



LoRaWAN Configuration Guide

by Cuongkim

Thông tin Tài liệu

Tài liệu này được tạo để hỗ trợ thiết lập chức năng uplink data message từng bước một. Hướng dẫn cung cấp các bước hướng dẫn chi tiết về cách kết nối cổng sử dụng LoRa LPS8 LoRaWAN Gateway đến LoRaWAN Network Servers (LNS) là The Things Network với STM32 :3

1. Phần cứng sử dụng
 - a. Ví điều khiển NUCLEO-F303RE STM32 Nucleo-64
 - b. module RFM95W 915MHz LoRa + Breakout
 - c. Anten Lora 868/915Mhz 5dBi SMA Đực + Cái

- d. Dragino LPS8 LoRaWAN Gateway. Để biết thêm thông tin, vui lòng tham khảo user manual của [Dragino](#)
- 2. Hướng dẫn thiết lập Gateway

Cách đặt lại LPS8 Gateway về cài đặt mặc định ban đầu:



Nhấn giữ nút Toggle trong 5-10s. Khi đèn trạng thái LED hệ thống chuyển sang màu đỏ, có thể thả nút Toggle ra.

Thiết bị LPS8 có tổng cộng bốn đèn LED, bao gồm:

Đèn LED nguồn : Đèn LED **màu đỏ** này sẽ sáng khi thiết bị được cấp nguồn đúng cách.

Đèn LED LoRa : Đèn LED RGB này sẽ nhấp nháy **màu xanh lá cây** khi module LoRaWAN bắt đầu hoặc truyền một gói tin.

Đèn LED Hệ thống  : Đèn LED RGB này sẽ hiển thị các màu khác nhau tùy theo trạng thái:

Màu xanh lam sáng: Thiết bị đang hoạt động với kết nối máy chủ LoRaWAN.

Màu xanh lam nhấp nháy:

Thiết bị có kết nối internet nhưng không có kết nối LoRaWAN.

Thiết bị đang trong giai đoạn khởi động, trong giai đoạn này, nó sẽ nhấp nháy màu xanh lam trong vài giây và sau đó cùng với **màu đỏ sáng** và nhấp nháy màu xanh lam.

Màu đỏ sáng: Thiết bị không có kết nối internet.

Đèn LED Ethernet  : Đèn LED này hiển thị trạng thái kết nối giao diện Ethernet.



2.1 Kết nối Gateway vào Internet:

Cách 1:



Sau khi reset Gateway, thiết bị sẽ tự động phát ra Wifi **dragino-168xxx**. Mật khẩu đăng nhập là **dragino+dragino**. Sau khi kết nối PC vào WiFi này, PC sẽ nhận được địa chỉ IP là 10.130.1.xxx và LPS8 có địa chỉ IP mặc định là 10.130.1.1

Cách 2:

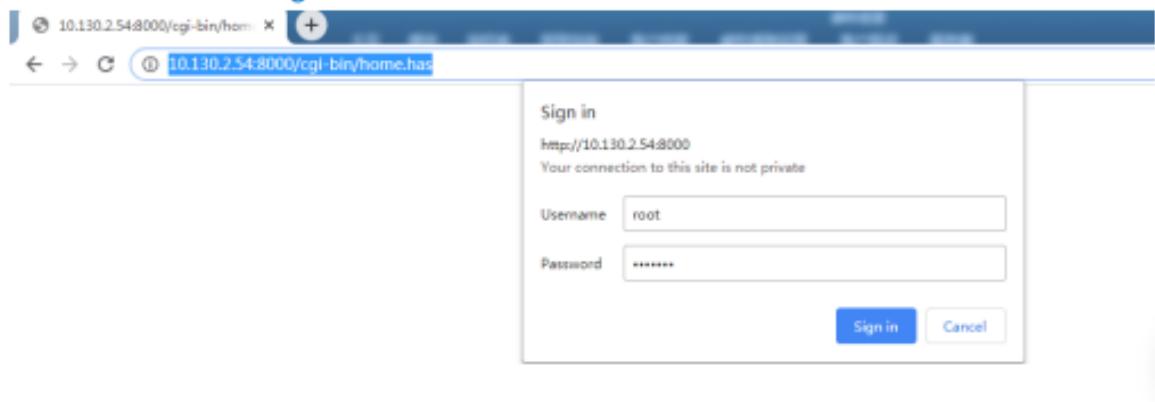
Sau khi reset, kết nối Gateway và dây mạng LAN.

2.2 Truy cập trang web cấu hình thiết bị

Mở trình duyệt ở trên máy tính và gõ địa chỉ IP LPS8:

<http://10.130.1.1/home.html>

Giao diện sẽ xuất hiện như bên dưới:

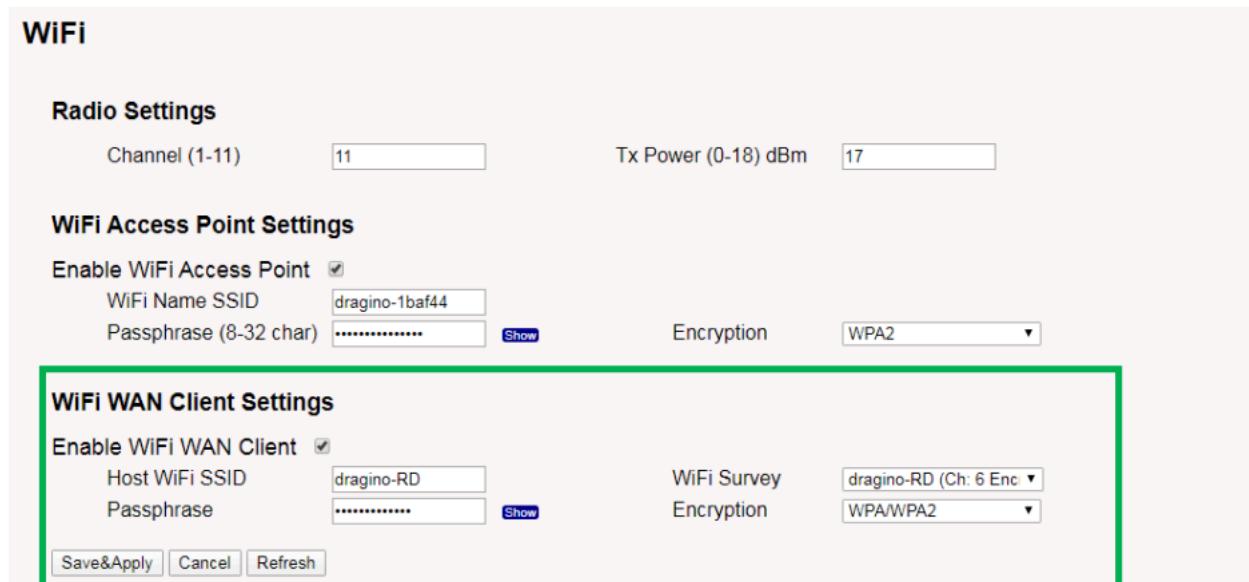


Tài khoản mặc định cho trang web là:

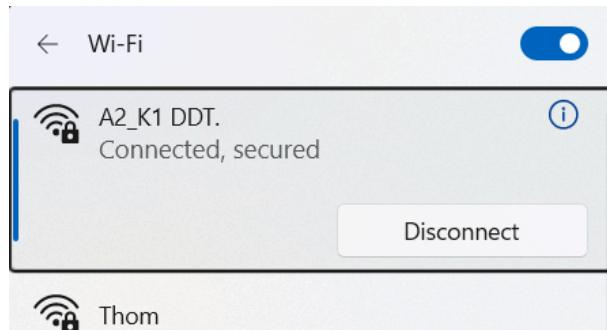
- User: root
- Password: dragino

2.3 Truy cập Internet với tư cách Wifi Client (bước này có thể bỏ qua nếu thiết bị gateway được kết nối bằng mạng LAN) nhưng bắt buộc để có thể cấu hình gateway sau này.

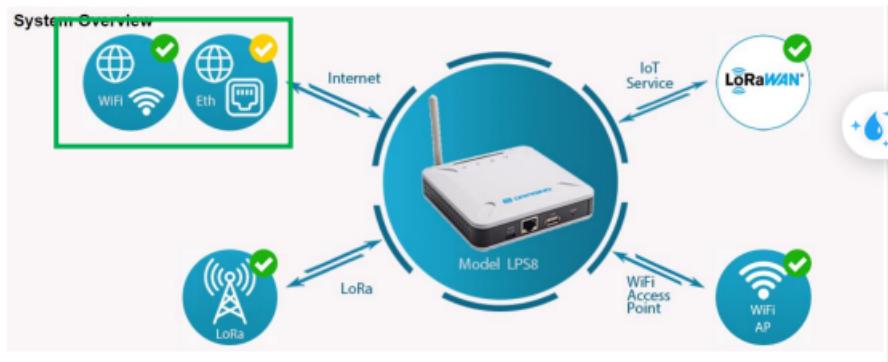
Ở chế độ này, LPS8 hoạt động như một WiFi client và nhận DHCP từ một router nguồn từ xa thông qua kết nối WiFi. Các thiết lập cho Chế độ WiFi client nằm trong taskbar System → WiFi → WiFi WAN client Settings.



Ở mục WiFi Survey, chọn tên wifi hiện tại cần kết nối. Chú ý encryption để điền vào ô bên dưới. Sau đó nhập tên wifi cần kết nối vào ô Host WiFi SSID và mật khẩu vào ô Passphrase. Sau đó nhấn Save&Apply.



Chú ý: Ở mục wifi hiện tại của máy tính, nếu sau kí nhấn Save&Apply, PC mất kết nối với mạng dragino-168 nhưng có hiện tên wifi ở cài đặt trên, thì kết nối vào. Chú ý đợi trong 2-3 phút, nếu wifi trên không xuất hiện, cần nhấn reset và thực hiện lại các thao tác ở mục 2.1. Ở đây, cần kết nối vào mạng dragino-168xx để có thể tiếp tục vào trang web 10.130.1.1 để tiếp tục cấu hình.



2.3 Kiểm tra kết nối Internet

Trong trang chủ, chúng ta có thể kiểm tra kết nối Internet.

- GREEN Tick Giao diện này có kết nối Internet.
- Yellow Tick Giao diện này có địa chỉ IP nhưng không sử dụng cho kết nối Internet.
- RED Cross Giao diện này không kết nối.

3. Hướng dẫn thiết lập The Thing Network

Cần tạo tài khoản và đăng nhập vào tài khoản

Please log in to continue

Log in to manage your The Things Industries subscriptions and payment information.

Email

Password

Log in

Don't have an account? [Sign up](#)

[I forgot my password](#)

Subscriptions

The Things Stack Cloud Discovery

Started 7th Nov 2023

AutonomousRobot (autonomousrobot)



Add a subscription



Chọn khu vực là North America 1

Existing clusters

- Europe 1**
eu1 – Dublin, Ireland 
- Europe 2**
eu2 – London, UK 
- North America 1**
nam1 – California, USA 
- Australia 1 (recommended)**
au1 – Sydney, Australia 

3.1 Cấu hình gateway và The Thing Network

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.

Learn more in our guide on [Adding Gateways](#).

Gateway EUI ⓘ

No gateway EUI

Reset

Gateway ID ⓘ *

my-new-gateway

Gateway name ⓘ

My new gateway

Frequency plan ⓘ *

Select a frequency plan...

+ Add frequency plan

Note: most gateways use a single frequency plan. Some 16 and 64 channel gateways however allow setting multiple within the same band.

Require authenticated connection ⓘ

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

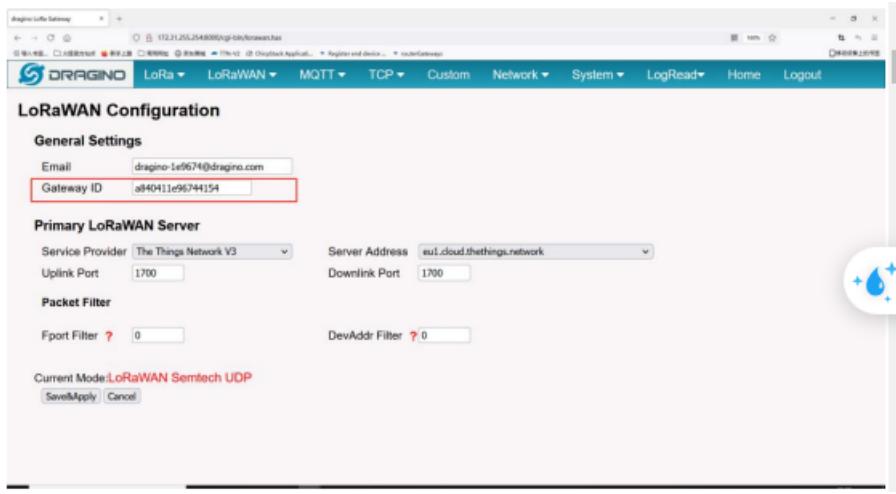
Select which information can be seen by other network participants, including [Packet Broker](#)

Share status within network ⓘ

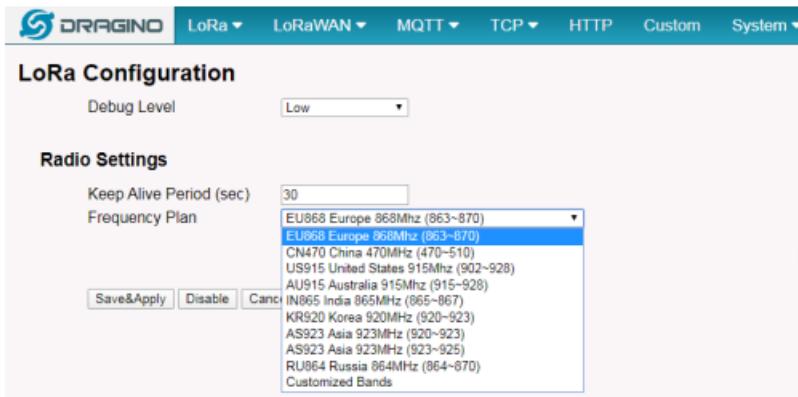
Share location within network ⓘ

Register gateway

Trước tiên, đăng ký gateway với No gateway EUI, tiến hành nhập các mục ở trên và nhấn đăng ký. Sau khi đã đăng ký gateway, chú ý Gateway EUI và Gateway Server address để tiến hành kết nối Gateway và server TTN.



Quay lại trang web cấu hình dragino, nhập Gateway ID đã được tạo ở trên vào. Ở ô Service Provider, chọn custom. Ở ô Server Address, nhập Gatewat Server address đã được đăng ký lúc nãy ở TTN. Sau đó nhấn save&apply. Chú ý đèn lora chuyển sang màu xanh lá và Status gateway ở TTN hiện connected.



Sau đó cấu hình tần số đúng với thiết bị end device

3.2 Cấu hình end device.

Owned applications		All (Admin)	Deleted (Admin)	<input type="text"/> Search	+ Create application
ID	Name			End devices	Created at
autonomousrobot	autonomousrobot			6	Nov 7, 2023

Tạo và đăng ký application, sau đó truy cập vào application đã tạo

Applications > autonomousrobot > End devices				
End devices (6)		Search	Import end devices	Register end device
ID	Name	DevEUI	JoinEUI	Last activity
eui-70b3d57ed8002243		70 B3 D5 7E D8 00...	none	Never
lothenho		70 B3 D5 7E D8 00...	none	Nov 17, 2023
fuckyoudat		70 B3 D5 7E D8 00...	none	15 days ago
eui-70b3d57ed8002106		70 B3 D5 7E D8 00...	none	Nov 15, 2023
eui-70b3d57ed8002105		70 B3 D5 7E D8 00...	none	Nov 15, 2023
eui-70b3d57ed8002104		70 B3 D5 7E D8 00...	none	Nov 15, 2023

End device type

Input method ?

- Select the end device in the LoRaWAN Device Repository
- Enter end device specifics manually

Frequency plan ? *

Australia 915-928 MHz, FSB 2 (used by TTN)

LoRaWAN version ? *

LoRaWAN Specification 1.0.1

Regional Parameters version ? *

TS001 Technical Specification 1.0.1

Tiến hành đăng ký end device với **input method** là enter end device specifics manually. Chọn cấu hình phù hợp với LoRaWAN version là 1.0.3 (bắt buộc với thư viện LMIC này). Ở chế độ nâng cao, chọn Activation mode là ABP và class A:

- OTAA (Over-the-Air Activation)

OTAA là cơ chế kích hoạt thiết bị LoRaWAN được sử dụng rộng rãi nhất. Trong OTAA, thiết bị sẽ gửi yêu cầu tham gia (join request) đến mạng LoRaWAN, bao gồm AppEUI, DevEUI và JoinEUI của thiết bị. Mạng LoRaWAN sau đó sẽ xác thực thiết bị và gửi lại DevAddr và session keys cho thiết bị. Thiết bị sẽ sử dụng DevAddr và session keys này để gửi và nhận dữ liệu với mạng LoRaWAN.

- ABP (Activation by Personalization)

ABP là cơ chế kích hoạt thiết bị LoRaWAN đơn giản hơn OTAA. Trong ABP, nhà phát triển thiết bị sẽ cấu hình sẵn DevAddr và session keys cho thiết bị. Khi khởi động, thiết bị sẽ sử dụng DevAddr và session keys này để tham gia mạng LoRaWAN. ABP thường được sử dụng cho các thiết bị LoRaWAN được triển khai trong môi trường riêng tư, nơi nhà phát triển có quyền kiểm soát mạng LoRaWAN.

- Define multicast group (ABP & Multicast)

Multicast là tính năng cho phép gửi một tin nhắn LoRaWAN đến nhiều thiết bị LoRaWAN cùng lúc. Để sử dụng multicast, nhà phát triển thiết bị cần định nghĩa một nhóm multicast (multicast group). Mỗi nhóm multicast sẽ có một DevAddr riêng. Các thiết bị LoRaWAN muốn nhận tin nhắn multicast cần được cấu hình sẵn để tham gia vào nhóm multicast đó.

Multicast thường được sử dụng cho các ứng dụng như điều khiển chiếu sáng, điều khiển nhiệt độ hoặc gửi thông báo đến nhiều thiết bị LoRaWAN cùng lúc.

Tiến hành generate các key bên dưới và bấm đăng ký:

Provisioning information

DevEUI ⓘ

70 B3 D5 7E D8 00 1F 5A

 Generate

3/50 used

Device address ⓘ *

27 FD 0E 49

 Generate

AppSKey ⓘ *

F7 65 F3 1A 99 04 C7 0E 83 8B 7C 6A 64 27 13 1D

 Generate

NwkSKey ⓘ *

0C 54 D1 85 09 4C 04 4C 66 19 30 5B E7 F4 B9 D8

 Generate

End device ID ⓘ *

eui-cuongtuandatvinh

This value is automatically prefilled using the DevEUI

After registration

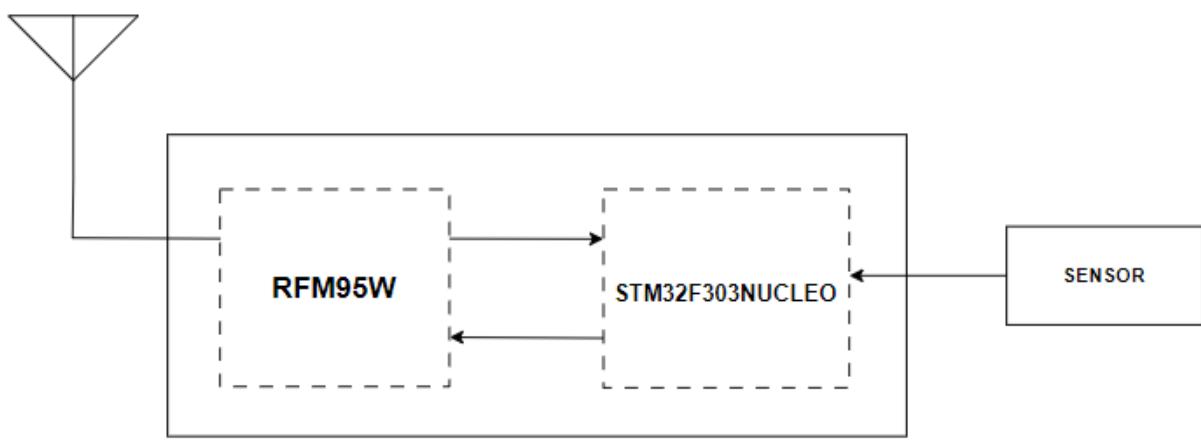
View registered end device

Register another end device of this type

Register end device

4. Thiết bị sử dụng trong hệ thống

4.1 End node



a. Module LoRa RFM95W:

RFM95W là một module thu phát dữ liệu không dây sử dụng sóng LoRa, được thiết kế bởi hãng Hope RF. Module này có độ nhạy thu lên đến -148dBm, đi kèm với bộ khuếch đại công suất tích hợp +20dBm. Module hoạt động ở điện áp nguồn từ 1.8V đến 3.7V, với dòng tiêu thụ cao nhất khi truyền là 120mA.

Để giao tiếp với module RFM95W, chúng ta sử dụng giao thức SPI. Các chân giao tiếp SPI bao gồm MOSI, MISO, SCK. Ngoài ra, module còn có các chân cấu hình khác gồm NSS, RESET, DIO0.

Dưới đây là một số thông số kỹ thuật chi tiết của module RFM95:

Dải tần hoạt động: 433MHz, 868MHz, 915MHz

Độ nhạy thu: -148dBm

Công suất truyền: +20dBm

Tiêu thụ điện năng: 10.3mA (thu), 120mA (truyền)

Giao thức giao tiếp: SPI

Module RFM95W được sử dụng trong nhiều ứng dụng IoT, chẳng hạn như:

Giám sát môi trường

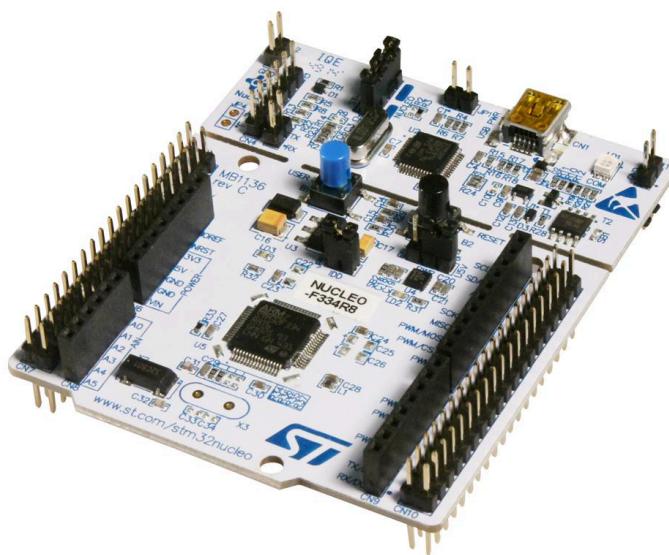
Theo dõi tài sản

Kết nối thiết bị di động

Truyền dữ liệu trong các khu vực có nhiễu cao

b. Ví điều khiển STM32F303 Nucleo RE

Việc sử dụng các chip lõi ARM đã trở nên phổ biến và là xu hướng trong thiết kế mạch. Những ưu điểm như chi phí thấp, tiết kiệm năng lượng, tốc độ xử lý nhanh (36-72MHz), hỗ trợ nhiều giao diện ngoại vi (GPIO, I2C, SPI, ADC, USB, Ethernet, CAN, v.v...), và thư viện lập trình liên tục được cập nhật đã làm cho dòng ARM STM32 chiếm lĩnh thị trường và thay thế các dòng vi điều khiển cũ như 8051, PIC, v.v.



Một số cấu hình chi tiết của STM32F303RE bao gồm:

- Lõi: ARM Cortex-M4 CPU 32-bit với 72 MHz FPU, nhân chu kỳ đơn và phép chia HW, 90 DMIPS, lệnh DSP và MPU.
 - Điện áp hoạt động, Clock: Dải điện áp từ 2.0 V đến 3.6 V, sử dụng thạch anh ngoài từ 4 MHz đến 32 MHz.
 - Bộ nhớ: Flash lên đến 512 Kbyte, 64 Kbyte SRAM.

- Đặt lại và quản lý nguồn cung cấp: Đặt lại nguồn bật/tắt nguồn (POR/PDR), chế độ năng lượng thấp.
- Bộ điều khiển DMA 12 kênh, hỗ trợ DMA cho ADC, I2C, SPI, UART.
- Bốn ADC (lên đến 40 kênh) với độ phân giải có thể lựa chọn 12/10/8/6 bit, phạm vi chuyển đổi 0 đến 3.6 V.
- Hai kênh DAC 12-bit với nguồn cung cấp tương tự từ 2.4 đến 3.6 V.
- Timer: Một bộ định thời 32 bit và hai bộ định thời 16 bit hỗ trợ các IC/OC/PWM, ba bộ hẹn giờ điều khiển nâng cao 16 bit 6 kênh và một bộ hẹn giờ 16 bit với hai IC/OC.
- Các kênh giao tiếp bao gồm: 1 bộ CAN interface, ba chế độ I2C Fast công, lên đến năm USART/UART, lên đến bốn SPI với giao diện I2S bán song công/toàn phần, USB 2.0 full-speed interface và kiểm tra lỗi CRC và 96-bit ID.

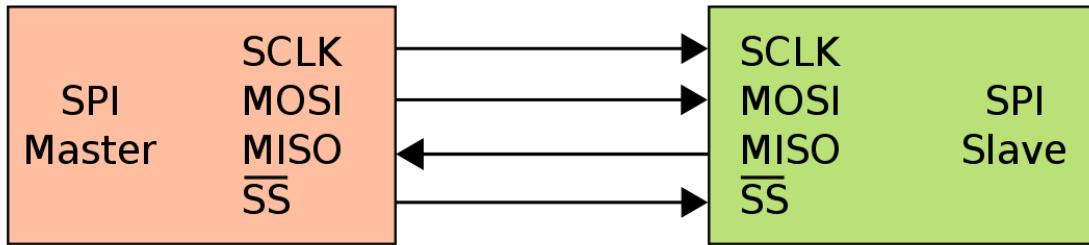
4.2 Giao thức SPI

Vì điều khiển thực hiện giao tiếp với LoRa Module thông qua giao thức SPI. Giao thức SPI, hay còn gọi là giao tiếp ngoại vi nối tiếp, đã được Motorola phát triển vào những năm 1980 như một tiêu chuẩn giao diện có chi phí thấp và đáng tin cậy giữa vi điều khiển (ban đầu là vi điều khiển của Motorola) và các vi mạch ngoại vi của nó.

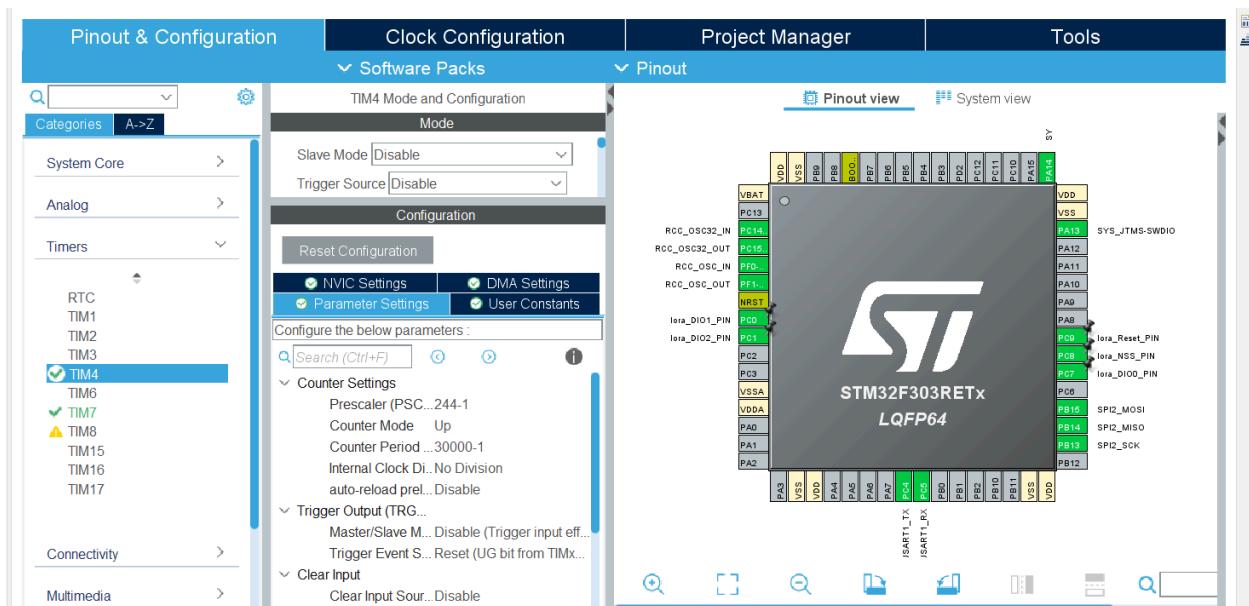
Do sự đơn giản, linh hoạt và dễ sử dụng của giao diện này, SPI đã trở thành một tiêu chuẩn và các nhà sản xuất bán dẫn khác cũng bắt đầu tích hợp giao thức này vào các chip của họ. Trong giao thức SPI, các thiết bị được kết nối theo một mối quan hệ Master – Slave trong một giao diện đa điểm.

Trong kiểu giao diện này, một thiết bị được xác định là Master của bus (thường là một vi điều khiển), trong khi tất cả các thiết bị khác, bao gồm cả IC ngoại vi và

thậm chí là các vi điều khiển khác, đều được xác định là Slave. Giao thức SPI sử dụng bốn dây chính bao gồm dây SCK clock đồng bộ, dây MISO – Master to Slave, dây MOSI – Slave to Master, dây SS – Slave Select.

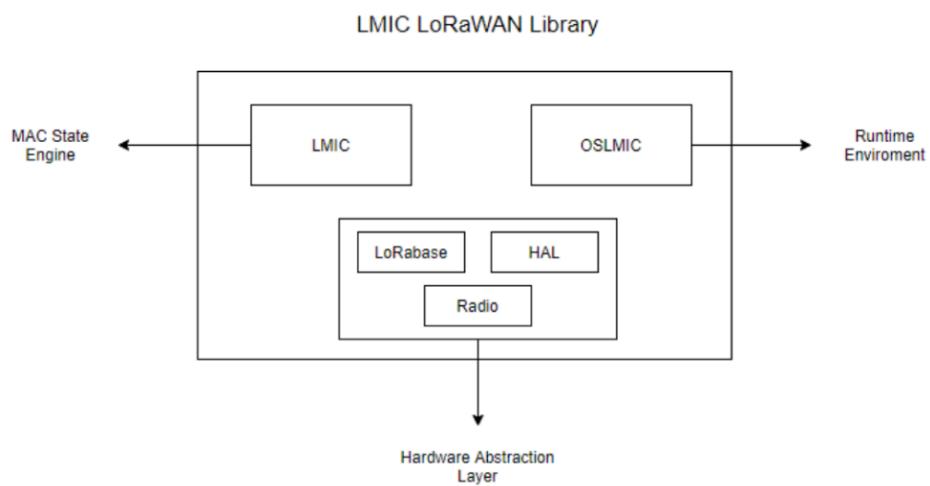


5. Cấu hình trên STM32



Cấu hình pinout & configuration. Chú ý không kết nối lora_reset_PIN với module. Khi kết nối, module chỉ truyền dữ liệu 1 lần, không thể reset.

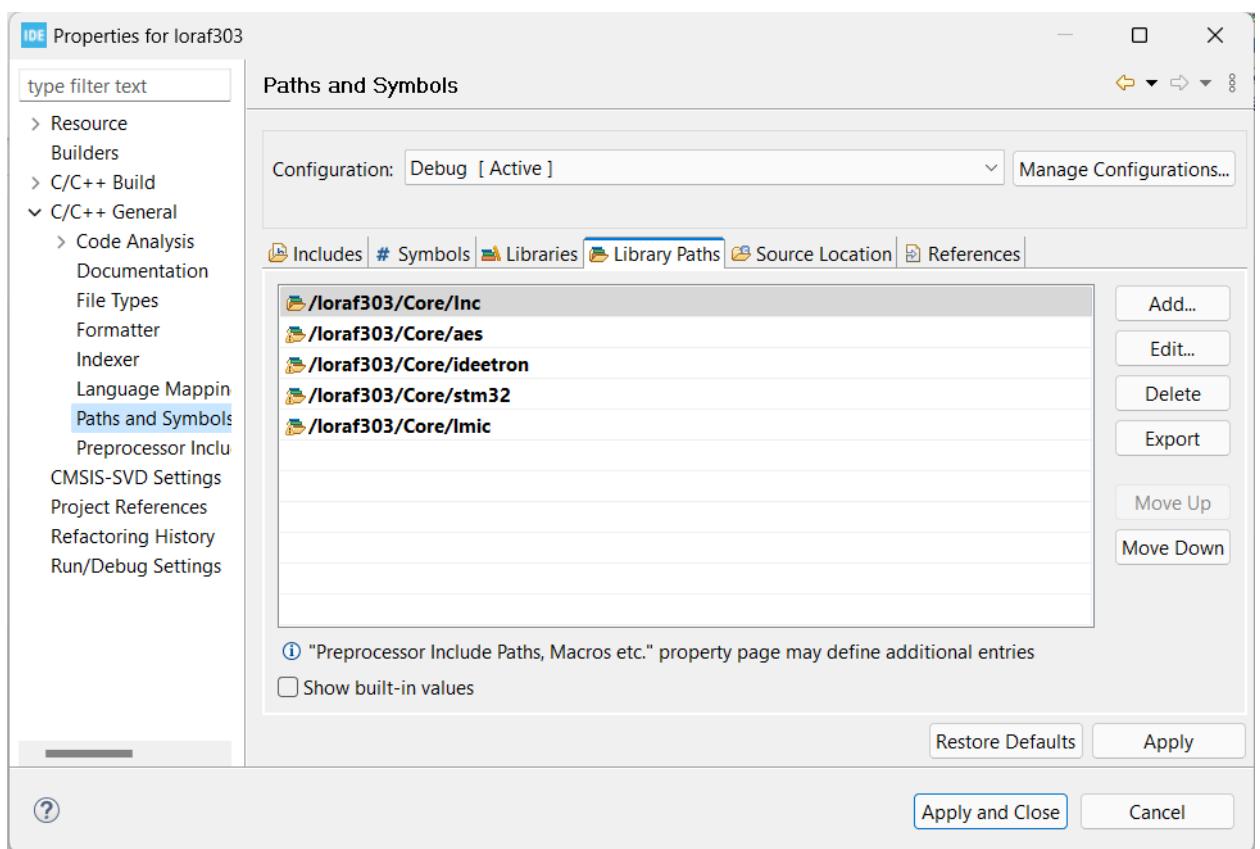
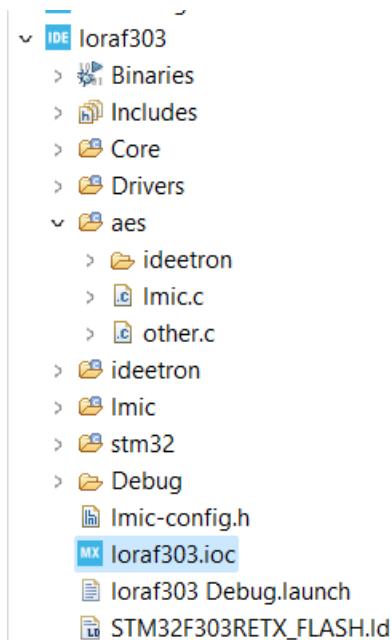
5.1 Cấu trúc thư viện LORAWAN LMIC



- MAC State Engine: Media Access Control bao gồm các chức năng truy cập dữ liệu, được sắp xếp để chuyển đổi dữ liệu từ Application Layer sang MAC Layer.
- Runtime Environment: Lớp môi trường thực thi bao gồm các chức năng quản lý thứ tự, thời gian của chương trình, và các chức năng giao tiếp giữa MCU và LoRa Module.
- Hardware Abstraction Layer: Lớp trừu tượng hóa phần cứng thực hiện việc cấu hình tần số truyền, chuyển đổi dữ liệu theo định dạng đúng với Physical Layer và truyền tải sang LoRa Module.

5.2 Cấu hình thư viện LMIC

Tiến hành add các thư mục thư viện LMIC.



5.3 Cấu hình các file ở thư viện LMIC

a. Lorabase.h

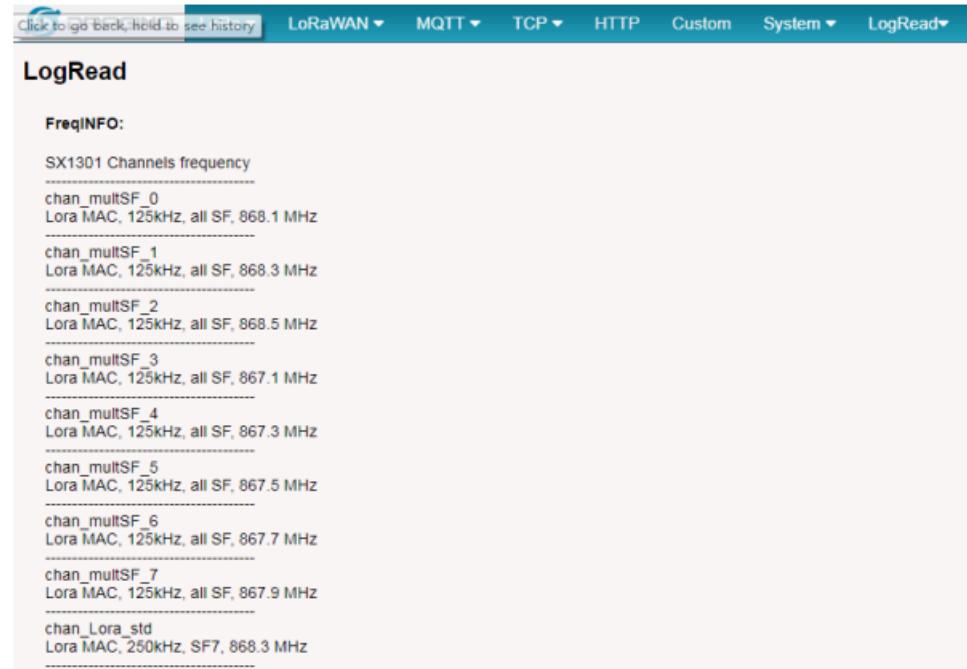
```
1
2 // Default frequency plan for US 915MHz
3 enum { US915_125kHz_UPFBASE = 902300000,
4         US915_125kHz_UPFSTEP = 200000,
5         US915_500kHz_UPFBASE = 903000000,
6         US915_500kHz_UPFSTEP = 1600000,
7         US915_500kHz_DNFBASE = 923300000,
8         US915_500kHz_DNFSTEP = 600000
9 };
```

Nhập tần số đã cấu hình vào và sửa cấu hình chân ở 1 file nào đó trong thư viện :3. Sau khi cấu hình xong, nhập đúng tần số vào hàm khởi tạo

```
#if defined(CFG_eu868)
// Set up the channels used by the Things Network, which corresponds
// to the defaults of most gateways. Without this, only three base
// channels from the LoRAWAN specification are used, which certainly
// works, so it is good for debugging, but can overload those
// frequencies, so be sure to configure the full frequency range of
// your network here (unless your network autoconfigures them).
// Setting up channels should happen after LMIC_setSession, as that
// configures the minimal channel set.
// NA-US channels 0-71 are configured automatically
LMIC_setupChannel(0, 903900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(1, 904100000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI); // g-band
LMIC_setupChannel(2, 904300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(3, 904500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(4, 904700000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(5, 904900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(6, 905100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(7, 905300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI); // g-band
LMIC_setupChannel(8, 904600000, DR_RANGE_MAP(DR_FSK, DR_FSK), BAND_MILLI); // g2-band
// TTN defines an additional channel at 869.525Mhz using SF9 for class B
// devices' ping slots. LMIC does not have an easy way to define set this
// frequency and support for class B is spotty and untested, so this
// frequency is not configured here.
#elif defined(CFG_us915)
// ... more channel definitions for US 915MHz ...
```

Tần số này ở trang web cấu hình dragino, mục LogRead

In logread page, user can check the frequency actually used.



The screenshot shows the 'LogRead' page with the following content:

FreqINFO:

SX1301 Channels frequency

chan_multSF_0
Lora MAC, 125kHz, all SF, 868.1 MHz

chan_multSF_1
Lora MAC, 125kHz, all SF, 868.3 MHz

chan_multSF_2
Lora MAC, 125kHz, all SF, 868.5 MHz

chan_multSF_3
Lora MAC, 125kHz, all SF, 867.1 MHz

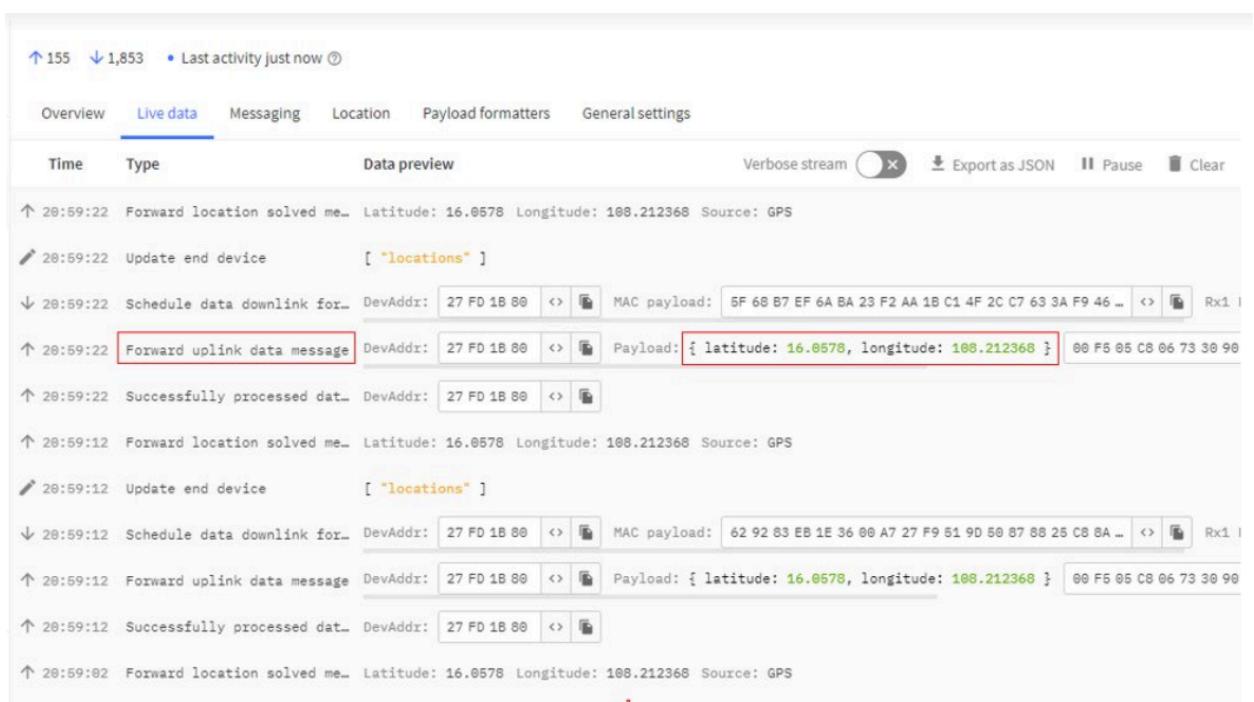
chan_multSF_4
Lora MAC, 125kHz, all SF, 867.3 MHz

chan_multSF_5
Lora MAC, 125kHz, all SF, 867.5 MHz

chan_multSF_6
Lora MAC, 125kHz, all SF, 867.7 MHz

chan_multSF_7
Lora MAC, 125kHz, all SF, 867.9 MHz

chan_Lora_std
Lora MAC, 250kHz, SF7, 868.3 MHz



The screenshot shows the 'Live data' page with the following interface elements:

↑ 155 ↓ 1,853 • Last activity just now ⓘ

Overview Live data Messaging Location Payload formatters General settings

Time Type Data preview Verbose stream Export as JSON ⏸ Pause 🗑 Clear

Time	Type	Data preview	Verbose stream <input checked="" type="checkbox"/>	Export as JSON ⏸	Pause	Clear
↑ 20:59:22	Forward location solved message	Latitude: 16.0578 Longitude: 108.212368 Source: GPS				
📝 20:59:22	Update end device	["locations"]				
↓ 20:59:22	Schedule data downlink for...	DevAddr: 27 FD 1B 80 <>  MAC payload: 6F 68 B7 EF 6A BA 23 F2 AA 1B C1 4F 2C C7 63 3A F9 46 ... <>  Rx1				
↑ 20:59:22	Forward uplink data message	DevAddr: 27 FD 1B 80 <>  Payload: { latitude: 16.0578, longitude: 108.212368 } 00 F5 05 C8 06 73 30 90				
↑ 20:59:22	Successfully processed data...	DevAddr: 27 FD 1B 80 <> 				
↑ 20:59:12	Forward location solved message	Latitude: 16.0578 Longitude: 108.212368 Source: GPS				
📝 20:59:12	Update end device	["locations"]				
↓ 20:59:12	Schedule data downlink for...	DevAddr: 27 FD 1B 80 <>  MAC payload: 62 92 83 EB 1E 36 00 A7 27 F9 51 9D 50 87 88 25 C8 8A ... <>  Rx1				
↑ 20:59:12	Forward uplink data message	DevAddr: 27 FD 1B 80 <>  Payload: { latitude: 16.0578, longitude: 108.212368 } 00 F5 05 C8 06 73 30 90				
↑ 20:59:12	Successfully processed data...	DevAddr: 27 FD 1B 80 <> 				
↑ 20:59:02	Forward location solved message	Latitude: 16.0578 Longitude: 108.212368 Source: GPS				

Sau đó, code các hàm gửi data ở file main và debug. Dữ liệu sẽ được thay đổi ở biến **deadline**. Lúc này dữ liệu đã được uplink liên tục. Ở cấu hình end device, có thể thay đổi số messages được gửi trong quá trình hoạt động của thiết bị.

TẠM HẾT