Overview

Introduction

This series of products are LoRa modules using the new generation of SX1262 RF chip, with the features of long communication distance and strong anti-interference ability. Suitable for Sub-GHz frequency band networks, and is available in LF (410~510MHz) or HF (850~930MHz) frequency band versions. Combined with a LoRa gateway, it can be connected to servers such as TTN to build a LoRaWAN network. In addition to the basic LoRaWAN version, it also provides an optional GNSS positioning function with GPS/BD support.

Features

Common features:

Standard Raspberry Pi 40PIN GPIO header, supports Raspberry Pi series boards.

The new generation SX1262 has higher power efficiency and longer transmission distance than the SX1278.

Suitable for the Sub-GHz band, combined with LoRa gateway, can be quickly connected to a cloud server such as TTN to build a LoRaWAN network.

GNSS version features:

Onboard L76K module with GPS/BD support, provides accurate clock and location info for node module.

Onboard battery holder ,supports ML1220 rechargeable battery, for power off preserving information and hot starts.

Onboard 4 LED indicators for module operating status.

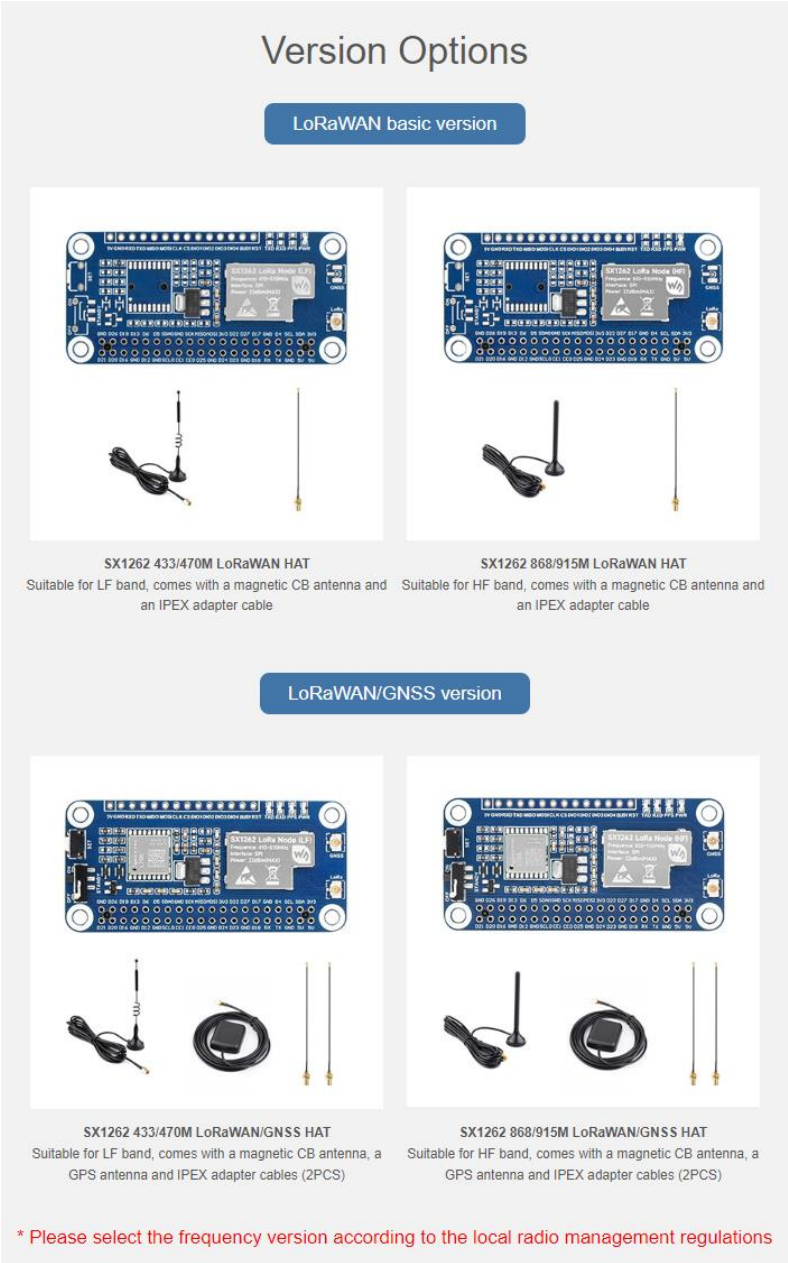
Comes with online development resources and manual(example in C)

Parameters

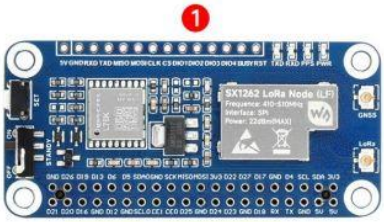
LORA PARAMETERS	PARAMETERS
RF CHIP	SX1262
FREQUENCY BAND	Sub-GHz: SX1262 433/470M LoRaWAN/GNSS HAT: 410~490MHz SX1262 868/915M LoRaWAN/GNSS HAT: 850~930MHz 433M&868M band for EU 915M band for US 470M band for China
MODULATION	LoRa/(G)FSK
EMIT POWER	22dBm@3.3V
INPUT	DC 5V/3A
OPERATING VOLTAGE	3.3V
MODULE CURRENT CONSUMPTION	transmitting current: 45mA@14dBm receiving current: 5.3mA@125KHz
COMMUNICATION BUS	SPI
OPERATING TEMPERATURE	-40 ~ 85℃
DIMENSIONS	19.00 ×22.00mm

Hardware Description

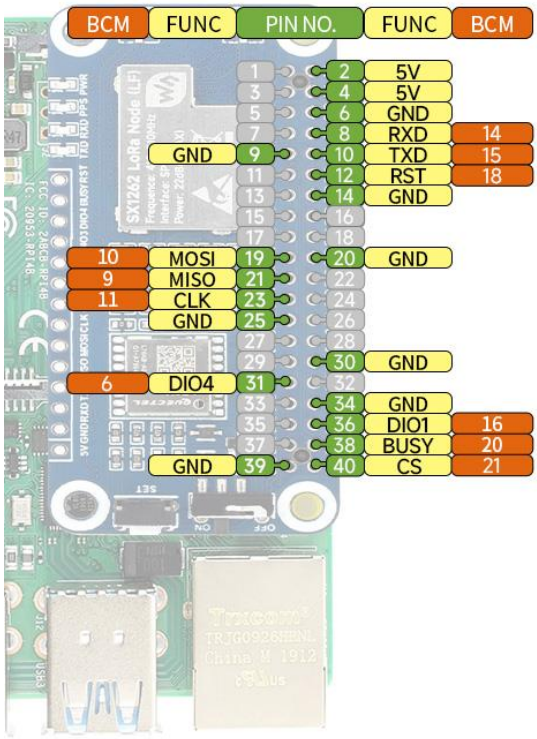
Hardware Selection



Hardware Connection



Pinout Definition



LoRa node	
CS	LoRa CS pin
CLK	LoRa CLK pin
MOSI	LoRa MOSI pin
MISO	LoRa MISO pin
BUSY	LoRa BUSY pin
DIO1	LoRa DIO1 pin
DIO4	LoRa transmit enable pin
RST	LoRa reset pin

GNSS module	
5V	5V power supply
GND	Ground
RXD	L76K UART RX
TXD	L76K UART TX

CS MOSI MISO CLK is the SPI interface of SX1262, and BUSY is the status pin of SX1262. Use MCU's SPI bus to communicate with SX1262 XXXM LoRaWAN/GNSS HAT. When MCU

reads and writes SX1262 registers, it needs to read and write parameters in the order of Opcode + Address + Data. For more information, please refer to chapters 8, 10, 11, and 12 of the datasheet
When the MCU reads and writes the SX1262 register, it needs to detect the BUSY pin first. The low level indicates that it is idle and can be read and written normally, and the high level indicates that it is busy and cannot read and write the register, as shown in the figure below.

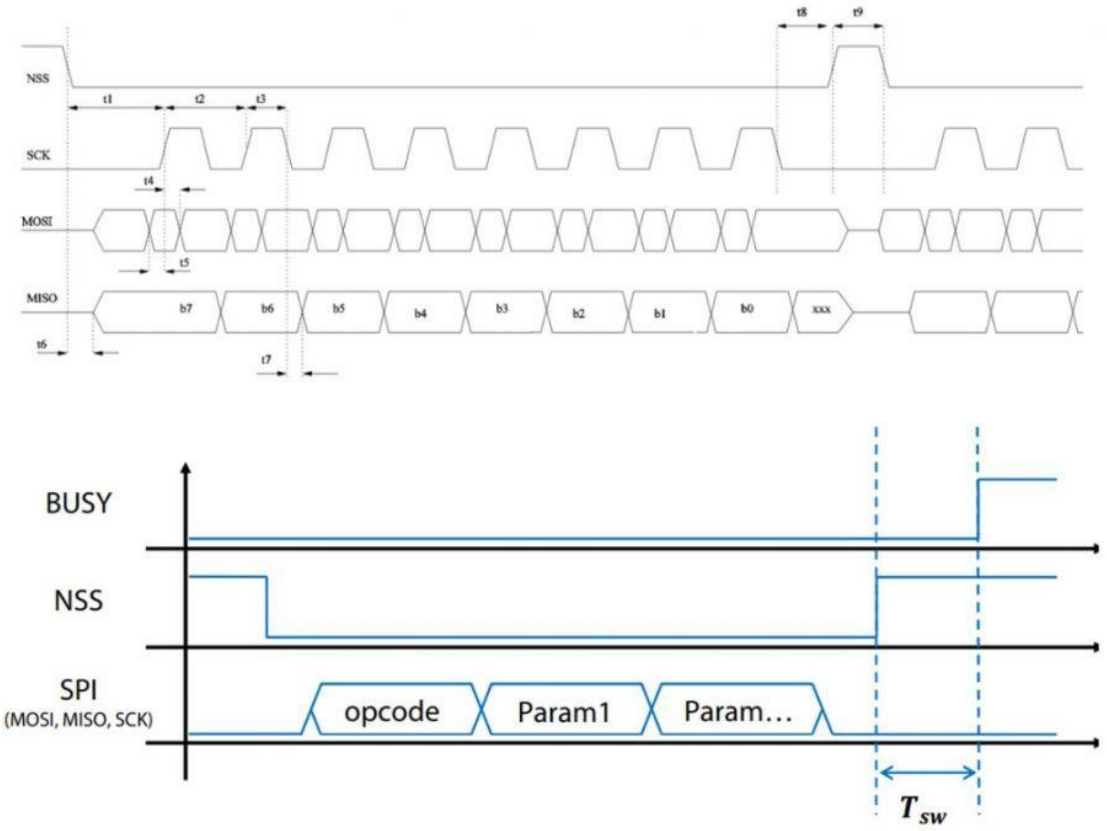


Figure 8-3: Switching Time Definition

Table 11-2: Commands to Access the Radio Registers and FIFO Buffer

Command	Opcode	Parameters	Description
WriteRegister	0x0D	address[15:0], data[0:n]	Write into one or several registers
ReadRegister	0x1D	address[15:0]	Read one or several registers
WriteBuffer	0x0E	offset, data[0:n]	Write data into the FIFO
ReadBuffer	0x1E	offset	Read data from the FIFO

RESET is the factory reset pin of SX1262, pull it low for 100us to restore the default parameters of the register, and keep a high level when working.
RXEN, TXEN are RF single-pole switch (SPDT
TXEN pin is connected to BCM(6), and RXEN pin is not used, which is -1 by default. Refer to the schmatic diagram
DIO1, DIO2, and DIO3 are SX1262 functional GPIO pins, which can be set as input and output to indicate various states of SX1262 (8.5 IRQ Handling in the figure below), usually, DIO1 is set as an interrupt output in the state shown in the figure below, and DIO2 is connected to RXEN and set

as the control pin of the RF single-pole switch, DIO3 is set to supply power to the TCXO, for details, please refer to Datasheet 8.3.2, 8.5 summary.

8.3.2 Digital Input/Output

Any of the 3 DIOs can be selected as an output interrupt source for the application. When the application receives ar interrupt, it can determine the source by using the command *GetIrqStatus(...)*. The interrupt can then be cleared using the *ClearIrqStatus(...)* command. The Pin Description is as follows:

DIO1 is the generic IRQ line, any interrupt can be mapped to DIO1. The complete list of available IRQ can be found ir Section 8.4 "Digital Interface Status versus Chip modes" on page 54.

DIO2 has a double functionality. As DIO1, DIO2 can be used as a generic IRQ line and any IRQ can be routed through this pin. Also, DIO2 can be configured to drive an RF switch through the use of the command *SetDio2AsRfSwitchCtrl(...)*. In this mode, DIO2 will be at a logical 1 during Tx and at a logical 0 in any other mode.

DIO3 also has a double functionality and as DIO1 or DIO2, it can be used as a generic IRQ line. Also, DIO3 can be used to automatically control a TCXO through the command *SetDio3AsTCXOCtrl(...)*. In this case, the device will automatically powe cycle the TCXO when needed.

8.5 IRQ Handling

In total there are 10 possible interrupt sources depending on the selected frame and chip mode. Each one can be enabled or masked. In addition, each one can be mapped to DIO1, DIO2 or DIO3.

Table 8-4: IRQ Status Registers

Bit	IRQ	Description	Modulation
0	TxDone	Packet transmission completed	All
1	RxDone	Packet received	All
2	PreambleDetected	Preamble detected	All
3	SyncWordValid	Valid Sync Word detected	FSK
4	HeaderValid	Valid LoRa® Header received	LoRa®
5	HeaderErr	LoRa® Header CRC error	LoRa®
6	CrcErr	Wrong CRC received	All
7	CadDone	Channel activity detection finished	LoRa®
8	CadDetected	Channel activity detected	LoRa®
9	Timeout	Rx or Tx Timeout	All

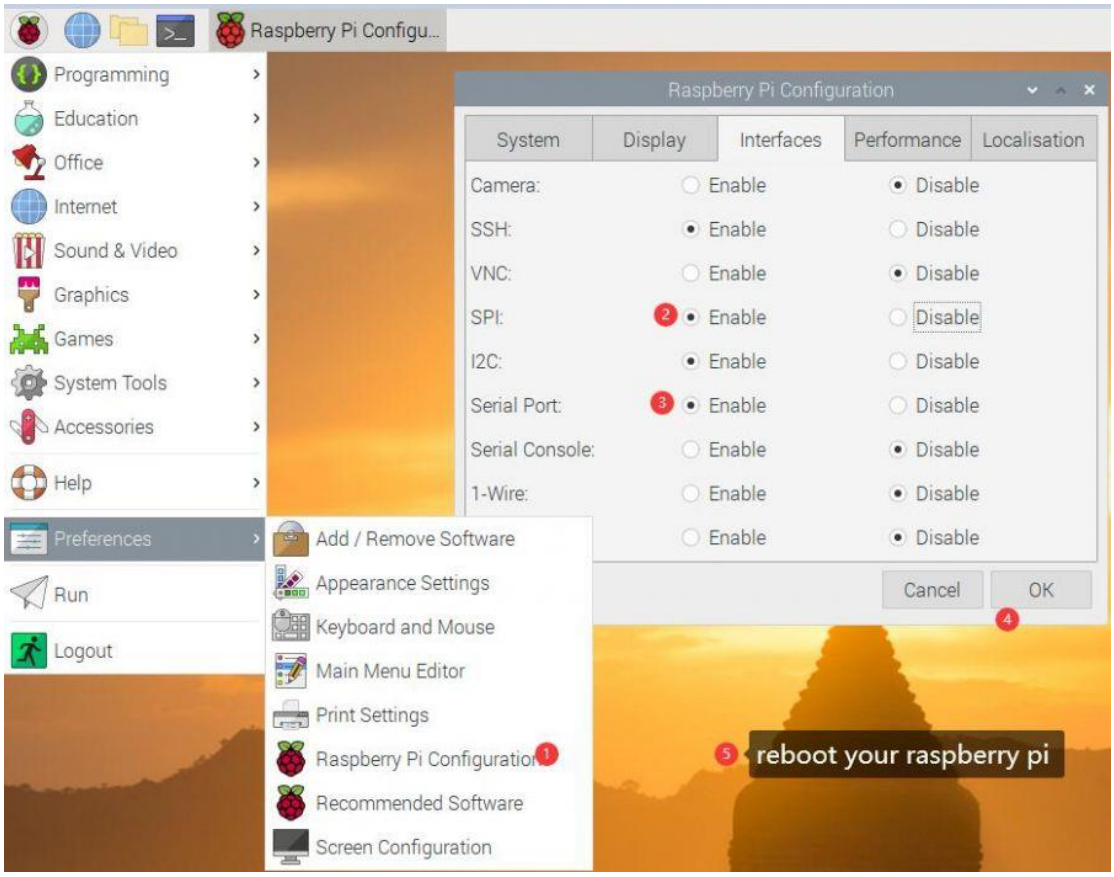
For more information on how to setup IRQ and DIOs, refer to the function *SetDioIrqParams()* in Section 13.3.1 "SetDioIrqParams" on page 79

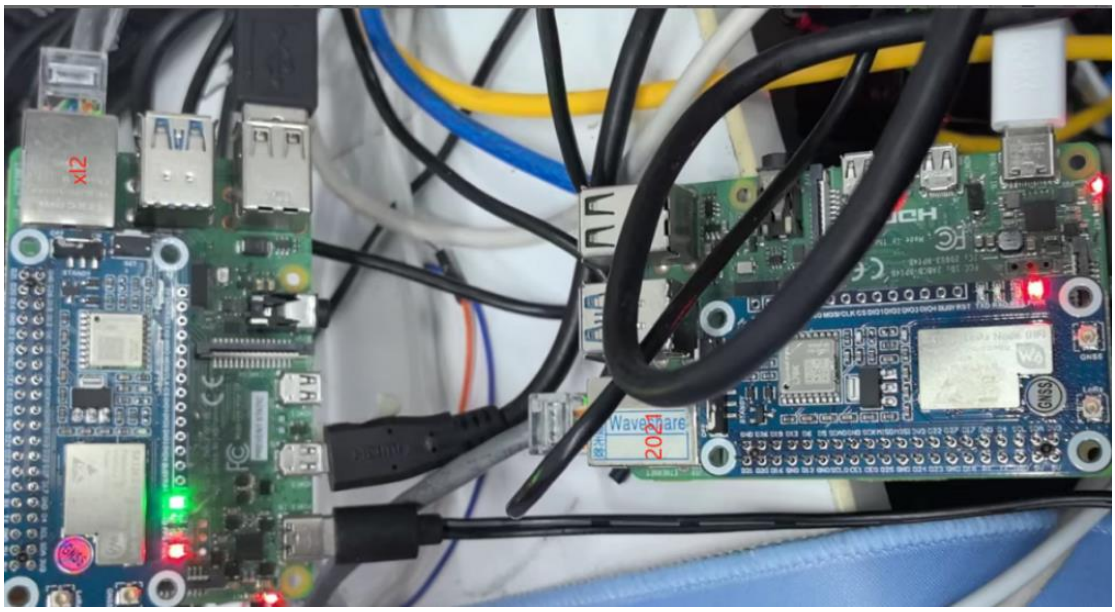
Dimensions

Unit: mm

Lora Beginner Test

If you are not familiar with Linux (or Raspberry Pi), please refer to the Raspberry Pi Documentation to set up the Raspberry Pi to turn on the SPI bus, etc., as shown in the following figure:





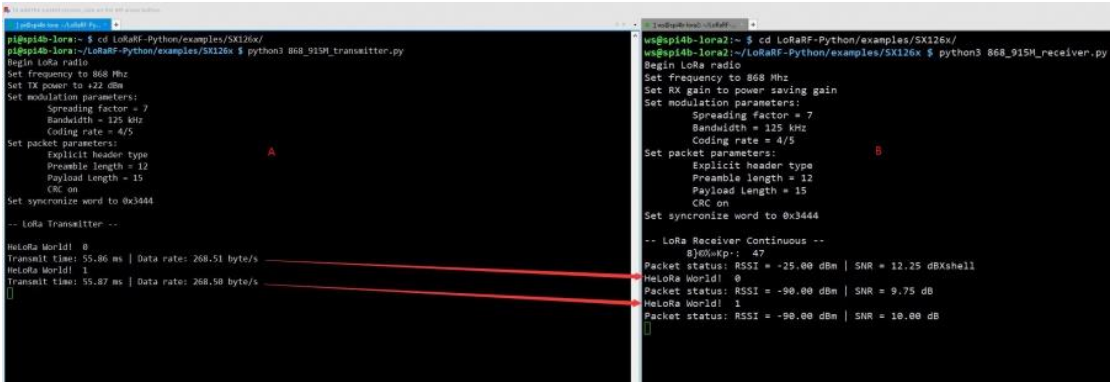
Demo Download

Open the Raspberry Pi terminal, and enter the "root" mode. Use the following commands to download sample demo and unzip:

```
sudo su

wget https://files.waveshare.com/wiki/SX1262-XXXM-LoRaWAN-GNSS-HAT/Sx126x_lorawan_hat_code.zip

unzip Sx126x_lorawan_hat_code.zip

cd cd sx126x_lorawan_hat_code/python/lora/

python3 setup.py install # If you need "sudo" privileges, add "sudo" in front of python 3, and be sure to install the libraries the first time you download the demo

cd example/SX126x/

python3 transmission.py # This sample demo requires two SX1262 XXXM LoRaWAN/GNSS HAT for send/receive tests, this command is for the transmitter module

python3 receiver_continuous.py # This sample demo requires two SX1262 XXXM LoRaWAN/GNSS HAT transceiver tests, this command is for the receiver module
```

sx126x_lorawan_hat_code/python/lora/examples/network, under the directory example demo for a single-channel gateway and node

```
cd sx126x_lorawan_hat_code/python/lora/examples/network

python3 LoRa_simple_node.py # This sample demo requires two SX1262 XXXM LoRaWAN/GNSS HAT for send/receive tests, this command is for the node module

python3 LoRa_simple_gateway.py # This sample demo requires two SX1262 XXXM LoRaWAN/GNSS HAT for send/receive tests, this command is for the single-channel gateway module
```



Example Analysis

This summarizes the code in transmission.py and receiver_continuous.py.

LoRa & LoRaWAN

What is LoRa ?

Semtech's LoRa is a long-distance, low-power wireless platform for the Internet of Things (IoT), which generally refers to radio frequency chips using LoRa technology. The main features are as follows：

The spread spectrum modulation technology adopted by LoRa (abbreviation of long range) is derived from Chirp Spread Spectrum (CSS) technology, which is one of the long-distance wireless transmission technology and LPWAN communication technology. Spread spectrum technology uses bandwidth for sensitivity technology, Wi-Fi, ZigBee, etc. all use spread spectrum technology, but the characteristic of LoRa modulation is that it is close to the limit of Shannon's theorem, and the sensitivity can be improved with maximum efficiency. Compared with traditional FSK technology, at the same communication rate, LoRa is more sensitive than FSK by 8 ~12dBm. At present, LoRa mainly operates in the ISM frequency band of Sub-GHz.

LoRa technology integrates technologies such as digital spread spectrum, digital signal processing,

and forward error correction coding, which greatly improves the performance of long-distance communication. The link budget of LoRa is better than any other standardized communication technology. Link budget refers to the main factors that determine the distance in a given environment.

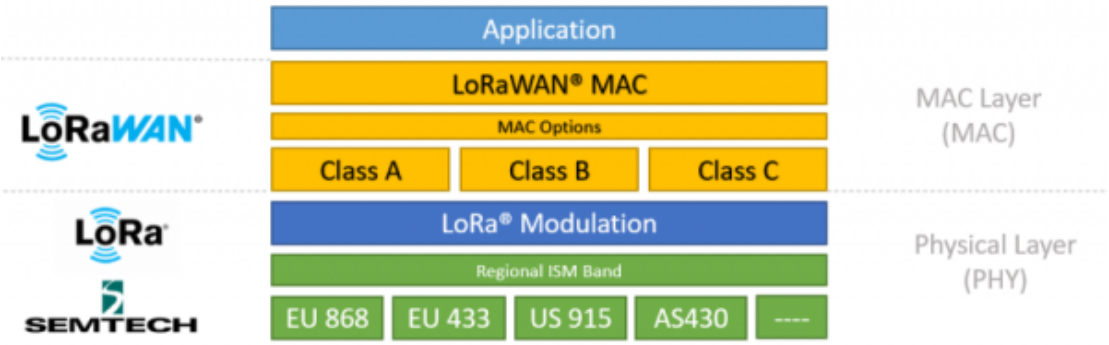
LoRa RF chips mainly include SX127X series, SX126X series, SX130X series, of which SX127X, SX126X series are used for LoRa nodes, and SX130X is used for LoRa gateways. For details, please refer to Semtech

What is LoRaWAN ?

LoRaWAN is an open protocol for low-power Wide Area Network(WAN), built on LoRa radio modulation technology. Designed to wirelessly connect battery-powered "things" to the Internet in regional, national, or global networks, and target critical Internet of Things (IoT) requirements such as bidirectional communication, end-to-end security, mobility, and localized services. The node wirelessly connects to the Internet with network access authentication, which is equivalent to establishing an encrypted communication channel between the node and the server. The LoRaWAN protocol level is shown in the figure below.

The Class A/B/C node devices in the MAC layer basically cover all the application scenarios of Internet of Things. The difference among them is that the time slots for nodes to send and receive are different.

EU868 and AS430 in the Modulation layer show that frequency band parameters are different in different countries. Please click the reference link for regional parameters.

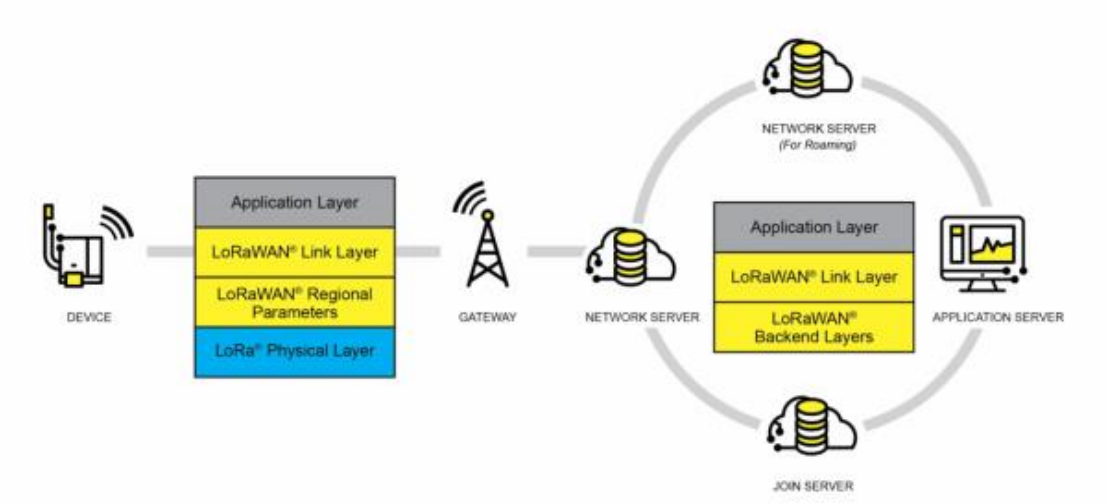


To achieve LoRaWAN network coverage in cities or other areas, it needs to be composed of four parts: node (LoRa node radio frequency chip), gateway (or base station, LoRa gateway radio frequency chip), server, and cloud, as shown in the following figure.

The DEVICE (node device) needs to initiate a network access request packet to the GATEWAY (gateway) and then to the server. After the authentication is passed, it can send and receive application data with the server normally.

GATEWAY (gateway) can communicate with the server through a wired network, 3/4/5G wireless network

The main operators on the server side are TTN, etc. For building cloud services by yourself, please refer to lorawan-stack, chirpstack



There are two methods for Raspberry Pico and Pico-LoRa-SX1262 to connect to the network via LoRaWAN: OTAA (Over-The-Air-Activation) and ABP (Activation By Personalization). Here we use OTAA, as shown below. Also you can refer to link 1, link 2 and source code

Step 1: Please send the "Join-Request" message to the network to join, and note that the join process is always initiated by the end device. The join-Request message can be transmitted at any rate and in one of the region-specific join channels. For example: in Europe, the end device can send the join-Request message at 868.10 MHz, 868.30MHz, or 838.50MHz. Also, the message can be sent to the network server by one or more gateway. In addition, you need to pay attention to choosing the applicable frequency band according to the local radio management regulations. You can refer to link or visitLoRa Alliance to search for the frequency distribution table. The Join-Request message is combined by the following field, AppEUI, and DevEUI are generated by registering on the server side.

AppEUI: A 64-bit globally unique application identifier in the IEEE EUI64 address space that uniquely identifies an entity capable of processing Join-Request frames.

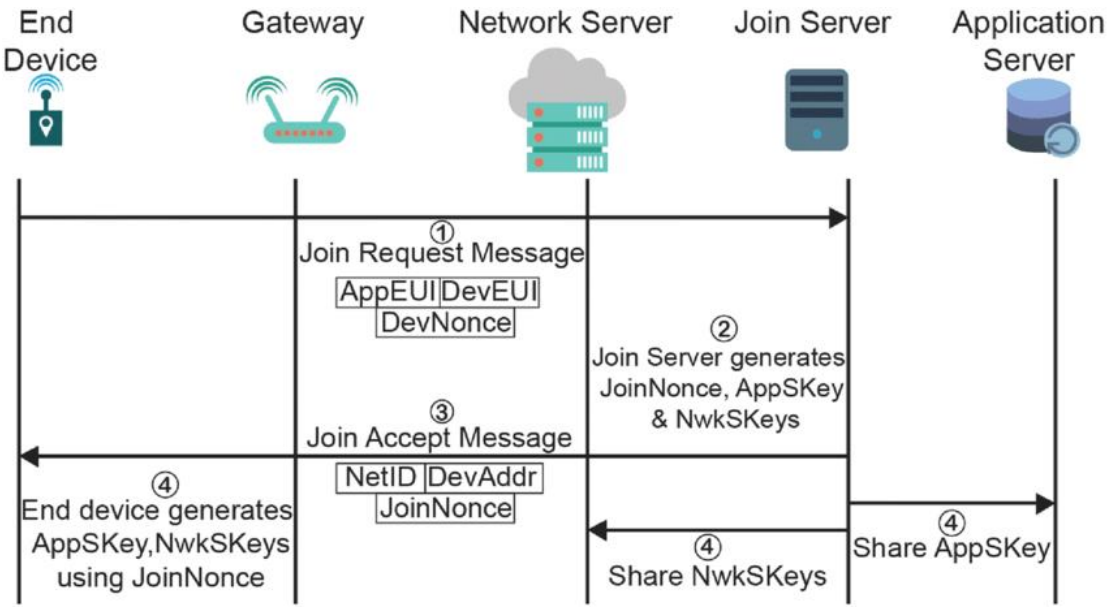
DevEUI: A 64-bit globally unique device identifier that uniquely identifies an end device in the IEEE EUI64 address space.

DevNonce: A unique random 2-byte value generated by the end device. The web server uses the DevNonce of each end device to keep track of their join requests. If the end device sends a join request with the previously used DevNonce (this case is called a replay attack), the network server will reject the join request and will not allow the end device to register with the network.

Step 2: The web server handles the Join-Request-Message. If the end device is allowed to join the network, the web server will generate two session keys (NwkSKey and AppSKey) and the Join-accept message. The Join-accept message itself is then encrypted using AppKey. The web server uses AES decryption operation in ECB mode to encrypt the Join-accept message.

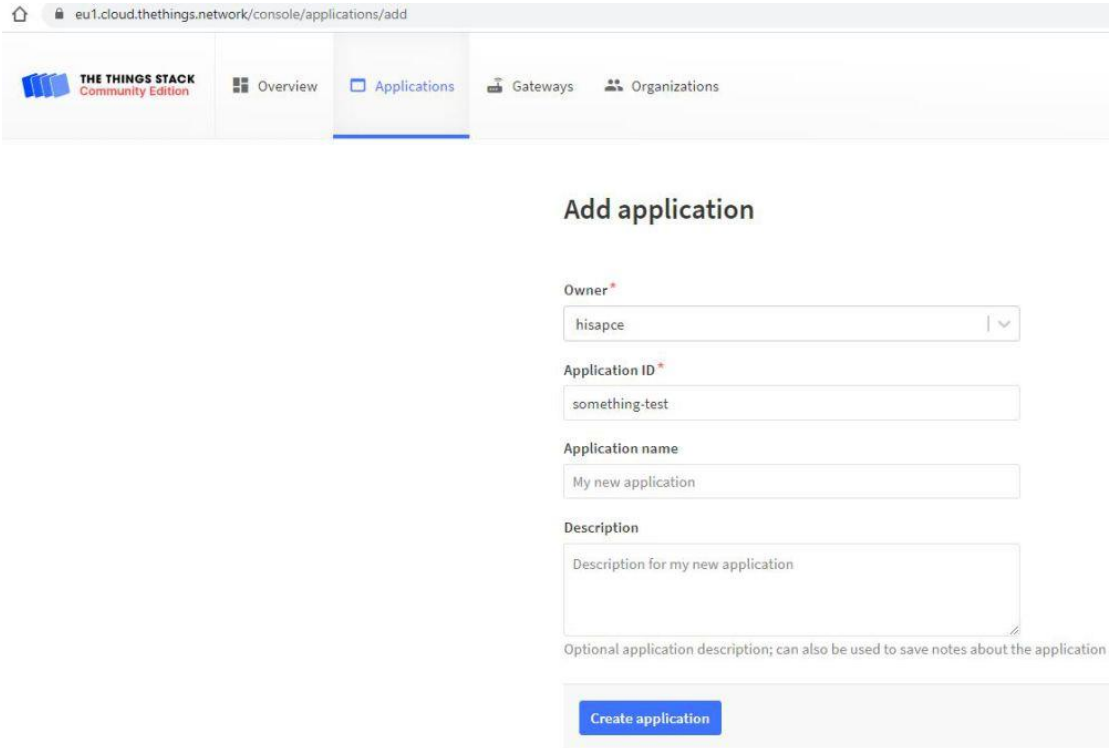
Step 3: The network server sends the encrypted Join-accept message back to the end device as a normal downlink.

Step 4: The end device uses AES to decrypt the Join-Accept message. And uses AppKey and AppNonce to generate two session keys(NwkSKey and AppSKey) for subsequent communication with the Networking server. Network Server also saves kSKey, Join server distributes AppSKey to Application Server.

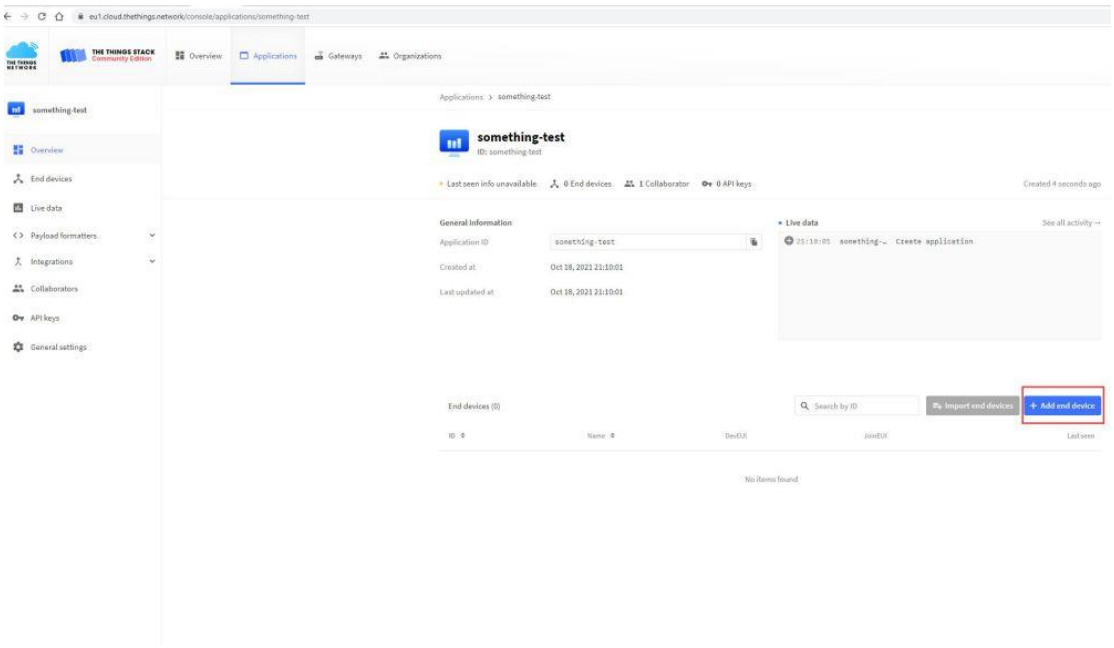


The DevEUI and AppEUI parameters used as terminal devices to access the Internet need to be registered and generated by the server. The specific process is as follows:

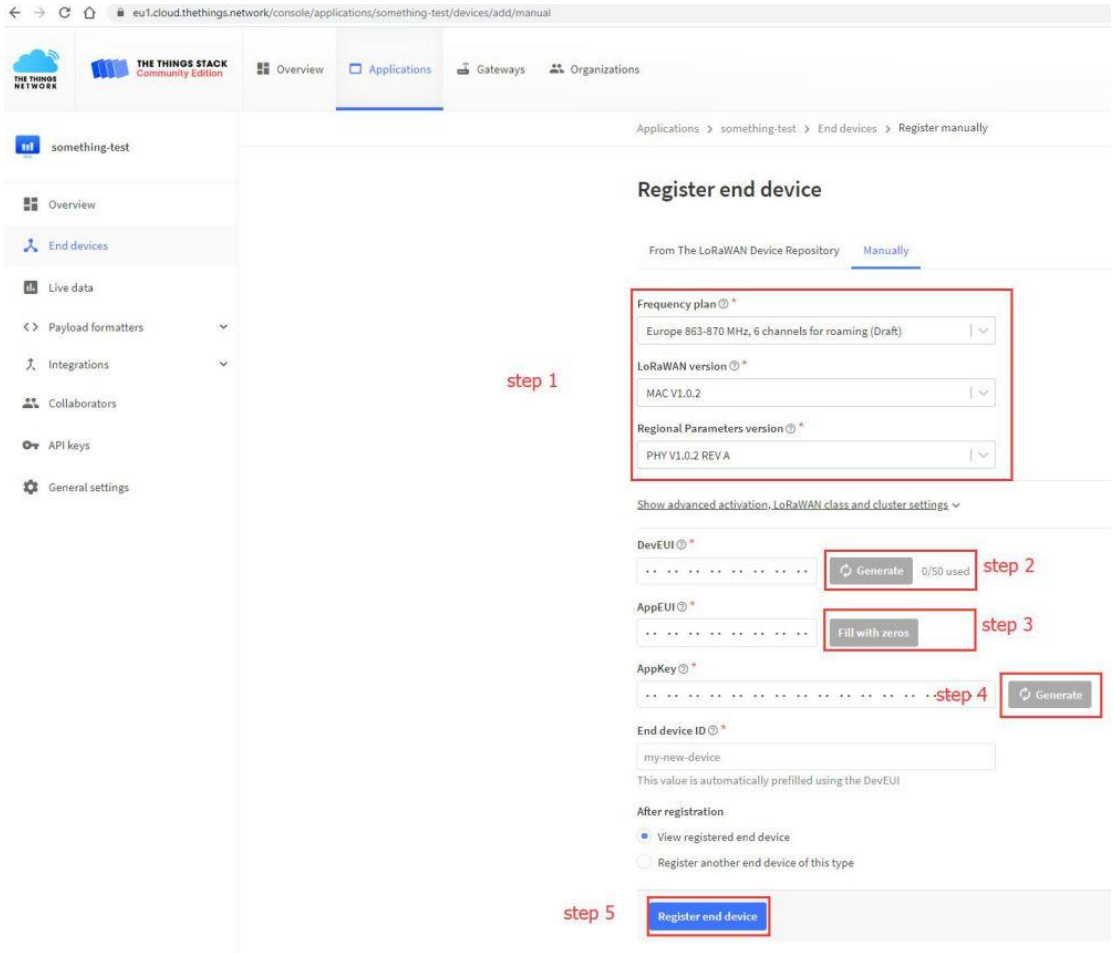
Register an account in TTS and log in. Create an Application.



Add an End Device

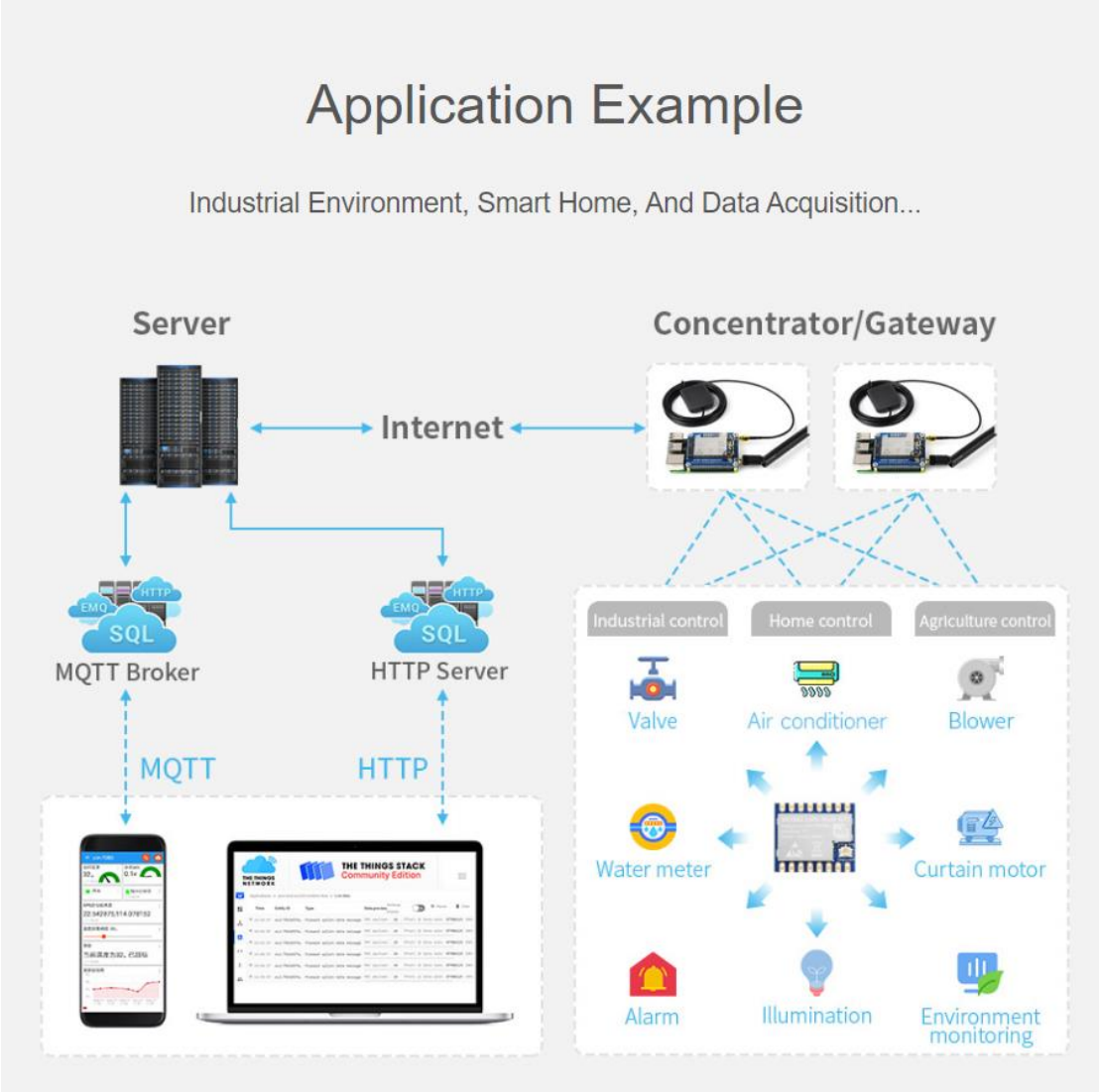


Configure the End Device as in the picture. Please save the DevEUI of Step2 and the AppEUI of Step3 for further use.



Application

LoRa devices and networks such as LoRaWAN enable smart IoT applications to help solve the planet's formidable challenges in energy management, natural resource reduction, pollution control, infrastructure efficiency, disaster prevention, and more. Semtech's LoRa devices have achieved hundreds of successful use cases in smart cities, homes and buildings, communities, metrology, supply chain, logistics, agriculture, and more. LoRa networks have covered hundreds of millions of devices in more than 100 countries and are committed to a smarter planet.



This product test is based on TTS(The THINGS STACK) and Semtech SX1302 official library, if users build their own cloud server, please click to refer to lorawan-stack, chirpstack.

Lorawan test

Use "sudo apt-get installconnectionpi" to install the GPIO access library written in C language for the BCM2835 used in Raspberry Pi;

1.2 Officially start

Connect the LoRa shield to the Raspberry Pi

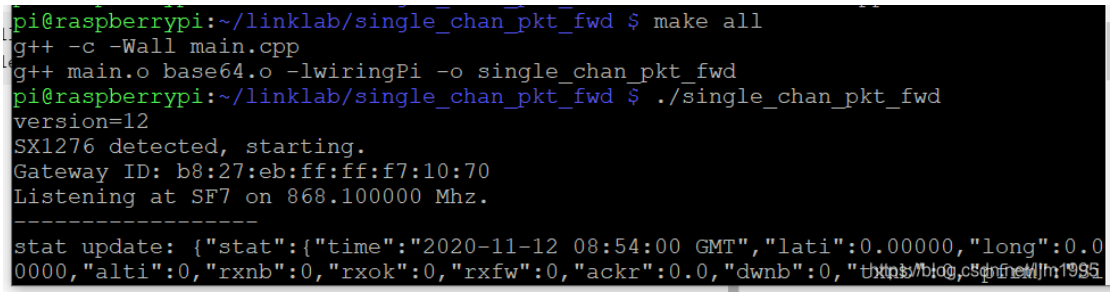
Download the single-channel LoRa gateway code to the Raspberry Pi

Download the single-channel LoRa gateway code to the Raspberry Pi

Source code address: https://github.com/tftelkamp/single_chan_pkt_fwd

Compile the code and run

```
$ make all
$ ./single_chan_pkt_fwd
```

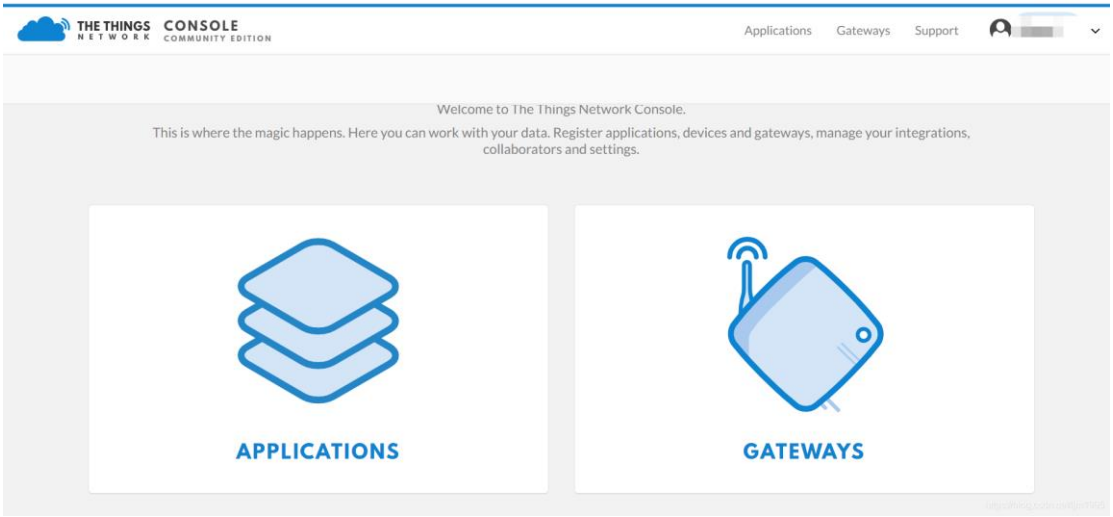


As shown above, the Raspberry Pi indicates that it has found the LoRa shield connected to itself. Record the "Gateway ID" part.

The work here is not yet completed, so put it aside for now and come back to do it.

TTN website related operations

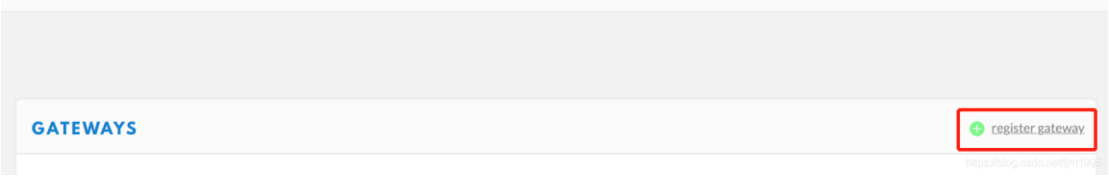
2.1 Register and log in to TTN



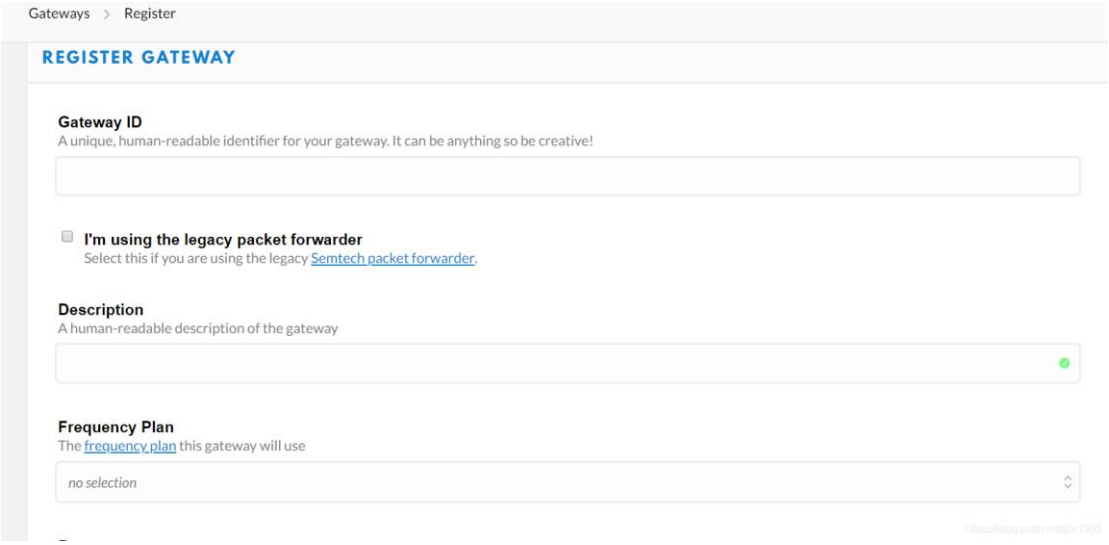
After entering TTN, click the drop-down arrow next to the avatar and select "console" to enter the console, as shown below.

2.2 Register Gateway

Select "GATEWAYS", and then select "register gateway" on the page that appears.

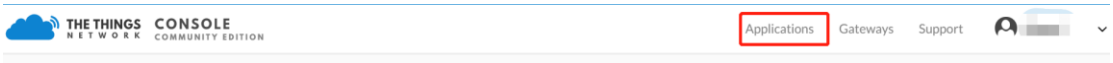


In the "Gateway ID" section, fill in the ID recorded above and tick the box below. After selecting the corresponding frequency band, click "Register Gateway" at the bottom of the page.



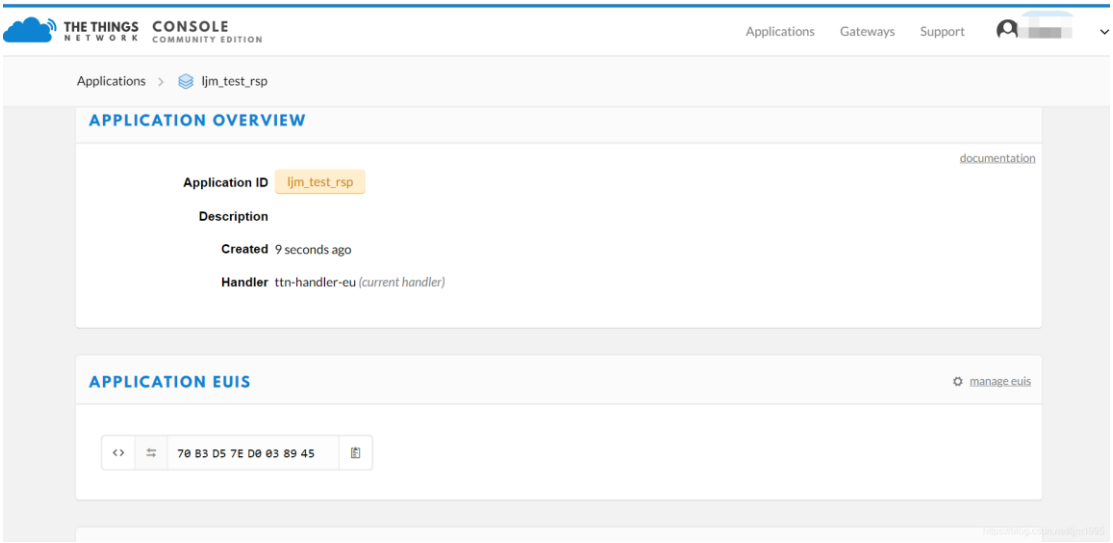
2.3 Register Application

Click "Applications" to enter the page. Click "add application" to enter the configuration page. After filling in the "Application ID", click the button "Add application" at the bottom of the page to generate the corresponding application, as shown in the figure below.

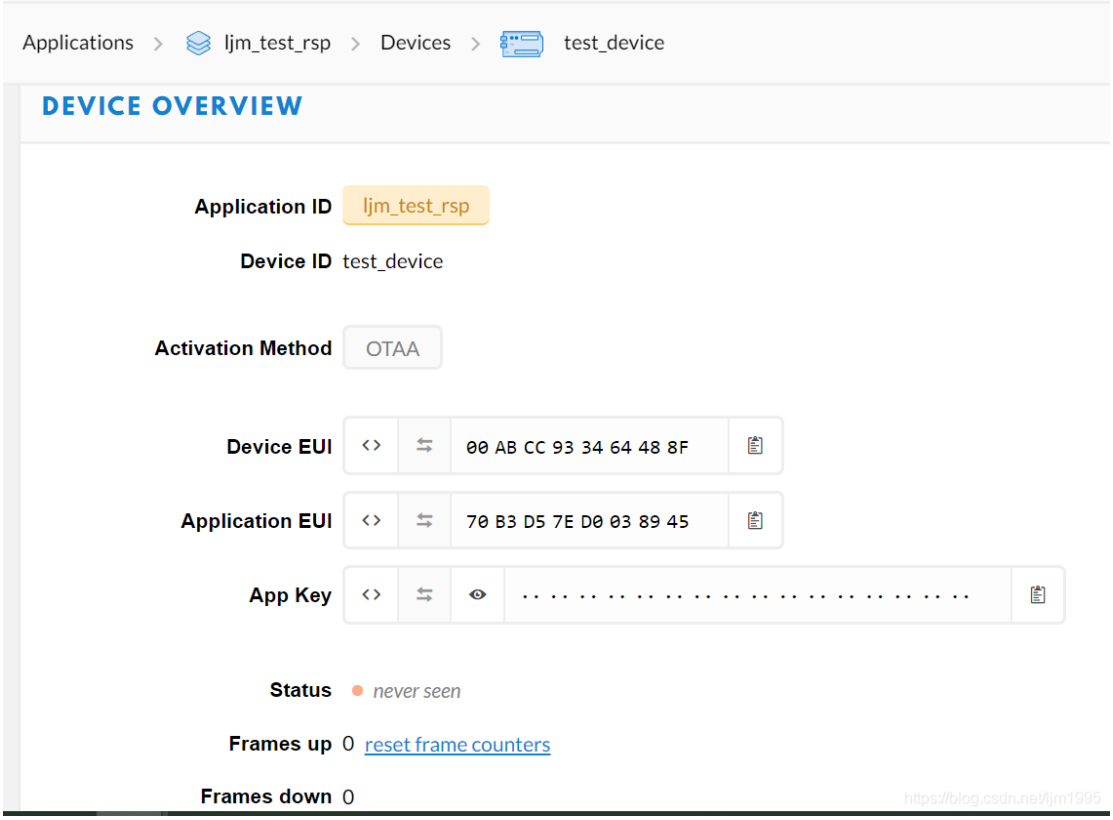


2.4 Register device

Register the device under the corresponding application and click "register device".



After entering the "Application ID", click "Register" to generate the corresponding device.



Modify the code of the corresponding part of the gateway code main.cpp and recompile and run.
Modify the center frequency:
uint32_t freq = 868100000;

Change to

uint32_t freq = 433175000;

Modify the server address: Modify according to the TTN server you choose. Reference link: TTN server list. The website gives the domain name, which is converted to the IP address according to the tool. Domain name/IP query tool.
The server I use is router.eu.thethings.network, and the corresponding IP address is: 52.169.76.203
#define SERVER1 "54.72.145.119"

Change to
#define SERVER1 "52.169.76.203"

3. Client operation

3.1 Download code

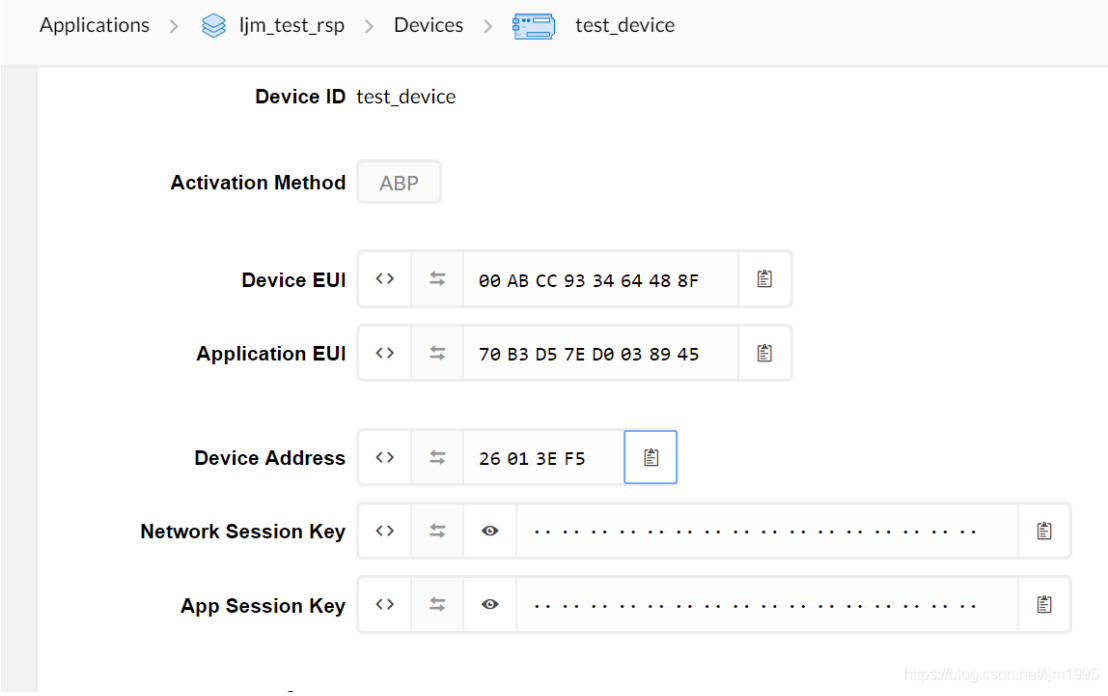
Code link: https://github.com/dragino/Lora/tree/master/Lora%20Shield/Examples/lora_shield_ttn

3.2 Modify code

3.2.1 Client code modification

Modify NWKSKEY, APPSKEY, DEVADDR and other parameters according to the device information on TTN.

```
// LoRaWAN NwkSKey, network session key
// This is the default Semtech key, which is used by the prototype TTN
// network initially.
static const PROGMEM u1_t NWKSKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, (

// LoRaWAN AppSKey, application session key
// This is the default Semtech key, which is used by the prototype TTN
// network initially.
static const u1_t PROGMEM APPSKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, (

// LoRaWAN end-device address (DevAddr)
// See http://thethingsnetwork.org/wiki/AddressSpace
static const u4_t DEVADDR = 0x09984507 ; // <-- Change this address for every node!
```

2. When the default operating frequency of the client is 433MHz, we need to modify the LMIC library file to ensure communication between the two parties.

① Add 433MHz related codes to the code. For details, refer to how to add other frequency bands, such as EU433.

② Force the center frequency of the transmission to be converted to 433MHz in lmic.c.

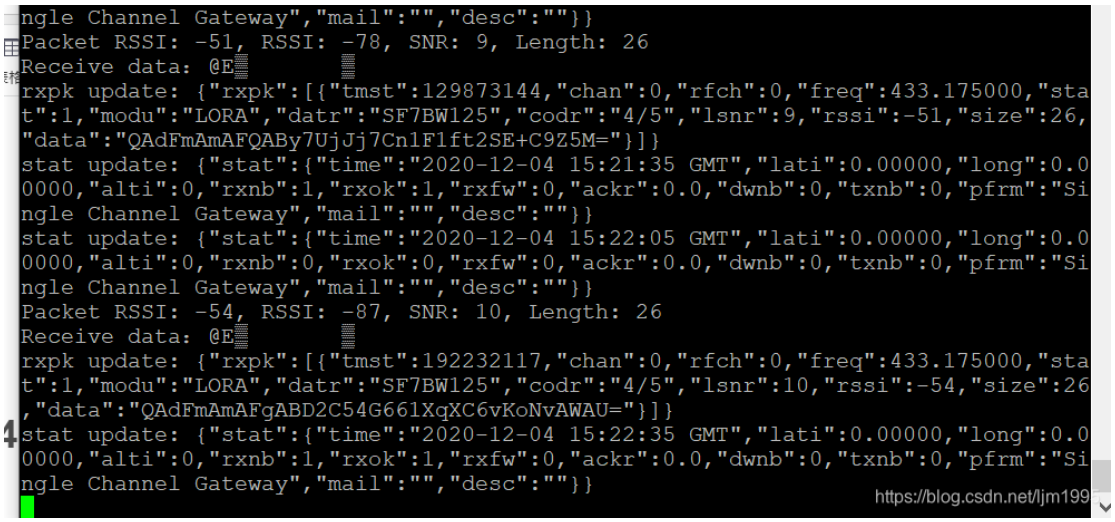
```
bit_t LMIC_setupChannel (u1_t chidx, u4_t freq, u2_t drmap, s1_t band) {
...
LMIC.channelFreq [chidx] = freq;
}
```

Change to

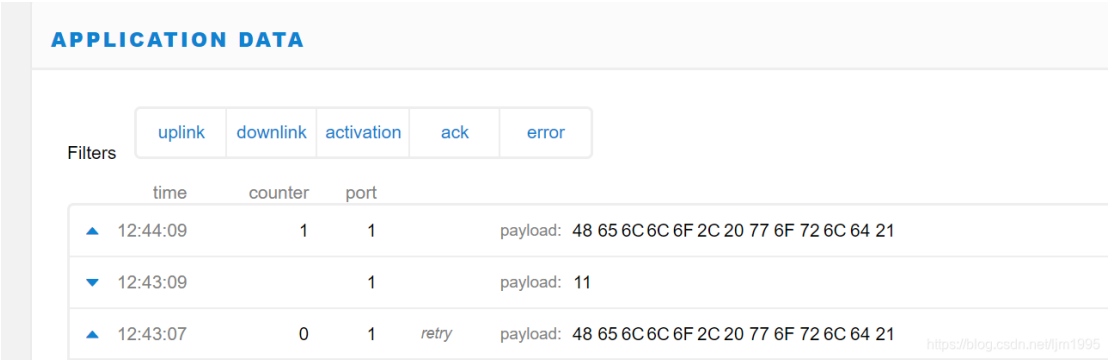
```
bit_t LMIC_setupChannel (u1_t chidx, u4_t freq, u2_t drmap, s1_t band) {
...
freq = 433175000;//This frequency should be the same as the frequency monitored by the gateway
LMIC.channelFreq [chidx] = freq;
}
```

4. Observe the experimental results

The packet receiving phenomenon observed on the built gateway:



The packet receiving phenomenon observed on TTN:



FCC Statement

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: 1) this device may not cause harmful interference, and 2) this device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

--Reorient or relocate the receiving antenna.

--Increase the separation between the equipment and receiver.

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

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

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

FCC Radiation Exposure Statement

This device complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

This device must operate with a minimum distance of 20 cm between the radiator and user body.