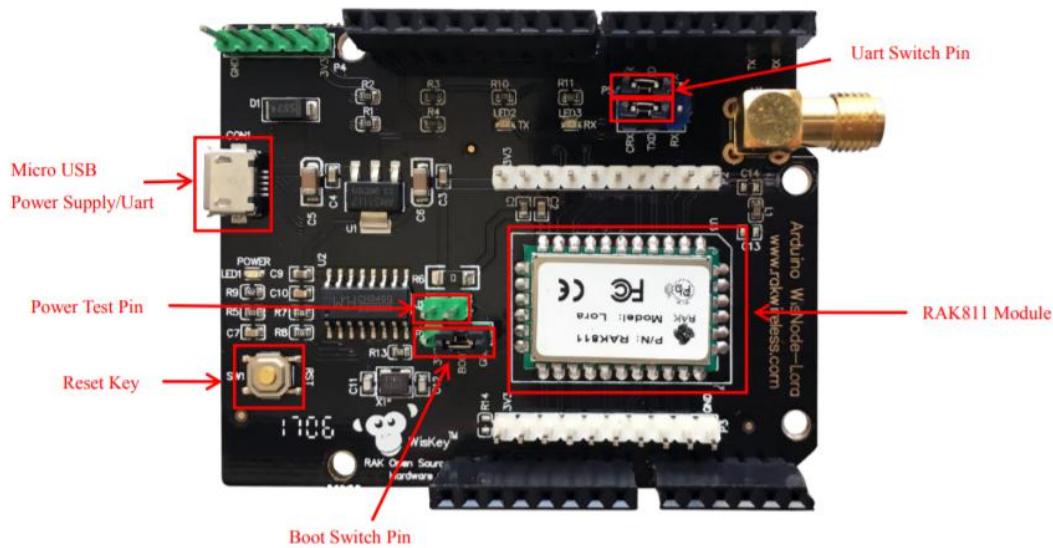


Get Start with RAK811 WisNode-LoRa



1. Where is the latest firmware file?

If you want to get a pre-compiled firmware instead of compiling the source code by yourself, you can find the latest firmware on RAK website after it is released.

<https://www.rakwireless.com/en/download/LoRa/WisNode#Firmware>

2. How to burn the bootloader into RAK811 WisNode?

Usually, you needn't to burn the bootloader because there will be a bootloader in RAK811 WisNode from V3.0.0.0 firmware on, so if the **firmware of your RAK811 WisNode is V3.0.0.0 or a newer one, just jump this section.**

But if you are using an old firmware of RAK811, maybe you need to burn the bootloader as follow:

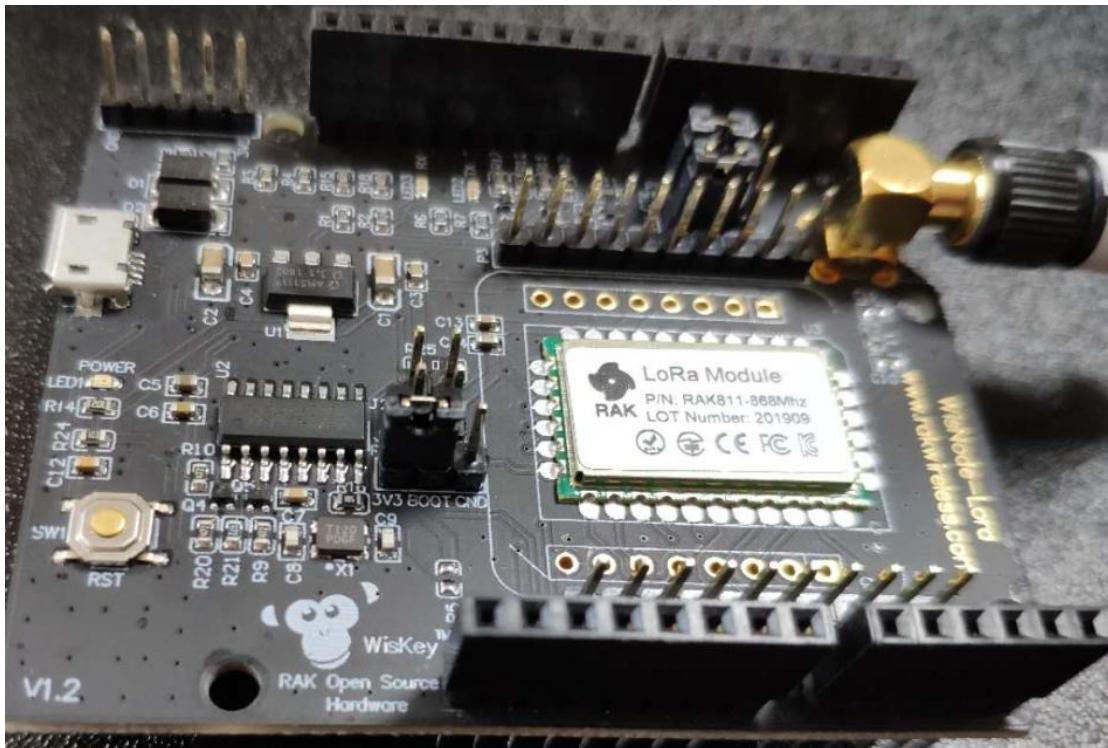
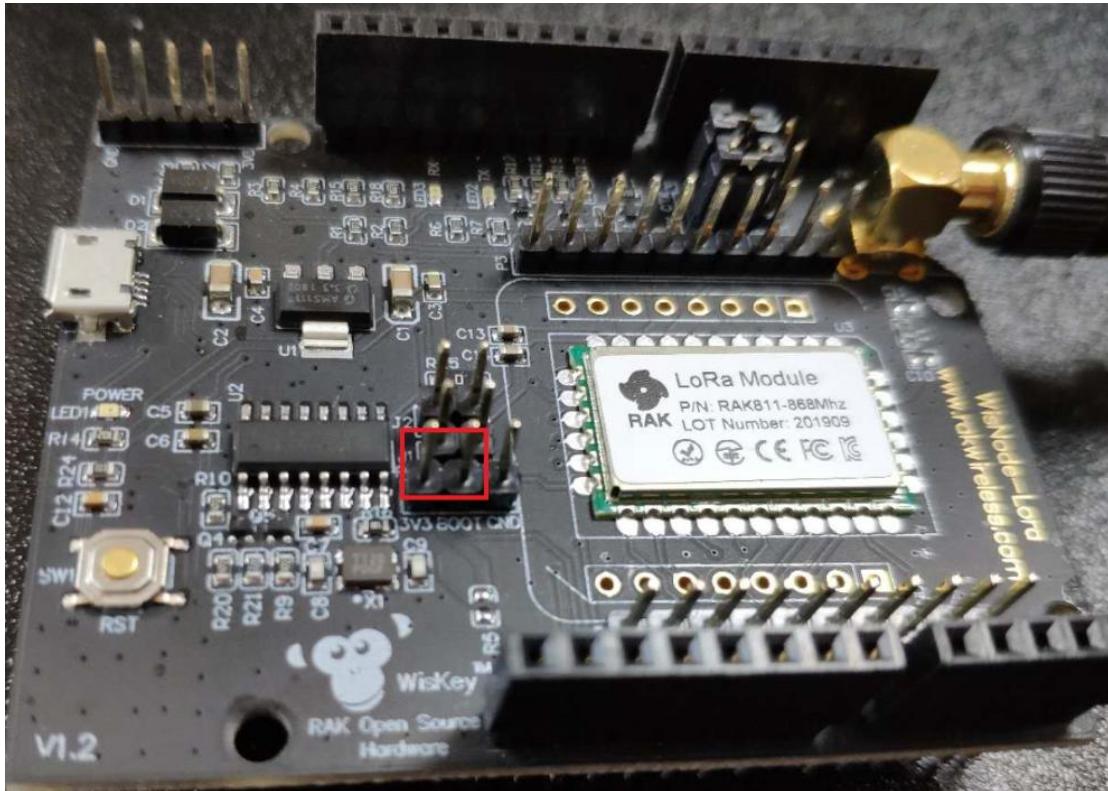
Firstly, please install the “STM32CubeProgrammer” tool on your Windows PC. You can download it from here:

https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stm32cubeprog.html#overview

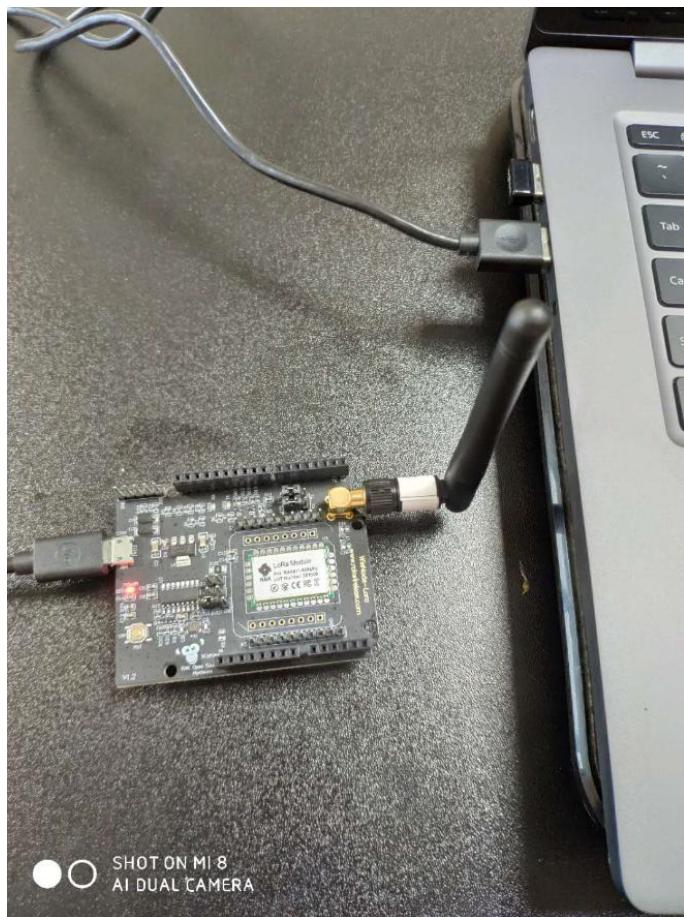
or from RAK website:

<https://downloads.rakwireless.com/en/LoRa/RAK811/Tools/SetupSTM32CubeProgrammer-2.1.0.rar>

Secondly, jump the “BOOT” pin and “3V3” pin for boot mode as following:

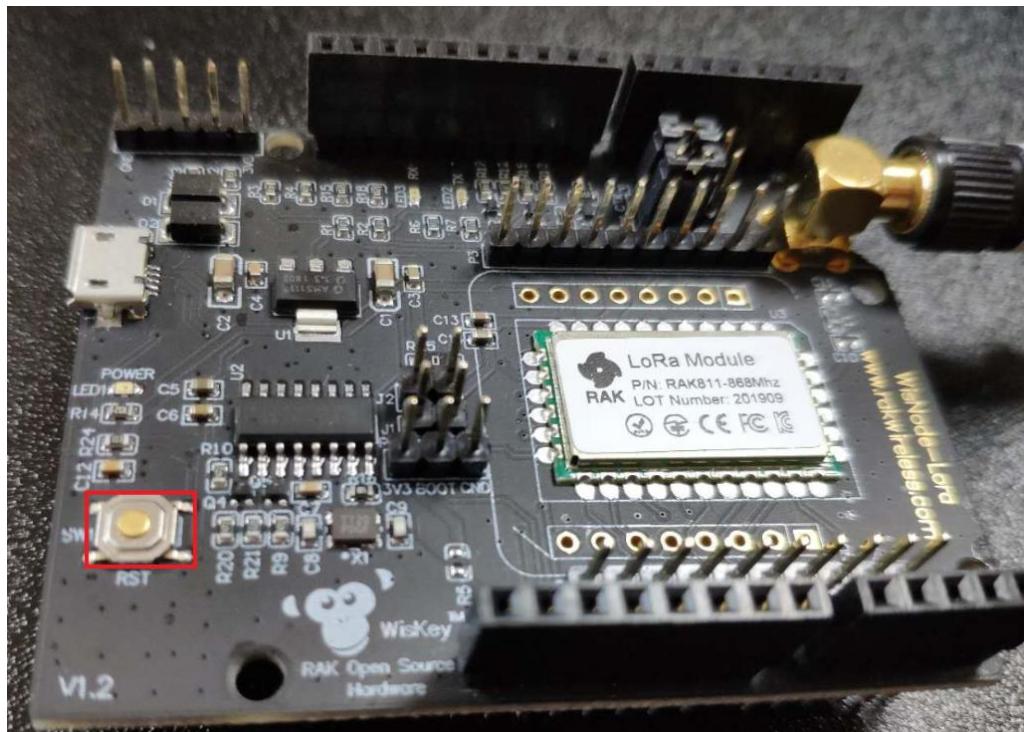


Connect RAK811 WisNode with your PC's USB interface as follow:

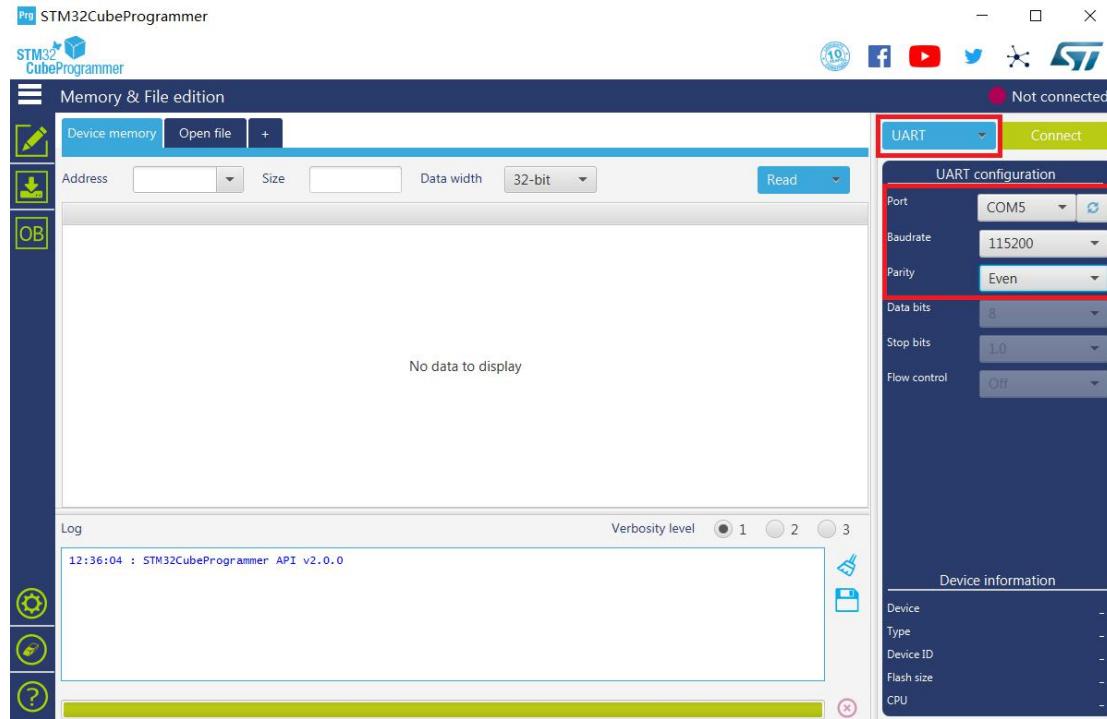


SHOT ON MI 8
AI DUAL CAMERA

Press the RST button on RAK811 WisNode:

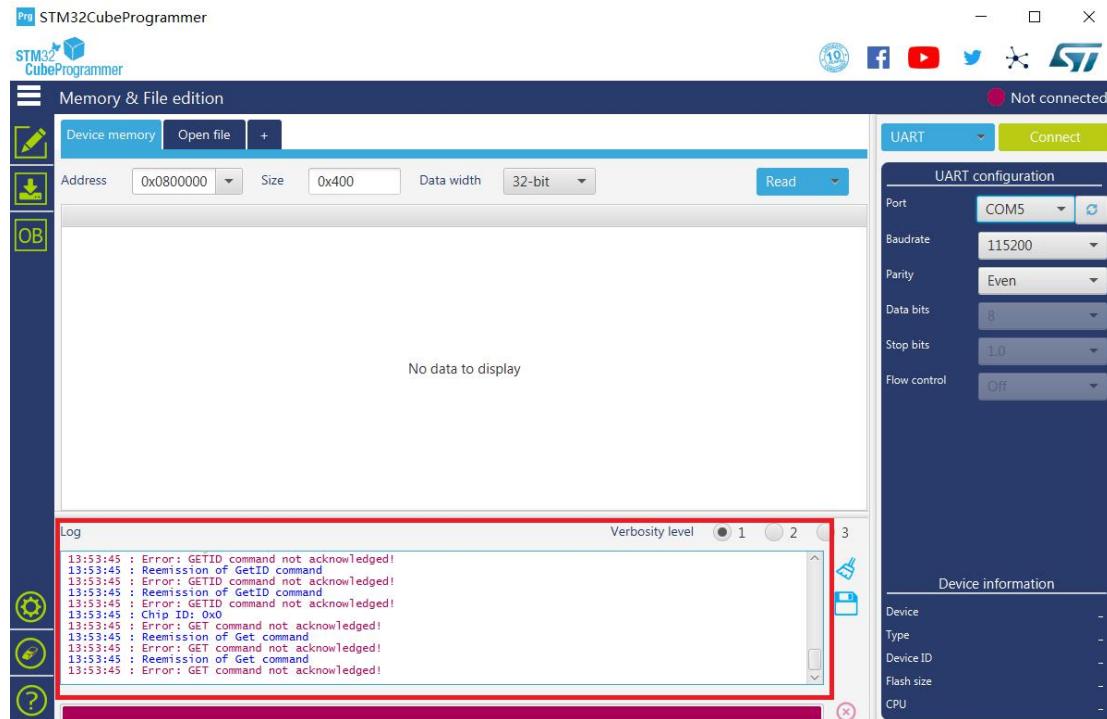


Open the “STM32CubeProgrammer” tool, and select UART type, then configure the Port, Baudrate, and Parity as the following picture shows:

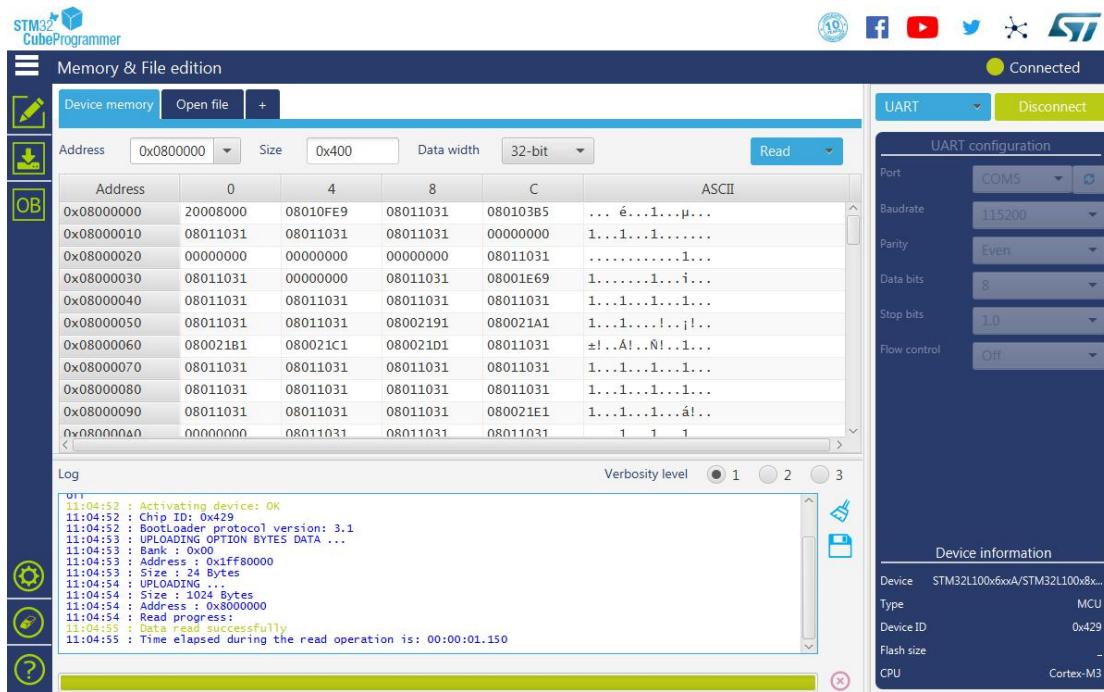


Note: It is COM5 on my computer, and it may be another COM port on your computer.

Then press “Connect” button at the top right corner. If there are some errors in the Log box or it can't connect, please close the STM32CubeProgrammer and reset RAK811 WisNode, then open the STM32CubeProgrammer and connect again.

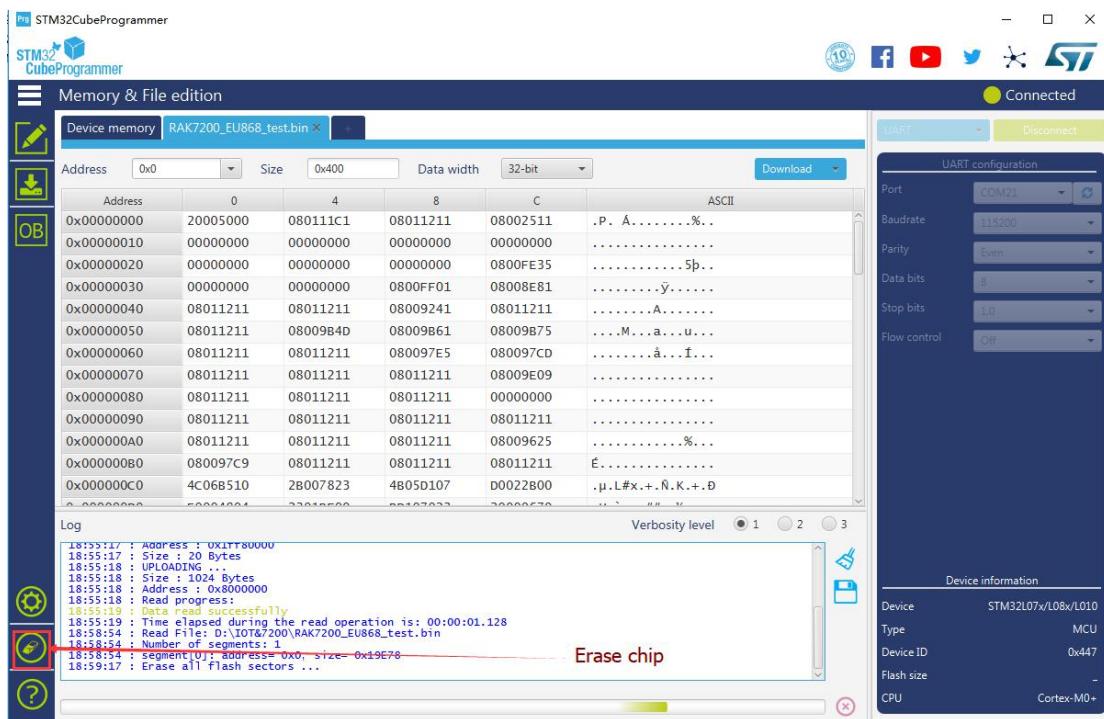


The correct Log you should see is the information like the following picture shows:

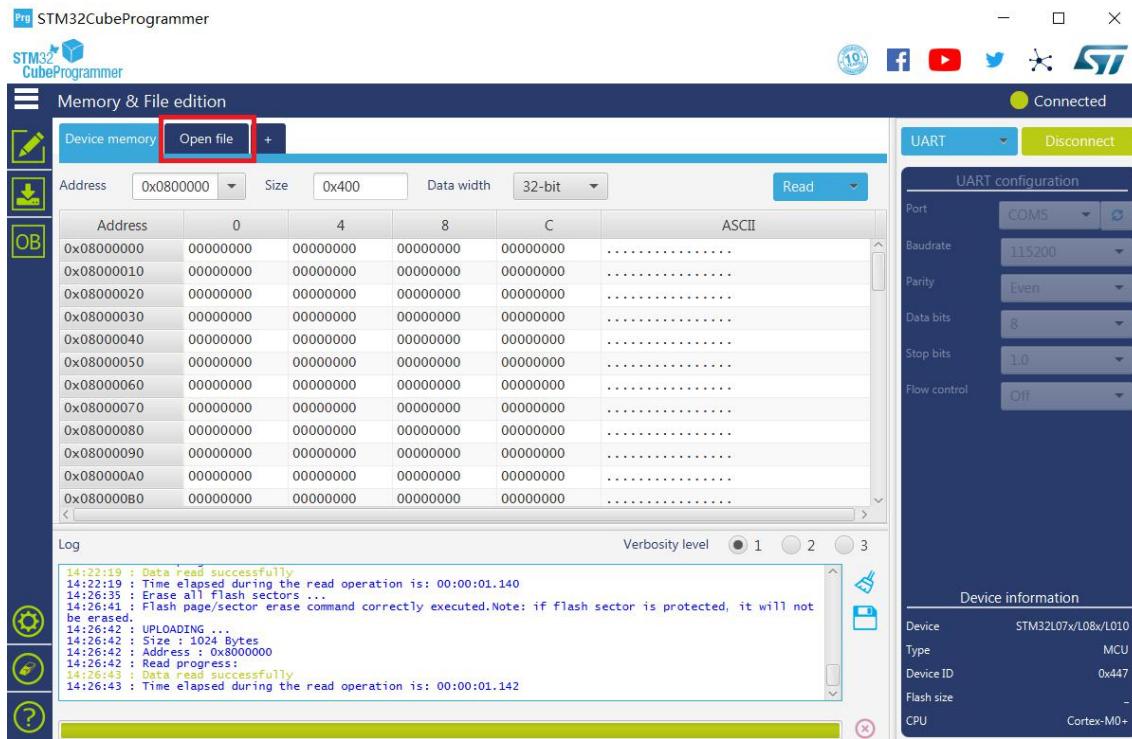


Now, let's start to burn a bootloader into RAK811 WisNode.

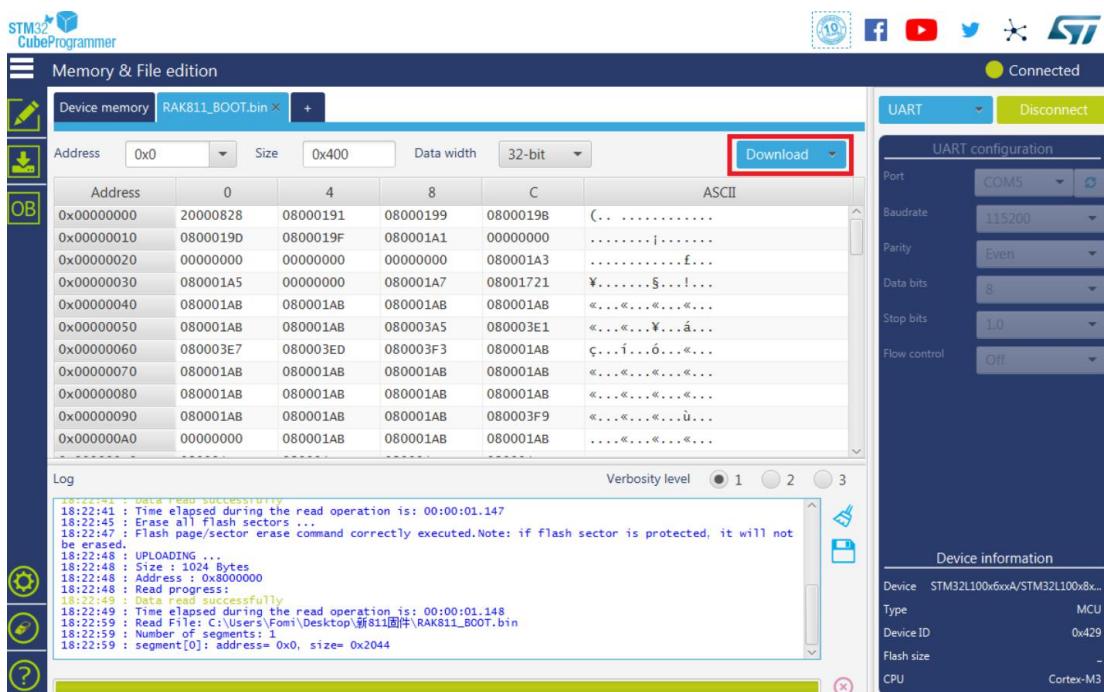
At first, erase all data on RAK811 WisNode according the following picture shows, it may take several seconds:

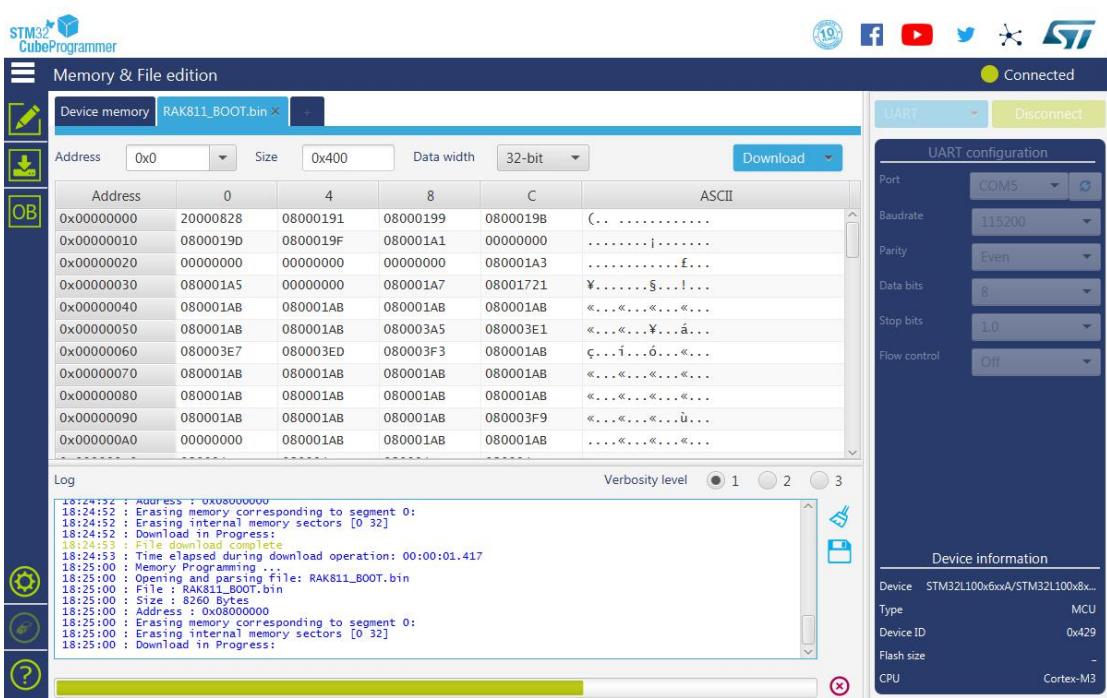


Press "Open file" and select the bootloader file in the pop-up window as follow:

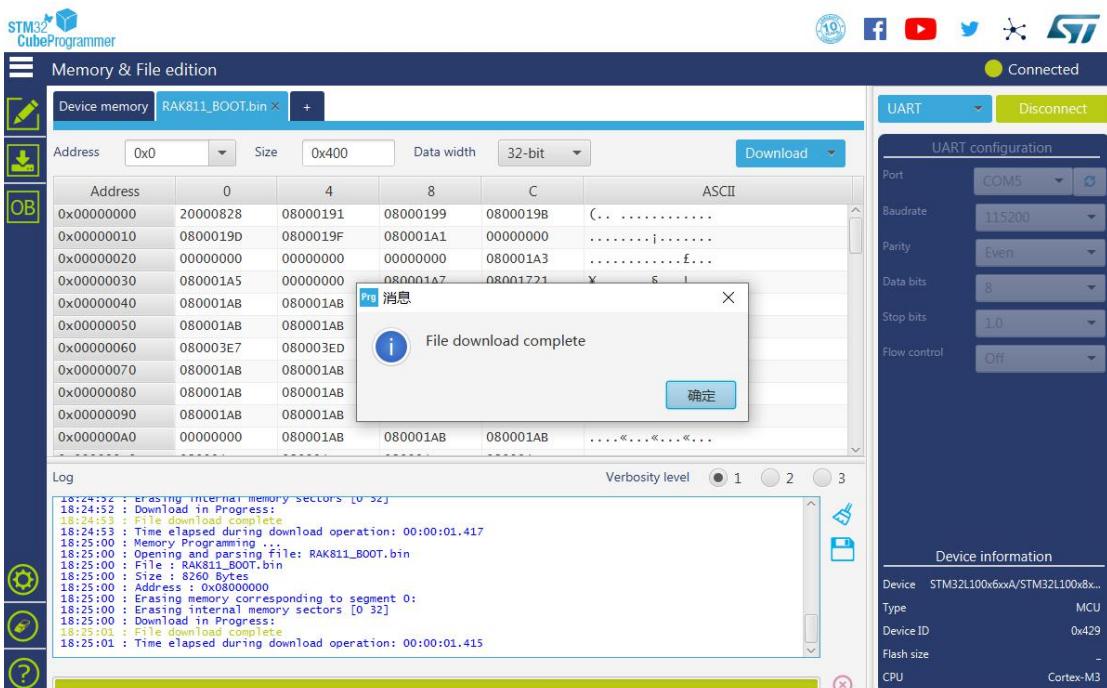


Press the “Download” button to start the burning process:



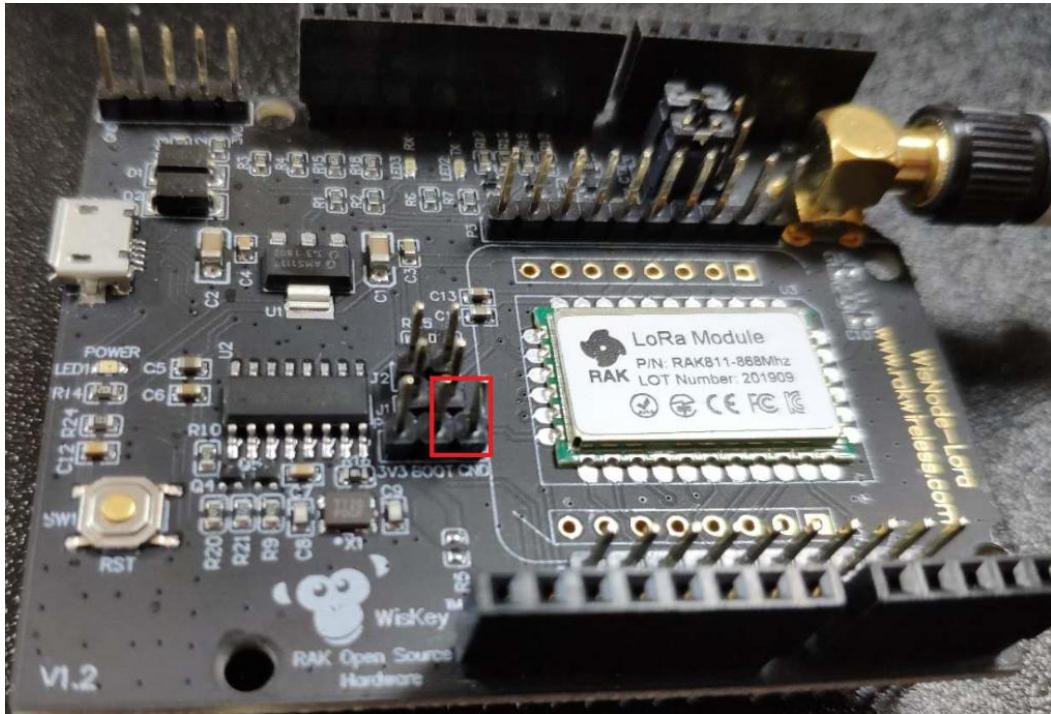


OK, you have burned a bootloader into RAK811 WisNode successfully!



“Disconnect” and close the “STM32CubeProgrammer” tool.

Now, power down and jump the “BOOT” pin and the “CND” pin as follow:



Then, connect RAK811 WisNode with your PC's USB interface again.

If you have opened the serial port tool, you can see some content like this:



It means that you have burned the bootloader into RAK811 WisNode successfully.

Note: Please download this serial port tool from RAK website:

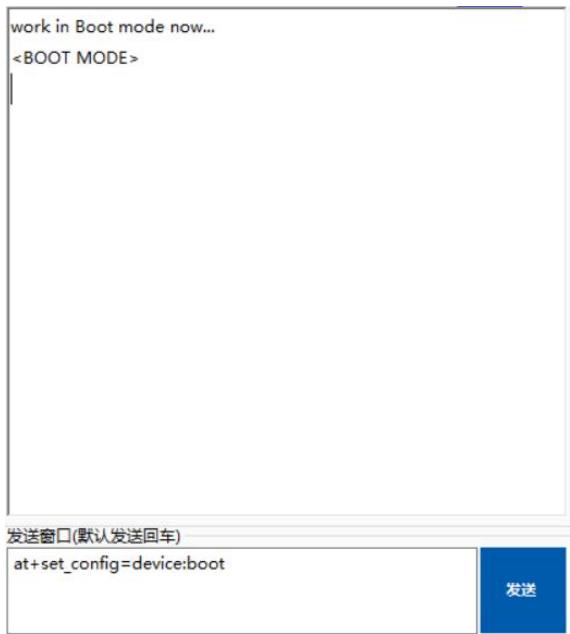
https://downloads.rakwireless.com/en/LoRa/RAK811/Tools/RAK_SERIAL_PORT_TOOL_V1.2.1.zip

OK, you should continue to burn a firmware into RAK811 WisNode now!

3. How to burn the firmware into RAK811?

If the current version of the RAK811 WisNode's firmware is newer than V3.0.0.0 or you have just burned a bootloader into RAK811 WisNode according to the section 2, you just need to burn a firmware according to the following steps now:

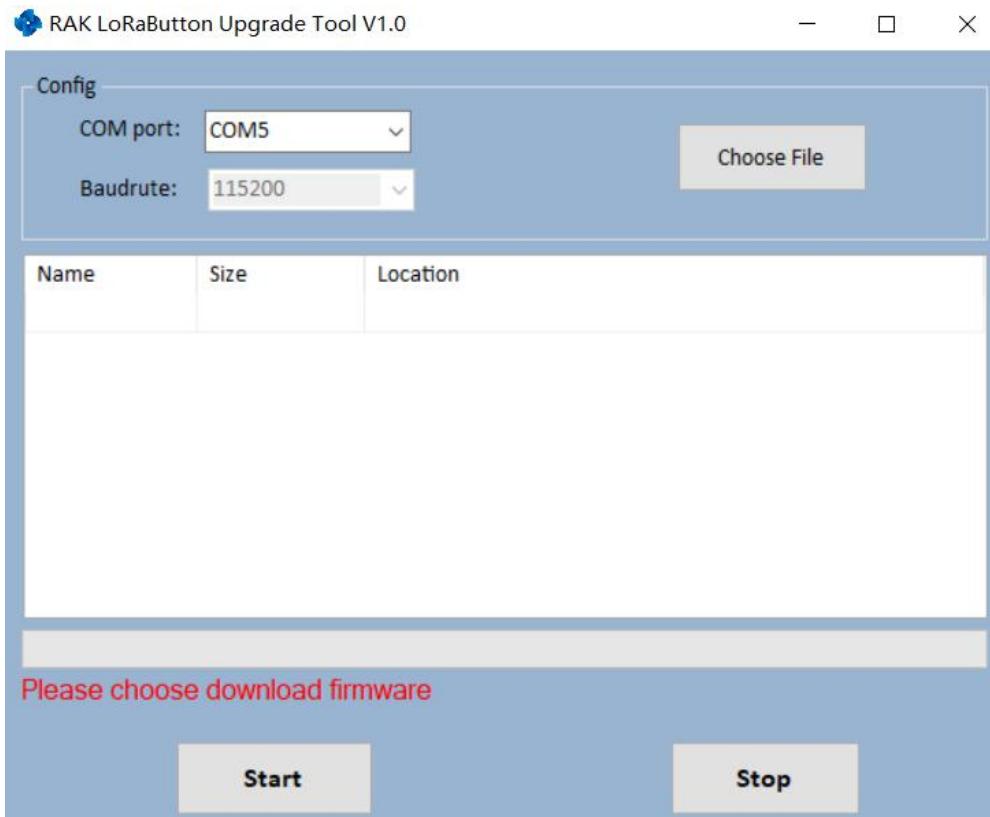
Firstly, let the RAK811 WisNode work in boot mode. If you have just burned a bootloader by yourself according to the section 2, it works in boot mode now. If the current version of the RAK811 WisNode's firmware is newer than V3.0.0.0, you need to set an AT command to let it work in boot mode as the following picture shows:



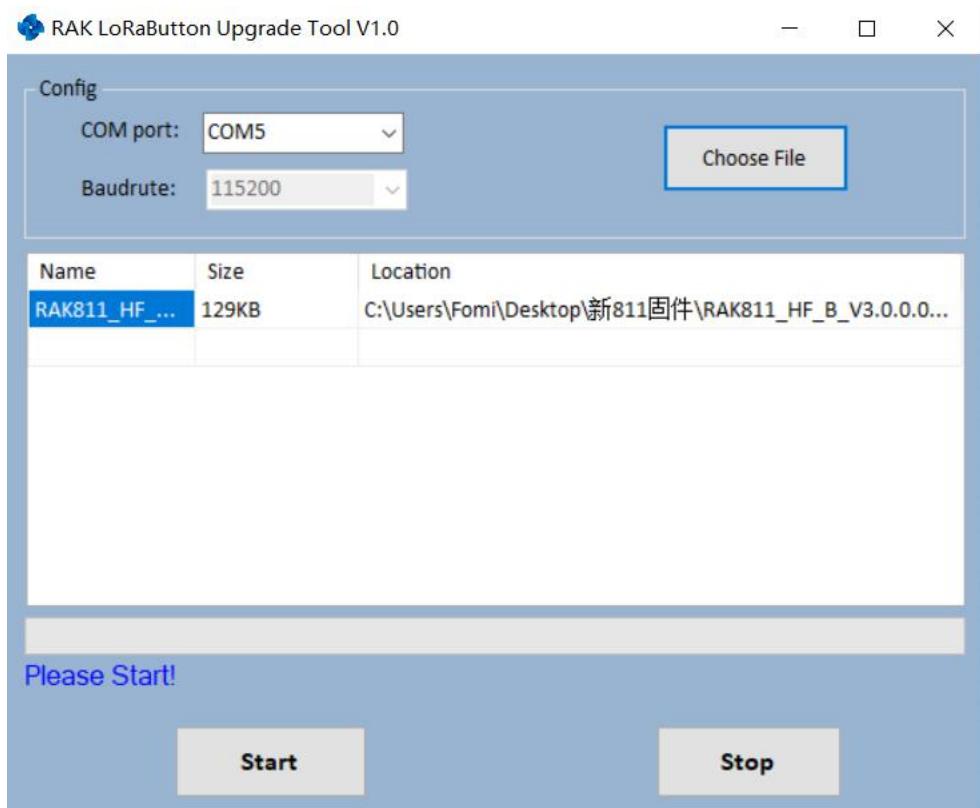
Secondly, close serial port tool and download the RAK Upgrade Tool from RAK website:

<https://www.rakwireless.com/en/download/LoRa/RAK612-LoRaButton#Tools>

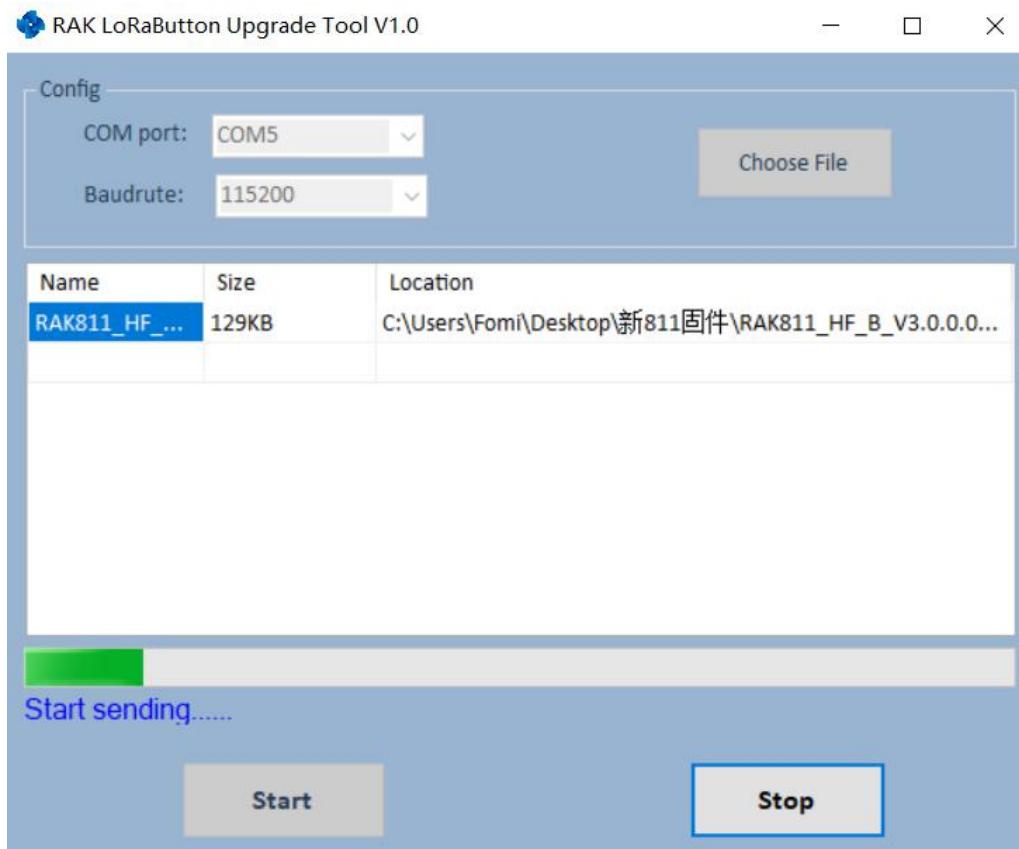
Open this tool:

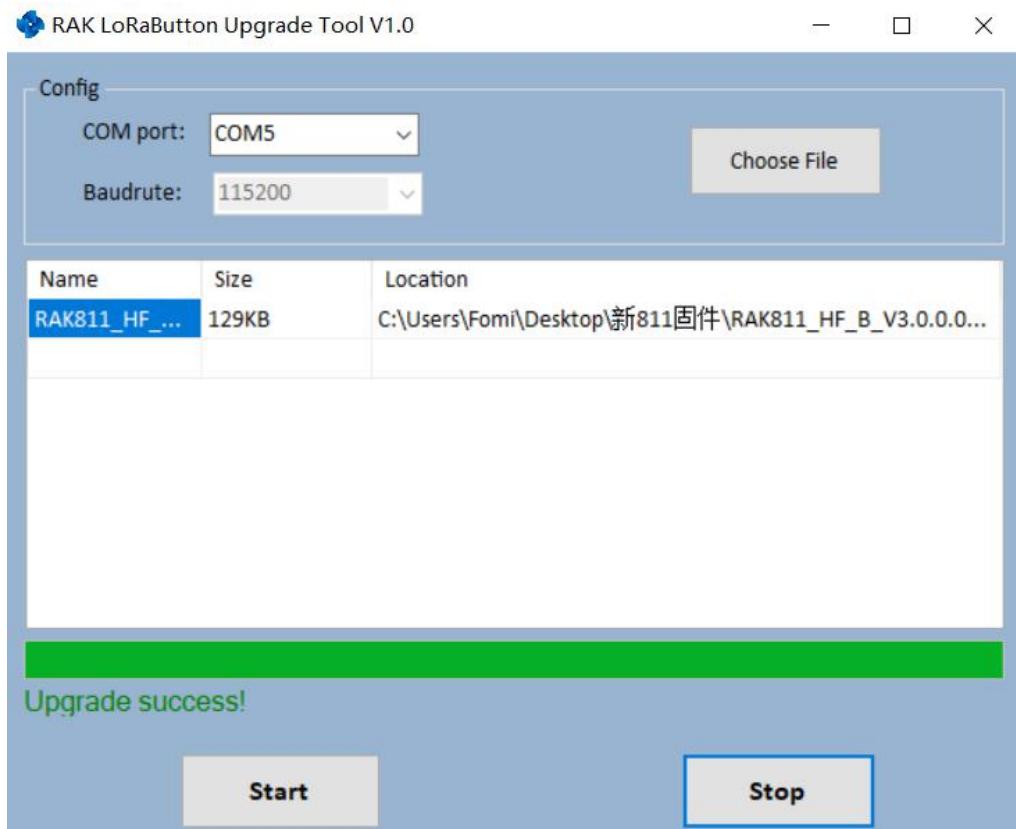


Click "Choose File" button to choose a correct upgrade file:



“Start” to upgrade, it may take one minutes:





Now, close the upgrade tool and open a serial port tool.

We recommend you to use RAK serial port tool, because there are some ready AT commands in this tool and this will be very useful for you. You can get it from RAK website and use it freely:

http://docs.rakwireless.com/en/LoRa/RAK811/Tools/RAK_SERIAL_PORT_TOOL_V1.2.1.zip

Choose the correct COM port and baud rate is 115200. Then press the RST button on RAK811 WisNode, you'll see the following log on serial port tool:

```
[11:58:08.737]收-<◆
=====
[RAK WIRELESS]
=====
*****  
RAK811 version:3.0.0.0.H  
*****  

=====
Selected LoRaWAN 1.0.2 Region: EU868
[11:58:08.858]收-<◆
LoRa Initialization OK.
```

It means that you have upgraded the new firmware successfully!

4. How to configure RAK811 WisNode?

You can configure RAK811 WisNode by sending AT commands into it from a serial port tool running on your PC.

The following list shows the AT commands:

AT Command	Description
at+help	This AT command can show all available AT commands of this module/product for you.
at+version	Get the current firmware version number.
at+get_config=device:status	Get all information about the device's hardware components and their current status.
at+set_config=device:restart	After set, the device will restart.
at+set_config=device:boot	Let the device work in boot mode.
at+run	Stop boot mode and run as normal. It is valid when the device works in boot mode.
at+set_config=device:sleep:X	After setting, the device will go to sleep mode or wake up immediately. X definition: 0: wake up, 1: sleep
at+set_config=device:gpio:X:Y	Set a certain GPIO pin to high/low level. X definition: the pin number of a certain GPIO on RAK811 module. Y definition: 0: low level, 1: high level.
at+get_config=device:gpio:X	Get a certain GPIO's level. X definition: the pin number of the GPIO on RAK811 module.
at+get_config=device:adc:X	Get the ADC value. X definition: the pin number of the ADC on RAK811 module.

at+set_config=device:iic:X:YY:ZZ:AAA	Read data from I2C or write a data to I2C. X definition: 0: read, 1: write. YY definition: device address, in HEX format. ZZ definition: sensor's register address, in HEX format. AAA definition: if read, this parameter means the length you want to read. If write, this parameter means the data you want to write. It must be in HEX format too.
at+set_config=device:uart_mode:X:Y	Set the UART work mode. X definition: UART number on RAK811 module. Y definition: 0: Configuration mode; 1: Passthrough mode.
at+set_config=device:uart:X:Y	Set a certain UART's Baud rate. X definition: the UART number. Y definition: the Baud rate value.
at+send=uart:X:YYY	Send data through UART. X definition: the UART number of RAK811 module. YYY definition: the data you want to send through UART.
at+join	Start to join LoRa network.
at+send=lora:X:YYY	Send a customized data. X definition: LoRa port YYY definition: the data which you want to send. The limited length is 50 Bytes, and the data must be in HEX format.
at+set_config=lora:work_mode:X	Set the work mode for LoRa. X definition: 0: LoRaWAN, 1: LoRaP2P, 2:

	Test Mode.
at+set_config=lora:join_mode:X	Set the join mode for LoRaWAN. X definition: 0: OTAA, 1: ABP
at+set_config=lora:class:X	Set the class for LoRa. X definition: 0: Class A, 1: Class B, 2: Class C
at+set_config=lora:region:XXX	Set the region for LoRa. XXX define: one of the following items: EU868 EU433, CN470, IN865, EU868, AU915, US915, KR920, AS923.
at+set_config=lora:confirm:X	Set the type of messages which will be sent out through LoRa. X definition: 0: unconfirm, 1: confirm
at+set_config=lora:dev_eui:XXXX	Set the device EUI for OTAA. XXXX definition: the device EUI.
at+set_config=lora:app_eui:XXXX	Set the application EUI for OTAA. XXXX definition: the application EUI.
at+set_config=lora:app_key:XXXX	Set the application key for OTAA. XXXX definition: the application key.
at+set_config=lora:dev_addr:XXXX	Set the device address for ABP. XXXX definition: the device address.
at+set_config=lora:apps_key:XXXX	Set the application session key for ABP. XXXX definition: the application session key.
at+set_config=lora:nwks_key:XXXX	Set the network session key for ABP. XXXX definition: the network session key.
at+set_config=lora:ch_mask:X:Y	Set a certain channel on or off. X definition: the channel number, and you can check which channel can be set before you set it.

	Y definition: 0: off, 1: on
at+set_config=lora:adr:X	<p>Open or close the ADR function of LoRa Node.</p> <p>X definition: 0: Close ADR; 1: Open ADR.</p>
at+set_config=lora:dr:X	<p>Set the DR of LoRa Node.</p> <p>X definition: the number of DR. Generally, the value of X can be 0~5. More details, please check the LoRaWAN 1.0.2 specification.</p>
at+set_config=lora:tx_power:X	<p>Set the TX power level.</p> <p>X definition: The level of TX power. If you want to know the relationship between TX power level and dbm, please have a look at LoRaWAN 1.0.2 region specification.</p> <p>https://github.com/RAKWireless/Update-File/blob/master/LoRaWANRegionalParametersv1.0.2.pdf</p>
at+get_config=lora:status	It will return all of the current information of LoRa, except LoRa channel.
at+get_config=lora:channel	It will return the state of all LoRa channels, then you can see which channel is closed and which channel is open very clearly.
at+set_config=lorap2p:XXX:Y:Z:A:B: C	<p>Set the parameters for LoRa P2P mode. This AT command is valid when the work mode is ·LoRaP2P.</p> <p>XXX definition: Frequency in Hz.</p> <p>Y definition: Spreading factor, [6, 7, 8, 9, 10, 11, 12].</p> <p>Z definition: Bandwidth,</p> <p>0: 125 kHz,</p> <p>1: 250 kHz,</p> <p>2: 500kHz.</p>

	<p>A definition: Coding Rate,</p> <p>1: 4/5,</p> <p>2: 4/6,</p> <p>3: 4/7,</p> <p>4: 4/8.</p> <p>B definition: Preamble Length, 5~65535.</p> <p>C definition: Power in dbm, 5~20.</p>
at+send=lorap2p:XXX	<p>Send data through LoRaP2P. This AT command is valid when it works in LoRaP2P mode.</p> <p>XXX definition: the data in HEX.</p>

More information, please have a look at the next section <How to Connect with TTN?>.

5. How to connect with TTN?

In this section, we'll do some practice to show how to connect RAK811 WisNode with TTN.

Firstly, connect RAK811 WisNode with your PC and open the serial port tool on your PC.

Open the serial port by click the following button:

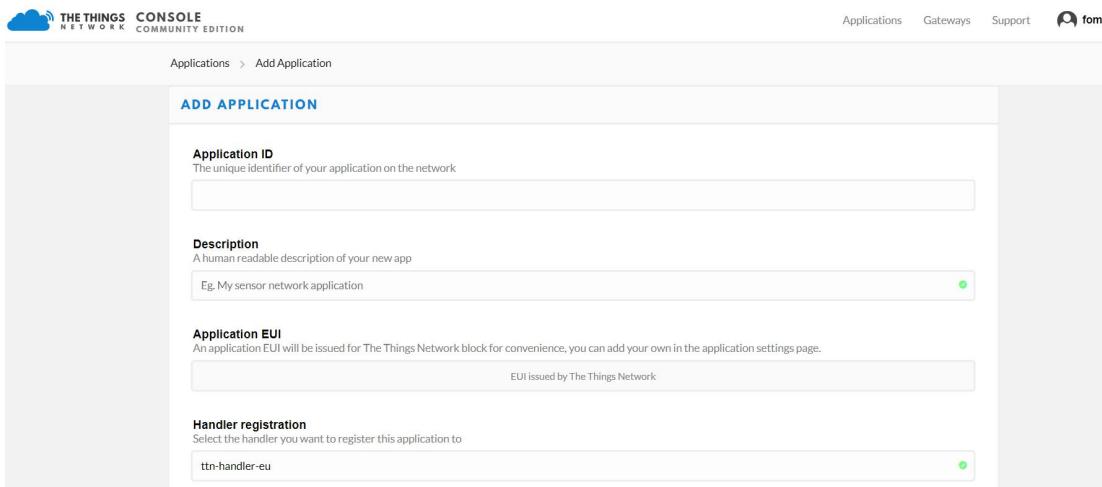


In this document, i assume that you have connected a LoRa gateway with TTN correctly. If not, please have a look at the document of RAK LoRa gateway.

OK, let's continue. Open the link <https://www.thethingsnetwork.org/> and login, then open the “Console” page from the right corner at the top:

Press “APPLICATIONS”:

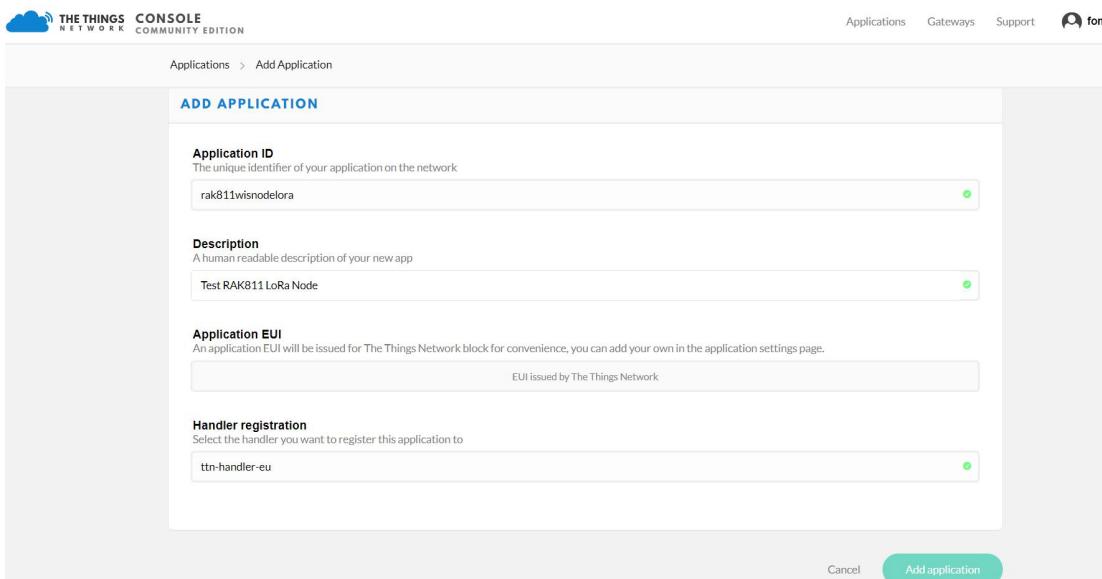
“add application”:



The screenshot shows the 'Add Application' page in The Things Network Console. The top navigation bar includes 'Applications', 'Gateways', 'Support', and a 'fom' icon. The main form has sections for 'Application ID', 'Description', 'Application EUI', and 'Handler registration'. The 'Application ID' field contains 'ttt-testapp'. The 'Description' field contains 'Test application for RAK811'. The 'Application EUI' field contains 'EUI issued by The Things Network'. The 'Handler registration' field contains 'ttt-handler-eu'. A green checkmark icon is visible next to each input field.

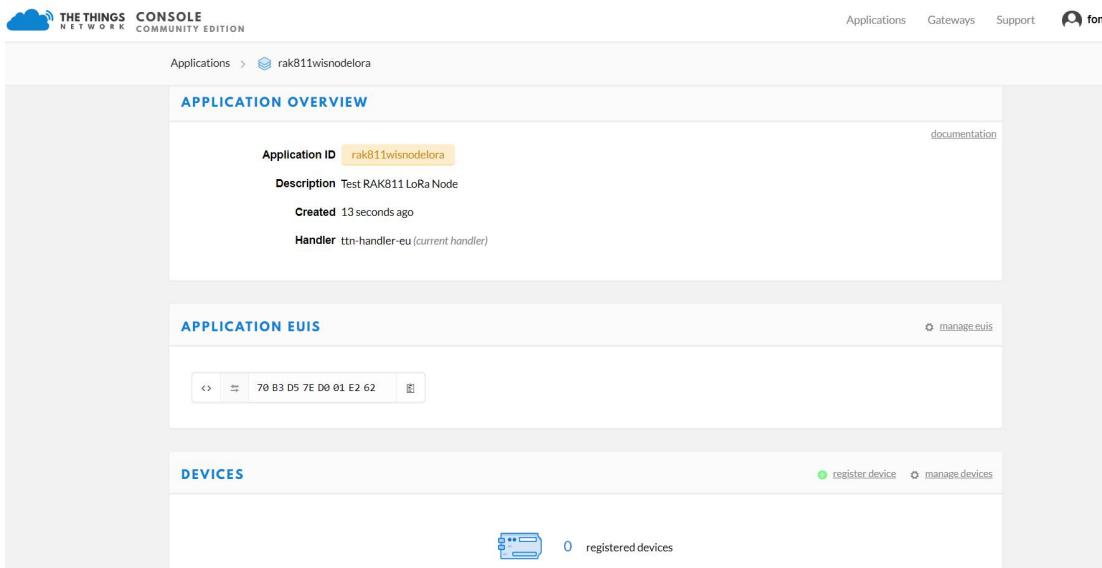
Fill in the correct contents.

Please note that the content you fill in “Application ID” item should be in low case, and it must be the unique ID on TTN network.



The screenshot shows the 'Add Application' page in The Things Network Console with filled fields. The 'Application ID' field contains 'rak811wisnode1ora'. The 'Description' field contains 'Test RAK811 LoRa Node'. The 'Application EUI' field contains 'EUI issued by The Things Network'. The 'Handler registration' field contains 'ttt-handler-eu'. At the bottom right, there are 'Cancel' and 'Add application' buttons, with 'Add application' being highlighted in green.

Then press the “Add application” button at the bottom of this page, and you can see the following page:



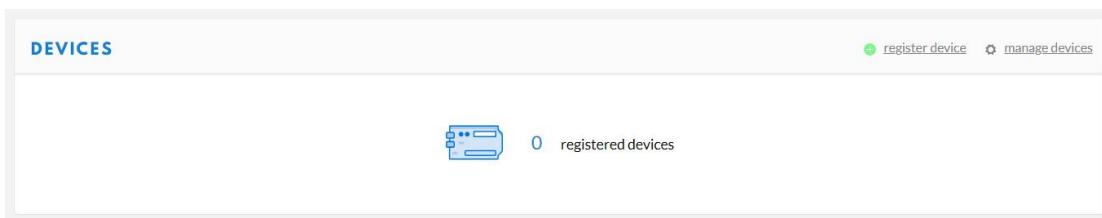
The screenshot shows the 'APPLICATION OVERVIEW' section with the following details:

- Application ID:** rak811wisnodelora
- Description:** Test RAK811 LoRa Node
- Created:** 13 seconds ago
- Handler:** ttn-handler-eu (current handler)

The 'APPLICATION EUIS' section shows a hex string: 70 B3 D5 7E D0 01 E2 62.

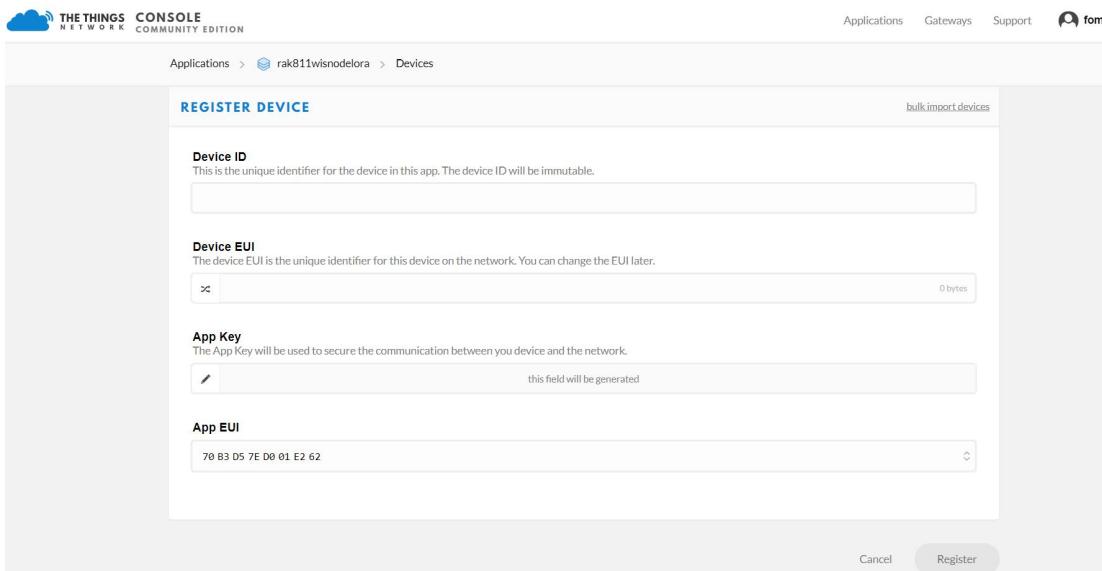
The 'DEVICES' section shows 0 registered devices.

At the middle of this page, you can find the box named “DEVICES”:



The 'DEVICES' section shows 0 registered devices.

Just “register device”:

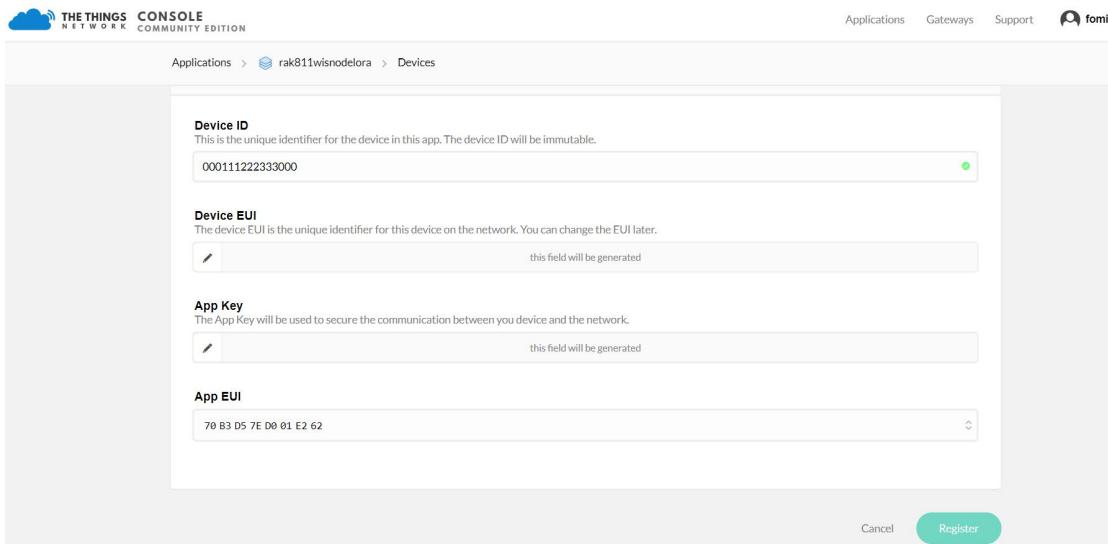


The 'REGISTER DEVICE' form includes the following fields:

- Device ID:** A text input field for the unique identifier for the device.
- Device EUI:** A text input field for the unique identifier for this device on the network, with a note that it can be changed later.
- App Key:** A text input field for the App Key used to secure communication, with a note that it will be generated.
- App EUI:** A text input field containing the hex string 70 B3 D5 7E D0 01 E2 62.

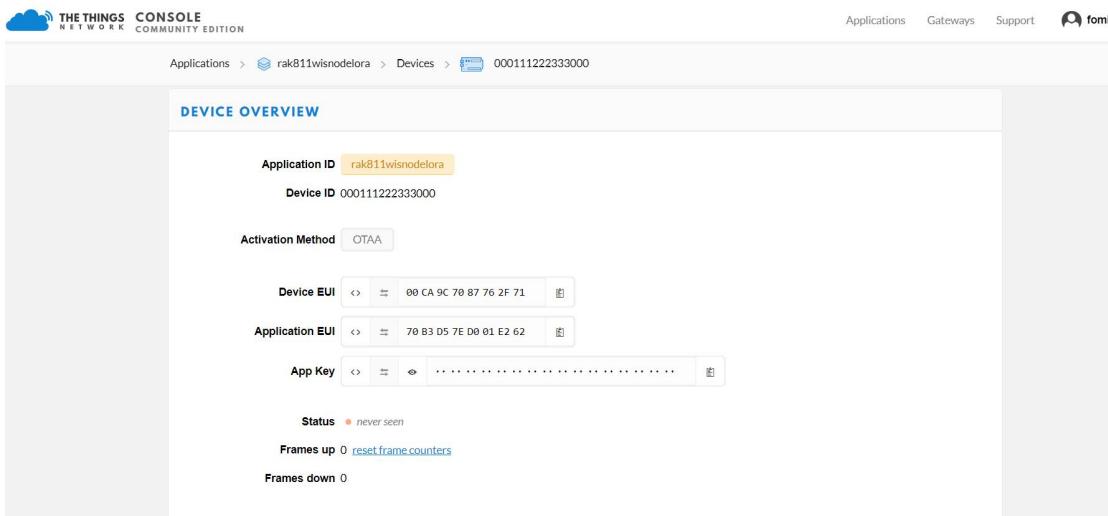
Buttons at the bottom right include 'Cancel' and 'Register'.

You can click the icon and “Device EUI” will be generated automatically in the next step:



The screenshot shows the 'Device ID' field containing '000111222333000'. Below it, the 'Device EUI' field has a placeholder 'this field will be generated'. The 'App Key' field also has a placeholder 'this field will be generated'. The 'App EUI' field contains '70 B3 D5 7E D0 01 E2 62'. At the bottom right are 'Cancel' and 'Register' buttons.

Then press the “Register” button at the bottom of this page to finish.

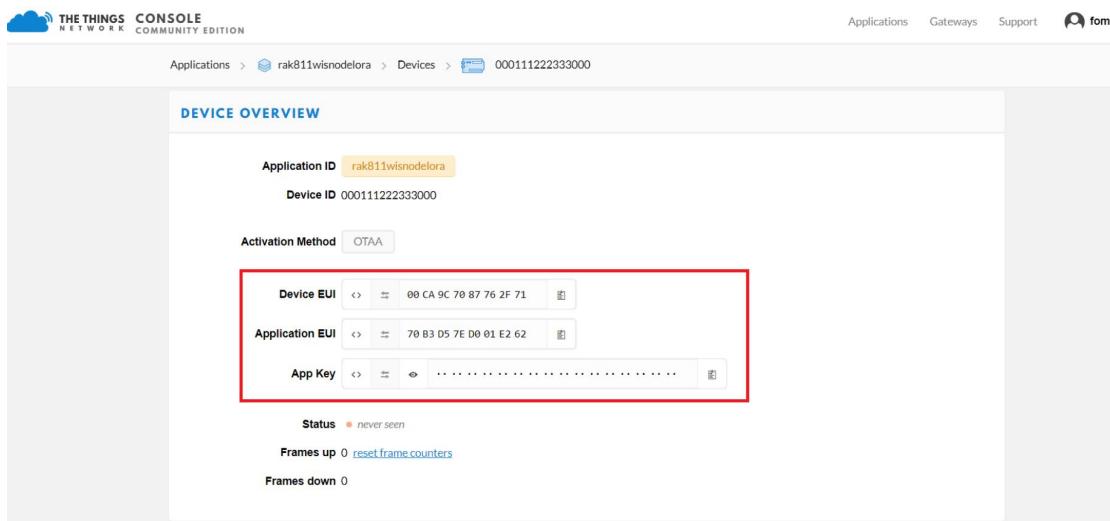


The screenshot shows the 'Application ID' as 'rak811wisnode'. Other fields include 'Device ID' (000111222333000), 'Activation Method' (OTAA), 'Device EUI' (00 CA 9C 70 87 76 2F 71), 'Application EUI' (70 B3 D5 7E D0 01 E2 62), and 'App Key' (redacted). Status information shows 'never seen' for frames up (0) and down (0). Buttons for 'reset frame counters' and 'Frames up' are also visible.

5.1 Join in OTAA mode

As you see in the above page, the default activation method is OTAA.

These three parameters will be used on RAK811 WisNode:



The screenshot shows the 'Device Overview' page for a device named 'rak811wisnodelora'. The 'Activation Method' is listed as 'OTAA'. The 'Device EUI', 'Application EUI', and 'App Key' fields are highlighted with a red box. Below these fields, the 'Status' is shown as 'never seen'. Under the 'Frames up' section, there is a link to 'reset frame counters'.

OK! Now, let's join in OTAA mode and EU868 frequency for example!

The default LoRa work mode is LoRaWAN 1.0.2, the default LoRa join mode is OTAA, and the default LoRa class is Class A.

If the join mode is not OTAA now, just set the LoRa join mode to OTAA as follow:



```

RAK811 version:3.0.0.0.H
Selected LoRaWAN 1.0.2 Region: AU915
LoRa Initialization OK.
>>at+set_config=lora:join_mode:0
OK

```

发送窗口(默认发送回车)
at+set_config=lora:join_mode:0

Set the LoRa class to Class A:

```
=====
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK
>>at+set_config=lora:class:0
OK
|
```

发送窗口(默认发送回车)
at+set_config=lora:class:0 发送

Set the frequency/region to EU868:

```
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868

Selected LoRaWAN 1.0.2 Region: EU868
Band switch success.
OK
|
```

发送窗口(默认发送回车)
at+set_config=lora:region:EU868 发送

Set the Device EUI:

```
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: EU868

LoRa Initialization OK.
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_eui:00CA9C7087762F71
OK

发送窗口(默认发送回车)
at+set_config=lora:dev_eui:00CA9C7087762F71
```

发送

Set the Application EUI:

```
RAK811 version:3.0.0.0.H
*****
=====
Selected LoRaWAN 1.0.2 Region: EU868

LoRa Initialization OK.
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_eui:00CA9C7087762F71
OK
>>at+set_config=lora:app_eui:70B3D57ED001E262
OK

发送窗口(默认发送回车)
at+set_config=lora:app_eui:70B3D57ED001E262
```

发送

Set the Application Key:

```

Selected LoRaWAN 1.0.2 Region: EU868

LoRa Initialization OK.
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_eui:00CA9C7087762F71
OK
>>at+set_config=lora:app_eui:70B3D57ED001E262
OK
>>
at+set_config=lora:app_key:C85E8499381501391B7727393781472
C
OK

发送窗口(默认发送回车)
at
+set_config=lora:app_key:C85E8499381501391B7727393
781472C
    发送

```

Join in OTAA mode:

```

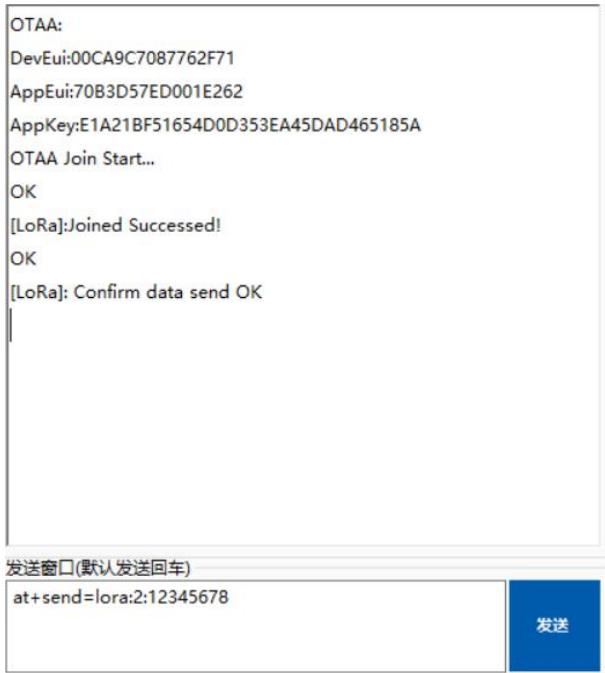
OTAA:
DevEui:00CA9C7087762F71
AppEui:70B3D57ED001E262
AppKey:E1A21BF51654D0D353EA45DAD465185A
OTAA Join Start...
OK
[LoRa]:Joined Successed!

发送窗口(默认发送回车)
at+join
    发送

```

Join successfully!

Now, let's try to send a data from RAK811 WisNode to TTN:



You can see the data sending from RAK811 WisNode on TTN website as follow:

The screenshot shows the TTN console interface. The top navigation bar includes "THE THINGS NETWORK CONSOLE COMMUNITY EDITION", "Applications", "Gateways", "Support", and a user icon. The main area displays the path: Applications > rak811wisnode > Devices > 0001122233000 > Data. Below this, there are tabs for "Overview", "Data" (which is selected), and "Settings". The "APPLICATION DATA" section has a header with "uplink", "downlink", "activation", "ack", and "error" buttons, and a "Filters" dropdown set to "uplink". It lists three data entries:

time	counter	port	
17:15:07	0		
17:15:07	0	2	confirmed payload: 12 34 56 78
17:13:58			dev addr: 26012D78 app eui: 70B3D57ED001E262 dev eui: 00CA9C7087762F71

Great! That's all about OTAA mode.

5.2 Join in ABP mode

First you need to switch the activation method is ABP.

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications > rak811wisnodeora > Devices > 00011222333000 > Settings

Overview Data **Settings**

DEVICE SETTINGS	SETTINGS
General	Description A human-readable description of the device
Location	
	Device EUI The serial number of your radio module, similar to a MAC address 00 CA 9C 70 87 76 2F 71 8 bytes
	Application EUI 70 B3 D5 7E D0 01 E2 62
	Activation Method OTAA ABP
	App Key The key your device will use to set up sessions with the network

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications > rak811wisnodeora > Devices > 00011222333000 > Settings

Device EUI The serial number of your radio module, similar to a MAC address 00 CA 9C 70 87 76 2F 71 8 bytes
Application EUI 70 B3 D5 7E D0 01 E2 62
Activation Method OTAA ABP
Device Address The device address will be assigned by the network server
Network Session Key Network Session Key will be generated
App Session Key App Session Key will be generated

These three parameters will be used on RAK811 WisNode:

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications > rak811wisnodeora > Devices > 00011222333000

DEVICE OVERVIEW

Application ID rak811wisnodeora
Device ID 00011222333000
Activation Method ABP
Device EUI 00 CA 9C 70 87 76 2F 71
Application EUI 70 B3 D5 7E D0 01 E2 62
Device Address 26 01 10 0F
Network Session Key 4 days ago
App Session Key 4 days ago
Status 4 days ago
Frames up 0 reset frame counters
Frames down 0

OK! Now, let's join in ABP mode and EU868 frequency for example!

The default LoRa work mode is LoRaWAN 1.0.2, the default LoRa join mode is ABP, and the default LoRa class is Class A.

If the join mode is not ABP now, just set the LoRa join mode to ABP as follow:

```

=====
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: US915

LoRa Initialization OK.
>>at+set_config=lora:join_mode:1
OK
|-----|
发送窗口(默认发送回车)
at+set_config=lora:join_mode:1
|-----|发送

```

Set the LoRa class to Class A:

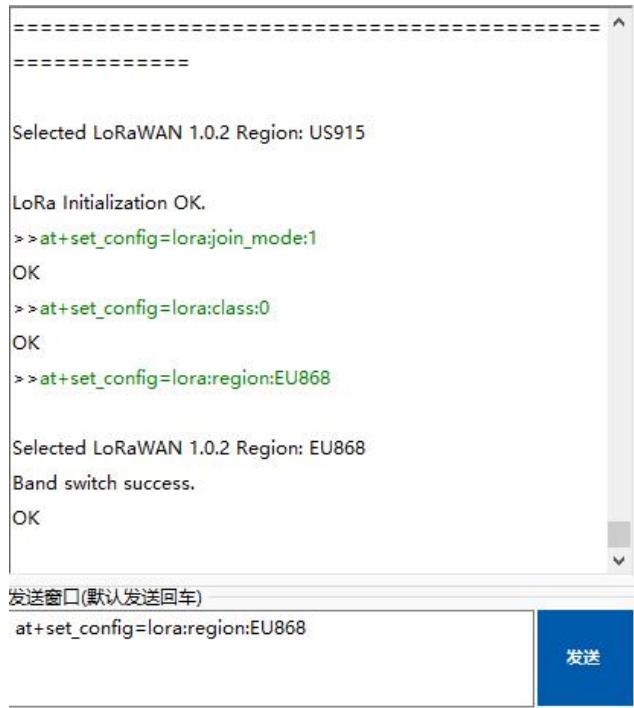
```

=====
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: US915

LoRa Initialization OK.
>>at+set_config=lora:join_mode:1
OK
>>at+set_config=lora:class:0
OK
|-----|
发送窗口(默认发送回车)
at+set_config=lora:class:0
|-----|发送

```

Set the frequency/region to EU868:



```
=====
=====
Selected LoRaWAN 1.0.2 Region: US915

LoRa Initialization OK.
>>at+set_config=lora:join_mode:1
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868

Selected LoRaWAN 1.0.2 Region: EU868
Band switch success.
OK
```

发送窗口(默认发送回车)
at+set_config=lora:region:EU868 发送

Set the dev_addr:



```
>>at+set_config=lora:dev_addr:2601100F
OK
```

发送窗口(默认发送回车)
at+set_config=lora:dev_addr:2601100F 发送

Set the nwks_key:

```
>>at+set_config=lora:dev_addr:2601100F  
OK  
>>  
at+set_config=lora:nwks_key:5CAE46002AC3A799764A6F511575A4DE  
OK
```

发送窗口(默认发送回车)

```
at  
+set_config=lora:nwks_key:5CAE46002AC3A799764A6F51  
1575A4DE
```

发送

Set the apps_key:

```
>>at+set_config=lora:dev_addr:2601100F  
OK  
>>  
at+set_config=lora:nwks_key:5CAE46002AC3A799764A6F511575A4DE  
OK  
>>  
at+set_config=lora:apps_key:4A1D7677AFCF14CFF4B7AE127CDC92D9  
OK
```

发送窗口(默认发送回车)

```
at  
+set_config=lora:apps_key:4A1D7677AFCF14CFF4B7AE12  
7CDC92D9
```

发送

Join in ABP mode:

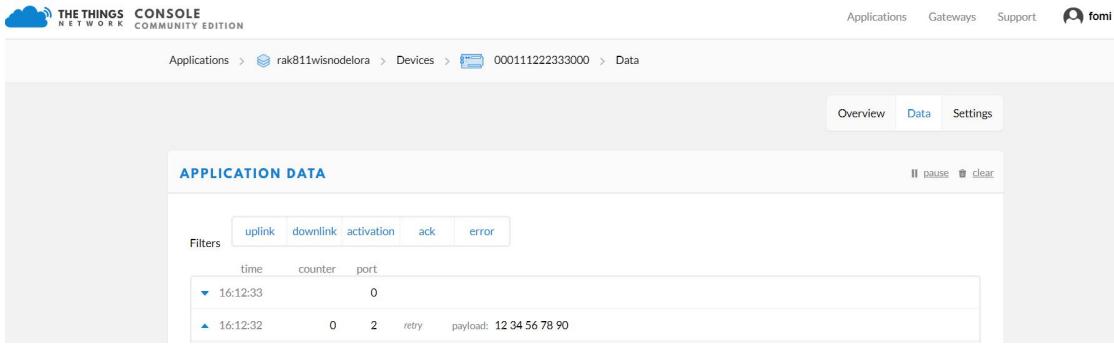
```
>> at+set_config=lora:dev_addr:2601100F
OK
>>
at+set_config=lora:nwks_key:5CAE46002AC3A799764A6F511575A4DE
OK
>>
at+set_config=lora:apps_key:4A1D7677AFCF14CFF4B7AE127CDC92D9
OK
ABP:
DevEui: 3238333560385D04
DevAddr: 2601100F
AppsKey: 4A1D7677AFCF14CFF4B7AE127CDC92D9
NwksKey: 5CAE46002AC3A799764A6F511575A4DE
OK
发送窗口(默认发送回车)
at+join
发送
```

Note: Actually, it is no need to join in ABP mode. But you still need to set this AT command to valid the parameters which you just set for ABP mode.

Now, let's try to send a data from RAK811 WisNode to TTN in ABP mode.

```
at+set_config=lora:nwks_key:5CAE46002AC3A799764A6F511575A4DE
OK
>>
at+set_config=lora:apps_key:4A1D7677AFCF14CFF4B7AE127CDC92D9
OK
ABP:
DevEui: 3238333560385D04
DevAddr: 2601100F
AppsKey: 4A1D7677AFCF14CFF4B7AE127CDC92D9
NwksKey: 5CAE46002AC3A799764A6F511575A4DE
OK
OK
[LoRa]: Unconfirm data send OK
发送窗口(默认发送回车)
at+send=lora:2:1234567890
发送
```

You can see that TTN has received the data you just sent, as the following picture shows:



The screenshot shows the 'Data' tab of the application's device page. It lists two data entries:

- 16:12:33, counter 0
- 16:12:32, counter 2, retry, payload: 12 34 56 78 90

Great! That's all about ABP mode.

6. How to connect with LoRaServer?

The LoRaServer is an open source project which you can find in

<https://www.loraserver.io/>.

You can use RAK811 WisNode to connect with LoRaServer according to the following steps:

In this document, I assume that you are using RAK LoRa gateway and its built-in LoRaServer or RAK cloud testing LoRaServer. I also assume that you have configured a LoRa gateway with the LoRaServer successfully. If not, please have a look at RAK's documents for More details about RAK LoRa gateway and RAK cloud testing LoRaServer:

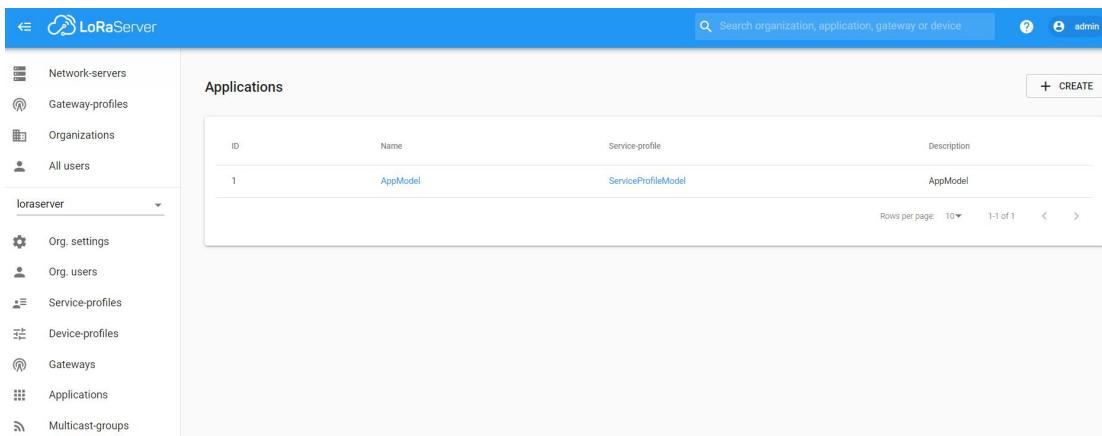
<https://downloads.rakwireless.com/en/LoRa/>

Also, RAK supply a free cloud LoRaServer for you to test, just go to RAK forum to submit an application:

<https://forum.rakwireless.com/t/rak-free-cloud-loraserver-for-testing/344/45>

OK! Let's get start!

Open the web page of the LoRaServer which you want to connect with and login.

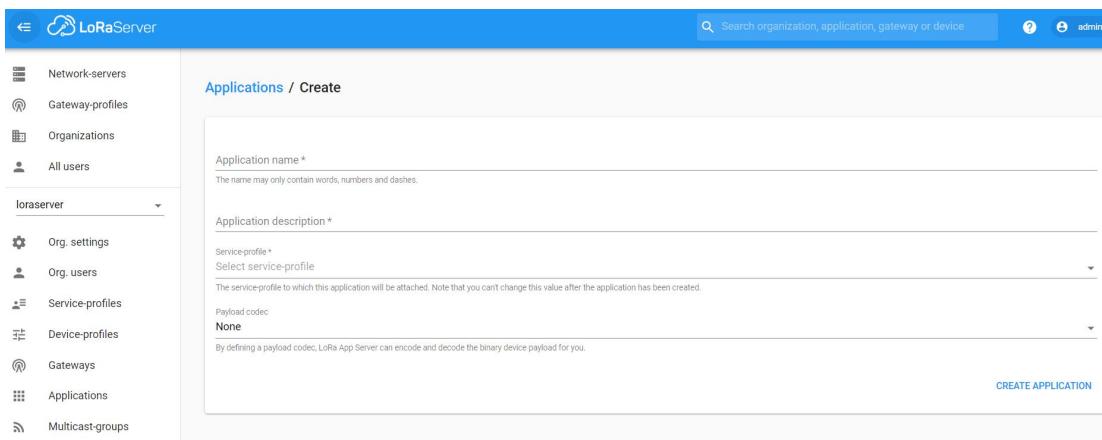


The screenshot shows the LoRaServer interface with the title "Applications". On the left, there is a sidebar with a tree view of the organization structure under "loraserver". The main area displays a table with one item:

ID	Name	Service-profile	Description
1	AppModel	ServiceProfileModel	AppModel

Below the table, there are pagination controls: "Rows per page: 10", "1-1 of 1", and navigation arrows.

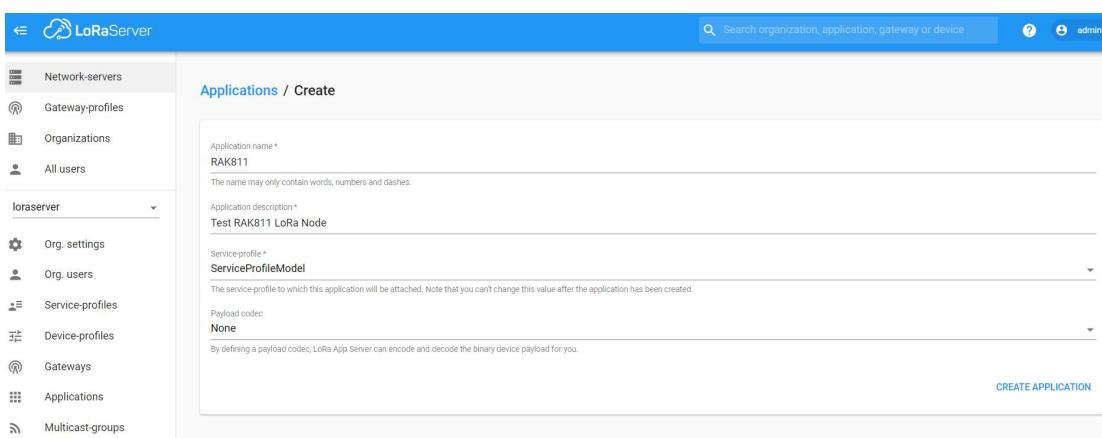
By default, there is already one or more items in this page, you can use it or create a new item. Now, let's create a new item by click the “CREATE” button, and fill in them.



The screenshot shows the "Applications / Create" form. The sidebar on the left is identical to the previous screenshot. The main form fields are:

- Application name ***: A text input field containing "AppModel". A note below says: "The name may only contain words, numbers and dashes."
- Application description ***: A text input field containing "Test Application".
- Service-profile ***: A dropdown menu set to "ServiceProfileModel". A note below says: "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**: A dropdown menu set to "None". A note below says: "By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you."

A "CREATE APPLICATION" button is located at the bottom right of the form.

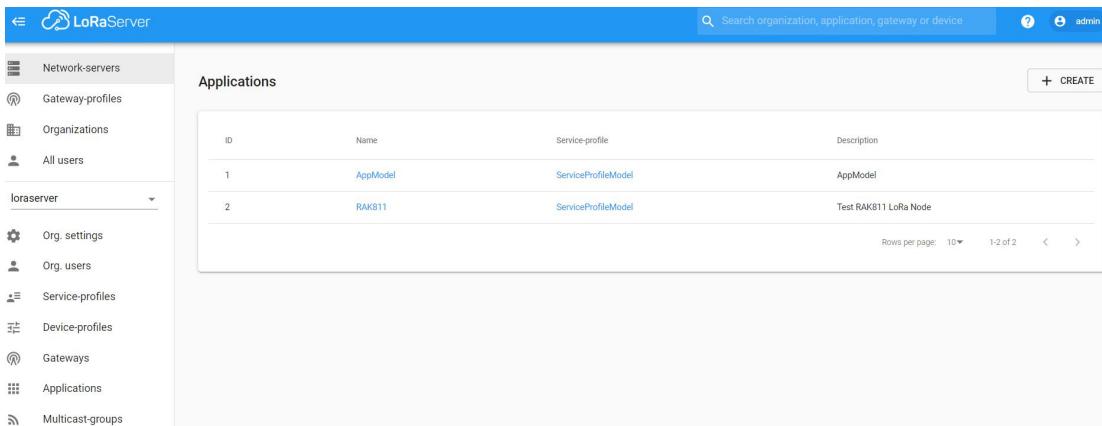


The screenshot shows the "Applications / Create" form with the following filled fields:

- Application name ***: "RAK811". A note below says: "The name may only contain words, numbers and dashes."
- Application description ***: "Test RAK811 LoRa Node".
- Service-profile ***: "ServiceProfileModel". A note below says: "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". A note below says: "By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you."

A "CREATE APPLICATION" button is located at the bottom right of the form.

“CREATE APPLICATION”.

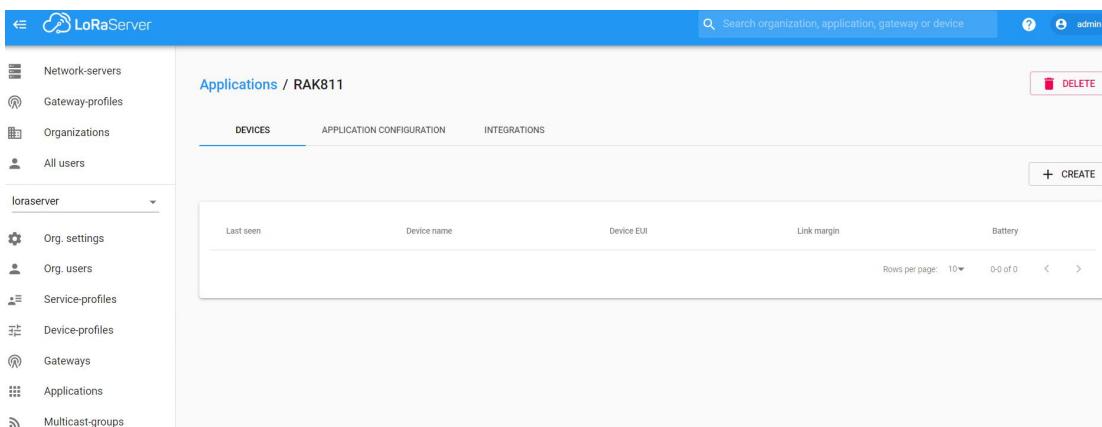


The screenshot shows the LoRaServer application management interface. On the left is a sidebar with navigation links: Network-servers, Gateway-profiles, Organizations, All users, loraserver (selected), Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main area is titled "Applications" and contains a table with two rows:

ID	Name	Service-profile	Description
1	AppModel	ServiceProfileModel	AppModel
2	RAK811	ServiceProfileModel	Test RAK811 LoRa Node

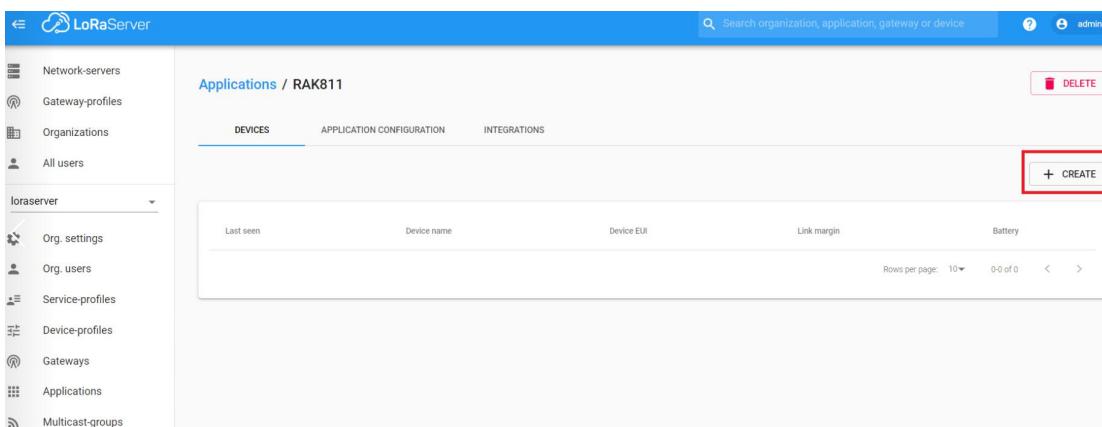
At the bottom right of the table are buttons for "Rows per page" (10), "1-2 of 2", and navigation arrows.

Click the new item name “RAK811”:

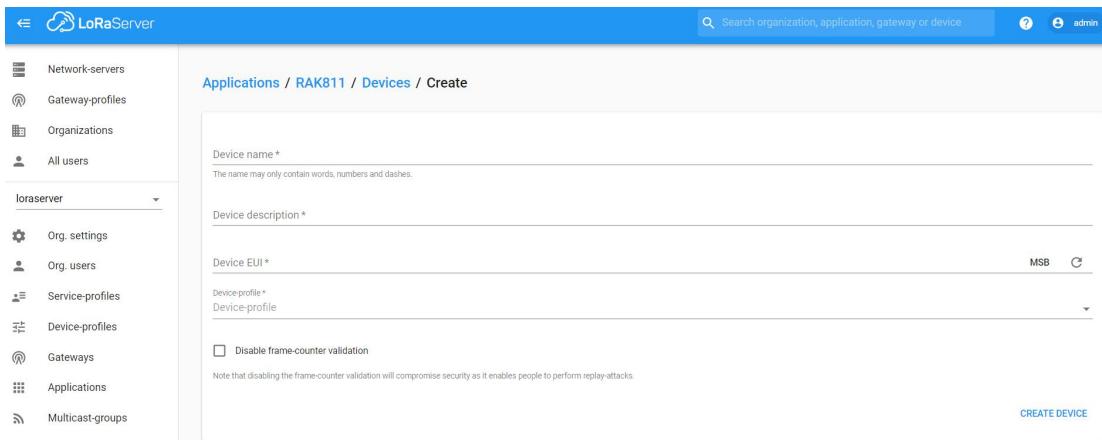


The screenshot shows the detailed view for the RAK811 application. The sidebar and main navigation are identical to the previous screenshot. The main area is titled "Applications / RAK811" and has three tabs: DEVICES (selected), APPLICATION CONFIGURATION, and INTEGRATIONS. Below the tabs is a table with columns: Last seen, Device name, Device EUI, Link margin, and Battery. A red "DELETE" button is located at the top right of the table area.

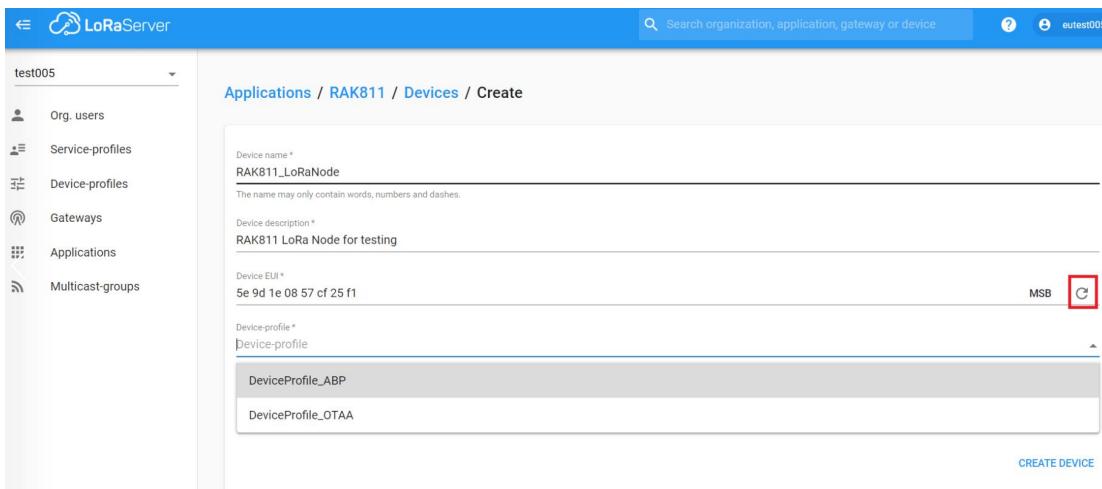
Add a LoRa node device into LoRaServer by clicking the “CREATE” button:



This screenshot is similar to the previous one, showing the RAK811 application details. However, the "+ CREATE" button in the top right corner of the table area is now highlighted with a red box.



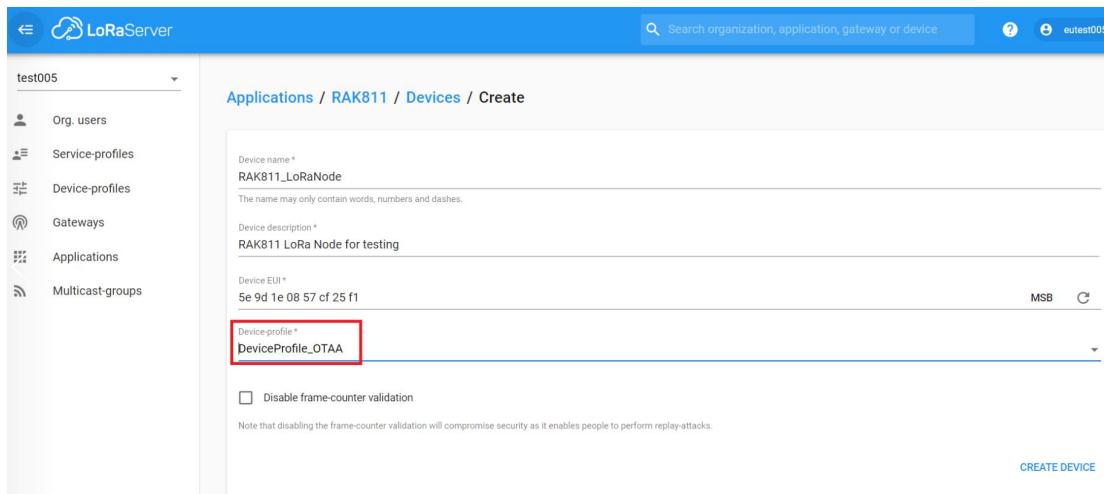
Fill in them. You can generate a Device EUI automatically by click the following icon, or you can write a correct Device EUI in the edit box.



Note: If you want to join in OTAA mode, you should select “**DeviceProfile_OTAA**” in the “Device-profile” item. If you want to join in ABP mode and CN470 frequency, you should select “**DeviceProfile_ABP_CN470**” in the “Device-profile” item. If you want to join in ABP mode and other frequencies except AS923 and CN470, you should select “**DeviceProfile_ABP**” in the “Device-profile” item. What about AS923 in ABP mode? Sorry! LoRaServer can not support it now.

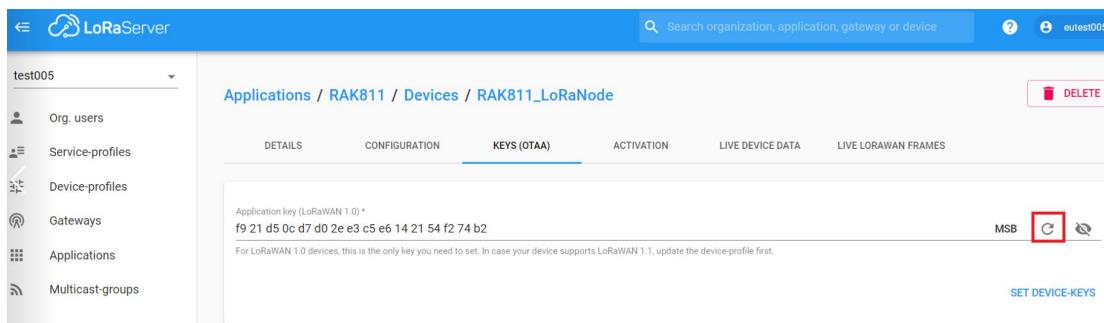
6.1 Join in OTAA mode

If you select “DeviceProfile_OTAA”, it means you want to join LoRaServer in OTAA mode.



The screenshot shows the LoRaServer interface for creating a new device. In the left sidebar, under the 'Devices' section, 'Device-profile' is selected. A red box highlights the 'Device-profile' dropdown menu, which contains the entry 'DeviceProfile_OTAAC'. The main form fields include 'Device name' (RAK811_LoRaNode), 'Device description' (RAK811 LoRa Node for testing), 'Device EUI' (5e 9d 1e 08 57 cf 25 f1), and 'Device-profile' (DeviceProfile_OTAAC). A note about frame-counter validation is present at the bottom.

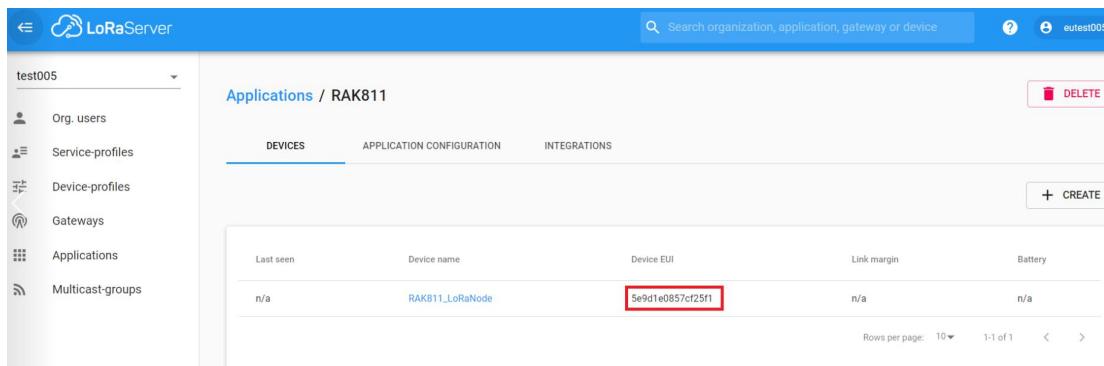
“CREATE DEVICE”. Then generate the application key in this page. You can write it by yourself or generate it automatically by clicking the following icon:



The screenshot shows the LoRaServer interface for managing a specific device named 'RAK811_LoRaNode'. Under the 'KEYS (OTAA)' tab, the 'Application key (LoRaWAN 1.0)' field is populated with the value 'f9 21 d5 0c d7 d0 2e e3 c5 e6 14 21 54 f2 74 b2'. A red box highlights the 'C' icon next to the 'MSB' switch, indicating a copy operation.

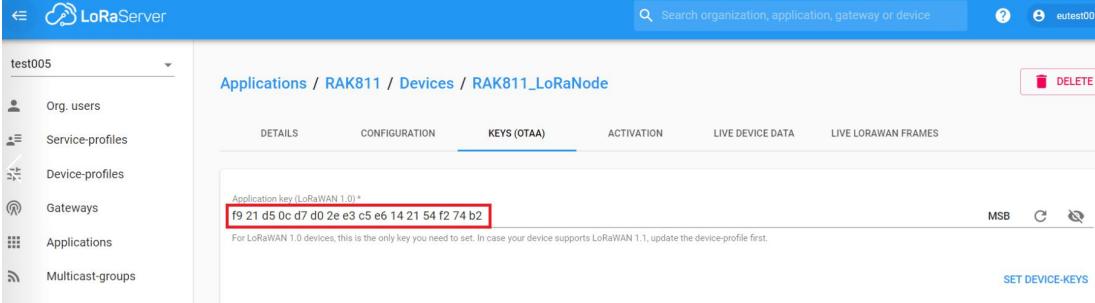
“SET DEVICE-KEYS”. That’s OK! You’ve complete the configuration on LoRaServer.

As you see, the Device EUI which will be set into WisNode as “dev_eui” is this one:



The screenshot shows the LoRaServer interface for managing devices under the 'RAK811' application. The 'DEVICES' tab is selected. A red box highlights the 'Device EUI' column for the single listed device, which is '5e9d1e0857cf25f1'.

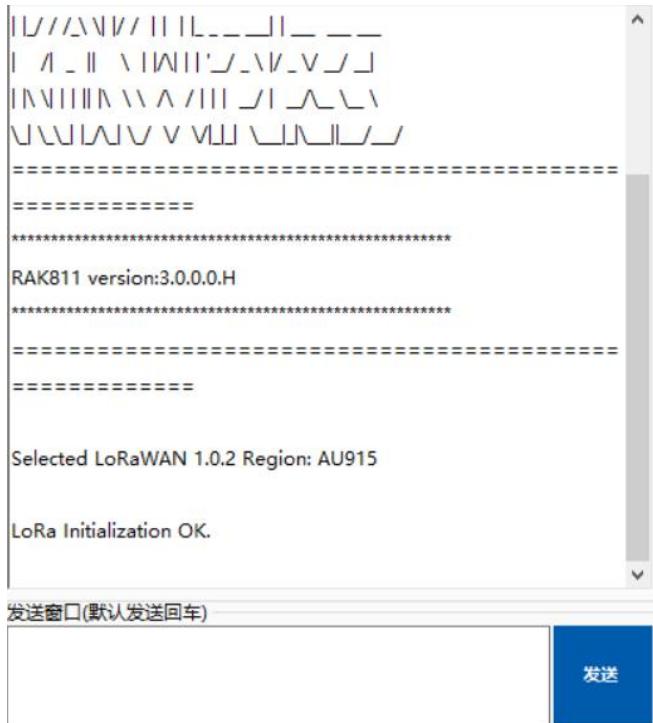
The Application Key which will be set into WisNode as “app_key” is this one:



The Application EUI which will be set into WisNode as “app_eui” is useless for LoRaServer, and you can set it to any value with a correct format, even if you can set a value as same as the dev_eui.

Next, let's configure WisNode by using AT command.

Connect your WisNode with a PC, power on it and open RAK Serial Port Tool on your PC.



The default LoRa work mode is LoRaWAN 1.0.2, and the default join mode is OTAA.

If the join mode is not OTAA now, just set the LoRa join mode to OTAA as follow:

```

=====
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK

发送窗口(默认发送回车)
at+set_config=lorajoin_mode:0

```

发送

Set the LoRa class to Class A:

```

=====
=====
*****
RAK811 version:3.0.0.0.H
*****
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK
>>at+set_config=lora:class:0
OK

发送窗口(默认发送回车)
at+set_config=lora:class:0

```

发送

Set the frequency/region to EU868:

```
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868

Selected LoRaWAN 1.0.2 Region: EU868
Band switch success.
OK
```

发送窗口(默认发送回车)
at+set_config=lora:region:EU868 发送

Set “dev_eui”:

```
=====
=====
Selected LoRaWAN 1.0.2 Region: AU915

LoRa Initialization OK.
>>at+set_config=lorajoin_mode:0
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868

Selected LoRaWAN 1.0.2 Region: EU868
Band switch success.
OK
>>at+set_config=lora:dev_eui:5e9d1e0857cf25f1
OK
```

发送窗口(默认发送回车)
at+set_config=lora:dev_eui:5e9d1e0857cf25f1 发送

Set “app_eui”:

```
LoRa Initialization OK.  
>>at+set_config=lorajoin_mode:0  
OK  
>>at+set_config=lora:class:0  
OK  
>>at+set_config=lora:region:EU868  
  
Selected LoRaWAN 1.0.2 Region: EU868  
Band switch success.  
OK  
>>at+set_config=lora:dev_eui:5e9d1e0857cf25f1  
OK  
>>at+set_config=lora:app_eui:5e9d1e0857cf25f1  
OK
```

发送窗口(默认发送回车)
at+set_config=lora:app_eui:5e9d1e0857cf25f1 发送

Set “app_key”:

```
OK  
>>at+set_config=lora:class:0  
OK  
>>at+set_config=lora:region:EU868  
  
Selected LoRaWAN 1.0.2 Region: EU868  
Band switch success.  
OK  
>>at+set_config=lora:dev_eui:5e9d1e0857cf25f1  
OK  
>>at+set_config=lora:app_eui:5e9d1e0857cf25f1  
OK  
>>  
at+set_config=lora:app_key:f921d50cd7d02ee3c5e6142154f274b2  
OK
```

发送窗口(默认发送回车)
at+set_config=lora:app_key:f921d50cd7d02ee3c5e6142154f274b2 发送

Start to join:

```

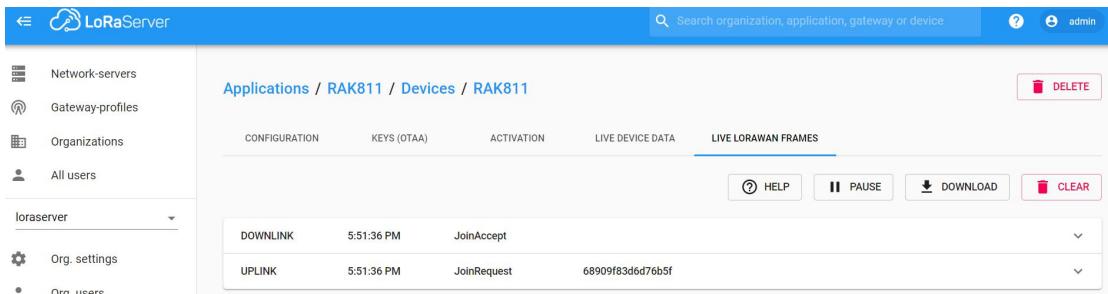
LoRa Initialization OK.
>>at+set_config=lora:dev_eui:68909f83d6d76b5f
OK
>>at+set_config=lora:app_eui:68909f83d6d76b5f
OK
>>
at+set_config=lora:app_key:14179eea4f5250c1324389943020fc9d
OK
OTAA:
DevEui:68909F83D6D76B5F
AppEui:68909F83D6D76B5F
AppKey:14179EEA4F5250C1324389943020FC9D
OTAA Join Start...
OK
[LoRa]:Joined Successed!

```

发送窗口(默认发送回车)

Join successfully!

You can see the JoinRequest and JoinAccept on LoRaServer page:



DOWNLINK	5:51:36 PM	JoinAccept
UPLINK	5:51:36 PM	JoinRequest 68909f83d6d76b5f

Let's try to send a data from WisNode to LoRaServer:

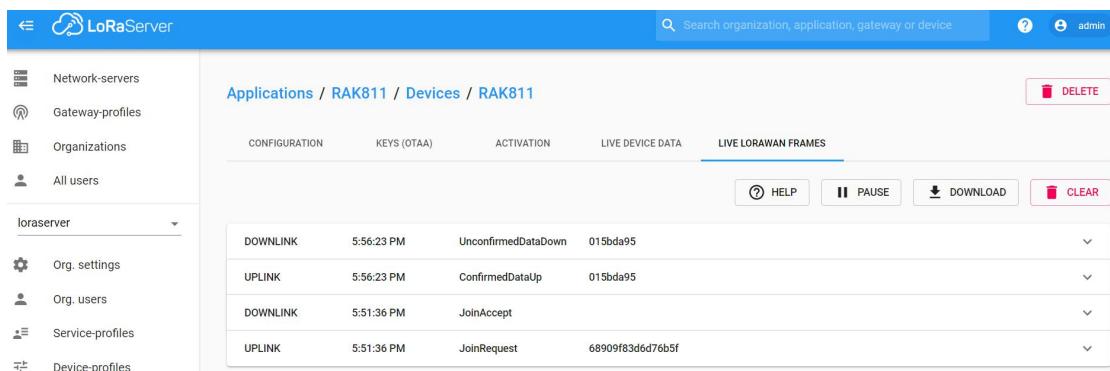
```

OK
>>
at+set_config=lora:app_key:14179eea4f5250c1324389943020fc9d
OK
OTAA:
DevEui:68909F83D6D76B5F
AppEui:68909F83D6D76B5F
AppKey:14179EEA4F5250C1324389943020FC9D
OTAA Join Start...
OK
[LoRa]:Joined Successed!
>>at+set_config=lora:confirm:1
OK
OK
[LoRa]: Confirm data send OK

```

发送窗口(默认发送回车)
at+send=lora:2:1234567890 发送

You can see the message on LoRaServer page:

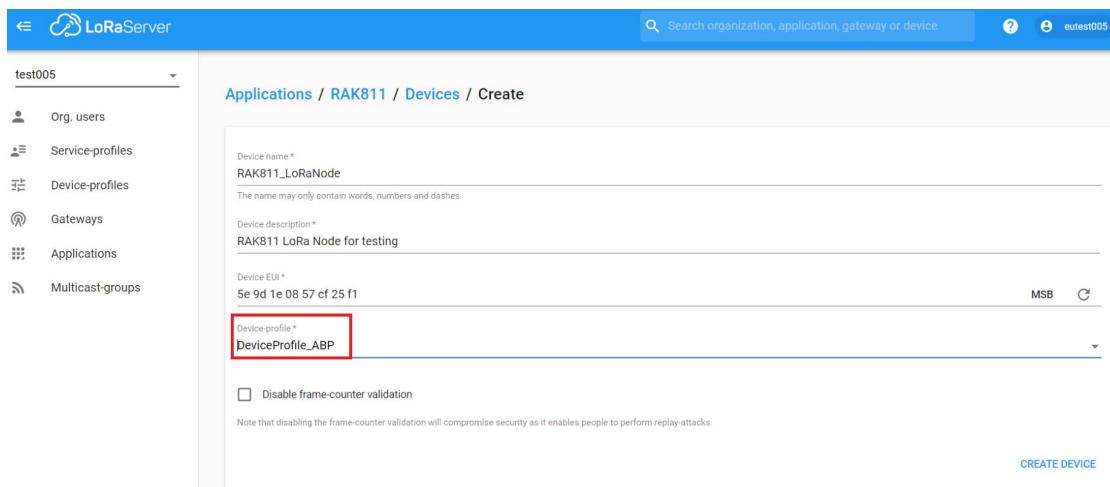


LINK	TIME	MESSAGE	DATA
DOWNLINK	5:56:23 PM	UnconfirmedDataDown	015bda95
UPLINK	5:56:23 PM	ConfirmedDataUp	015bda95
DOWNLINK	5:51:36 PM	JoinAccept	
UPLINK	5:51:36 PM	JoinRequest	68909f83d6d76b5f

OK, that's all about “Join in OTAA mode” with LoRaServer.

6.2 Join in ABP mode

If you select “DeviceProfile_ABP” or “DeviceProfile_ABP_CN470”, it means you want to join LoRaServer in OTAA mode.



Device name *
RAK811_LoRaNode
The name may only contain words, numbers and dashes.

Device description *
RAK811 LoRa Node for testing

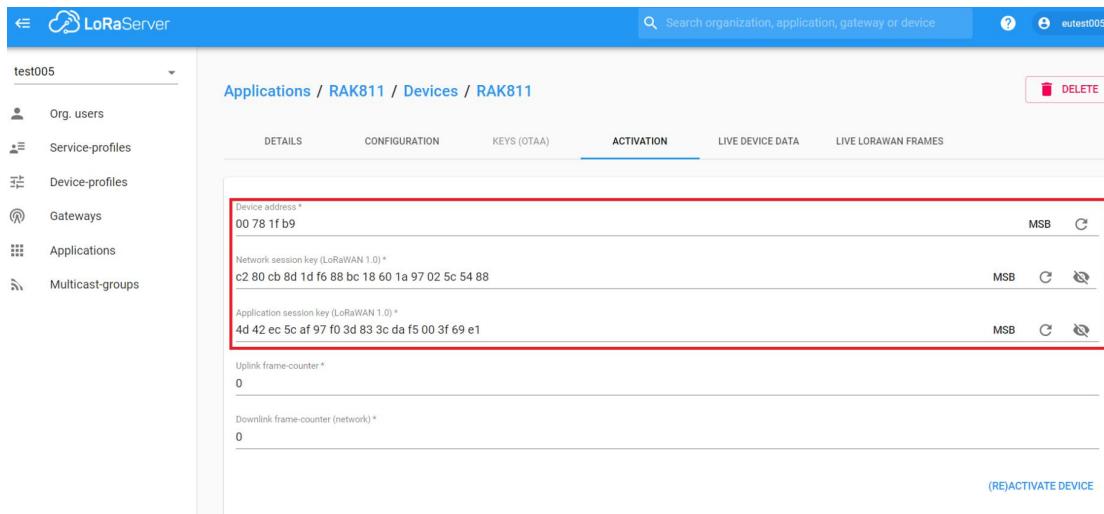
Device EUI *
5e 9d 1e 08 57 cf 25 f1 MSB

Device-profile *
DeviceProfile_ABP

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

CREATE DEVICE

Then you can see that there are some parameters for ABP in the “ACTIVATION” item:



Next, let's use these parameters to set WisNode by using AT command.

Set LoRa join mode to ABP:



```
>>at+set_config=lorajoin_mode:1
OK
```

Set LoRa class to Class A:

```
>>at+set_config=lora:join_mode:1  
OK  
>>at+set_config=lora:class:0  
OK
```

发送窗口(默认发送回车)

at+set_config=lora:class:0

发送

Set the frequency/region to EU868:

```
>>at+set_config=lora:join_mode:1  
OK  
>>at+set_config=lora:class:0  
OK  
>>at+set_config=lora:region:EU868  
No switch region.Current region:EU868  
OK
```

发送窗口(默认发送回车)

at+set_config=lora:region:EU868

发送

Set “dev_addr”:

```
>>at+set_config=lora:join_mode:1
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_addr:00781fb9
OK
```

发送窗口(默认发送回车)
at+set_config=lora:dev_addr:00781fb9

发送

Set “nwks_key”:

```
>>at+set_config=lora:join_mode:1
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_addr:00781fb9
OK
>>
at+set_config=lora:apps_key:c280cb8d1df688bc18601a97025c5488
OK
```

发送窗口(默认发送回车)
at+set_config=lora:apps_key:c280cb8d1df688bc18601a97025c5488

发送

Set “apps_key”:

```

>>at+set_config=lora:join_mode:1
OK
>>at+set_config=lora:class:0
OK
>>at+set_config=lora:region:EU868
No switch region.Current region:EU868
OK
>>at+set_config=lora:dev_addr:00781fb9
OK
>>
at+set_config=lora:apps_key:c280cb8d1df688bc18601a97025c5488
OK
>>at+set_config=lora:nwks_key:4d42ec5caf97f03d833cdaf5003f69e1
OK
    
```

发送窗口(默认发送回车)

at +set_config=lora:nwks_key:4d42ec5caf97f03d833cdaf5003f69e1	发送
--	-----------

Start to join:

```

OK
>>
at+set_config=lora:apps_key:c280cb8d1df688bc18601a97025c548
8
OK
>>
at+set_config=lora:nwks_key:4d42ec5caf97f03d833cdaf5003f69e1
OK
Join Network first and try again.
ABP:
DevEui: 5E9D1E0857CF25F1
DevAddr: 00781FB9
AppsKey: C280CB8D1DF688BC18601A97025C5488
NwksKey: 4D42EC5CAF97F03D833CDAF5003F69E1
OK
    
```

发送窗口(默认发送回车)

at+join	发送
---------	-----------

Note: Actually, it is no need to join in ABP mode. But you still need to set this AT command to valid the parameters which you just set for ABP mode.

Try to send a message:



You can see the data which is just sent from WisNode on LoRaServer page:

DOWNLINK	6:12:12 PM	UnconfirmedDataDown	0178cae9
UPLINK	6:12:11 PM	ConfirmedDataUp	0178cae9

That's all about “Join in ABP mode” with LoRaServer.

7. How to use LoRaP2P mode?

In this document, i'll show you how to use LoRaP2P mode on EU868 for example, other frequency is similar.

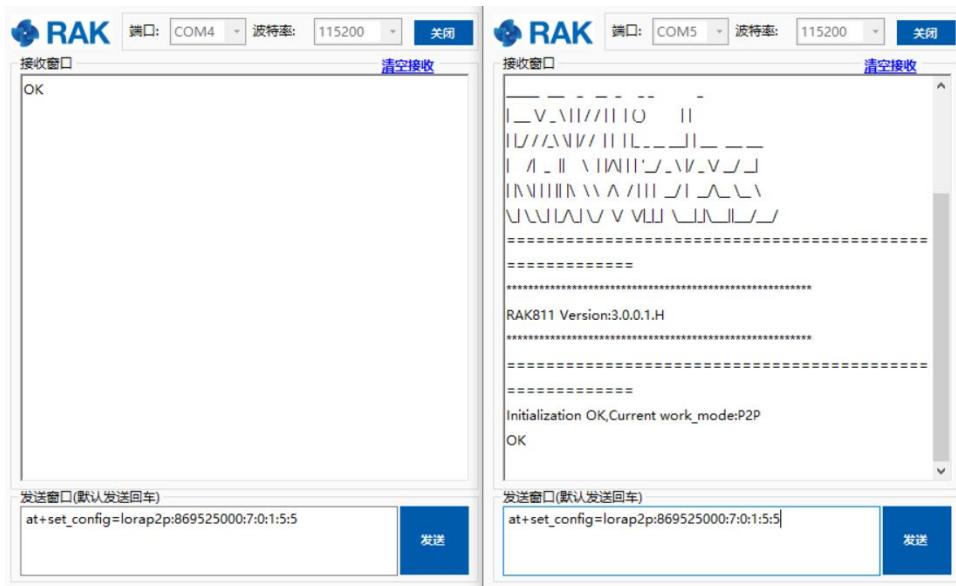
First, find two RAK811 WisNode which can work on EU868 frequency and make sure their firmware version isn't less than V3.0.0.1.

Second, connect these two RAK811 WisNode with PC through UART, and open two serial port tool on PC.

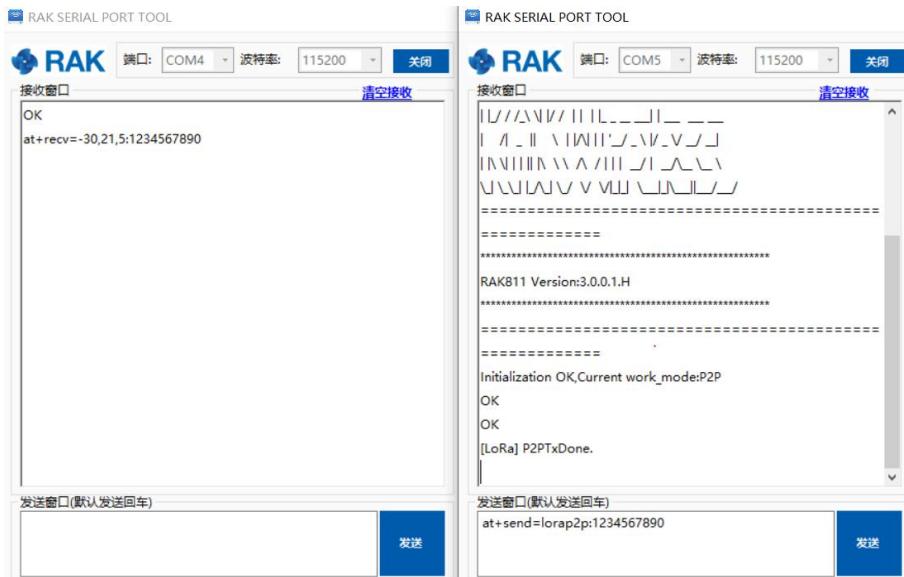
Now, configure them to both work in LoRaP2P mode as follow:



Then configure LoRaP2P parameters for both of them as follow for example:

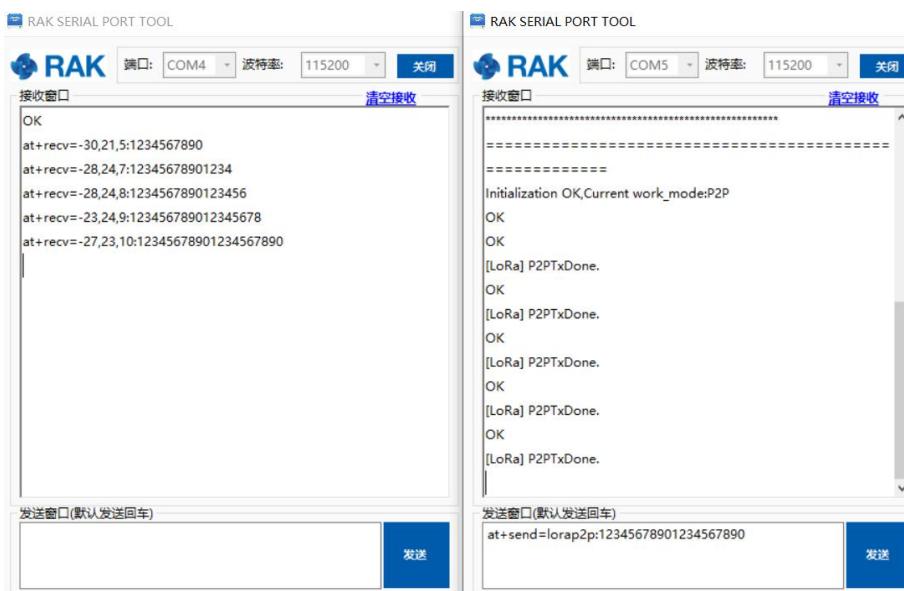


OK! Try to send a message from WisNode 2 (the right one) to WisNode 1 (the left one):



Successfully!

Send more messages:



Successfully too!

Similarly, you can send message from WisNode 1 to WisNode 2 surely. Just try it freely.

Great! We've done it, and that's all about how to use LoRaP2P on RAK811 WisNode.

Author	Date	Version
Fomi	2019.8.23	V3.0.0.4