打不开网页，尝试更换浏览器。



首先创建网关配置文件，选择“Gateway-profiles”，点击“Create”。

不同频段使用的通道不同，EU863-870 使用: 0, 1, 2 。

“Name” 自定义即可，“Enabled channels” 输入: 1, 2, 3



Gateway-profiles / seedtest-gwprof + DELETE

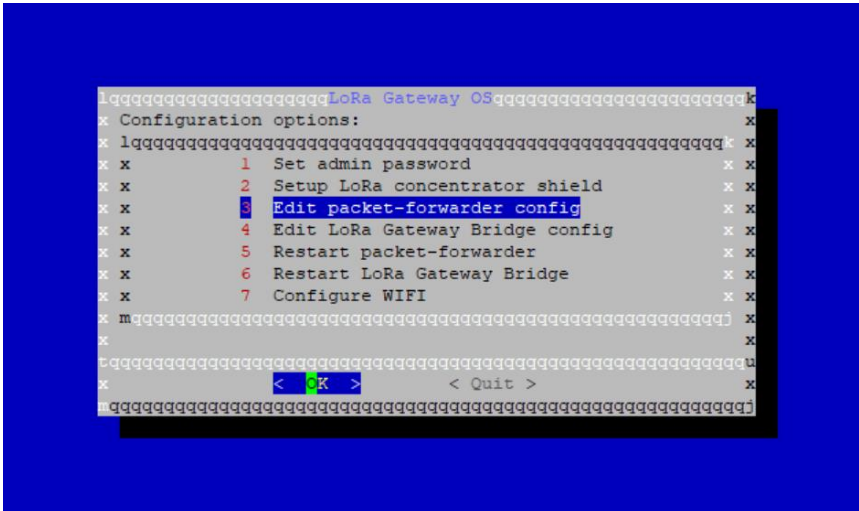
Name \*  
seedtest-gwprof  
A short name identifying the gateway-profile.

Enabled channels \*  
0, 1, 2  
The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters sepecification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

[ADD EXTRA CHANNEL](#) [UPDATE GATEWAY-PROFILE](#)

### 3) 网关本地文件的参数配置

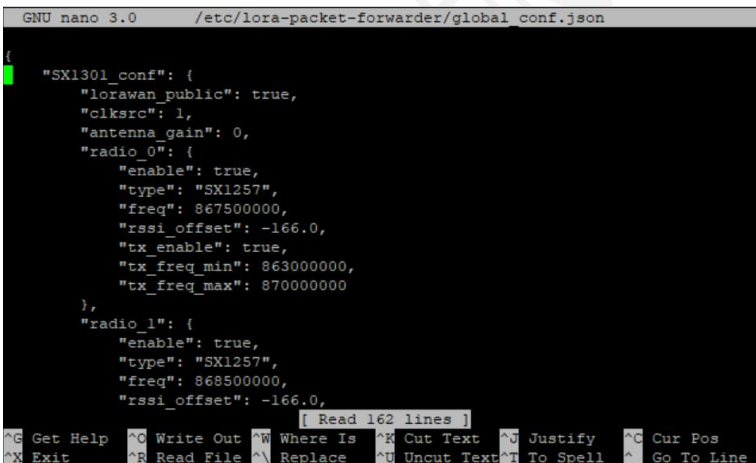
回到 SSH 界面，选择“Edit packet-forwarder config”。



可见文件中主要包含 “SX1301\_conf” 和 “gateway\_conf” 两部分。“SX1301\_conf” 是对频点等参数的配置，“gateway\_conf” 是对网关的基本配置。在这里需要修改 “SX1301\_conf”，先删除原本的配置，再将文档末尾的[附录 EU868](#) 粘贴到原位置，即使用附录中 EU868 的 “SX1301\_conf” 替换系统中原有的 “SX1301\_conf”。

快捷键：“Ctrl + K” 删除整行；“Ctrl + O” 保存；“Ctrl + X” 退出编辑

替换前



替换后

```

GNU nano 3.0 /etc/lorawan-packet-forwarder/global_conf.json Modified
{
  "SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1, /* radio_1 provides clock to concentrator */
    "lbt_cfg": {
      "enable": false,
      "rssi_target": 160, /* rssi in dBm = -lbt_rssi_target/2 */
      "nb_channel": 1,
      "start_freq": 869525000,
      "scan_time_us": 5000,
      "tx_delay_1ch_us": 4000000,
      "tx_delay_2ch_us": 4000000
    },
    "antenna_gain": 0, /* antenna gain, in dBi */
    "radio_0": {
      "enable": true,
      "type": "SX1257",
      "freq": 867500000,
      "rssi_offset": -166.0,
    },
  },
  "gateway_conf": {
    "gateway_ID": "b827ebFFFE91adb8",
    "server_address": "localhost",
    "serv_port_up": 1700,
    "serv_port_down": 1700,
    "keepalive_interval": 10,
    "stat_interval": 30,
    "push_timeout_ms": 100,
    "forward_crc_valid": true,
  },
  "tx_lut_15": {
    /* TX gain table, index 15 */
    "pa_gain": 3,
    "mix_gain": 14,
    "rf_power": 27,
    "dig_gain": 0
  }
},
}

```

注意：最后和 “gateway\_conf” 部分的分隔需要 “},”

记录下 “gateway\_ID” ，后续配置将使用。

```

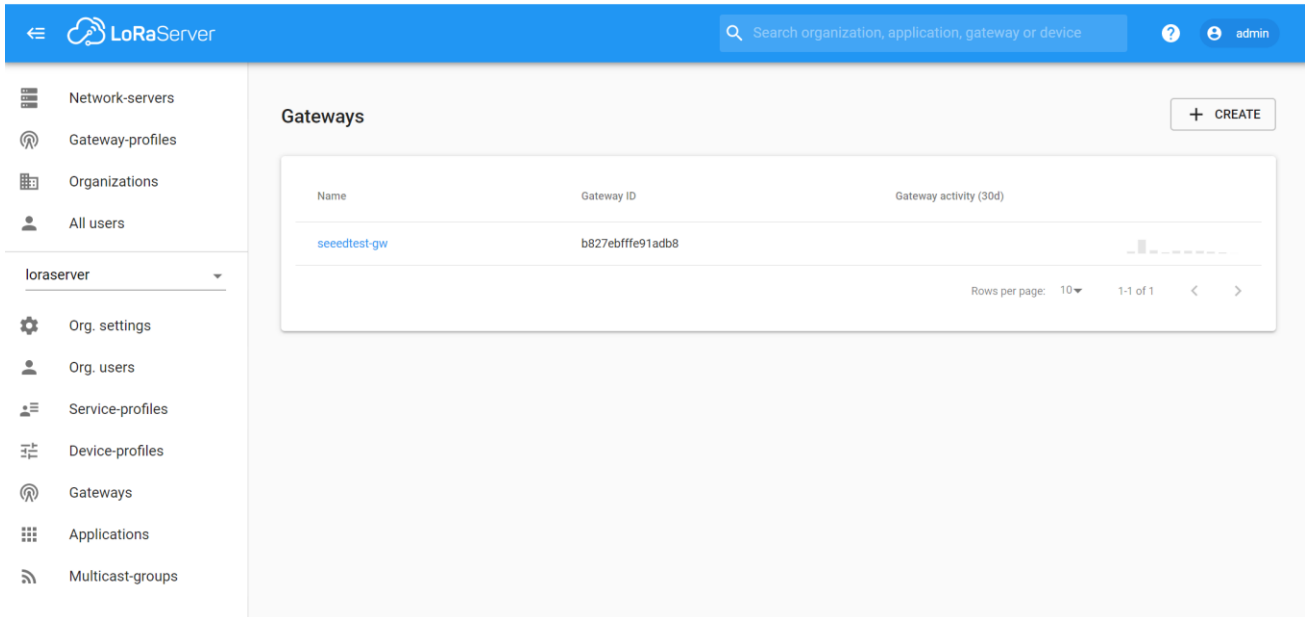
GNU nano 3.0 /etc/lorawan-packet-forwarder/global_conf.json Modified
},
"tx_lut_15": {
  /* TX gain table, index 15 */
  "pa_gain": 3,
  "mix_gain": 14,
  "rf_power": 27,
  "dig_gain": 0
}
},
"gateway_conf": {
  "gateway_ID": "b827ebFFFE91adb8",
  "server_address": "localhost",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
  "keepalive_interval": 10,
  "stat_interval": 30,
  "push_timeout_ms": 100,
  "forward_crc_valid": true,
}
},
}

```

回到 LoRa Gateway OS 界面，选择 “Restart packet-forwarder” ，更新配置。

#### 4) 创建网关

再次回到浏览器，选择 “Gateways” ，创建网关。

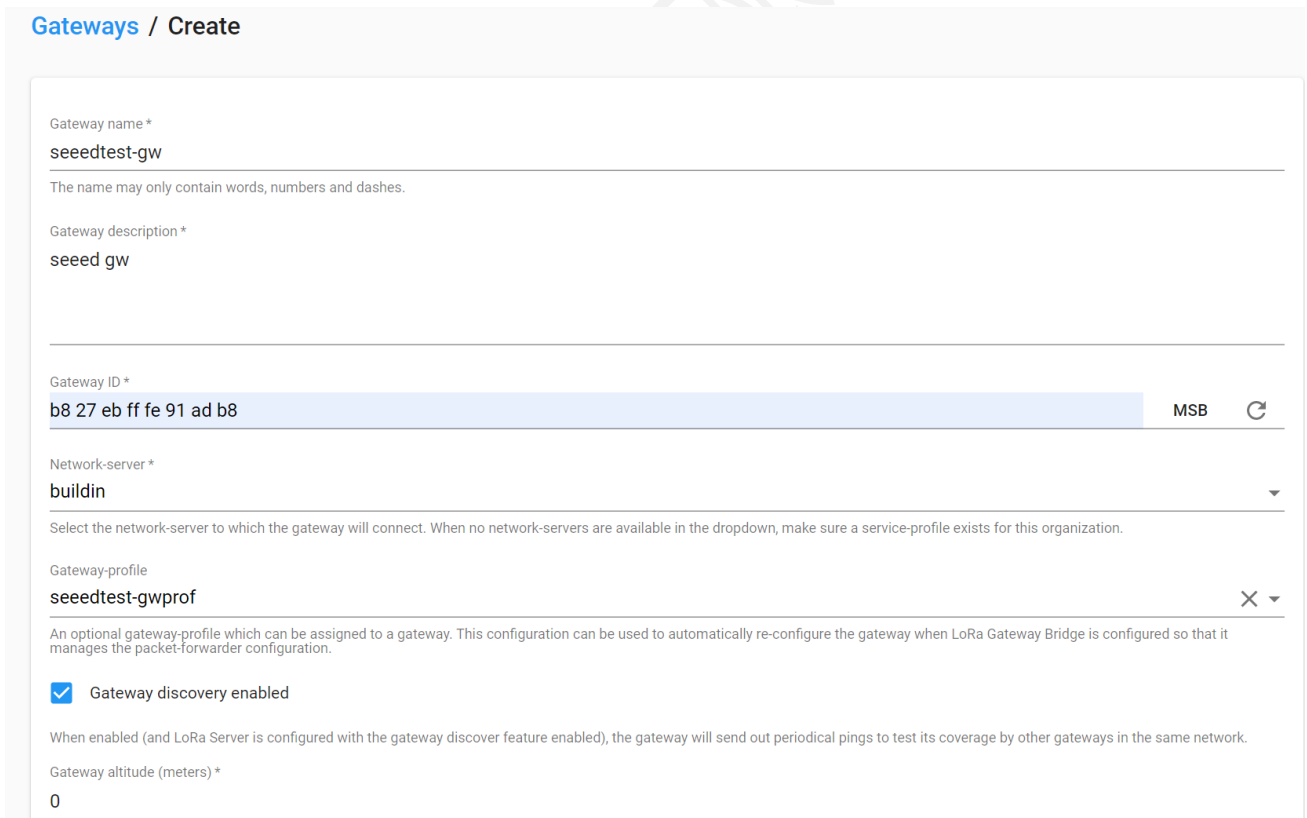


The screenshot shows the LoRaServer interface with a sidebar on the left containing navigation options like 'Network-servers', 'Gateway-profiles', 'Organizations', and 'All users'. The main area is titled 'Gateways' and features a '+ CREATE' button. A table displays gateway information:

Name	Gateway ID	Gateway activity (30d)
seedtest-gw	b827ebffe91adb8	

At the bottom of the table, there are controls for 'Rows per page: 10' and '1-1 of 1'.

自定义网关名字, 写关于该网关的描述, “Gateway ID”为上述步骤“gateway\_conf”中“gateway\_ID”, “Gateway-profile”选择之前配置的文件, 勾选“Gateway discovery enabled”, 其他选择默认, 完成创建。



The screenshot shows the 'Gateways / Create' form with the following fields and values:

- Gateway name \***: seedtest-gw
- Gateway description \***: seed gw
- Gateway ID \***: b8 27 eb ff fe 91 ad b8 (with MSB and refresh icons)
- Network-server \***: buildin
- Gateway-profile**: seedtest-gwprof
- Gateway discovery enabled**
- Gateway altitude (meters) \***: 0

Help text for 'Gateway discovery enabled': "When enabled (and LoRa Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network."



## 5.2.2 设备注册

1) 服务参数配置。自定义名字，选择“buildin”，勾选“Add gateway meta-data”，其他选择默认。

Service-profiles / Create

Service-profile name \*  
seedtest-serviceprof  
A name to identify the service-profile.

Network-server \*  
buildin  
The network-server on which this service-profile will be provisioned. After creating the service-profile, this value can't be changed.

Add gateway meta-data  
GW metadata (RSSI, SNR, GW geoloc., etc.) are added to the packet sent to the application-server.

Enable network geolocation  
When enabled, the network-server will try to resolve the location of the devices under this service-profile. Please note that you need to have gateways supporting the fine-timestamp feature and that the network-server needs to be configured in order to provide geolocation support.

Device-status request frequency  
0  
Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

Minimum allowed data-rate \*  
0  
Minimum allowed data rate. Used for ADR.

Maximum allowed data-rate \*  
0  
Maximum allowed data rate. Used for ADR.

2) 设备参数配置。

在“GENERAL”页面，自定义名字，“LoRaWAN MAC version”选择“1.0.2”版本，“LoRaWAN Regional Parameters revision”选择 B，其他选择默认。

Device-profiles / seedtest-deviceprof
DELETE

GENERAL
JOIN (OTAA / ABP)
CLASS-B
CLASS-C
CODEC

Device-profile name \*

**seedtest-deviceprof**

A name to identify the device-profile.

---

LoRaWAN MAC version \*

**1.0.2**

The LoRaWAN MAC version supported by the device.

---

LoRaWAN Regional Parameters revision \*

**B**

Revision of the Regional Parameters specification supported by the device.

---

Max EIRP \*

**0**

Maximum EIRP supported by the device.

[UPDATE DEVICE-PROFILE](#)

在“JOIN (OTAA/ABP)”页面，勾选“Device supports OTAA”。其他页面不做设置。

Device-profiles / seedtest-deviceprof
DELETE

GENERAL
JOIN (OTAA / ABP)
CLASS-B
CLASS-C
CODEC

Device supports OTAA

[UPDATE DEVICE-PROFILE](#)

3) 增加设备。在主界面列表，选择“Applications”，创建应用。

自定义名字和描述，选择之前已经配置的服务，完成创建。

## Applications / Create

Application name \*  
**seedtest-app**

The name may only contain words, numbers and dashes.

Application description \*  
**seedtest**

Service-profile \*  
**seedtest-serviceprof**

The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.

Payload codec  
**None**

By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you. **Important note:** they payload fields have moved to the device-profile. For backward-compatibility and migration, existing codec settings are still visible. Codec settings on the device-profile have priority over the application codec settings.

[CREATE APPLICATION](#)

点击应用名称，进入详情页，选择创建设备。

自定义名称和设备描述，输入在 [2.获取 SenseCAP 终端的参数](#) 部分得到的设备 EUI，选择配置好的设备参数，完成创建。

## Applications / seedtest-app / Devices / Create

**GENERAL**      VARIABLES      TAGS

Device name \*  
**868-device-2**

The name may only contain words, numbers and dashes.

Device description \*  
**868 test device #2**

Device EUI \*  
**2c f7 f1 22 04 40 00 00**      MSB      ↻

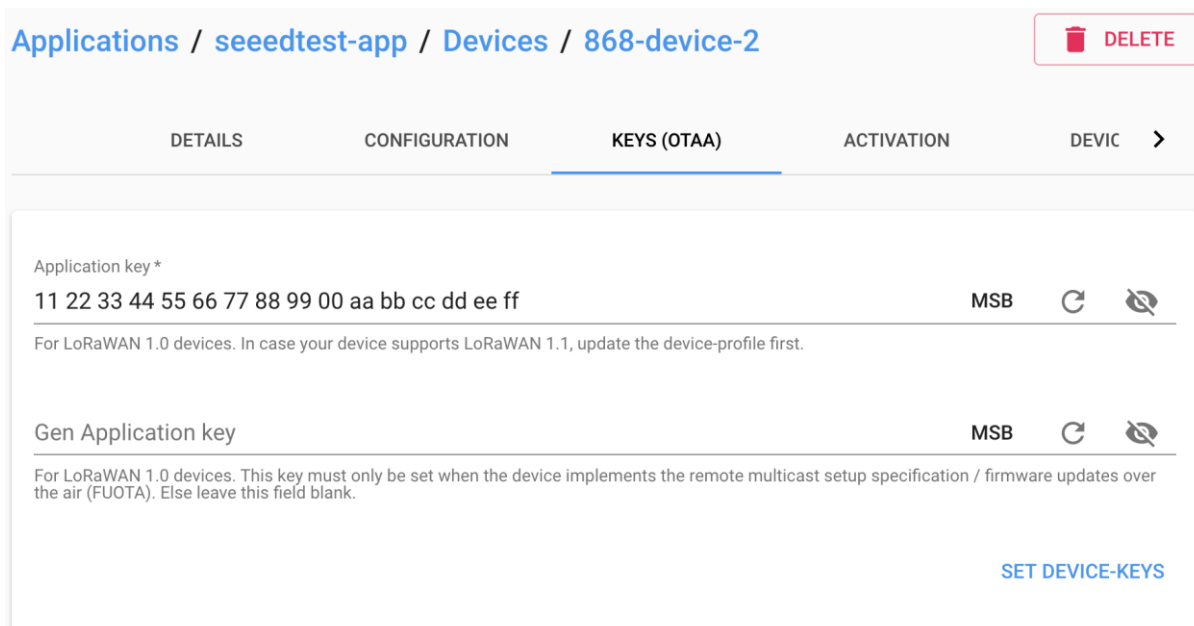
Device-profile \*  
**seedtest-deviceprof**

Disable frame-counter validation

Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

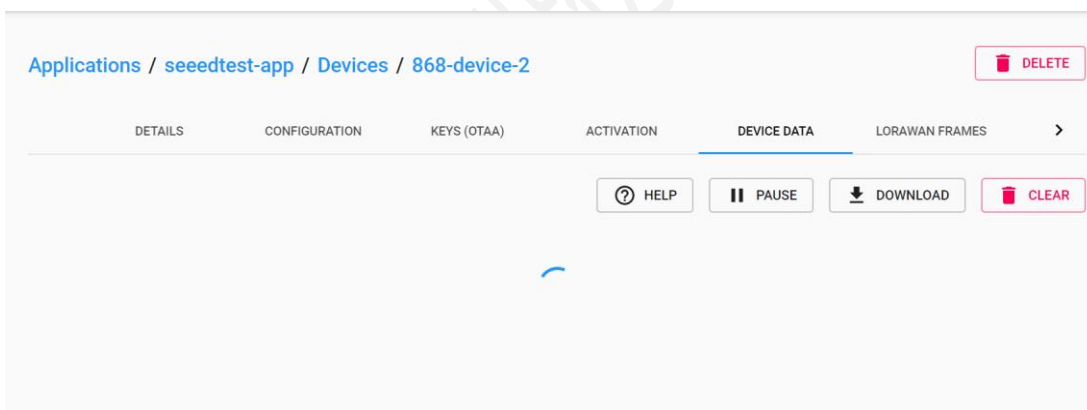
[CREATE DEVICE](#)

进入设备详情页，点击“KEYS (OTAA)”页面，在“Application key”输入在 [2.获取 SenseCAP 终端的参数](#) 部分通过 HTTP API 获取的 App Key，其他选择默认，完成设置。



#### 4) 传感器上电开机。

首先在浏览器中回到设备详情页的“DEVICE DATA”页面。此时处于刷新状态，无任何数据。



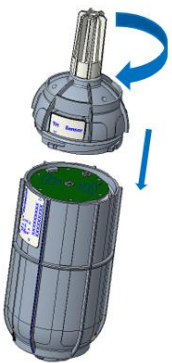
传感器的电源开关在设备内部，手握住逆时针转动即可很方便地拧开传感器头。



拧开后，拨动电源开关到 ‘ON’ 一侧，右边的 LED 灯会闪烁一下，表示上电成功。等待 10s 左右，LED 灯会连续快闪 2s，表示设备入网成功。同时，在页面中会出现入网数据。（若未出现数据，可按传感器板面正中的 RESET 按键，重新入网）

DETAILS	CONFIGURATION	KEYS (OTAA)	ACTIVATION	DEVICE DATA	LORAWAN FRAMES	FIRMWARE
<span>?</span> HELP <span>  </span> PAUSE <span>↓</span> DOWNLOAD <span>🗑️</span> CLEAR						
12:07:39 AM	uplink					▼
12:02:35 AM	uplink					▼
11:57:31 PM	uplink					▼
11:52:27 PM	uplink					▼
11:52:16 PM	uplink					▼
11:52:07 PM	uplink					▼
11:51:50 PM	uplink					▼
11:51:36 PM	uplink					▼
11:51:17 PM	uplink					▼
11:50:59 PM	uplink					▼
11:50:59 PM	join					▼

入网成功后，请快速将传感器头插入装回设备主体，顺时针将传感器头拧入，装回传感器头时注意两张标签要对齐，内部的连接器才是可靠连接状态，只有传感器头装回后，设备才能正常上传测量到的环境数据。



### 5) 添加更多的设备

再次添加新的 SenseCAP 设备，只需要重复以上第 3)、第 4) 步骤即可。

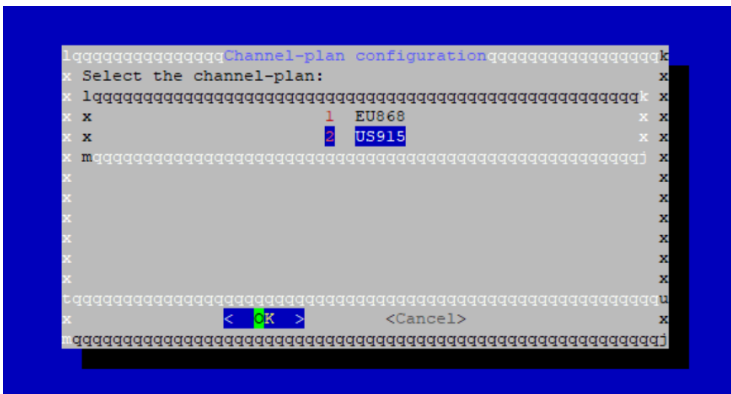
## 5.3 US915 网关配置和设备注册

在配置中，除网关配置与 EU868 不相同，其余步骤均与 [3.2 EU868 网关配置和设备注册](#) 相同。

参照 [3.2.1 网关配置](#)

### 1) 网关频段选择

选择 “US915” 。



### 2) 网页中的网关参数配置

可使用的通道：US902-908: 8, 9, 10, 11, 12, 13, 14, 15

**Gateway-profiles / Create**

Name \*

915-seeed

A short name identifying the gateway-profile.

Enabled channels \*

8, 9, 10, 11, 12, 13, 14, 15

The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters sepecification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

Network-server \*

buildin

[ADD EXTRA CHANNEL](#)   [CREATE GATEWAY-PROFILE](#)

### 3) 网关本地文件的参数配置

替代参数部分，使用附录 US915 提供的参数。

## 6. 数据包解析

### 1) 初始化数据包

time	counter	port		
16:30:52		0		
16:30:51	5	2	confirmed	payload: 00 19 00 BBFD 57 00 00 00 C9 79
16:30:33		0		
16:30:33	4	2	confirmed	payload: 01 06 00 00 00 00 00 2F 87
16:30:12		0		
16:30:11	3	2	confirmed	payload: 00 00 00 02 02 00 01 00 07 00 64 00 05 00 01 01 00 01 01 00 01 01 02 00 51 01 00 15 01 03 00 3C
16:30:00		0		
16:30:00	2	2	confirmed	payload: 00 00 00 00 00 00 00 00 00
16:29:40		0		
16:29:39	0	2	confirmed	payload: 00 00 00 00 00 00 00 00 00
16:29:30				dev addr: 27 00 34 03 app eui: 80 00 00 00 00 00 00 06 dev eui: 2C F7 F1 20 10 70 00 3F

SenseCAP 传感器终端在上电开机或者重启后都会进行 OTAA 入网，它会发送一系列的数据包到服务器。这些数据包依次包含：

1. 两个负载全 0 的数据包，用于清空来自于服务器的下行消息。
2. 两个数据包用于发送设备信息，包括硬件版本，软件版本，电池电量，每个通道的硬件和软件版本，每个通道的 sensor EUI，每个通道的电源时间计数。
3. 一个数据包用于向服务器请求 UTC 时间。
4. 一个数据包用于从服务器接受预定的 UTC 时间消息。
5. 一个数据包用于响应服务器发送的 UTC 时间（校验）。
6. 之后传感器终端将发送第一包传感器数据。

大多数情况下，我们不需要了解这些初始数据包（Seeed 的应用程序会解析这些包并使用），用户的应用程序可以忽略 UTC 时间请求，它是安全的，我们需要使用的应该是传感器测量数据包，它的帧计数通常是 7。

## 2) 数据包结构

如图所示，这里的数据包是 LoRaWAN 消息的 FRMPayload 部分。我们为数据包定义了另一种格式。

Radio PHY layer:

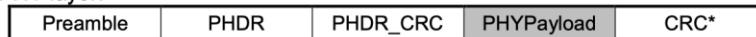
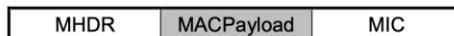
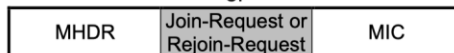


Figure 5: Radio PHY structure (CRC\* is only available on uplink messages)

PHYPayload:



or



or

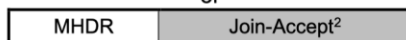


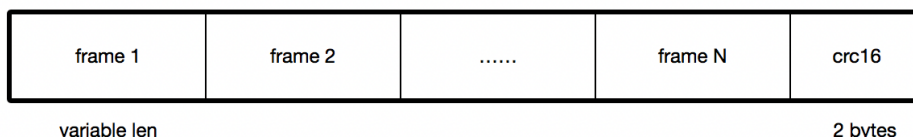
Figure 6: PHY payload structure

MACPayload:

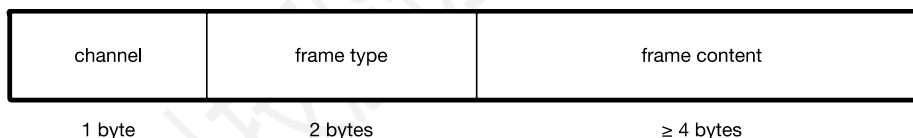


Figure 7: MAC payload structure

数据包的顶层结构如下图所示：



帧的长度是可变的，但是对于特定的帧类型，长度是已知的。框架结构如下图所示：



其中通道定义为下表：

Channel Number	Description
0x0	0x0 通道表示包含电池的节点的微控制器和 RF 部分。
0x1	0x1 通道是传感器探头连接到微控制器的第一个传感器通道。
0x2 及以上	Sensor hub 采用 0x2 通道及以上。

帧的类型字段和内容以及任何子字段都以小端模式的顺序发送，即先发送 LSB。帧字段及子字段等定义如下

表所示：

Frame Type	Description	1 <sup>st</sup> Byte	2 <sup>nd</sup> Byte	3 <sup>rd</sup> Byte	4 <sup>th</sup> Byte
Hex					
0x0000	控制器主板的版本	uint8, 硬件	uint8, 硬件	uint8, 软件	uint8, 软件



		版本（整数）	版本（小数）	版本（整数）	版本（小数）
0x0001	传感器主板的版本	uint8, 硬件版本（整数）	uint8, 硬件版本（小数）	uint8, 软件版本（整数）	uint8, 软件版本（小数）
0x0002	Sensor EUI（低位）	8 bytes EUI 中的低 4 bytes			
0x0003	Sensor EUI（高位）	8 bytes EUI 中的高 4 bytes			
0x0004	传感器生产日期	uint16, 年		uint8, 月	uint8, 日
0x0005	传感器上电时间（秒）	uint32			
0x0006	传感器空气闲置时间, 单位（天）	uint32			
0x0007	电量百分比	uint16, 终端电量的百分比, 0~100		uint16, 电量上报间隔（分钟）	
0x1200	传感器探头移除	-	-	-	-
0x0019	时间校准请求	特殊数据帧, 包含 6 bytes 内容			
0x0083	时间校准请求响应（下行）	特殊数据帧, 包含 6 bytes 内容			
0x001A	时间校准完成 ack	特殊数据帧, 包含 6 bytes 内容			
0x1001	空气温度	int32, 实际测量值 * 1000			
0x1002	空气湿度	后面均和 0x1001 相同, 都是测量值的 ID, 包含内容为处理后的测量值。 获取更多测量值, 请参考: <a href="https://sensecap-docs.seeed.cc/measurement_list.html">https://sensecap-docs.seeed.cc/measurement_list.html</a>			
0x1003	光照				
0x1004	二氧化碳				
0x1005	大气压力				
0x1006	土壤温湿度				
.....					

**例 1**

 光照传感器上传一包数据: **01** **03 10** A0 4E 02 00 **08 A0**
**08 A0** 是 CRC 校验部分

**01** 是 channel 编号

**03 10** 实际上是 0x1003, 它是光照的测量 ID

**A0 4E 02 00** 实际上是 0x00024EA0，转换为十进制的值为 151200，用该值除以 1000 即为实际测量值，由此可得出光照的实际测量值为 151.2Lux

### 例 2

温湿度传感器上传一包数据：01011090650000010210F4C90000345C

将数据分段：

**01 01 10 90 65 00 00**  
**01 02 10 F4 C9 00 00**  
**34 5C**

**01** 是 channel 编号

**01 10** 实际上是 0x1001，它是空气温度的测量 ID

**90 65 00 00** 实际上是 0x00006590，转换为十进制的值为 26000，用该值除以 1000 即为实际测量值，由此可得出空气温度的实际测量值为 26.0°C

**01** 是 channel 编号

**02 10** 实际上是 0x1002，它是空气湿度的测量 ID

**F4 C9 00 00** 实际上是 0x0000C9F4，转换为十进制的值为 51700，用该值除以 1000 即为实际测量值，由此可得出空气湿度的实际测量值为 51.7%RH

**34 5C** 是 CRC 校验部分

### 例 3

我们收到一个数据包的内容为：**00 07 00 64 00 05 00 xx xx**

**xx xx** 是 CRC 校验

**00** 是 channel 编号，这里 00 代表控制板

07 00 实际是 0x0007, 代表帧的类型, 对应上表为电池电来那个

64 00 实际是 0x0064, 代表电量百分比 (%), 十进制的值为 100, 即电量为 100%

05 00 实际是 0x0005, 代表电量上报间隔 (分钟), 十进制的值为 5, 即电量上报间隔为 5 分钟一次

综上, 可以总结处以下几点:

1. 最后两个字节为 CRC, 可以计算前两个字节的 CRC, 并比较 CRC
2. 从头部提取第 1 个字节, 这是 channel 编号
3. 之后的 2 个字节, 这是帧类型, 可以查找帧内容的长度, 并提取相应的长度
4. 剩下的根据帧类型去提取每个子字段的含义。
5. 重复步骤 2~4

# 代码附录

## EU868

```
"SX1301_conf": {
  "lorawan_public": true,
  "clksrc": 1, /* radio_1 provides clock to concentrator */
  "lbt_cfg": {
    "enable": false,
    "rssi_target": 160, /* rssi in dBm = -lbt_rssi_target/2 */
    "nb_channel": 1,
    "start_freq": 869525000,
    "scan_time_us": 5000,
    "tx_delay_1ch_us": 4000000,
    "tx_delay_2ch_us": 4000000
  },
  "antenna_gain": 0, /* antenna gain, in dBi */
  "radio_0": {
    "enable": true,
    "type": "SX1257",
    "freq": 867500000,
    "rssi_offset": -166.0,
    "tx_enable": true,
    "tx_notch_freq": 129000,
    "tx_freq_min": 863000000,
    "tx_freq_max": 870000000
  },
  "radio_1": {
    "enable": true,
    "type": "SX1257",
    "freq": 868500000,
    "rssi_offset": -166.0,
    "tx_enable": false
  },
  "chan_multiSF_0": {
    /* Lora MAC channel, 125kHz, all SF, 868.1 MHz */
    "enable": true,
    "radio": 1,
    "if": -400000
  },
  "chan_multiSF_1": {
    /* Lora MAC channel, 125kHz, all SF, 868.3 MHz */
```

```
"enable": true,
"radio": 1,
"if": -200000
},
"chan_multiSF_2": {
/* Lora MAC channel, 125kHz, all SF, 868.5 MHz */
"enable": true,
"radio": 1,
"if": 0
},
"chan_multiSF_3": {
/* Lora MAC channel, 125kHz, all SF, 867.1 MHz */
"enable": true,
"radio": 0, /* 0, -400000 */
"if": -400000
},
"chan_multiSF_4": {
/* Lora MAC channel, 125kHz, all SF, 867.3 MHz */
"enable": true,
"radio": 0, /* 0, -200000 */
"if": -200000
},
"chan_multiSF_5": {
/* Lora MAC channel, 125kHz, all SF, 867.5 MHz */
"enable": true,
"radio": 0, /* 0, 0 */
"if": 0
},
"chan_multiSF_6": {
/* Lora MAC channel, 125kHz, all SF, 867.7 MHz */
"enable": true,
"radio": 0,
"if": 200000
},
"chan_multiSF_7": {
/* Lora MAC channel, 125kHz, all SF, 867.9 MHz */
"enable": true,
"radio": 0,
"if": 400000
},
"chan_Lora_std": {
/* Lora standard channel, 250kHz, SF7, 868.3 MHz */
"enable": true,
"radio": 1,
```

```
"if": -200000,
"bandwidth": 250000,
"spread_factor": 7
},
"chan_FSK": {
  /* FSK 50kbps channel, 868.8 MHz */
  "enable": true,
  "radio": 1,
  "if": 300000,
  "bandwidth": 50000, /* 125000 */
  /*"freq_deviation": 25000,*/
  "datarate": 50000
},
"tx_lut_0": {
  /* TX gain table, index 0 */
  "pa_gain": 0,
  "mix_gain": 8,
  "rf_power": -6,
  "dig_gain": 0
},
"tx_lut_1": {
  /* TX gain table, index 1 */
  "pa_gain": 0,
  "mix_gain": 10,
  "rf_power": -3,
  "dig_gain": 0
},
"tx_lut_2": {
  /* TX gain table, index 2 */
  "pa_gain": 0,
  "mix_gain": 12,
  "rf_power": 0,
  "dig_gain": 0
},
"tx_lut_3": {
  /* TX gain table, index 3 */
  "pa_gain": 1,
  "mix_gain": 8,
  "rf_power": 3,
  "dig_gain": 0
},
"tx_lut_4": {
  /* TX gain table, index 4 */
  "pa_gain": 1,
```

```
"mix_gain": 10,
"rf_power": 6,
"dig_gain": 0
},
"tx_lut_5": {
  /* TX gain table, index 5 */
  "pa_gain": 1,
  "mix_gain": 12,
  "rf_power": 10,
  "dig_gain": 0
},
"tx_lut_6": {
  /* TX gain table, index 6 */
  "pa_gain": 1,
  "mix_gain": 13,
  "rf_power": 11,
  "dig_gain": 0
},
"tx_lut_7": {
  /* TX gain table, index 7 */
  "pa_gain": 2,
  "mix_gain": 9,
  "rf_power": 12,
  "dig_gain": 0
},
"tx_lut_8": {
  /* TX gain table, index 8 */
  "pa_gain": 1,
  "mix_gain": 15,
  "rf_power": 13,
  "dig_gain": 0
},
"tx_lut_9": {
  /* TX gain table, index 9 */
  "pa_gain": 2,
  "mix_gain": 10,
  "rf_power": 14,
  "dig_gain": 0
},
"tx_lut_10": {
  /* TX gain table, index 10 */
  "pa_gain": 2,
  "mix_gain": 11,
  "rf_power": 16,
```

```
        "dig_gain": 0
    },
    "tx_lut_11": {
        /* TX gain table, index 11 */
        "pa_gain": 3,
        "mix_gain": 9,
        "rf_power": 20,
        "dig_gain": 0
    },
    "tx_lut_12": {
        /* TX gain table, index 12 */
        "pa_gain": 3,
        "mix_gain": 10,
        "rf_power": 23,
        "dig_gain": 0
    },
    "tx_lut_13": {
        /* TX gain table, index 13 */
        "pa_gain": 3,
        "mix_gain": 11,
        "rf_power": 25,
        "dig_gain": 0
    },
    "tx_lut_14": {
        /* TX gain table, index 14 */
        "pa_gain": 3,
        "mix_gain": 12,
        "rf_power": 26,
        "dig_gain": 0
    },
    "tx_lut_15": {
        /* TX gain table, index 15 */
        "pa_gain": 3,
        "mix_gain": 14,
        "rf_power": 27,
        "dig_gain": 0
    }
},
```

## US915

```
"SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1,
    "clksrc_desc": "radio_1 provides clock to concentrator for most devices except MultiTech. For MultiTech set to 0.",
```



```
"antenna_gain": 0,
"antenna_gain_desc": "antenna gain, in dBi",
"radio_0": {
  "enable": true,
  "type": "SX1257",
  "freq": 904300000,
  "rssi_offset": -166.0,
  "tx_enable": true,
  "tx_freq_min": 923000000,
  "tx_freq_max": 928000000
},
"radio_1": {
  "enable": true,
  "type": "SX1257",
  "freq": 905000000,
  "rssi_offset": -166.0,
  "tx_enable": false
},
"chan_multiSF_0": {
  "desc": "Lora MAC, 125kHz, all SF, 903.9 MHz",
  "enable": true,
  "radio": 0,
  "if": -400000
},
"chan_multiSF_1": {
  "desc": "Lora MAC, 125kHz, all SF, 904.1 MHz",
  "enable": true,
  "radio": 0,
  "if": -200000
},
"chan_multiSF_2": {
  "desc": "Lora MAC, 125kHz, all SF, 904.3 MHz",
  "enable": true,
  "radio": 0,
  "if": 0
},
"chan_multiSF_3": {
  "desc": "Lora MAC, 125kHz, all SF, 904.5 MHz",
  "enable": true,
  "radio": 0,
  "if": 200000
},
"chan_multiSF_4": {
  "desc": "Lora MAC, 125kHz, all SF, 904.7 MHz",
```

```
"enable": true,
"radio": 1,
"if": -300000
},
"chan_multiSF_5": {
  "desc": "Lora MAC, 125kHz, all SF, 904.9 MHz",
  "enable": true,
  "radio": 1,
  "if": -100000
},
"chan_multiSF_6": {
  "desc": "Lora MAC, 125kHz, all SF, 905.1 MHz",
  "enable": true,
  "radio": 1,
  "if": 100000
},
"chan_multiSF_7": {
  "desc": "Lora MAC, 125kHz, all SF, 905.3 MHz",
  "enable": true,
  "radio": 1,
  "if": 300000
},
"chan_Lora_std": {
  "desc": "Lora MAC, 500kHz, SF8, 904.6 MHz",
  "enable": true,
  "radio": 0,
  "if": 300000,
  "bandwidth": 500000,
  "spread_factor": 8
},
"chan_FSK": {
  "desc": "FSK 100kbps channel, 903.0 MHz",
  "enable": false,
  "radio": 0,
  "if": 300000,
  "bandwidth": 250000,
  "datarate": 100000
},
"tx_lut_0": {
  "desc": "TX gain table, index 0",
  "pa_gain": 0,
  "mix_gain": 8,
  "rf_power": -6,
  "dig_gain": 0
```

```
},
"tx_lut_1": {
  "desc": "TX gain table, index 1",
  "pa_gain": 0,
  "mix_gain": 10,
  "rf_power": -3,
  "dig_gain": 0
},
"tx_lut_2": {
  "desc": "TX gain table, index 2",
  "pa_gain": 0,
  "mix_gain": 12,
  "rf_power": 0,
  "dig_gain": 0
},
"tx_lut_3": {
  "desc": "TX gain table, index 3",
  "pa_gain": 1,
  "mix_gain": 8,
  "rf_power": 3,
  "dig_gain": 0
},
"tx_lut_4": {
  "desc": "TX gain table, index 4",
  "pa_gain": 1,
  "mix_gain": 10,
  "rf_power": 6,
  "dig_gain": 0
},
"tx_lut_5": {
  "desc": "TX gain table, index 5",
  "pa_gain": 1,
  "mix_gain": 12,
  "rf_power": 10,
  "dig_gain": 0
},
"tx_lut_6": {
  "desc": "TX gain table, index 6",
  "pa_gain": 1,
  "mix_gain": 13,
  "rf_power": 11,
  "dig_gain": 0
},
"tx_lut_7": {
```

```
"desc": "TX gain table, index 7",
"pa_gain": 2,
"mix_gain": 9,
"rf_power": 12,
"dig_gain": 0
},
"tx_lut_8": {
"desc": "TX gain table, index 8",
"pa_gain": 1,
"mix_gain": 15,
"rf_power": 13,
"dig_gain": 0
},
"tx_lut_9": {
"desc": "TX gain table, index 9",
"pa_gain": 2,
"mix_gain": 10,
"rf_power": 14,
"dig_gain": 0
},
"tx_lut_10": {
"desc": "TX gain table, index 10",
"pa_gain": 2,
"mix_gain": 11,
"rf_power": 16,
"dig_gain": 0
},
"tx_lut_11": {
"desc": "TX gain table, index 11",
"pa_gain": 3,
"mix_gain": 9,
"rf_power": 20,
"dig_gain": 0
},
"tx_lut_12": {
"desc": "TX gain table, index 12",
"pa_gain": 3,
"mix_gain": 10,
"rf_power": 23,
"dig_gain": 0
},
"tx_lut_13": {
"desc": "TX gain table, index 13",
"pa_gain": 3,
```

```
"mix_gain": 11,
"rf_power": 25,
"dig_gain": 0
},
"tx_lut_14": {
  "desc": "TX gain table, index 14",
  "pa_gain": 3,
  "mix_gain": 12,
  "rf_power": 26,
  "dig_gain": 0
},
"tx_lut_15": {
  "desc": "TX gain table, index 15",
  "pa_gain": 3,
  "mix_gain": 14,
  "rf_power": 27,
  "dig_gain": 0
}
},
```

## CN470

```
"SX1301_conf": {
  "lorawan_public": true,
  "clksrc": 1, /* radio_1 provides clock to concentrator */
  "antenna_gain": 0, /* antenna gain, in dBi */
  "radio_0": {
    "enable": true,
    "type": "SX1255",
    "freq": 486600000,
    "rssi_offset": -176.0,
    "tx_enable": true,
    "tx_freq_min": 470000000,
    "tx_freq_max": 510000000
  },
  "radio_1": {
    "enable": true,
    "type": "SX1255",
    "freq": 487400000,
    "rssi_offset": -176.0,
    "tx_enable": false
  },
  "chan_multiSF_0": {
    "desc": "Lora MAC, 125kHz, all SF, 487.1 MHz",
    "enable": true,
```

```
"radio": 1,
  "if": -300000
},
"chan_multiSF_1": {
  "desc": "Lora MAC, 125kHz, all SF, 487.3 MHz",
  "enable": true,
  "radio": 1,
  "if": -100000
},
"chan_multiSF_2": {
  "desc": "Lora MAC, 125kHz, all SF, 487.5 MHz",
  "enable": true,
  "radio": 1,
  "if": 100000
},
"chan_multiSF_3": {
  "desc": "Lora MAC, 125kHz, all SF, 487.7 MHz",
  "enable": true,
  "radio": 1,
  "if": 300000
},
"chan_multiSF_4": {
  "desc": "Lora MAC, 125kHz, all SF, 486.3 MHz",
  "enable": true,
  "radio": 0,
  "if": -300000
},
"chan_multiSF_5": {
  "desc": "Lora MAC, 125kHz, all SF, 486.5 MHz",
  "enable": true,
  "radio": 0,
  "if": -100000
},
"chan_multiSF_6": {
  "desc": "Lora MAC, 125kHz, all SF, 486.7 MHz",
  "enable": true,
  "radio": 0,
  "if": 100000
},
"chan_multiSF_7": {
  "desc": "Lora MAC, 125kHz, all SF, 486.9 MHz",
  "enable": true,
  "radio": 0,
  "if": 300000
}
```

```
},
"chan_Lora_std": {
  "desc": "Lora MAC channel, 500kHz, SF8, 903.0 MHz",
  "enable": false,
  "radio": 1,
  "if": -200000,
  "bandwidth": 250000,
  "spread_factor": 7
},
"chan_FSK": {
  "desc": "FSK 100kbps channel, 903.0 MHz",
  "enable": false,
  "radio": 1,
  "if": 300000,
  "bandwidth": 125000,
  "datarate": 50000
},
"tx_lut_0": {
  "desc": "TX gain table, index 0",
  "pa_gain": 0,
  "mix_gain": 8,
  "rf_power": -6,
  "dig_gain": 0
},
"tx_lut_1": {
  "desc": "TX gain table, index 1",
  "pa_gain": 0,
  "mix_gain": 10,
  "rf_power": -3,
  "dig_gain": 0
},
"tx_lut_2": {
  "desc": "TX gain table, index 2",
  "pa_gain": 0,
  "mix_gain": 12,
  "rf_power": 0,
  "dig_gain": 0
},
"tx_lut_3": {
  "desc": "TX gain table, index 3",
  "pa_gain": 1,
  "mix_gain": 8,
  "rf_power": 3,
  "dig_gain": 0
}
```

```
},
"tx_lut_4": {
  "desc": "TX gain table, index 4",
  "pa_gain": 1,
  "mix_gain": 10,
  "rf_power": 6,
  "dig_gain": 0
},
"tx_lut_5": {
  "desc": "TX gain table, index 5",
  "pa_gain": 1,
  "mix_gain": 12,
  "rf_power": 10,
  "dig_gain": 0
},
"tx_lut_6": {
  "desc": "TX gain table, index 6",
  "pa_gain": 1,
  "mix_gain": 13,
  "rf_power": 11,
  "dig_gain": 0
},
"tx_lut_7": {
  "desc": "TX gain table, index 7",
  "pa_gain": 2,
  "mix_gain": 9,
  "rf_power": 12,
  "dig_gain": 0
},
"tx_lut_8": {
  "desc": "TX gain table, index 8",
  "pa_gain": 1,
  "mix_gain": 15,
  "rf_power": 13,
  "dig_gain": 0
},
"tx_lut_9": {
  "desc": "TX gain table, index 9",
  "pa_gain": 2,
  "mix_gain": 10,
  "rf_power": 14,
  "dig_gain": 0
},
"tx_lut_10": {
```



```
    "desc": "TX gain table, index 10",
    "pa_gain": 2,
    "mix_gain": 11,
    "rf_power": 16,
    "dig_gain": 0
  },
  "tx_lut_11": {
    "desc": "TX gain table, index 11",
    "pa_gain": 3,
    "mix_gain": 9,
    "rf_power": 20,
    "dig_gain": 0
  },
  "tx_lut_12": {
    "desc": "TX gain table, index 12",
    "pa_gain": 3,
    "mix_gain": 10,
    "rf_power": 23,
    "dig_gain": 0
  },
  "tx_lut_13": {
    "desc": "TX gain table, index 13",
    "pa_gain": 3,
    "mix_gain": 11,
    "rf_power": 25,
    "dig_gain": 0
  },
  "tx_lut_14": {
    "desc": "TX gain table, index 14",
    "pa_gain": 3,
    "mix_gain": 12,
    "rf_power": 26,
    "dig_gain": 0
  },
  "tx_lut_15": {
    "desc": "TX gain table, index 15",
    "pa_gain": 3,
    "mix_gain": 14,
    "rf_power": 27,
    "dig_gain": 0
  }
},
```