

Quick Start Guide

Prerequisites

What do you need?

Before going through each and every step in the installation guide of the RAK7200 WisNode Track Lite, make sure to prepare the necessary items listed below:

Hardware Tools

1. RAK7200 WisNode Track Lite
2. Micro USB Cable
3. Gateway in Range, for Testing
4. Windows PC

Software Tools

1. [STM32CubeProgrammer](#)
2. [RAK Serial Port Tool](#)

List of Acronyms

Acronyms	Definition
BLE	Bluetooth Low Energy
DFU	Device Firmware Upgrade
LoRa	Long Range
EUI	Electronic Unique Identifier
TTN	The Things Network

What's Included in the Package

1. RAK7200 WisNode Track Lite
2. Rubberized enclosure
3. Micro USB cable

Product Configuration

Interfacing with RAK7200

To interface with the RAK7200 WisNode Track Lite with your Windows Machine, you need to have a serial tool. Any serial tool will work, but it is recommended to use the [RAK Serial Port Tool](#).

Connecting to the RAK7200

Once done, follow the listed steps to interface your RAK7200 with your computer.

1. Connect the RAK7200 to the USB port of a general-purpose computer using a standard Micro - USB Cable.



Figure 1: RAK7200 Module Connection

NOTE

If this is your first time to connect your RAK7200 WisNode Track Lite to your computer, it should automatically download the CH340 driver in order for them to communicate properly. Make sure to have an internet access if you want such automatic installation to be successful. If such process fails, re-plug your Micro - USB cord and proceed to the next step.

2. Go to your **Device Manager** by pressing: **Windows + R** and type `devmgmt.msc`. Then, look for "**Other Devices**".

You can also use the following options:

- **Search in Start Menu**
- Right click "My Computer" or "This PC" and click **Manage**.

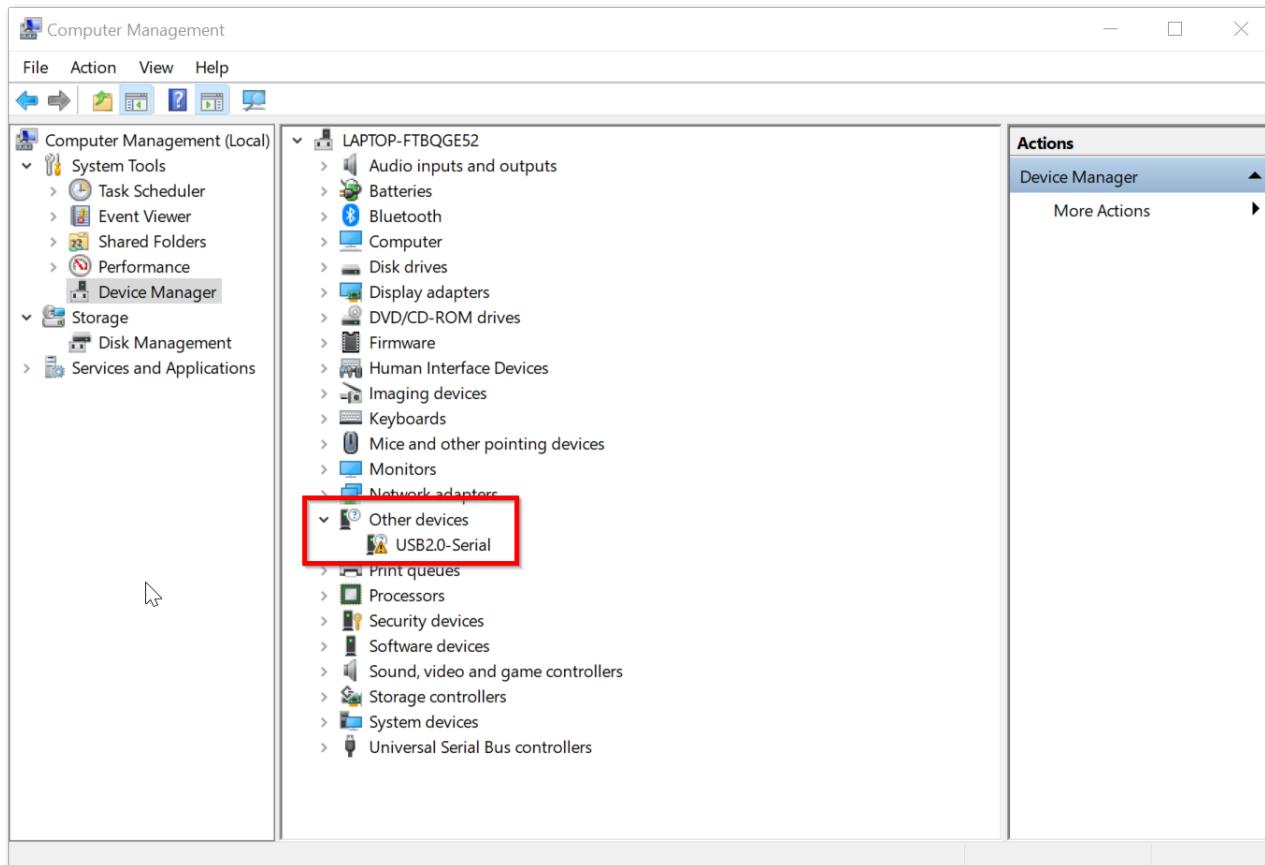


Figure 2: Missing Driver for the RAK811 LPWAN Evaluation Board

3. Under "**Other devices**" drop down list, right click the unknown **USB2.0-Serial** driver and choose "**Search automatically for updated driver software**". Again, before doing so, make sure to have an internet access or it will fail.

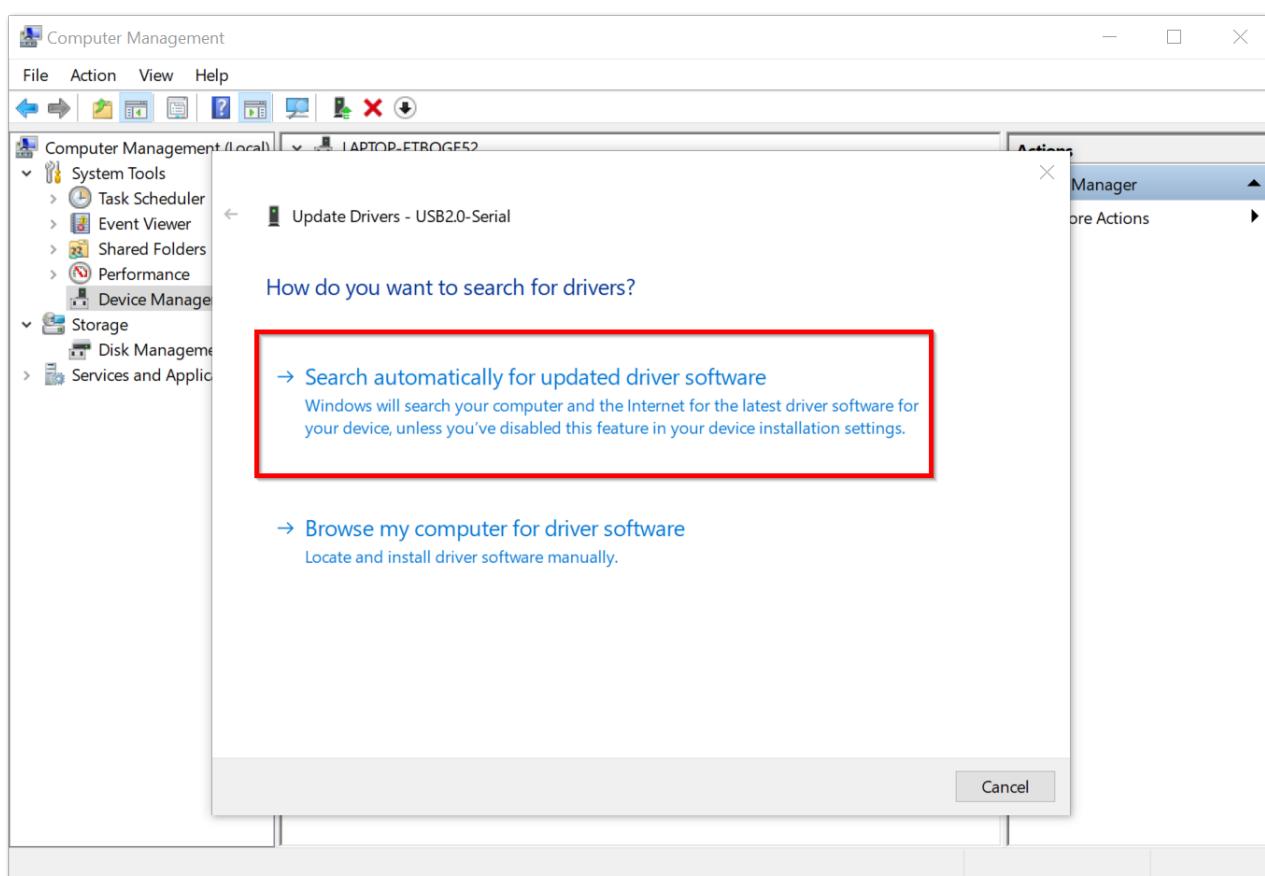


Figure 3: Automatic Driver Installation via Internet

4. Wait for it to automatically download and install the missing driver. Once installation is done, "USB-SERIAL CH340" must appear in the Ports (COM & LPT) drop down list. Take note of the COM Port associated with the driver as it will be used in the succeeding steps. For this sample process, the COM Port used by the USB-SERIAL CH340 driver is COM4.

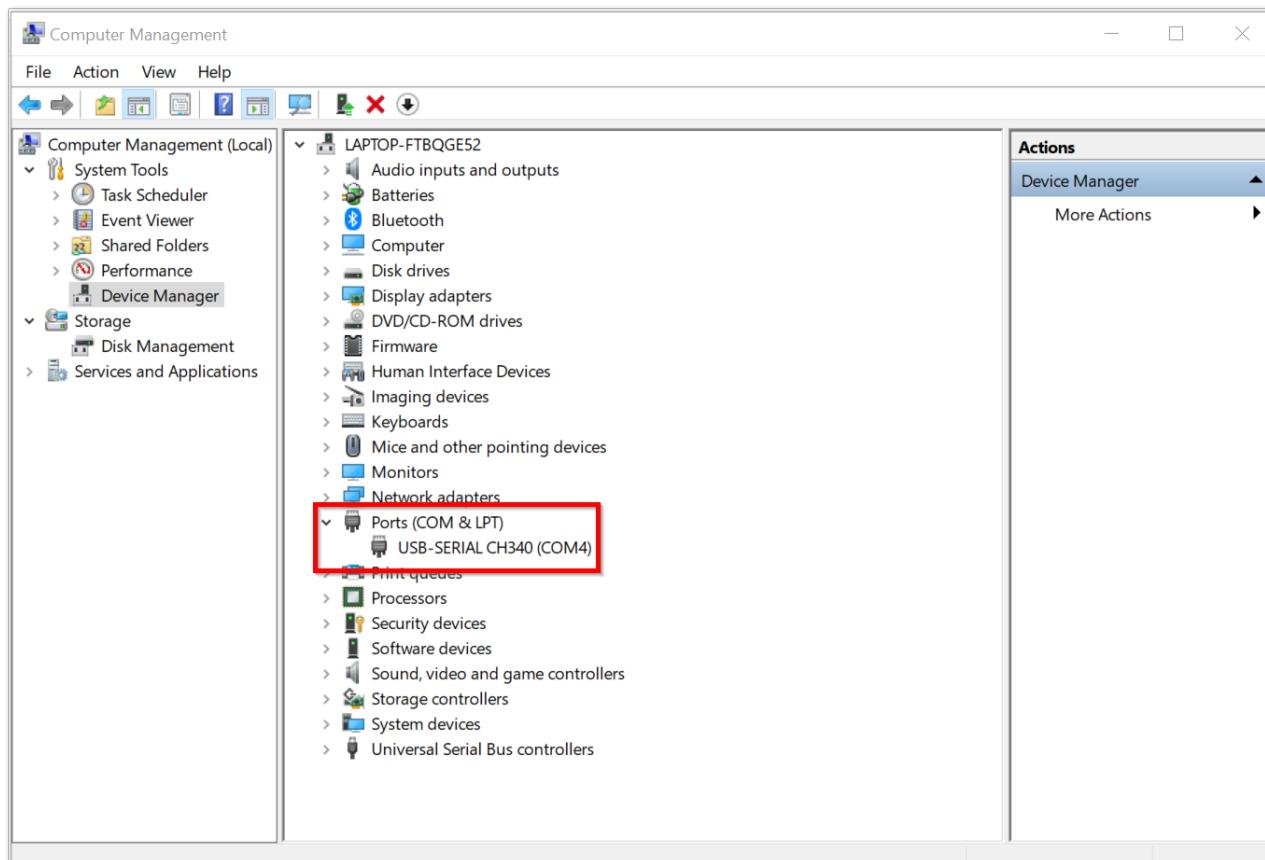


Figure 4: USB-SERIAL CH340 Driver Successfully Installed

NOTE

In case the driver is still not installed upon doing the previous steps, download the driver manually through this [link](#).

5. Test if your RAK7200 WisNode Track Lite can now communicate with the RAK Serial Port Tool. Configure the serial communication tool by selecting the proper port and configure the link as listed below and then click "Open".

- **COM:** Choose the COM Port associated with the USB-SERIAL CH340 from the previous step. For this tutorial, the COM Port is COM4.
- Baud Rate: **115200 bps**
- Data Bits: **8 bits**
- Stop Bits: **1 bit**
- Parity: **NONE**

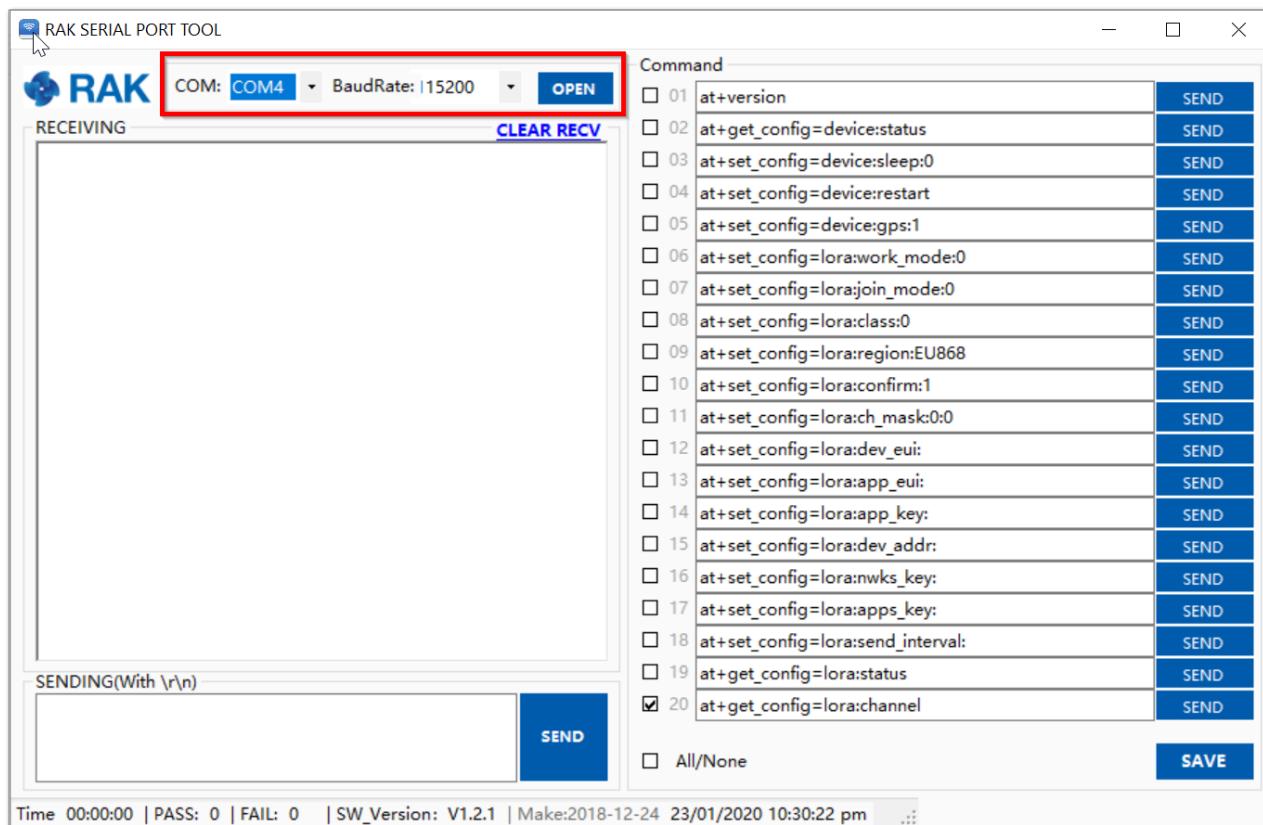


Figure 5: Connecting to the RAK Serial Port Tool

6. Verify the connectivity by sending an AT Command. In the RAK Serial Port Tool, there are built-in AT+Commands within it at the right side. Send the following command to check the firmware version of your RAK7200 WisNode Track Lite. If connection is successful, the firmware version should appear same with Figure 6 below:

```
at+version
```

sh

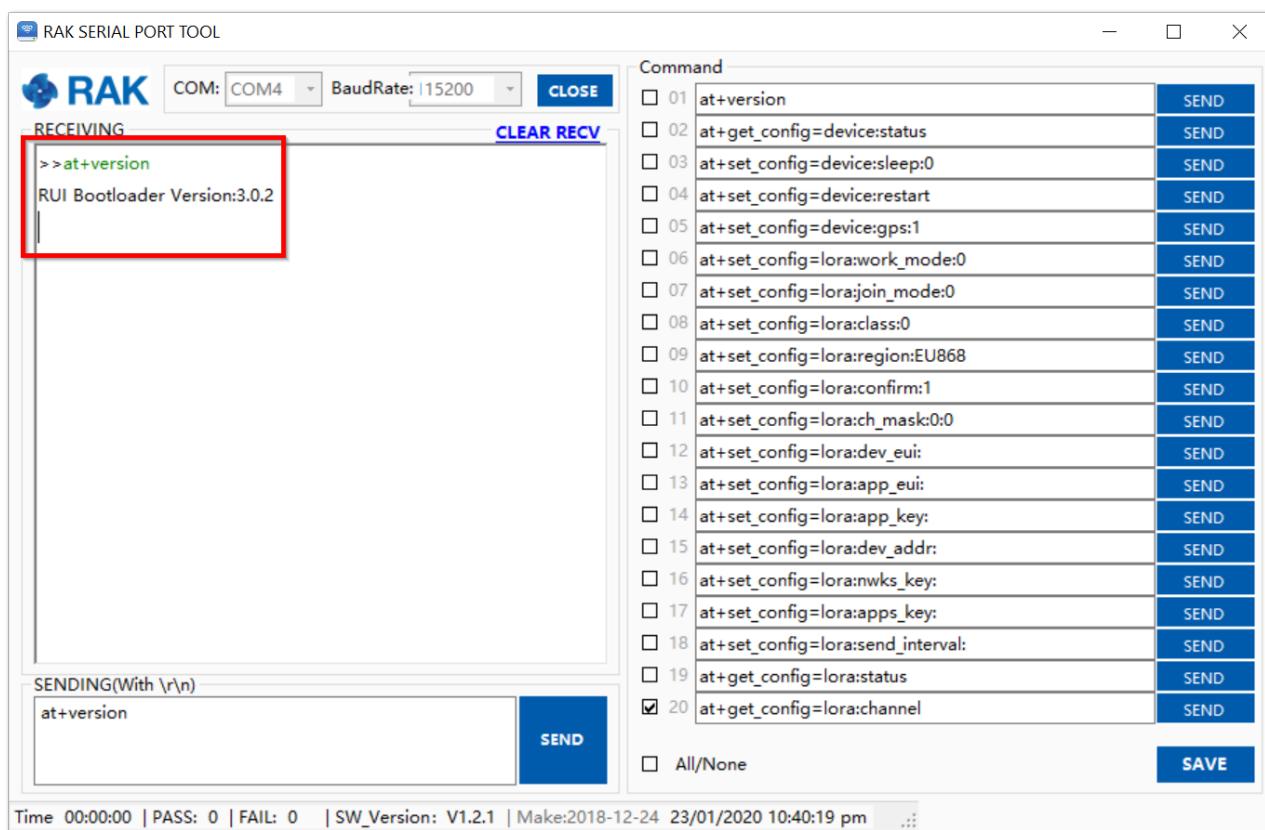


Figure 6: AT+Command Sample Serial Communication Test

Connecting to The Things Network (TTN)

In this section, a practical exercise will be performed to show how to connect the RAK7200 module to The Thing Network (TTN™) platform.

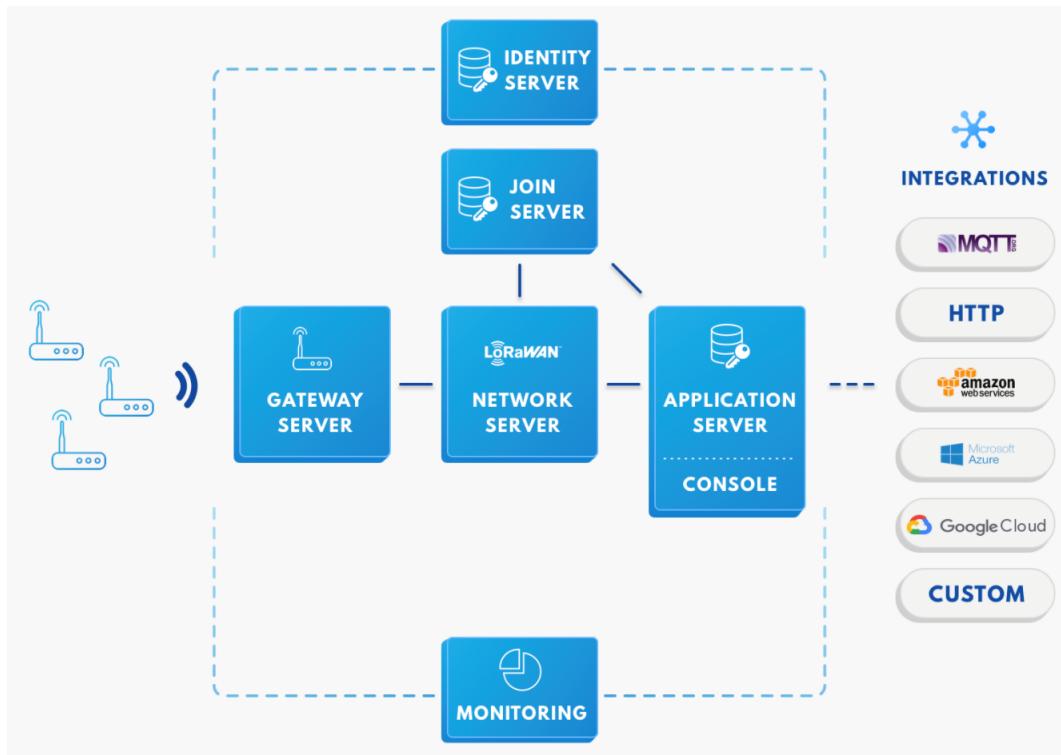


Figure 7: RAK7200 in the context of the TTN

As shown in Figure 7, the RAK7200 module is one of the devices located on the left side. In the context of an IoT solution, the objective is to deploy devices to sense the relevant process variables and transmit the data to the backend servers located in the cloud. The data will be processed and integrated as part of a larger solution that, ultimately, could generate efficiency, traceability, predictability capacity among others.

The RAK7200 module can be part of this ecosystem, and the objective of this section is to demonstrate how simple is to send data to the TTN using the LoRaWAN protocol. To achieve this, the RAK7200 module must be located inside of the coverage of a LoRaWAN gateway.

Sign up and login

If you don't have an account yet, head on to the [TTN website](#) and create one. Once done, login to your account and go to the Console.

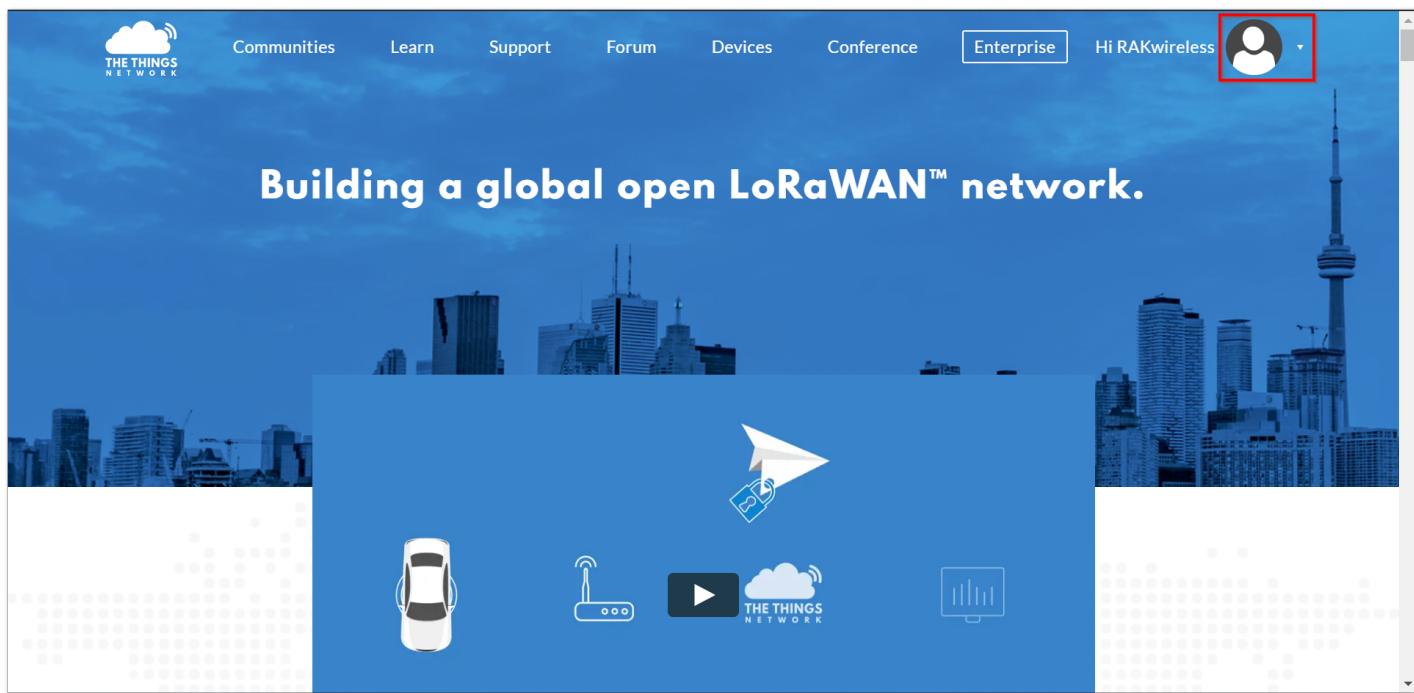


Figure 8: The Things Network Home Page

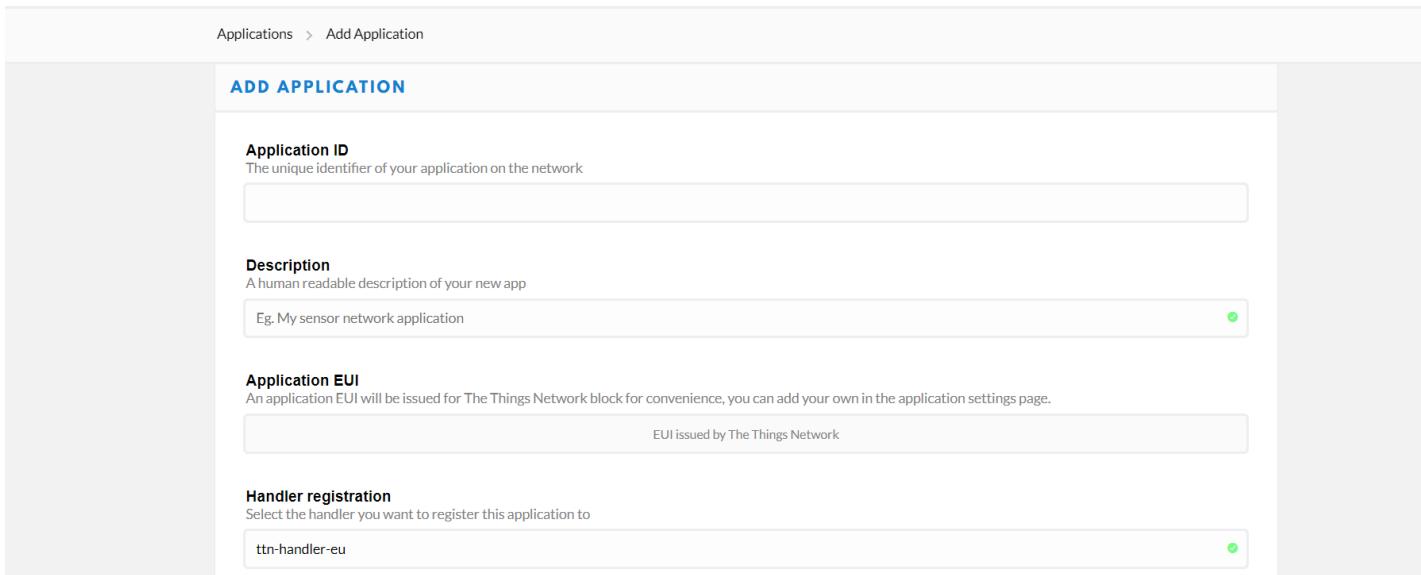
Figure 9: TTN Console Page

Create a New Application

1. Choose the **APPLICATIONS**.

Figure 10: Application Section

2. Click the "add application" button.



ADD APPLICATION

Application ID
The unique identifier of your application on the network

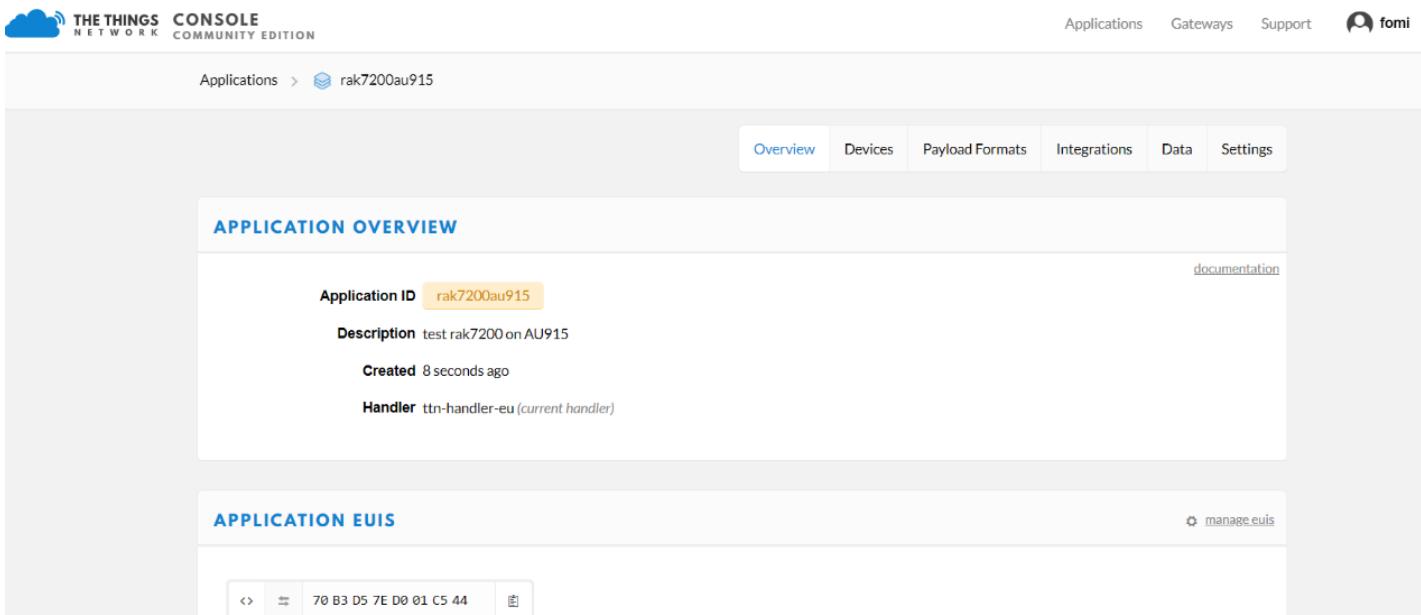
Description
A human readable description of your new app
E.g. My sensor network application

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

Handler registration
Select the handler you want to register this application to
ttn-handler-eu

Figure 11: Adding an Application

- Here are the things that you should take note in adding an application:
 - **Application ID** - this will be the unique ID of your application in the Network. Note that the characters should be in lower case, no spaces are allowed.
 - **Description** - this is a short and concise human readable description of your application.
 - **Application EUI** - this will be generated automatically by The Things Network for convenience.
 - **Handler Registration** - handler you want to register this application to.
3. After you fill in the necessary information, press the "Add application" button at the bottom of this page. If you see the same page as shown in Figure 12, this means that you have successfully registered your application.



APPLICATION OVERVIEW

Application ID rak7200au915 [documentation](#)

Description test rak7200 on AU915

Created 8 seconds ago

Handler ttn-handler-eu (*current handler*)

APPLICATION EUIS [manage euis](#)

<>	≡	70 B3 D5 7E D0 01 C5 44	☰
----	---	-------------------------	---

Figure 12: Application Overview

Register a New Device

1. Scroll down until you see the Devices section. Or, you can click the "Devices" button at the top.

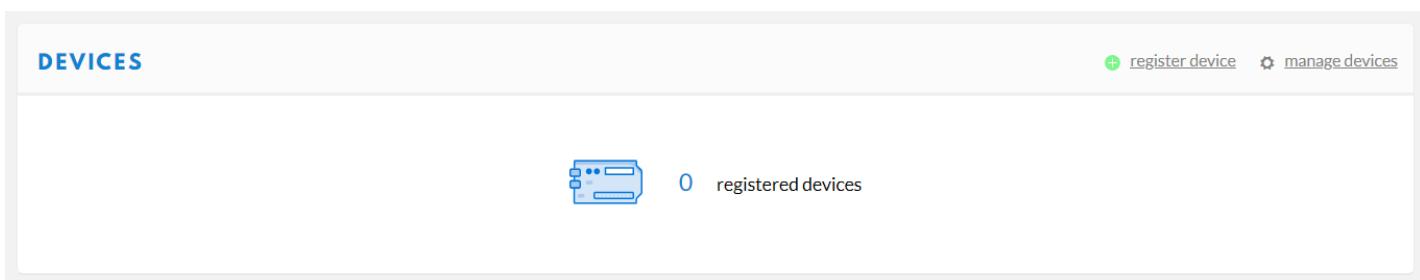
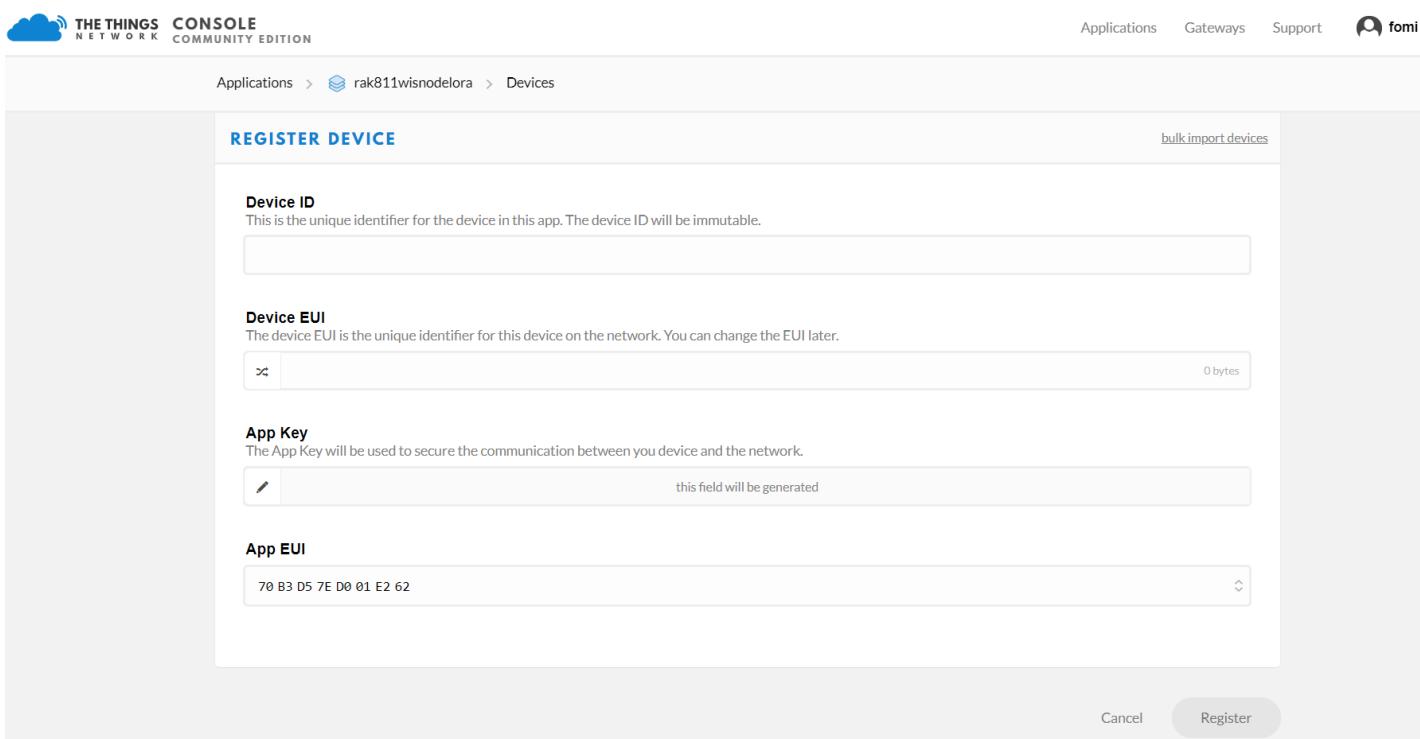


Figure 13: Register a New Device

2. Then, register a new device by clicking on the "register devices".



The screenshot shows the 'REGISTER DEVICE' page of the TTN console. At the top, there's a navigation bar with 'Applications', 'Gateways', 'Support', and a 'fomi' logo. Below the navigation, the path 'Applications > rak811wisnodelor > Devices' is shown. The main form has sections for 'Device ID', 'Device EUI', 'App Key', and 'App EUI'. Each section contains a description and an input field. At the bottom right of the form are 'Cancel' and 'Register' buttons.

Figure 14: Add your Device

In this form, the device ID must be unique for the application and must be completed with a lower case, alphanumeric characters. The rest of the parameters in the form are very important for the LoRaWAN protocol:

- Device EUI
- Application Key
- Application EUI

The TTN platform can generate these parameters randomly by leaving those fields empty. Or, you can enter already existing values.

3. Press the "Register" button at the bottom of this page to finish the process.

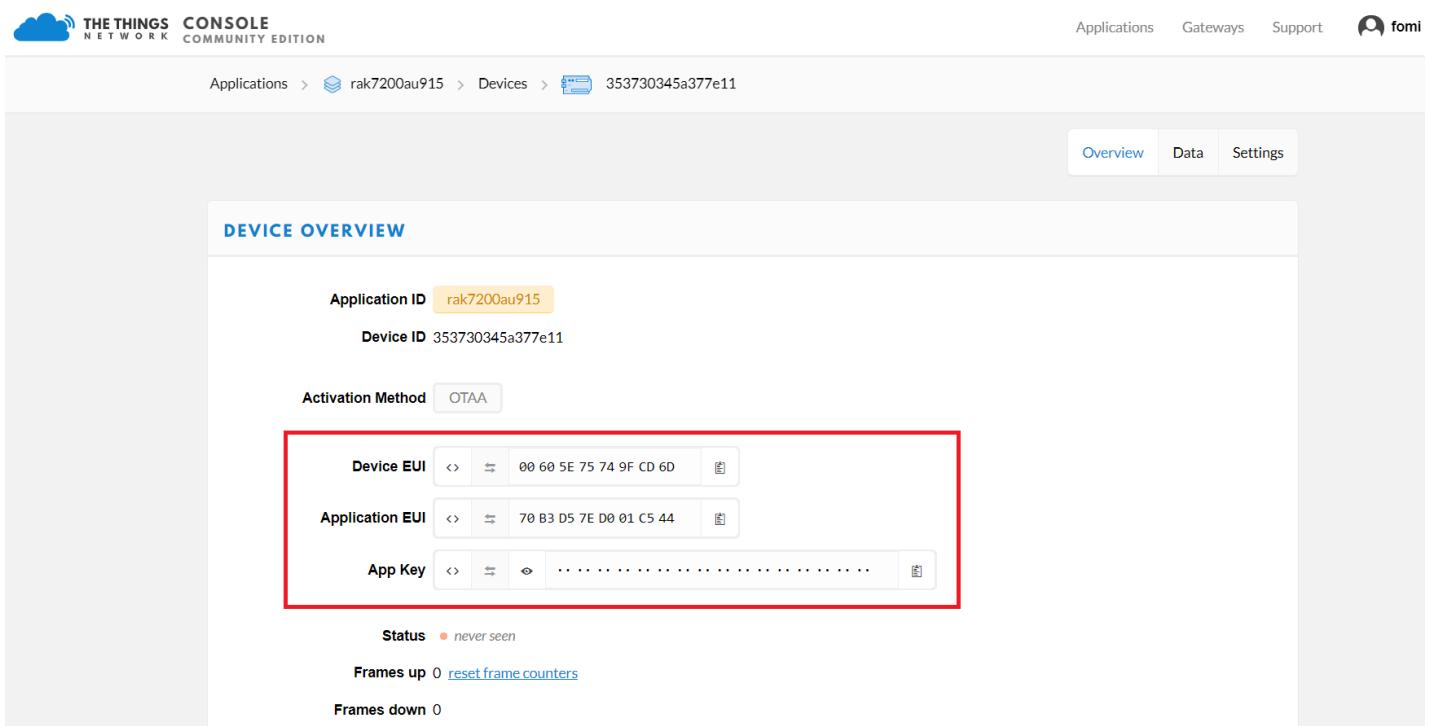


Figure 15: Device Overview

LoRaWAN Join Mode

The LoRaWAN specification defines that to join in a LoRaWAN network, each end-device has to be personalized and activated. Activation can be done either via Over-The-Air-Activation (OTAA) or Activation-By-Personalization (ABP). In OTAA, the previously personalized end-device is activated when it is deployed or reset. On the other hand, in ABP, the personalization and activation are done as a single step.

Hence, this is referred to as the "**Join Mode**". LoRaWAN allows the OTAA mode and the ABP mode. In this section, the configuration process of these two modes is explained, both on the platform side and the node side.

OTAA Mode

Configure the OTAA Mode on the Platform

As shown in the Figure 15, the default activation mode in TTN is the OTAA mode. Therefore, no further actions are required in the platform side.

To set up the RAK7200 module to join the TTN using OTAA, start by connecting the RAK7200 module to the Computer (see Figure 1) and open the RAK Serial Port Tool. Wait for the communication to start. It is recommended to test the serial communication and verify the current configuration by sending either of these two AT commands:

```
at+get_config=lora:status
```

```
at+version
```

By default, the parameters configured in the RAK7200 modules are:

- LoRa join mode: **OTTA**
- LoRa class: **Class A**
- LoRa region: **AU915**
- Device EUI: **00605E75749FCD6D**
- Application EUI: **00605E75749FCD6D**

- Application Key: 4E2003296FC5CD26F46940A6DAFA9D1

1. Set the LoRa join mode to OTAA.

```
at+set_config=lora:join_mode:0
```

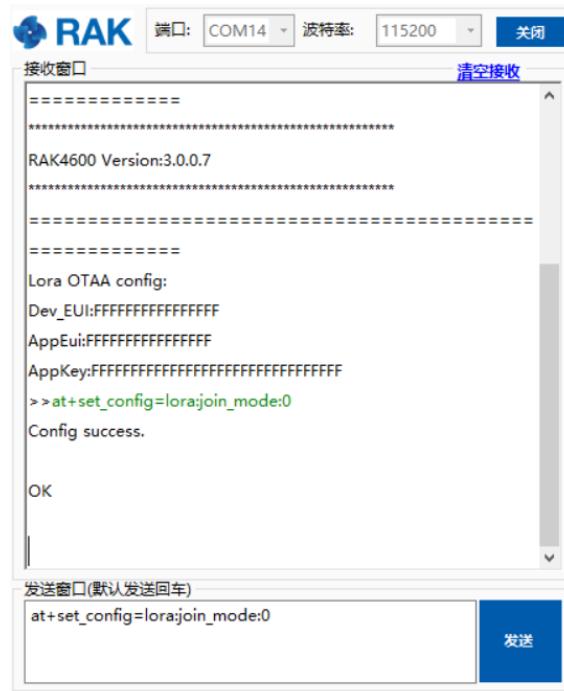


Figure 16: Setting LoRa Join Mode to OTAA Mode

2. Set the LoRa class to Class A.

```
at+set_config=lora:class:0
```

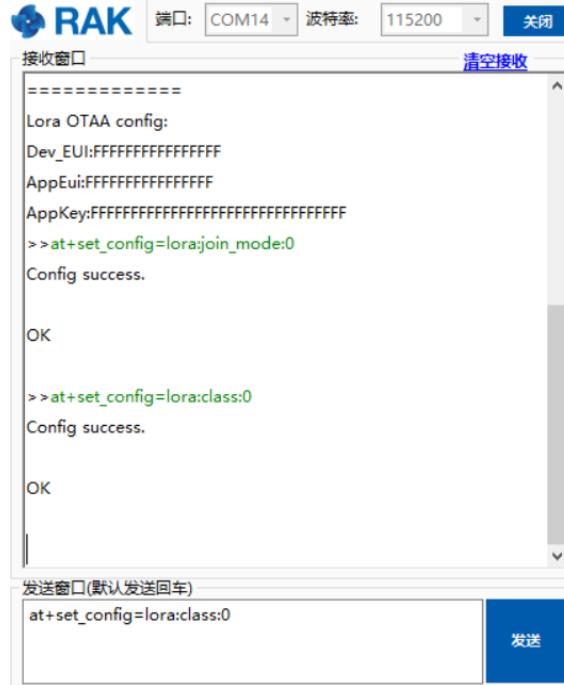


Figure 17: Setting LoRa Class Parameter

3. Set the frequency/region to AU915.

- Refer in the [RAK7200 Datasheet](#) for the list of supported frequencies.

```
at+set_config=lora:region:AU915
```



Figure 18: Setting LoRa Frequency/Region Parameter

4. Set the Device EUI.

```
at+set_config=lora:dev_eui:00605E75749FCD6D
```

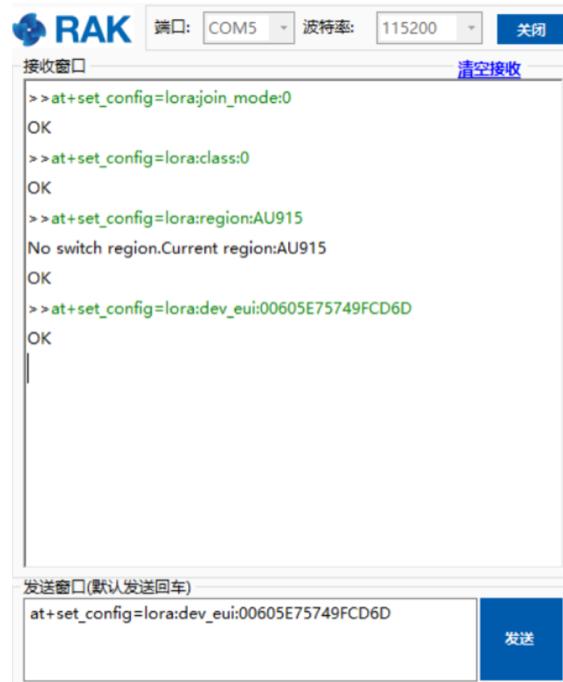


Figure 19: Setting LoRa Device EUI Parameter

5. Set the Application EUI.

```
at+set_config=lora:app_eui:00605E75749FCD6D
```

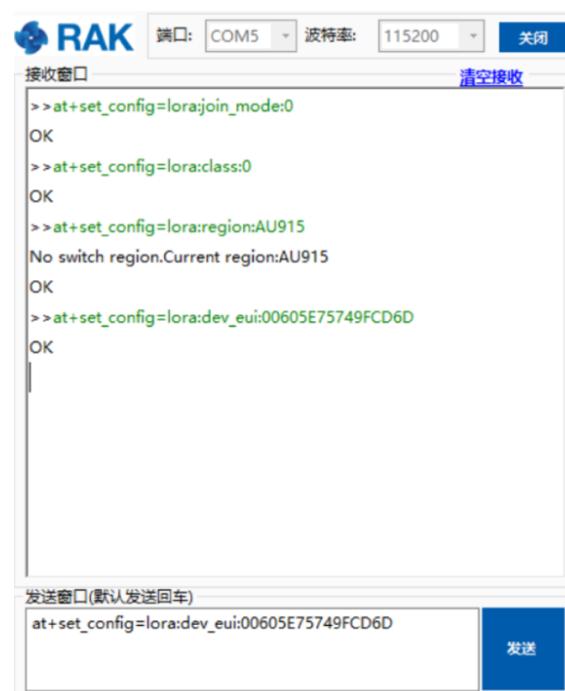


Figure 20: Setting LoRa Application EUI Parameter

 NOTE:

The Device EUI parameter is not needed for the ChirpStack platform, therefore you will use the same ID as the Device EUI. Otherwise, the firmware will compliant.

6. Set the Application Key.

```
at+set_config=lora:app_key:4E2003296FC5CD26F46940A6DAFA9D1
```



Figure 21: Setting LoRa Application Key Parameter

 NOTE:

After configuring all the parameters, you need to reset your RAK7200 Module for saving parameters.

7. After resetting, join in OTAA mode.

```
at+join
```

After 5 or 6 seconds, if the request was successfully received by a LoRa gateway, then you should see the messages shown in Figure 22.

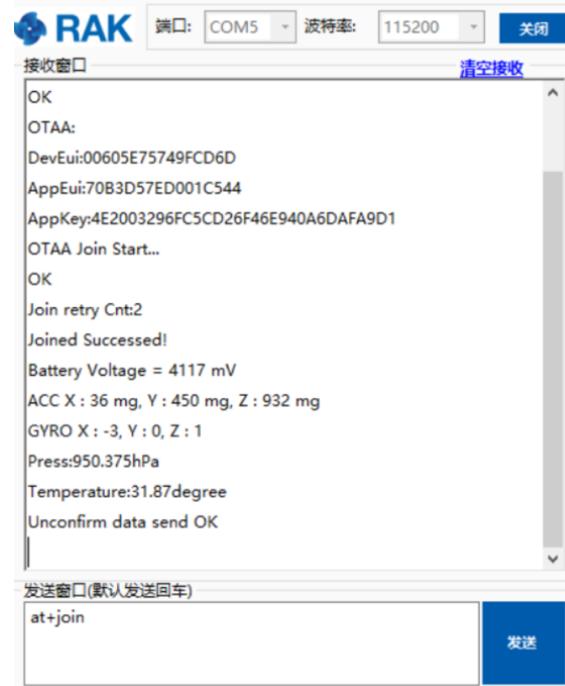


Figure 22: Join into a LoRaWAN Network

Once joined into the LoRaWAN network, the RAK7200 firmware will start to send the data periodically. As shown in Figure 23, data will be saved in the TTN platform.

Figure 23: OTAA Test Sample Data Sent Viewed in TTN

ABP Mode

Configure the ABP mode on the Platform

If the ABP mode is preferred, then the TTN platform needs to be configured first. At TTN, once a device is created, you can select the “SETTINGS” section of the “DEVICE” and switch the “Activation Method”.

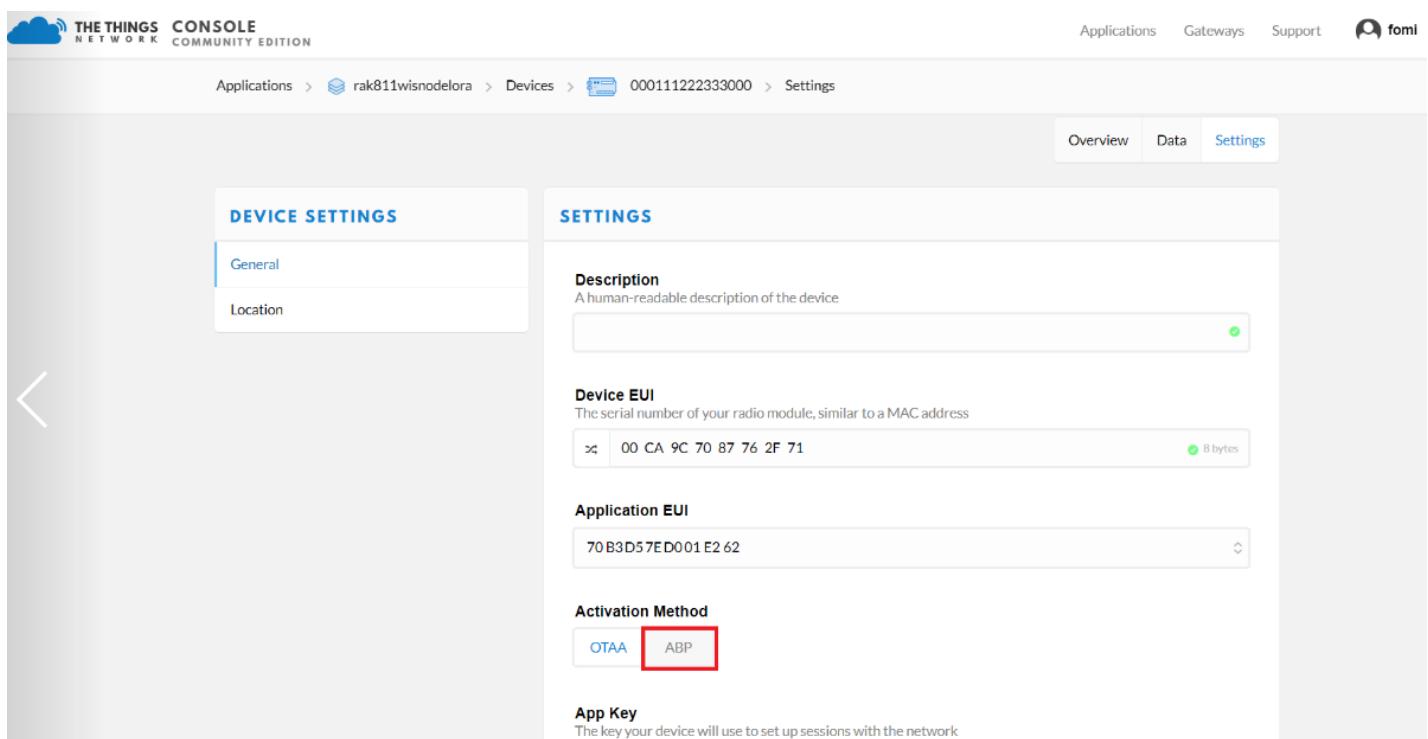


Figure 24: ABP Activation in TTN

- For ABP mode, the TTN parameters needed are the following:
 - Device Address
 - Network Session Key
 - App Session Key

 **NOTE:**

These fields can be left empty in the form. TTN will automatically generate random values. In other cases, you can fill them with specific values.

The screenshot shows the 'ABP Mode Parameters' form in the TTN Documentation Center. At the top, there's a navigation bar with 'THE THINGS NETWORK CONSOLE COMMUNITY EDITION', 'Applications', 'Gateways', 'Support', and a 'fomi' logo. Below the navigation, the path is 'Applications > rak811wisnodelora > Devices > 000111222333000 > Settings'. The form has several input fields:

- Device EUI:** The serial number of your radio module, similar to a MAC address. Value: '00 CA 9C 70 87 76 2F 71'.
- Application EUI:** Value: '70 B3 D5 7E D0 01 E2 62'.
- Activation Method:** Radio buttons for 'OTAA' and 'ABP', with 'ABP' selected.
- Device Address:** The device address will be assigned by the network server.
- Network Session Key:** A field with a pencil icon and placeholder text 'Network Session Key will be generated'.
- App Session Key:** A field with a pencil icon and placeholder text 'App Session Key will be generated'.

Figure 25: ABP Mode Parameters

The same as the OTAA form, you can leave these fields empty to allow TTN to generate random values or input the specific values that you want.

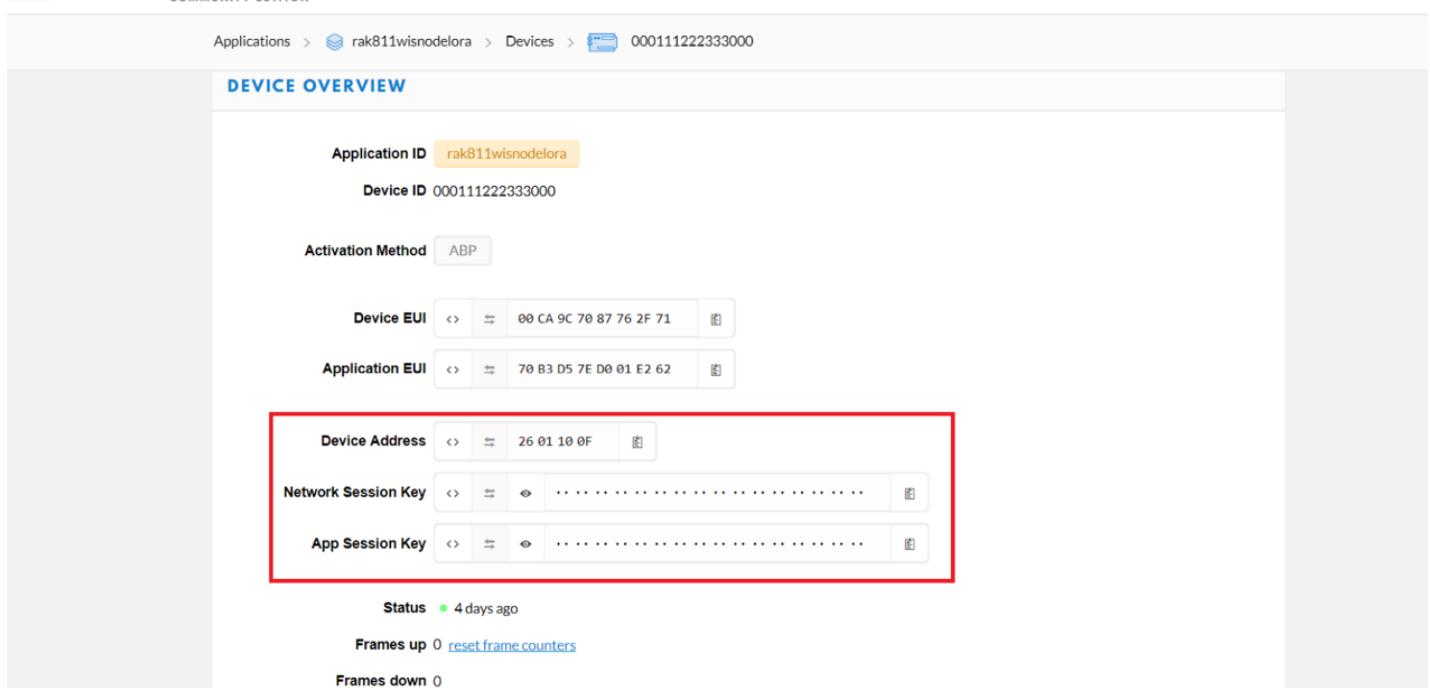


Figure 26: ABP Mode Parameters

Configure the ABP mode on the RAK7200

To set up the RAK7200 module in joining the TNN using ABP, start by connecting the RAK7200 module to the Computer (see Figure 1) and open the RAK Serial Port Tool. Wait for the communication to start. It is recommended to test the serial communication by sending either of these two AT commands:

```
at+get_config=lora:status
```

```
at+version
```

As an example, these are the list of the parameters you need to configure in RAK7200:

- LoRa join mode: **ABP**
- LoRa class: **Class A**
- LoRa region: **AU915**
- Device address: **000781fb9**
- Network Session Key: **c280cb8d1df688bc18601a97025c588**
- Application Session Key: **4d42ec5caf97f03d833cdaf5003f69e1**

1. Set the LoRa join mode to ABP.

```
at+set_config=lora:join_mode:1
```



Figure 27: Setting LoRa Join Mode to ABP Mode

2. Set the LoRa class to Class A.

```
at+set_config=lora:class:0
```

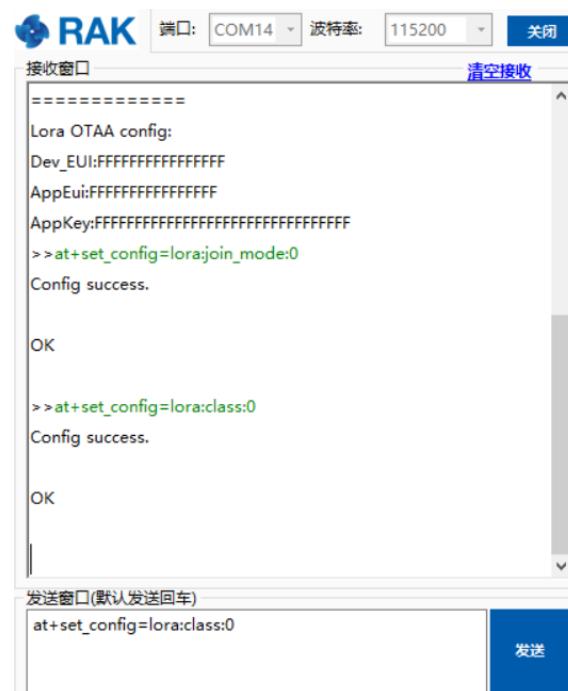


Figure 28: Setting LoRa Class Parameter

3. Set the frequency/region to AU915.

- Refer in the [RAK7200 Datasheet](#) for the list of supported frequencies.

```
at+set_config=lora:region:AU915
```

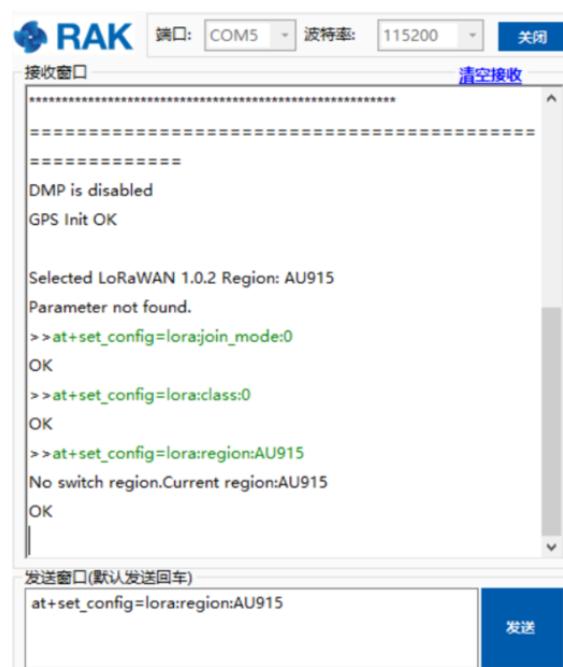


Figure 29: Setting LoRa Frequency/Region Parameter

4. Set the Device Address.

```
at+set_config=lora:dev_addr:00781fb9
```



Figure 30: Setting LoRa Device Address Parameter

5. Set the LoRa Network Session Key.

```
at+set_config=lora:nwks_key:c280cb8d1df688bc18601a97025c588
```

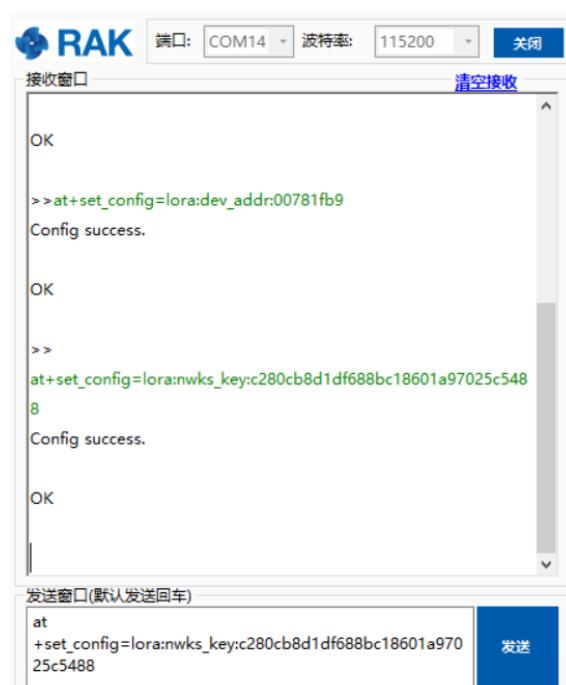


Figure 31: Setting LoRa Network Session Parameter

6. Set the LoRa Application Session Key.

```
at+set_config=lora:apps_key:4d42ec5caf97f03d833cdaf5003f69e1
```

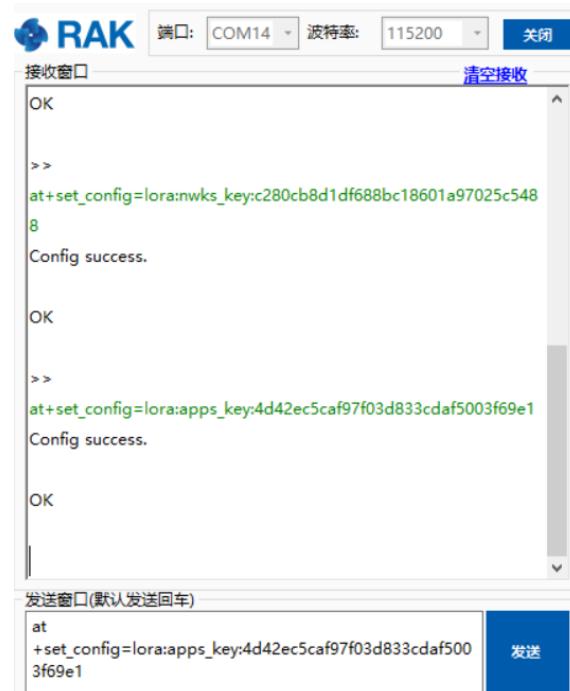


Figure 32: Setting LoRa Application Session Parameter

NOTE:

After configuring all parameters, you need to reset RAK7200 Module for saving parameters.

7. After resetting, join in OTAA mode.

```
at+join
```

 NOTE:

By using the ABP mode in LoRaWAN, it doesn't require to join a network before sending a LoRaWAN package. But to keep the consistency of internal states of the firmware of the RAK7200 module, it is still required to send at+join command in the ABP mode. This time, the firmware should reply almost immediately with an "OK".

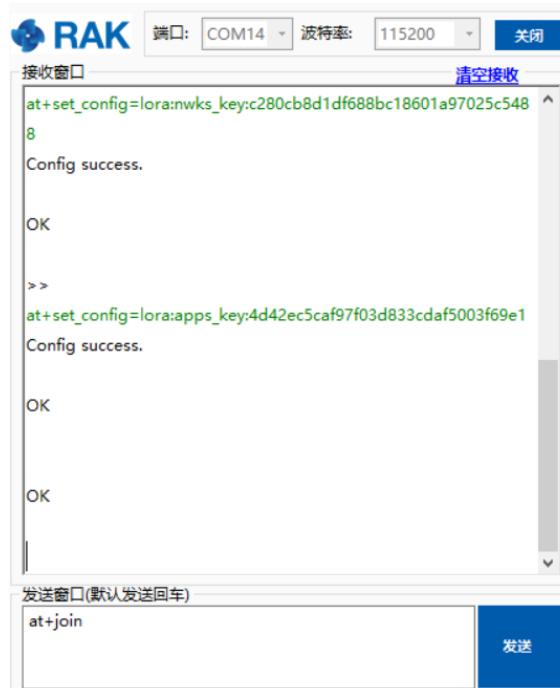


Figure 33: Join into a LoRaWAN Network

Once the RAK7200 joined into a LoRaWAN network, it will start to send periodically the data collected from the GPS and other sensors. Then, go to the TTN Console to confirm that the messages were properly received.

Connecting with ChirpStack

In this section, a practical exercise will be performed to show how to connect the RAK7200 module to the ChirpStack platform.

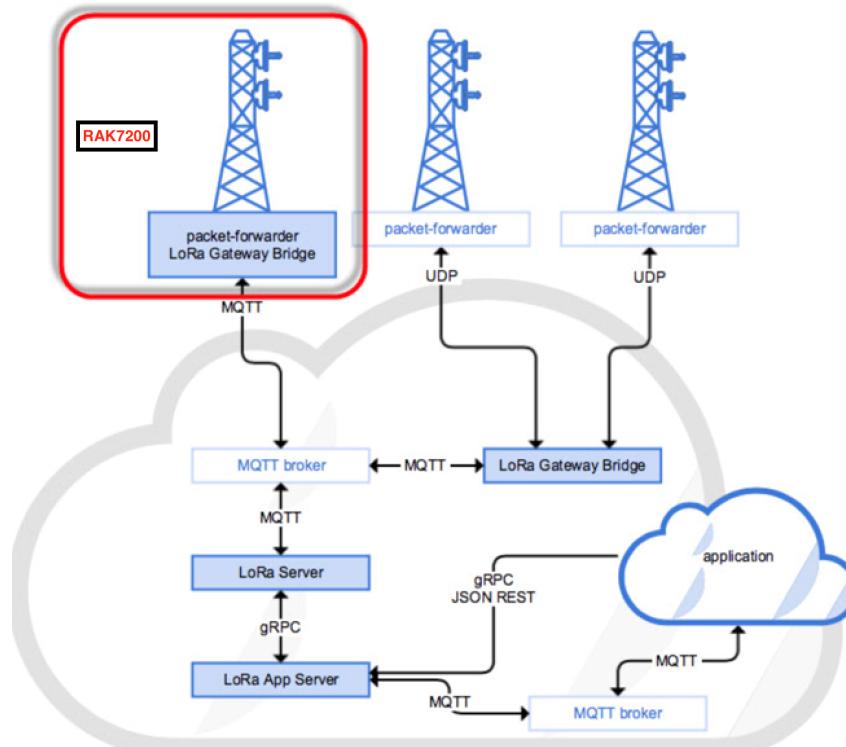


Figure 34: RAK7200 module in the context of the ChirpStack platform

The ChirpStack or previously known as LoRaServer project provides open-source components for building LoRaWAN networks. Like the case of TTN, the RAK7200 module is located in the periphery and transmit the data to the backend servers through a LoRa gateway. Learn more about [ChirpStack](#).

 **NOTE:**

In this document, it is assumed that you are using a RAK LoRa gateway, such as RAK7249. The gateway must be configured and registered previously to Chirpstack deployment. For further information, check the RAK documents for more details.

You can provide your own Chirpstack deployment or use the one provided by RAK for testing purposes. RAK has enabled a set of LoRaServer on the cloud to support customers to test their RAK LoRa gateway or RAK LoRa node. Table below shows the IP address for the supported regions. Submit your request in the following URL to get an account and password.

- [RAKwireless Forum](#)

Frequency	IP Address
CN470	106.15.233.112
EU868	47.88.62.184
US915	106.15.239.64
AS923	47.101.11.150
IN865	139.155.10.119

In this section, you need the following requirements:

1. Have an account at the ChirpStack deployment provided by RAK.
2. Have access to a LoRaWAN gateway registered at the RAK's ChirpStack server.
3. Serial Port Tool provided by RAK.
4. RAK7200 module

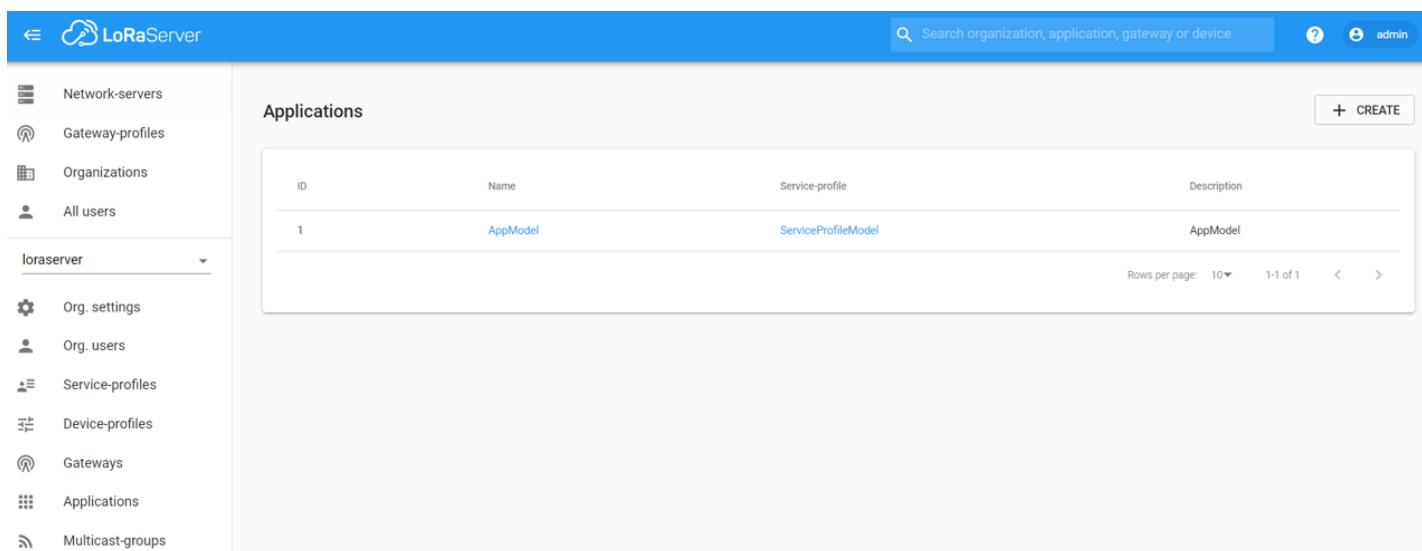
Before you start, you must choose which mode you are going to use, whether in OTAA or ABP mode, to register the device to the network server.

Sign up and login

Request an account in the forum of RAK, then access to server assigned for your region with your username and password.

Create a New Application

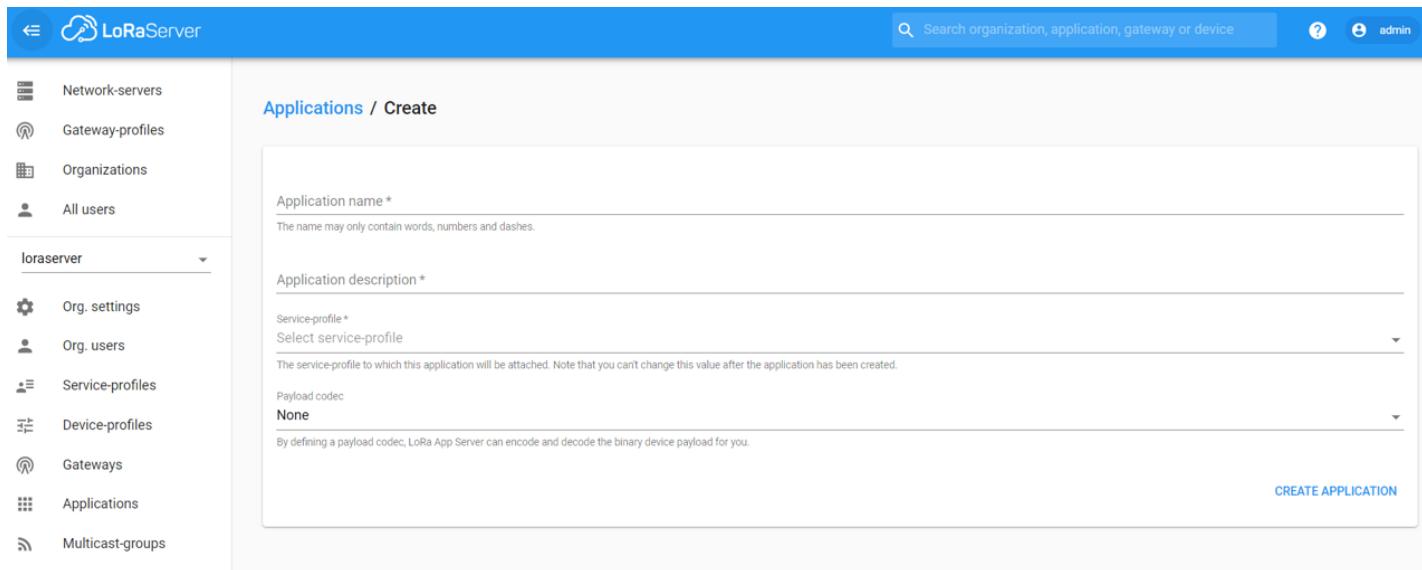
Go to the Application section as shown in the Figure 35.



The screenshot shows the 'Applications' section of the ChirpStack LoRaServer interface. On the left, there is a sidebar with various navigation options like Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'loraserver'. The main area displays a table with one row. The columns are ID, Name, Service-profile, and Description. The row contains ID 1, Name AppModel, Service-profile ServiceProfileModel, and Description AppModel. At the bottom right of the table, there are pagination controls showing 'Rows per page: 10', '1-1 of 1', and navigation arrows.

Figure 35: Application Section of ChirpStack LoRaServer

By default, you should create a new Application, although you can reuse the existing ones. For this setup, create a new Application by clicking on the “CREATE” button, and fill the required parameters as shown in the Figure 36 and Figure 37.



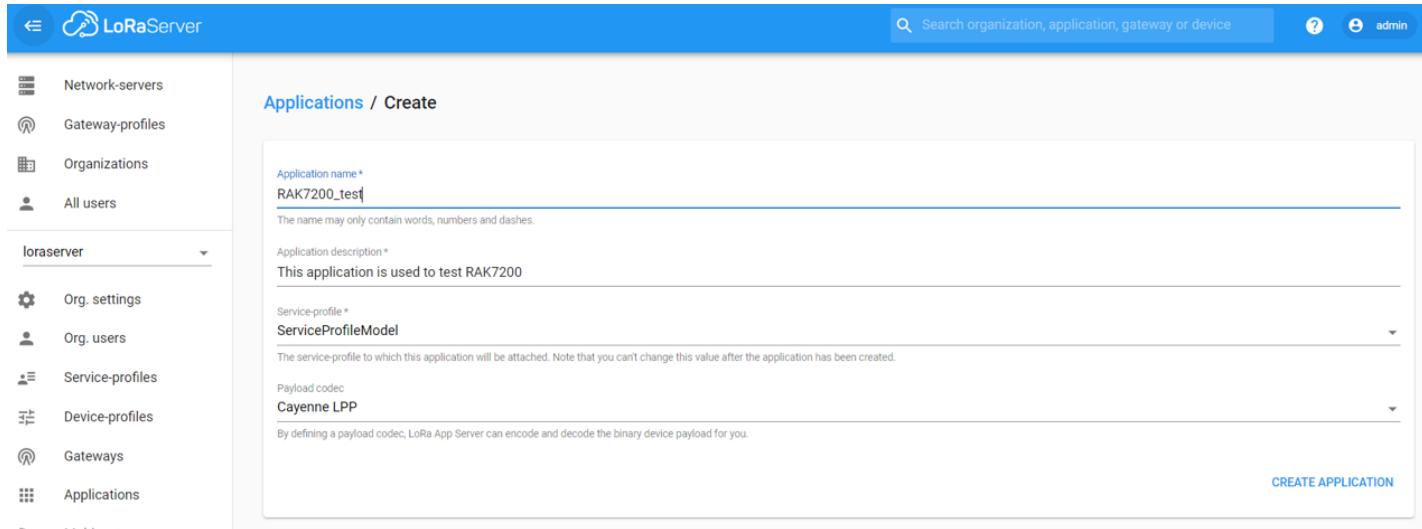
The screenshot shows the 'Applications / Create' form. It has fields for 'Application name *' (RAK7200_test), 'Application description *' (This application is used to test RAK7200), 'Service-profile *' (ServiceProfileModel), and 'Payload codec' (Cayenne LPP). A 'CREATE APPLICATION' button is located at the bottom right.

Figure 36: Creating a New Application

- For this setup, create an Application named “RAK7200_test”.

ChirpStack LoraServer supports multiple system configurations, with only one by default.

- **Service profile:** field is to select the system profile.
- **Payload codec:** is the parsing method for selecting load data. Such as parsing LPP format data.

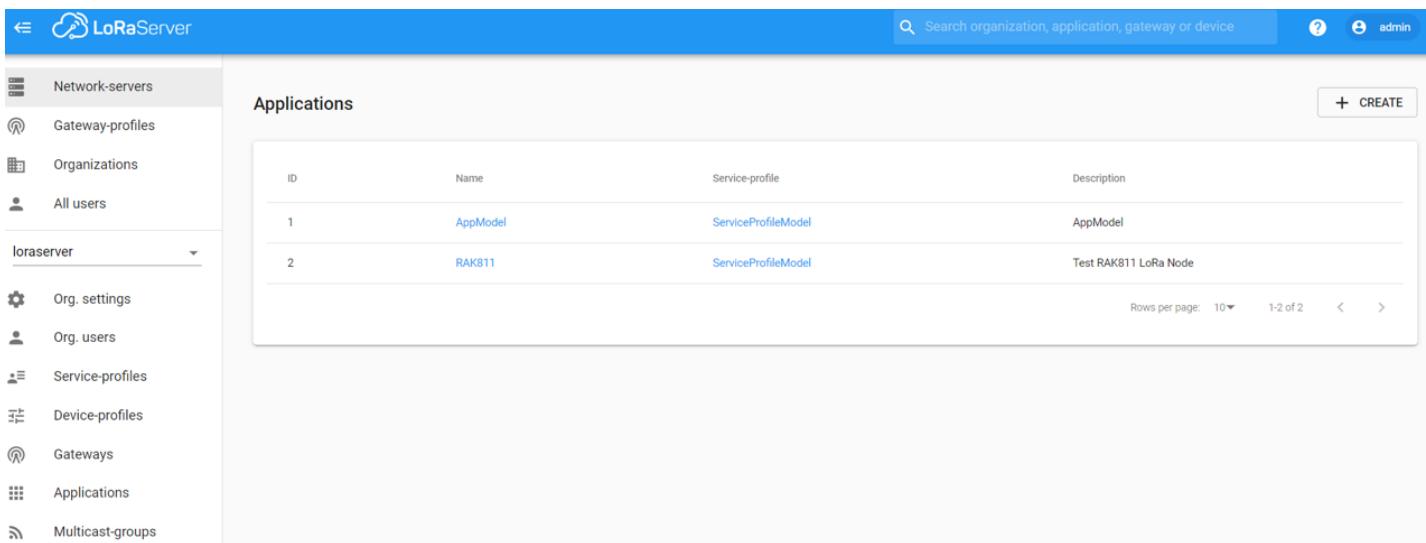


This screenshot shows the same 'Applications / Create' form as Figure 36, but with the fields filled in. The 'Application name *' field contains 'RAK7200_test', the 'Application description *' field contains 'This application is used to test RAK7200', the 'Service-profile *' field is set to 'ServiceProfileModel', and the 'Payload codec' field is set to 'Cayenne LPP'. The 'CREATE APPLICATION' button is visible at the bottom right.

Figure 37: Filling Parameters of an Application

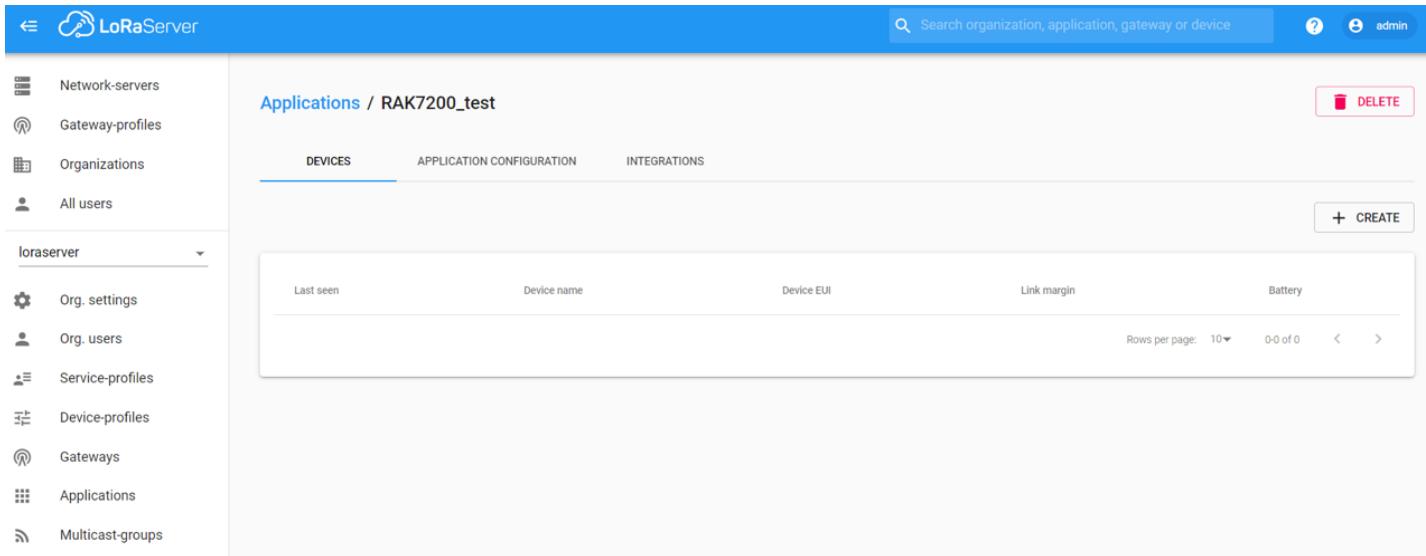
Register a New Device

1. Choose the **Application** created in the previous step, then select the **DEVICES** tab as shown in Figures 38 and 39.



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

Figure 38: List of Applications Created



Last seen	Device name	Device EUI	Link margin	Battery
-----------	-------------	------------	-------------	---------

Figure 39: Device Tab of an Application

3. Once inside of the DEVICE tab, create a new device (LoRa node) by clicking on the "+ CREATE" button.

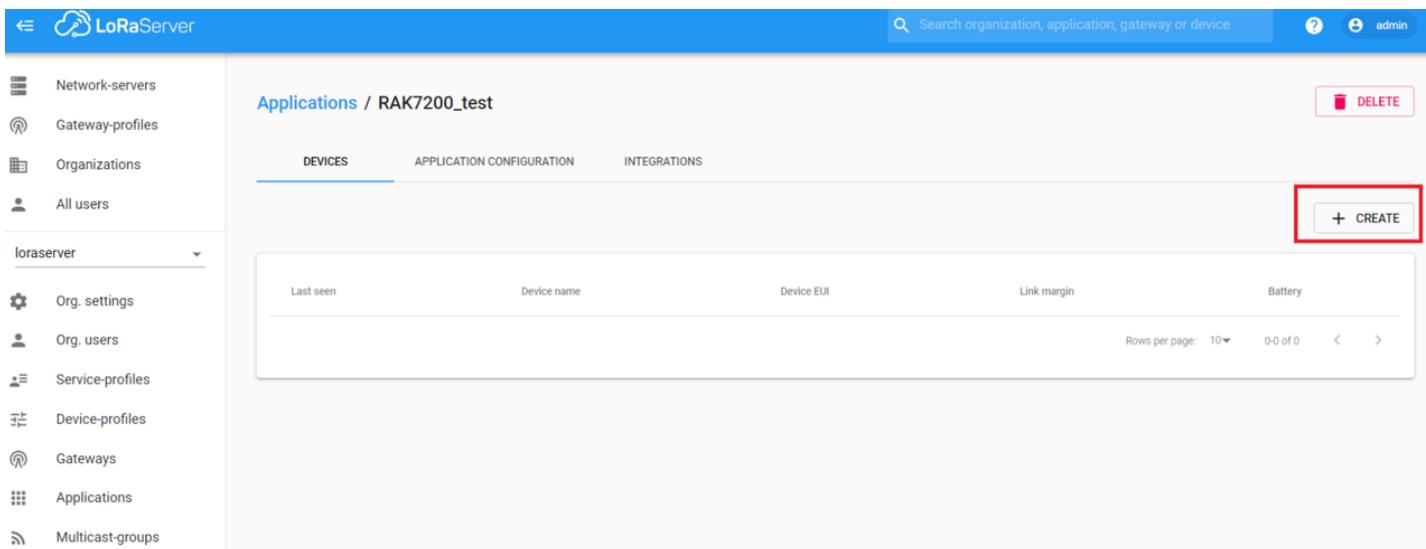


Figure 40: Add a New Device at Device Tab

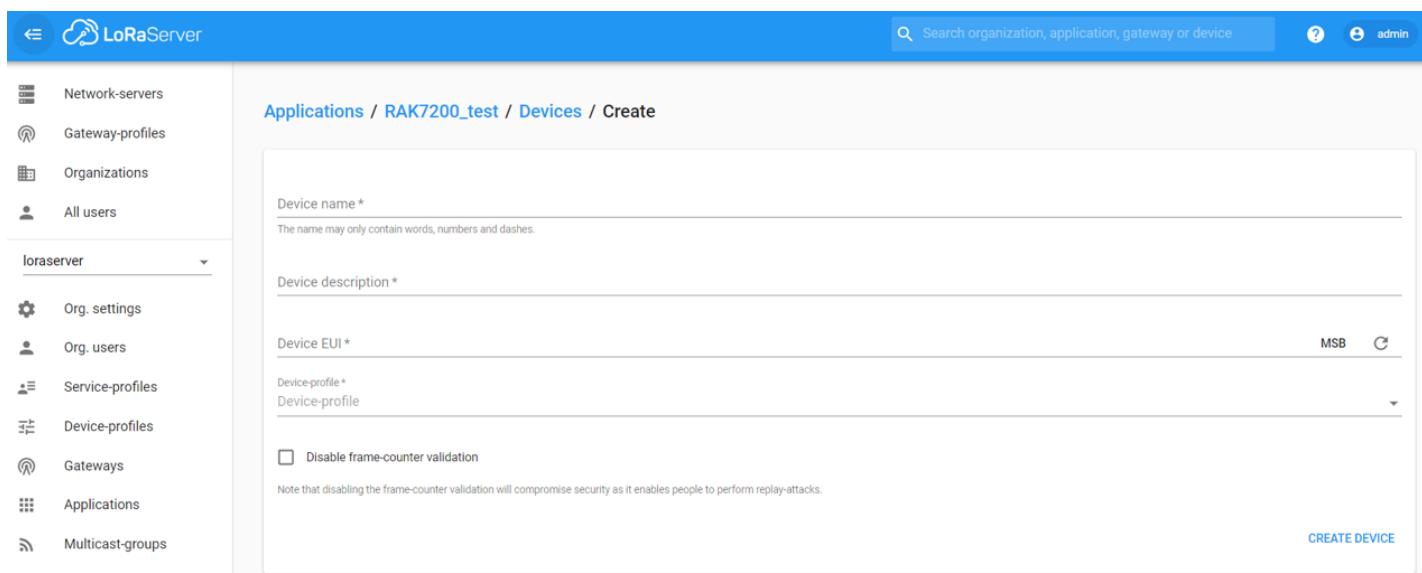


Figure 41: New Device Registration Form

6. Once the node is created, fill-in the necessary data. You can generate a Device EUI automatically by clicking the following icon, or you can write a correct Device EUI in the edit box.

- **Device name and Device description:** These are descriptive texts about your device.
- **Device EUI:** This interface allows you to generate a Device EUI automatically by clicking the icon highlighted in red in Figure 42. You can also add a specific Device EUI directly in the form.
- **Device Profile:**
 - If you want to join in OTAA mode, select “DeviceProfile_OTAA”.
 - If you want to join in ABP mode, select “DeviceProfile_ABP”.

NOTE:

ChirpStack does not support AS923 in ABP mode.

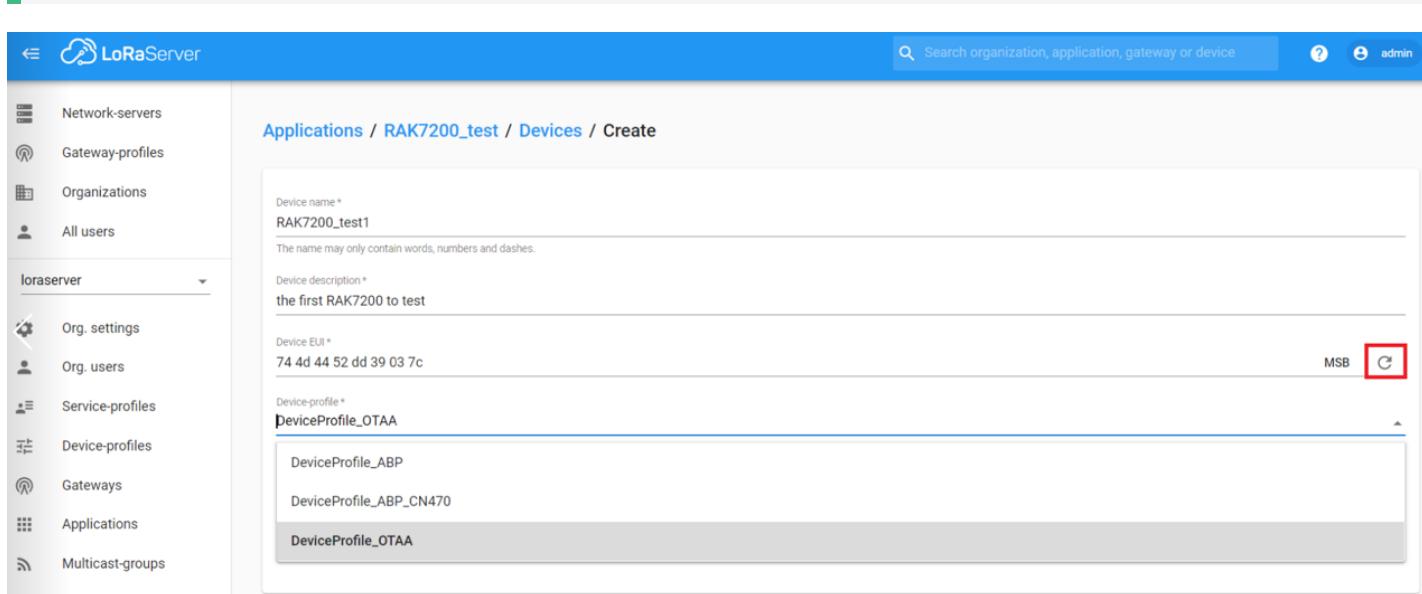


Figure 42: Generate a New Device EUI

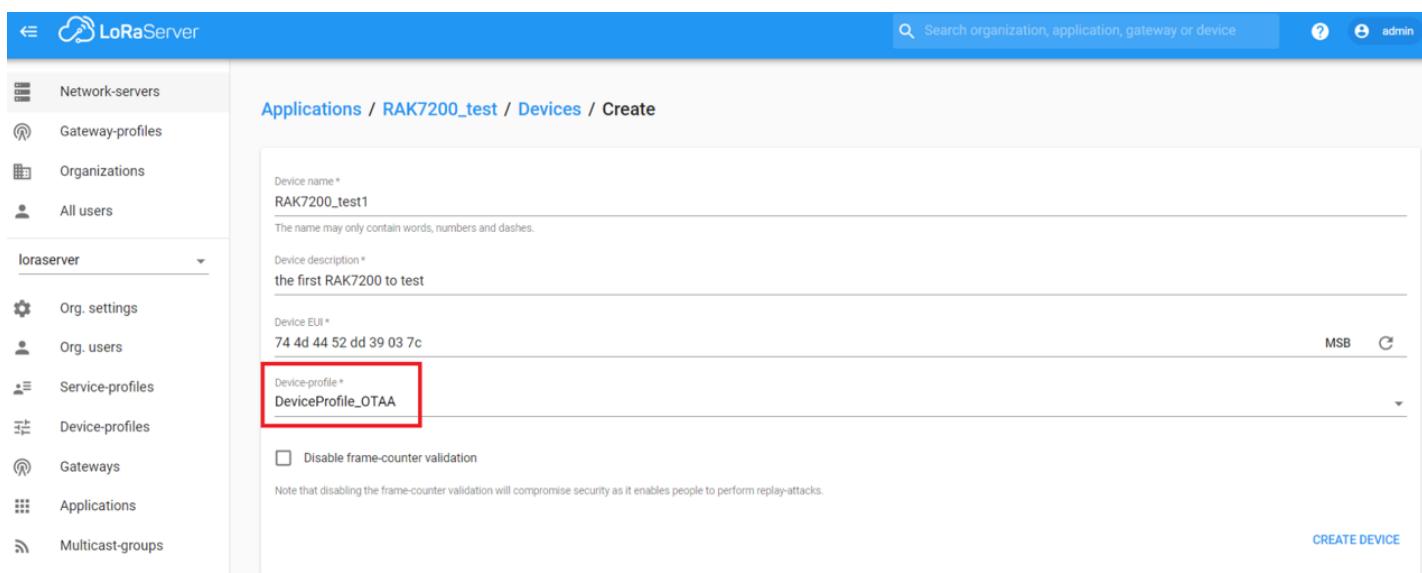
LoRaWAN Join Mode

In LoRaWAN, there are two ways a node can connect itself to the LoRaWAN network. This is referred to as **Join Mode**. LoRaWAN allows the OTAA mode and the ABP mode. In this section, the configuration process of these two modes, both on the platform side and the node side will be explained.

OTAA Mode

Configure the OTAA Mode on the Platform

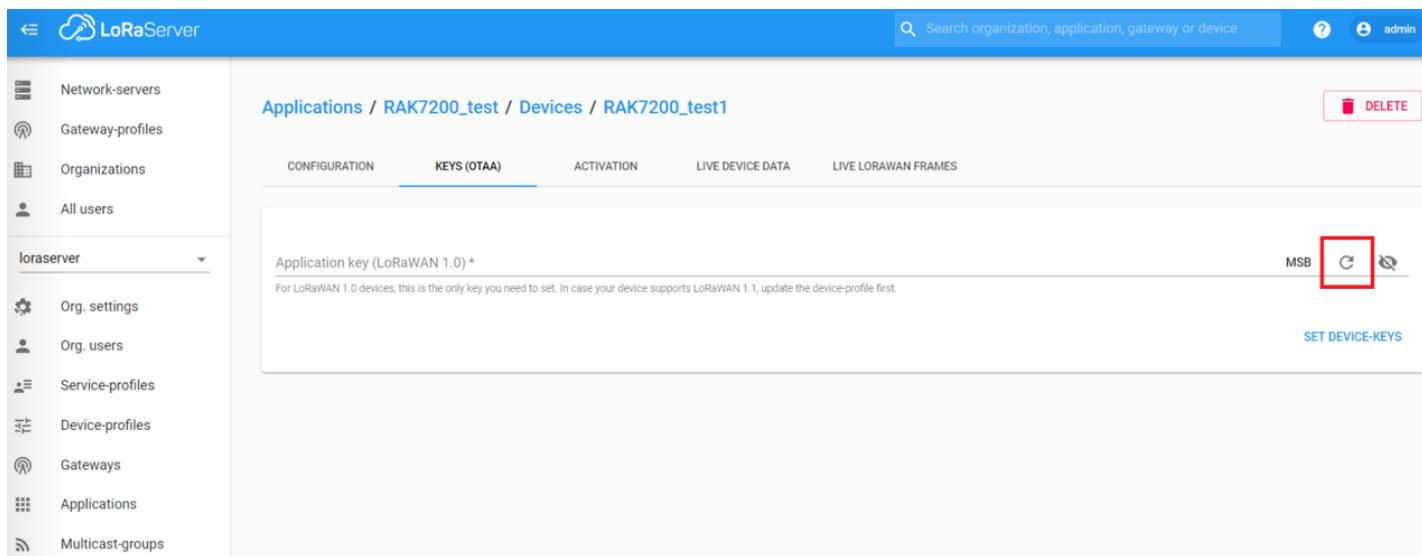
1. If you have selected “DeviceProfile_OTAA” as shown in Figure 43, then after the device is created, an Application Key must be also created for this device.



The screenshot shows the LoRaServer interface for creating a new device. On the left sidebar, under the 'loraserver' section, 'Device-profiles' is selected. In the main content area, the 'Device-profile*' dropdown is highlighted with a red box, showing 'DeviceProfile_OTAA' as the selected option. Other fields include 'Device name' (RAK7200_test1), 'Device description' (the first RAK7200 to test), and 'Device EUI' (74 4d 44 52 dd 39 03 7c). A note about frame-counter validation is present. At the bottom right is a 'CREATE DEVICE' button.

Figure 43: Chirpstack OTAA Activation

2. A previously created Application Key can be entered here, or a new one can be generated automatically by clicking the icon highlighted in red in Figure 44:

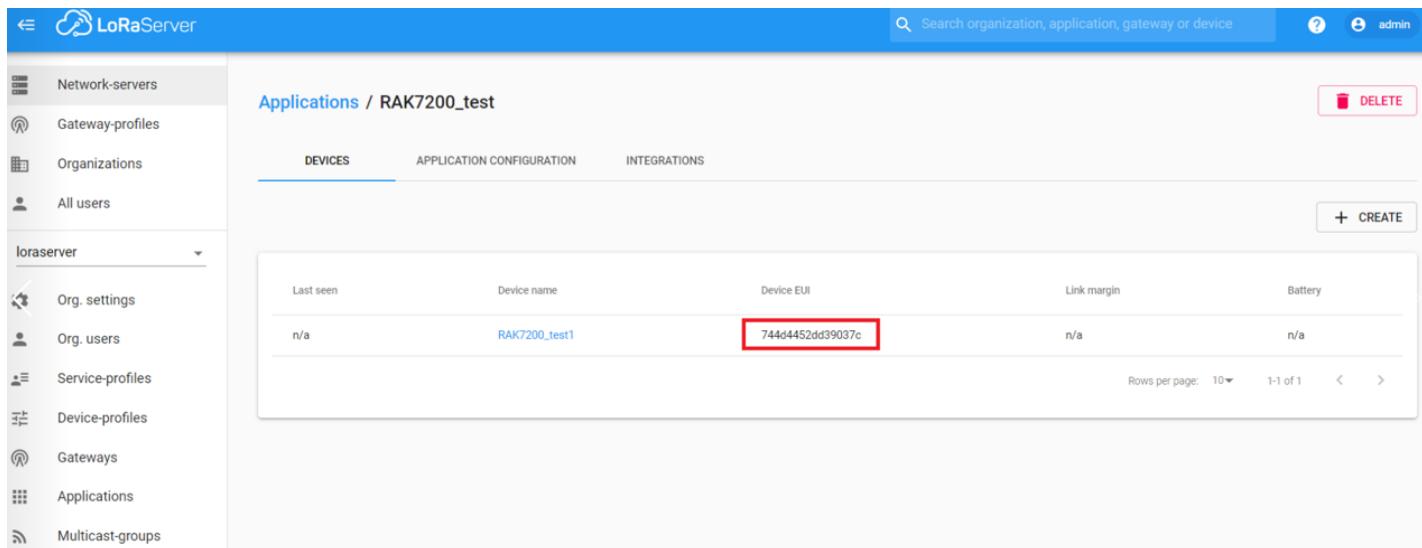


The screenshot shows the LoRaServer interface for managing a device named 'RAK7200_test1'. The 'KEYS (OTAA)' tab is active. Under the 'Application key (LoRaWAN 1.0)' field, there is a note: 'For LoRaWAN 1.0 devices, this is the only key you need to set. In case your device supports LoRaWAN 1.1, update the device-profile first.' To the right of the input field are icons for 'MSB' (highlighted with a red box), 'C' (copy), and 'D' (delete). Below the input field is a 'SET DEVICE-KEYS' button.

Figure 44: Chirpstack OTAA Set Application Keys

3. Once the Application Key is added in the form, the process can be finalized by clicking on the “SET DEVICE-KEYS” button.

- As shown in Figure 45, a new device should be listed in the DEVICES tab. The most important parameters, such as the Device EUI are shown in the summary.



The screenshot shows the LoRaServer interface for the 'RAK7200_test' application. The 'DEVICES' tab is active. A single device entry is listed: 'Last seen' (n/a), 'Device name' (RAK7200_test1), 'Device EUI' (744d4452dd39037c, highlighted with a red box), 'Link margin' (n/a), and 'Battery' (n/a). At the bottom right of the table are pagination controls: 'Rows per page: 10', '1-1 of 1', and navigation arrows. A '+ CREATE' button is located at the top right of the table area.

Figure 45: Chirpstack OTAA List of Device

4. To end the process, it is a good practice to review that the Application Key is properly associated with this device. The Application Key can be verified in the **KEYS(OTAA)** tab as shown in Figure 46.

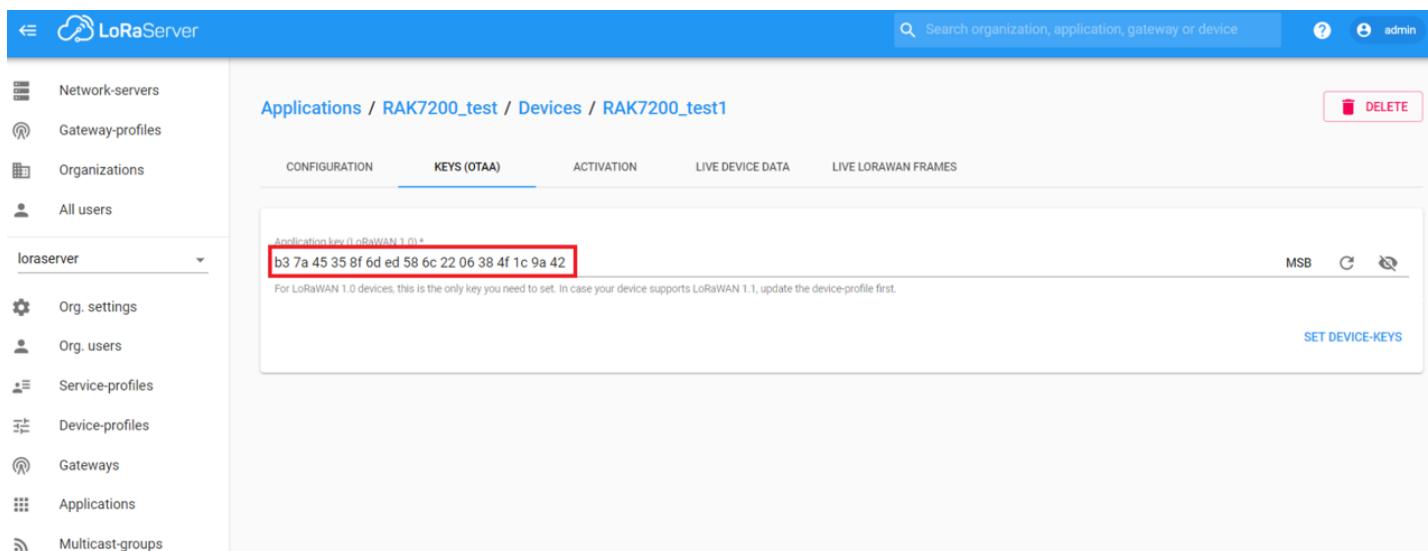


Figure 46: Chirpstack OTAA Set Application Key

NOTE:

Standard OTAA mode requires the **Device EUI**, **Application Key**, and the **Application EUI**. But in the ChirpStack's implementation, only the Device EUI and the Application Key are mandatory. The Application EUI is not required and is not recorded in the Application tab. Nevertheless, the Application EUI is a mandatory parameter in the RAK7200 module's firmware. To resolve this mismatch, you can reuse the Device EUI as the Application EUI during the configuration in the side of the node.

Configure the OTAA mode on the RAK7200

The RAK7200 module supports a series of AT commands to configure its internal parameters and control the functionalities of the module.

To set up the RAK7200 module to join ChirpStack using OTAA start by connecting the RAK7200 module to the Computer (see Figure 1) and open the RAK Serial Port Tool, wait for the communication to start. It is recommended to test the serial communication by sending either of these two AT commands:

```
at+get_config=lora:status
```

```
at+version
```

As an example, these are the list of the parameters you need to configure in RAK7200:

- LoRa join mode: **OTAA**
 - LoRa class: **Class A**
 - LoRa region: **AU915**
 - Device EUI: **744d4452dd39037c**
 - Application EUI: **70B3D57ED001C544**
 - Application Key: **4E2003296FC5CD26F46940A6DAFA9D1**
1. Set the LoRa join mode to OTAA.

```
at+set_config=lora:join_mode:0
```

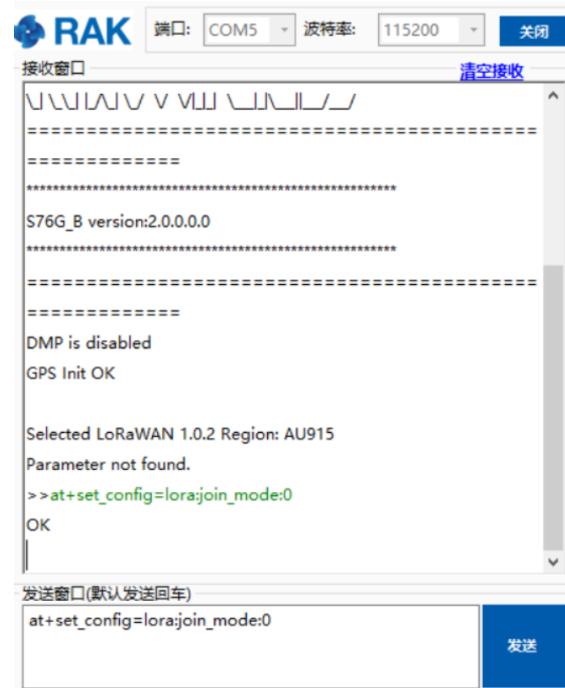


Figure 47: Setting LoRa Join Mode to OTAA Mode

2. Set the LoRa class to Class A.

```
at+set_config=lora:class:0
```

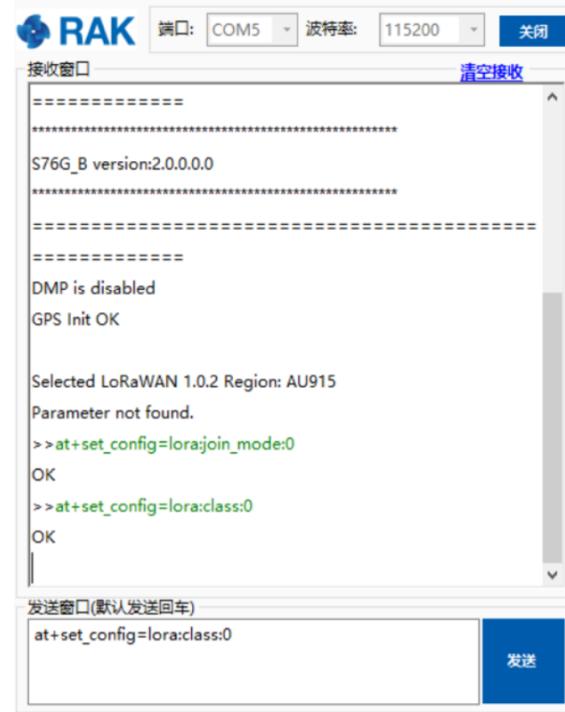


Figure 48: Setting LoRa Class Parameter

3. Set the frequency/region to AU915.

- Refer in the [RAK7200 Datasheet](#) for the list of supported frequencies.

```
at+set_config=lora:region:AU915
```

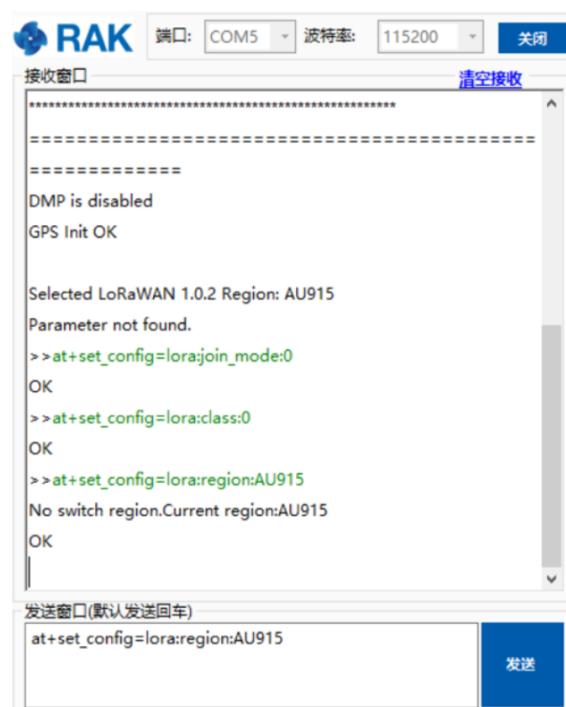


Figure 49: Setting LoRa Frequency/Region Parameter

4. Set the Device EUI.

```
at+set_config=lora:dev_eui:744d4452dd39037c
```



Figure 50: Setting LoRa Device EUI Parameter

5. Set the Application EUI.

```
at+set_config=lora:app_eui:70B3D57ED001C544
```

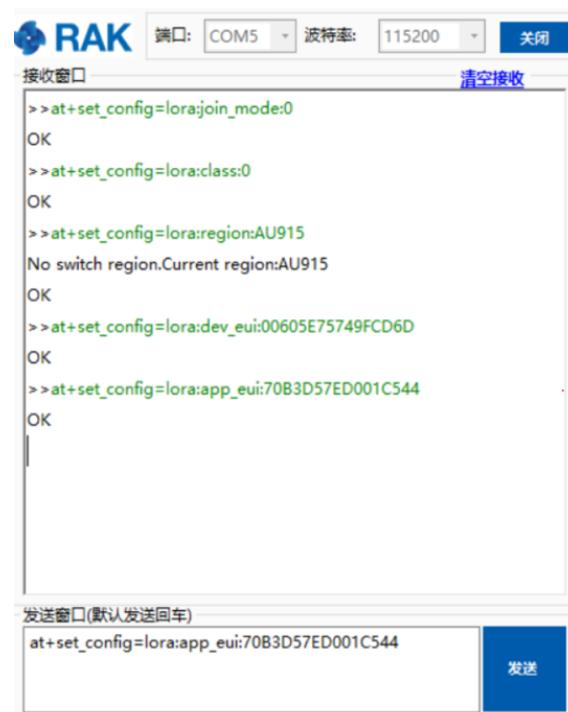


Figure 51: Setting LoRa Application EUI Parameter

 NOTE:

Remember, the Application EUI parameter was not required in the ChirpStack platform. Therefore, it is possible to use the same ID as the Device EUI. Otherwise, the firmware will complain.

6. Set the Application Key.

- Get the Application Key from the TTN register.

```
at+set_config=lora:app_key:4E2003296FC5CD26F46940A6DAFA9D1
```

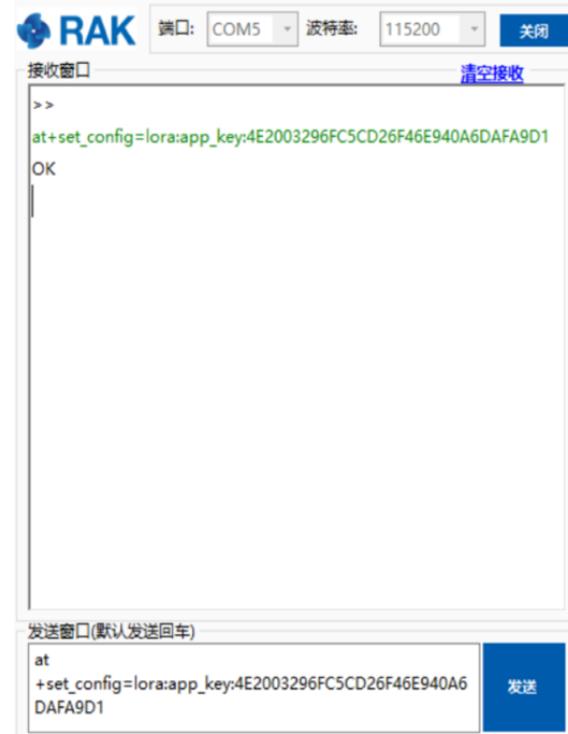


Figure 52: Setting LoRa Application Key Parameter

 NOTE:

After configuring all parameters, you need to reset RAK7200 Module to save the parameters.

7. After resetting, start to join.

```
at+join
```

After 5 or 6 seconds, if the request was successfully received by a LoRa gateway, then you should see the messages shown in Figure 53.



Figure 53: Chirpstack OTAA Join the Network via RAK Serial Port Tool

8. You can then see the **JoinRequest** and **JoinAccept** on ChirpStack page.

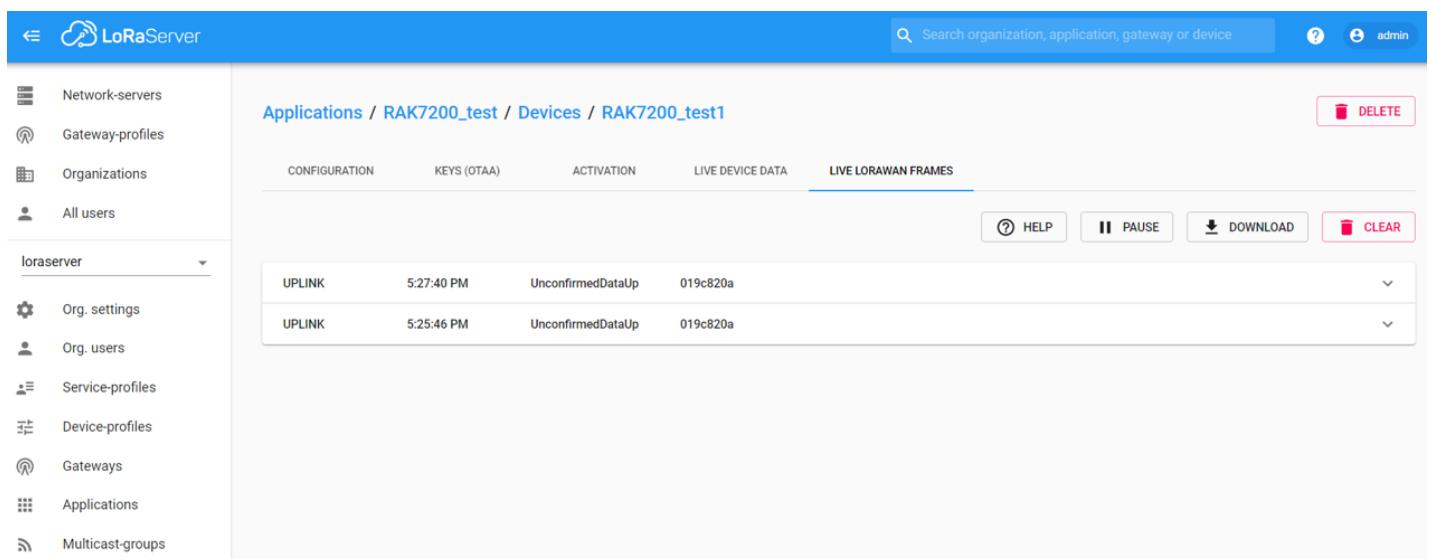
	DOWNLINK	5:42:29 PM	JoinAccept
	UPLINK	5:42:29 PM	JoinRequest 5e9d1e0857cf25f1

Figure 54: Checking LoRaWAN Joint Request in Chirpstack OTAA Console

- On the ChirpStack platform, you should also see the messages in the LORAWAN FRAMES tab as shown in Figure 55.

NOTE:

By convention, messages sent from nodes to gateways are considered as **Uplinks**. While messages sent by gateways to nodes are considered as **Downlinks**.



The screenshot shows the Chirpstack Data Received Preview 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 displays the path Applications / RAK7200_test / Devices / RAK7200_test1. Below this are tabs: CONFIGURATION, KEYS (OTAA), ACTIVATION, LIVE DEVICE DATA, and LIVE LORAWAN FRAMES (which is selected). Under LIVE LORAWAN FRAMES, there are two entries: "UPLINK 5:27:40 PM UnconfirmedDataUp 019c820a" and "UPLINK 5:25:46 PM UnconfirmedDataUp 019c820a". At the bottom right are buttons for HELP, PAUSE, DOWNLOAD, and CLEAR.

Figure 55: Chirpstack Data Received Preview

This concludes the exercise to send data in the OTAA mode.

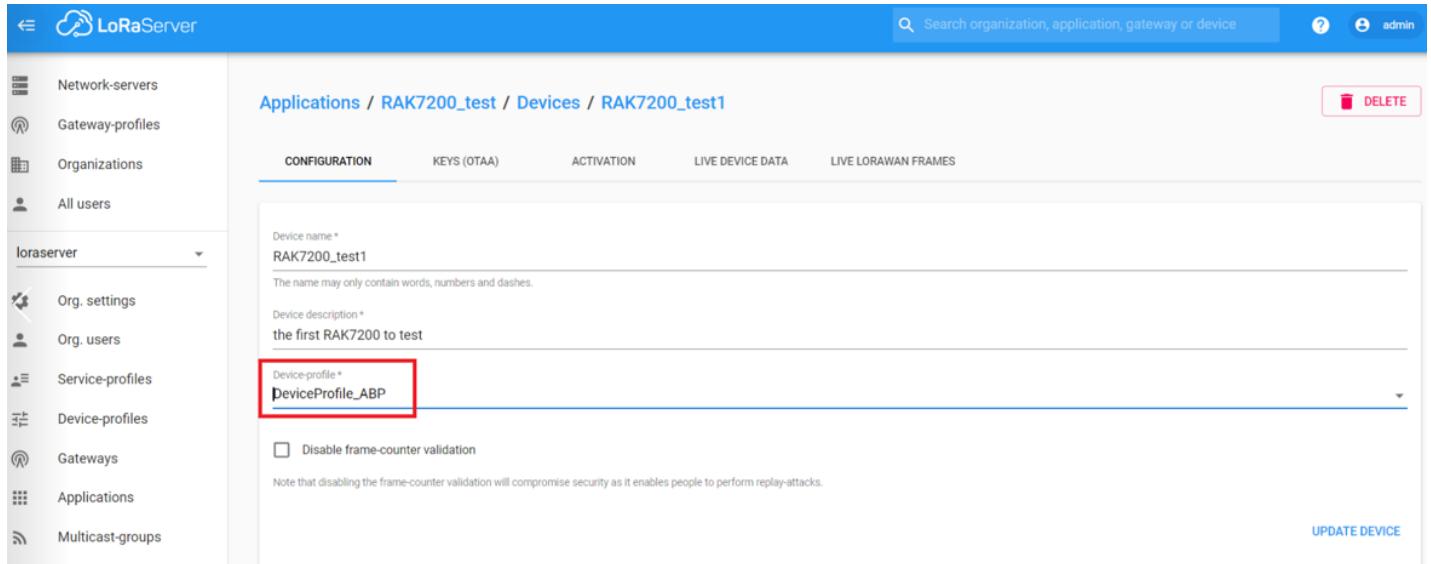
ABP Mode

Configure the ABP Mode on the Platform

During the registration of a new device, if you select "DeviceProfile_ABP" or "DeviceProfile_ABP_CN470", as shown in Figure 56, then the ChirpStack platform will assume that this device will join to the LoRaWAN network using the ABP mode.

 **NOTE:**

Check "Disable counting frame verification". During the test, when the module is restarted, the frame counting number will be also be restarted from zero. This would cause a synchronization problem with the ChirpStack server treating it as a replay attack. For the testing purpose, it is safe to disable this feature, but remember to activate it in a production environment.



The screenshot shows the ChirpStack Console configuration interface for a device named "RAK7200_test1". The sidebar includes links for Network-servers, Gateway-profiles, Organizations, All users, loraserver (selected), Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main configuration screen has tabs for CONFIGURATION, KEYS (OTAA), ACTIVATION, LIVE DEVICE DATA, and LIVE LORAWAN FRAMES. In the ACTIVATION tab, the "Device-profile" dropdown is set to "DeviceProfile_ABP" (which is highlighted with a red box). There is also a checkbox for "Disable frame-counter validation" with a note below stating: "Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks." A blue "UPDATE DEVICE" button is located at the bottom right.

Figure 56: ChirpStack Console, Configuring a Device

After selecting the ABP mode, the following parameters appear in the Activation tab:

- Device address
- Network Session Key
- Application Session Key

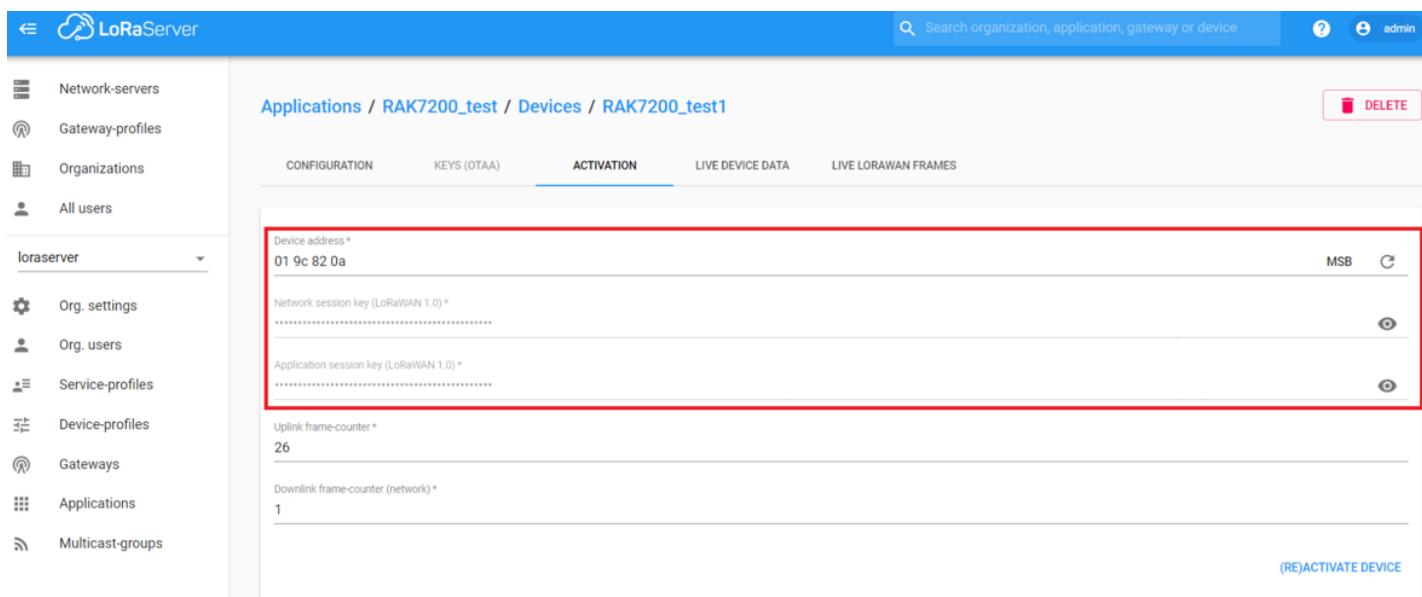


Figure 57: Chirpstack ABP Activation Parameters Needed

- The parameters can be generated as random numbers by the platform or can be set with user values. Once these parameters are filled properly, the process is completed by clicking on the “**ACTIVATE DEVICE**” button.

Configure the ABP mode on the RAK7200

In the following steps, you will configure the RAK7200 module to work in the ABP mode. To set up the RAK7200 module to join ChirpStack using ABP start by connecting the RAK7200 module to the Computer (see Figure 1) and open the RAK Serial Port Tool, wait for the communication to start. It is recommended to test the serial communication by sending either of these two AT commands:

```
at+get_config=lora:status
```

```
at+version
```

As an example, these are the list of the parameters you need to configure in RAK7200:

- LoRa join mode: **ABP**
- LoRa class: **Class A**
- LoRa region: **AU915**
- Device address: **019c820a**
- Network Session Key: **34fb174daa6dc34733495f73bd2b1ba1**
- Application Session Key: **d7fde8da53ded21216fe71d4f91ecf0e**

- Set LoRa join mode to ABP.

```
at+set_config=lora:join_mode:1
```



Figure 58: Setting LoRa Join Mode to ABP Mode

2. Set LoRa class to Class A.

```
at+set_config=lora:class:0
```

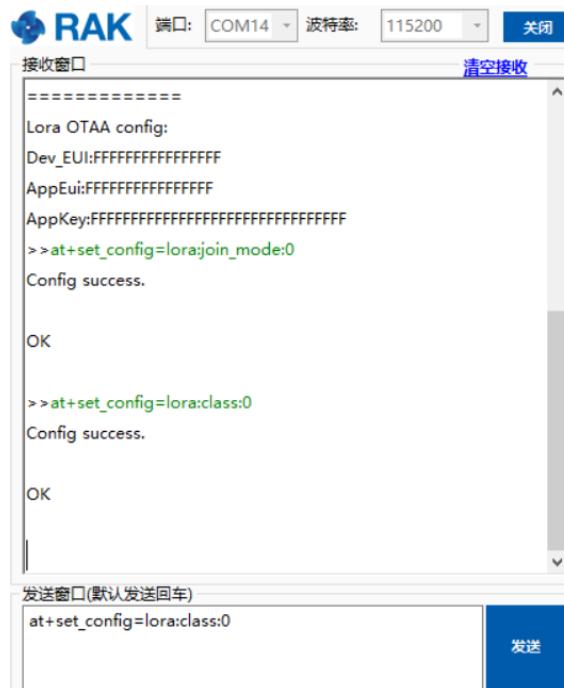


Figure 59: Setting LoRa Class Parameter

3. Set the frequency/region to AU915.

- Refer in the [RAK7200 Datasheet](#) for the list of supported frequencies.

```
at+set_config=lora:region:AU915
```

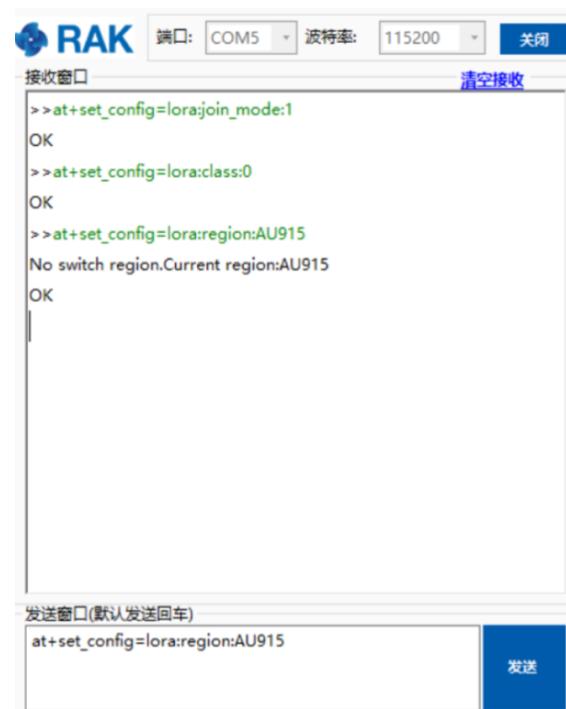


Figure 60: Setting LoRa Class Parameter

4. Set the Device Address.

```
at+set_config=lora:dev_addr:019c820a
```

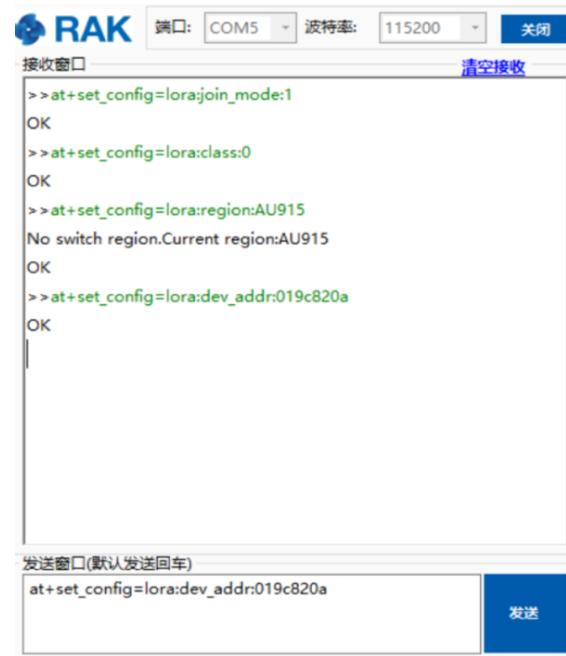


Figure 61: Setting LoRa Device Address Parameter

5. Set the Network Session Key.

```
at+set_config=lora:nwks_key:34fb174daa6dc34733495f73bd2b1ba1
```

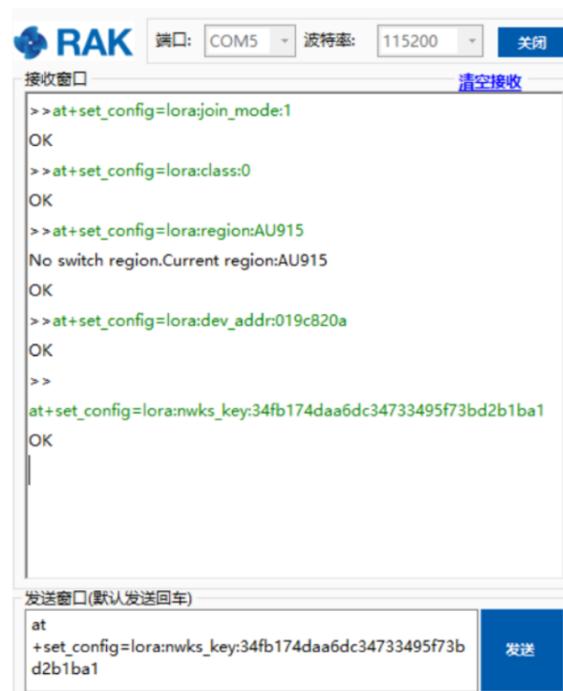


Figure 62: Setting LoRa Network Session Parameter

6. Set the Application Session Key.

```
at+set_config=lora:apps_key: d7fde8da53ded21216fe71d4f91ecf0e
```



Figure 63: Setting LoRa Network Session Parameter

NOTE:

After configuring all parameters, you need to reset RAK7200 Module for saving parameters.

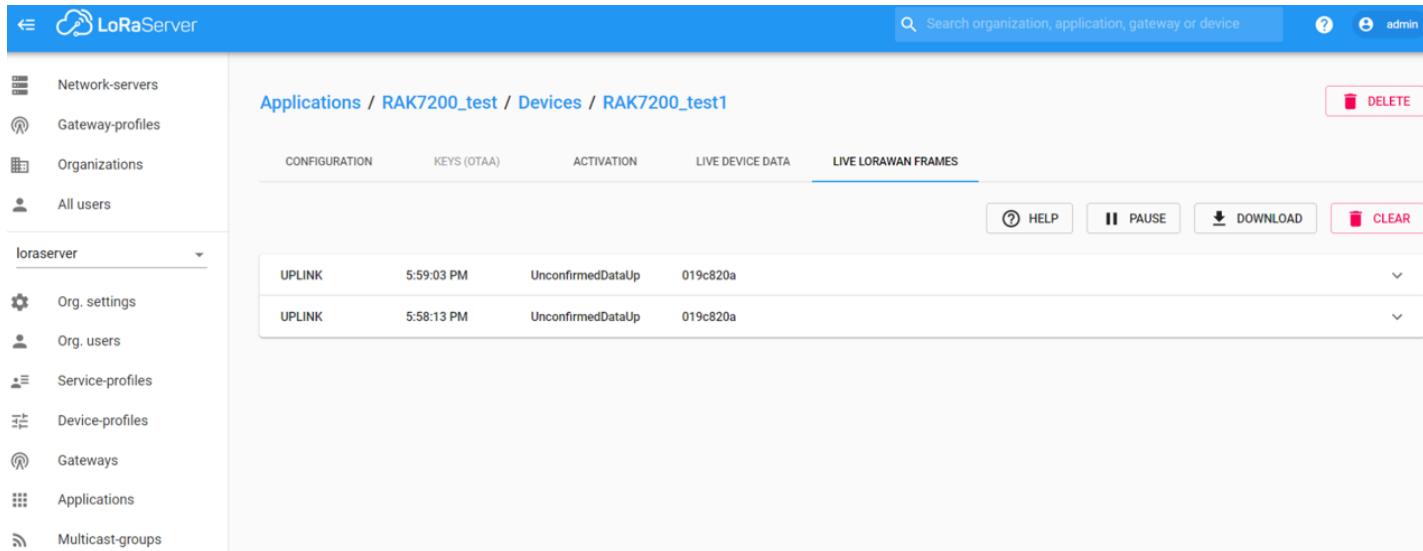
7. After resetting RAK7200 Module, join in ABP mode.

```
at+join
```

NOTE:

By using the ABP mode in LoRaWAN protocol, it doesn't require to join a network before sending LoRaWAN package. But to keep the consistency of internal states of the firmware of the RAK7200 module, it still required to send at+join command in the ABP mode. This time, the firmware should reply almost immediately with an "OK".

Once the RAK7200 joined into a LoRaWAN network, it will start to send periodically the data collected from the GPS and other sensors. Then, go to the ChirpStack platform to confirm that the messages were properly received as shown in the Figure 64.



Type	Time	Message Type	MAC Address
UPLINK	5:59:03 PM	UnconfirmedDataUp	019c820a
UPLINK	5:58:13 PM	UnconfirmedDataUp	019c820a

Figure 64: Data Collected in ChirpStack

Miscellaneous

Upgrading the Firmware

Before you start working with the RAK7200, it is recommended to keep the RAK7200 module updated to the latest version of the firmware. Download the latest [RAK7200 firmware](#).

In this section, you will upgrade the firmware by using the **STM32CubeProgrammer** provided by STM32. Execute the following steps in upgrading the firmware.

1. Install the [STM32CubeProgrammer](#) tool on your Windows PC.
2. Connect the RAK7200 module to the USB interface of a general-purpose computer as shown in Figure 1.
3. To program the firmware, the RAK7200 module must be entered into the boot mode. You can achieve it by following this procedure:
 - Hold down the BOOT0 button, and then press the Reset button.
 - Release the Reset button, then release the BOOT0 button.

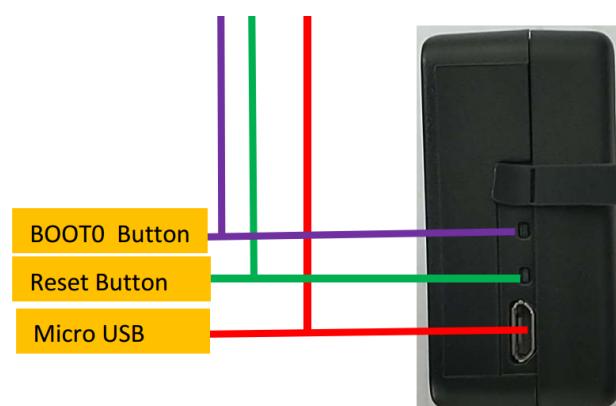


Figure 65: RAK7200 BOOT0 and Reset buttons

4. Open the **STM32CubeProgrammer** tool and select the UART as the type of connection to the RAK7200 module. Then, configure the following communication parameters:

- Port: **choose the port assigned by the operative system**
- Baud rate: **115200**.
- Parity: **Even**

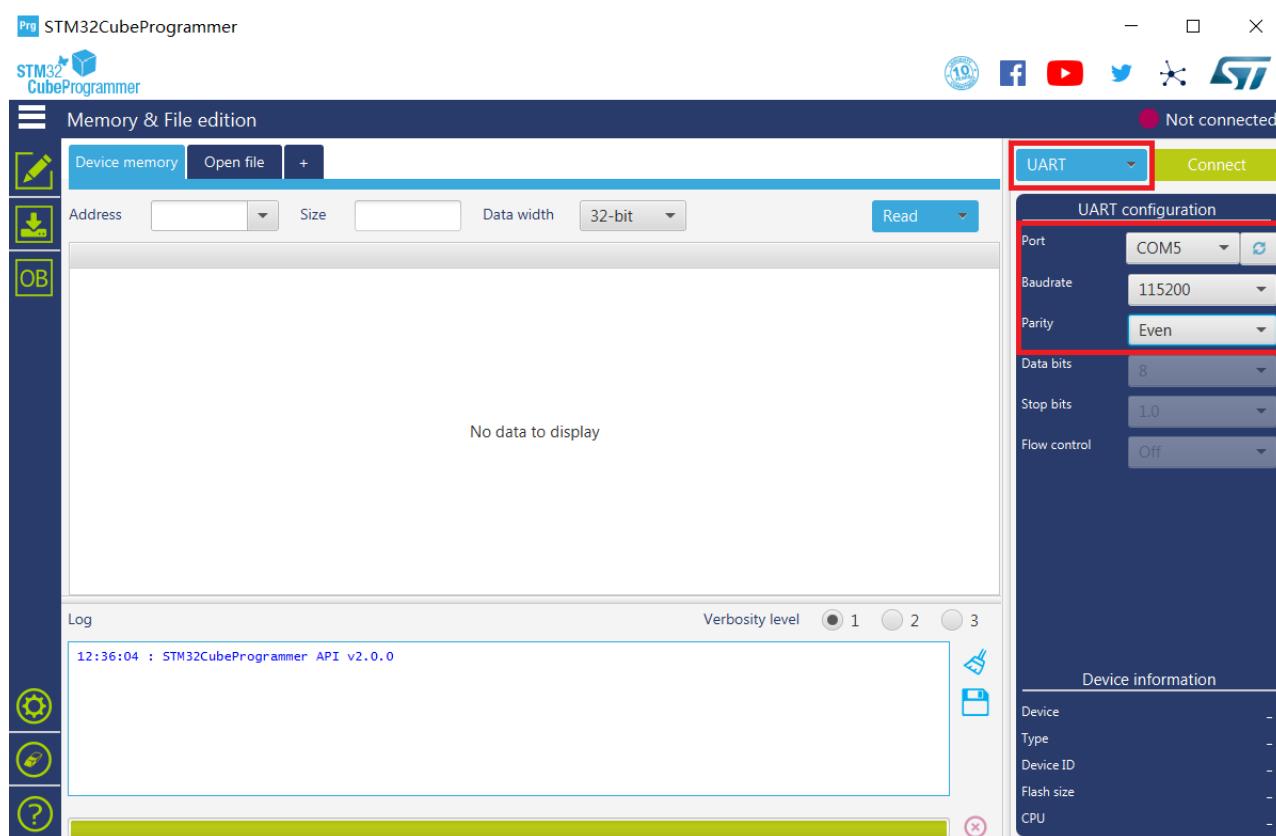


Figure 66: STM32CubeProgrammer UART Communication Parameters

5. Once the connection parameters were properly set, press the **Connect** button located at the top right corner. If there were problems during the connection, the errors would be shown in Log message box. The usual problem is because the RAK7200 didn't enter into the boot mode properly. In such cases, you will find similar error messages shown in the Figure 67.

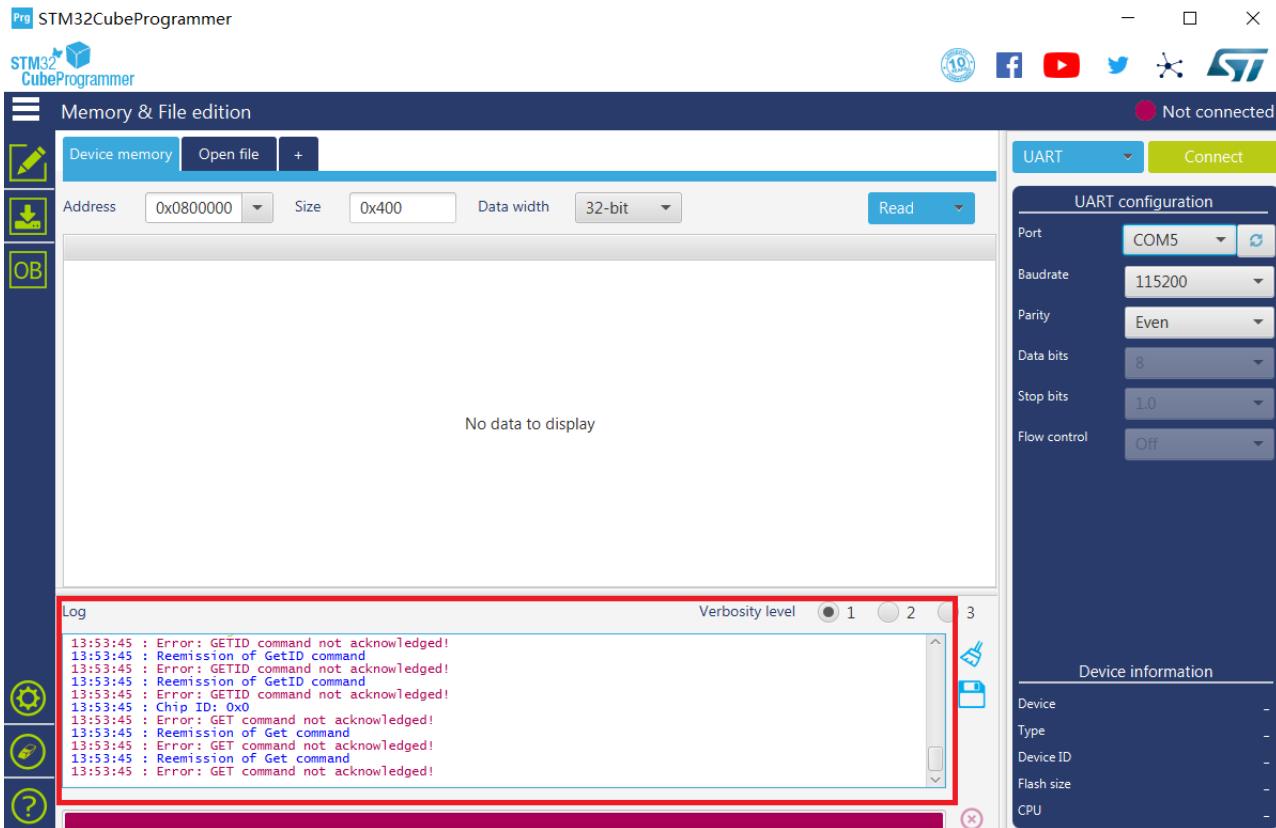


Figure 67: Error Messages Received when not in Boot Mode

- After connecting successfully, the STM32CubeProgrammer will display **Data read successfully** in the log message box.

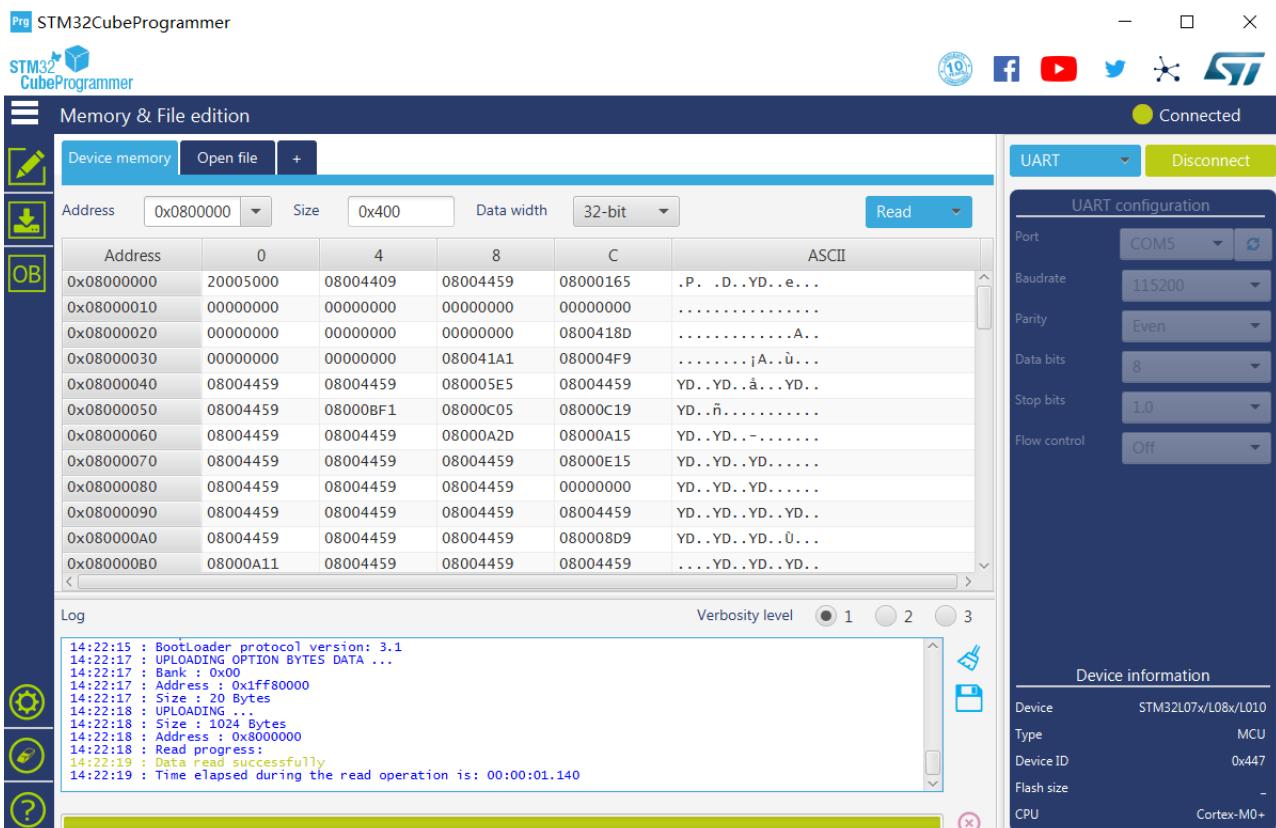


Figure 68: Messages Received when connected in Boot Mode

- Once the communication is established with the RAK7200, burn a newer version of firmware into the RAK7200 module:

- Erase the existing data:** To avoid unexpected problems during the future usage of the module, it's a good practice to erase completely the EEPROM before burning the new version of firmware. You can achieve it by clicking on the Erase button highlighted in red square in Figure 69.

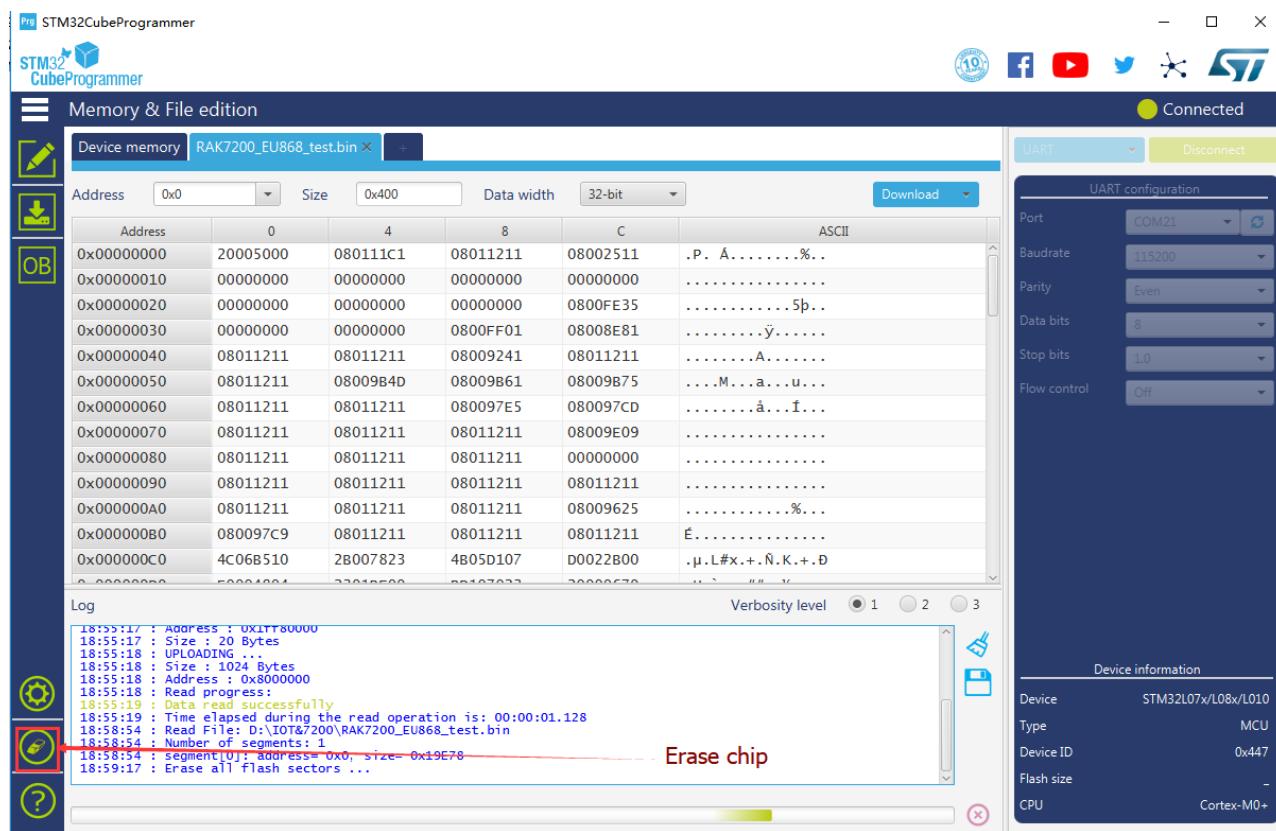


Figure 69: STM32CubeProgrammer Erase Button

- Select the desired firmware to burn: Press the “Open file” tab, see Figure 68, and select the correct firmware file in the pop-up window as shown in Figure 70.

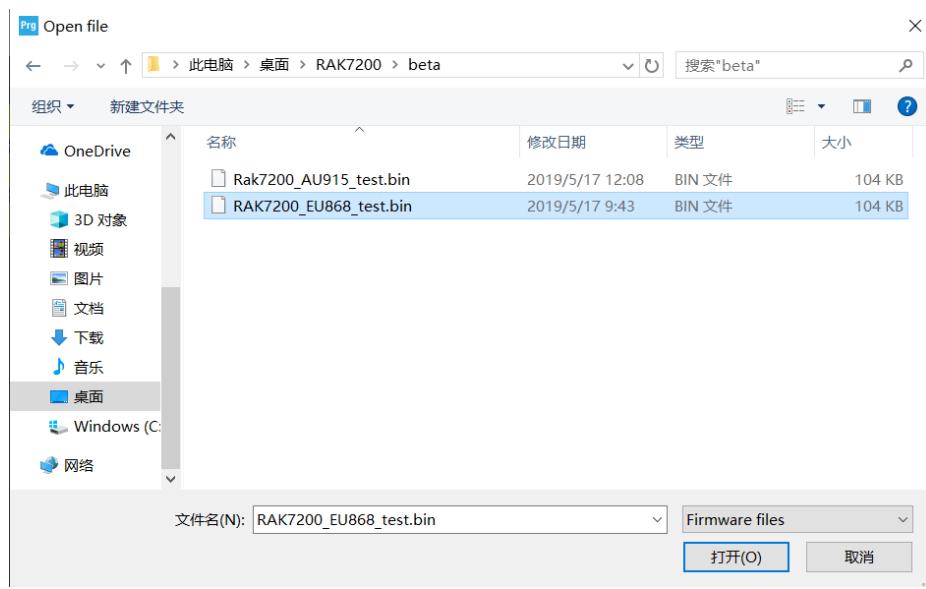


Figure 70: STM32CubeProgrammer Select New Firmware

- Once the new firmware is loaded, its content is displayed in the tool in hexadecimal format. Now, As shown in the Figure 71, press the “Download” button to start the burning process. Once finished, a pop-up message will appear.

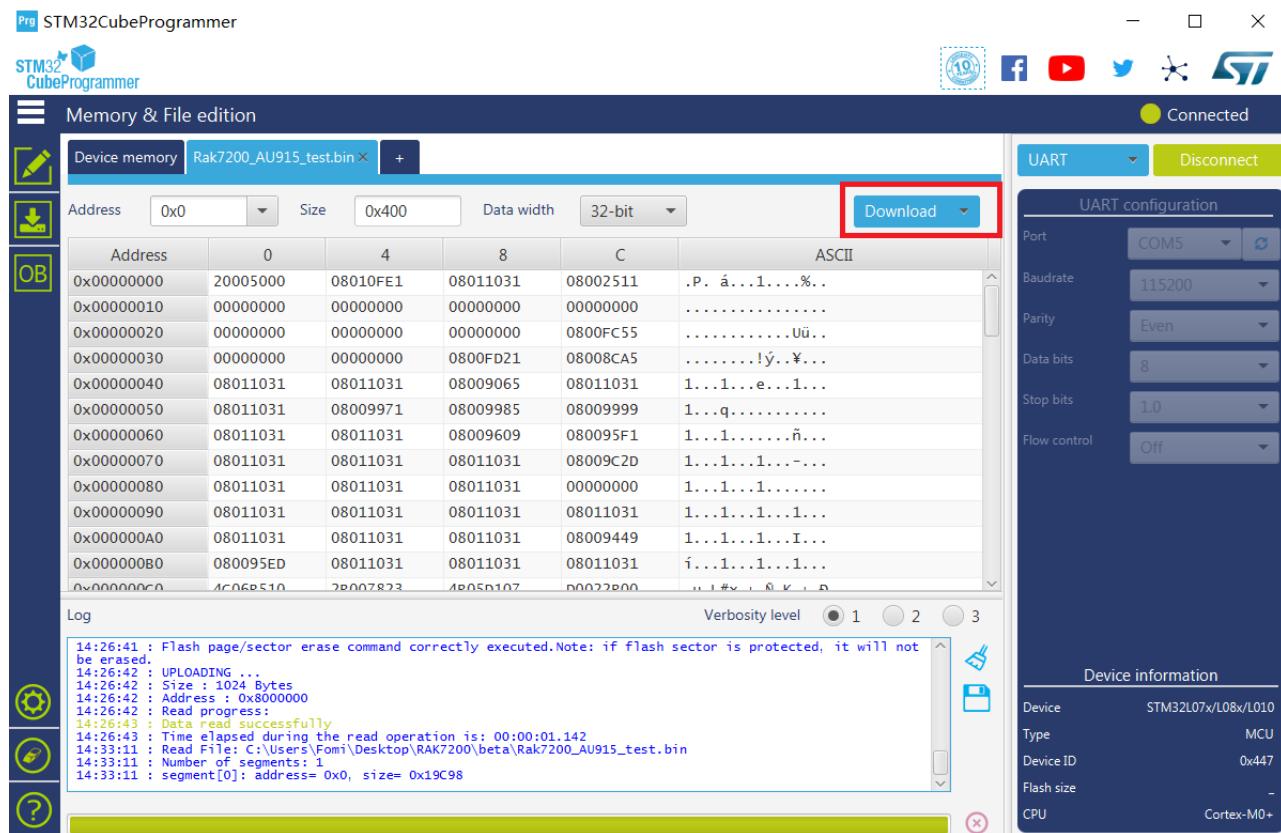


Figure 71: Download or save the new firmware

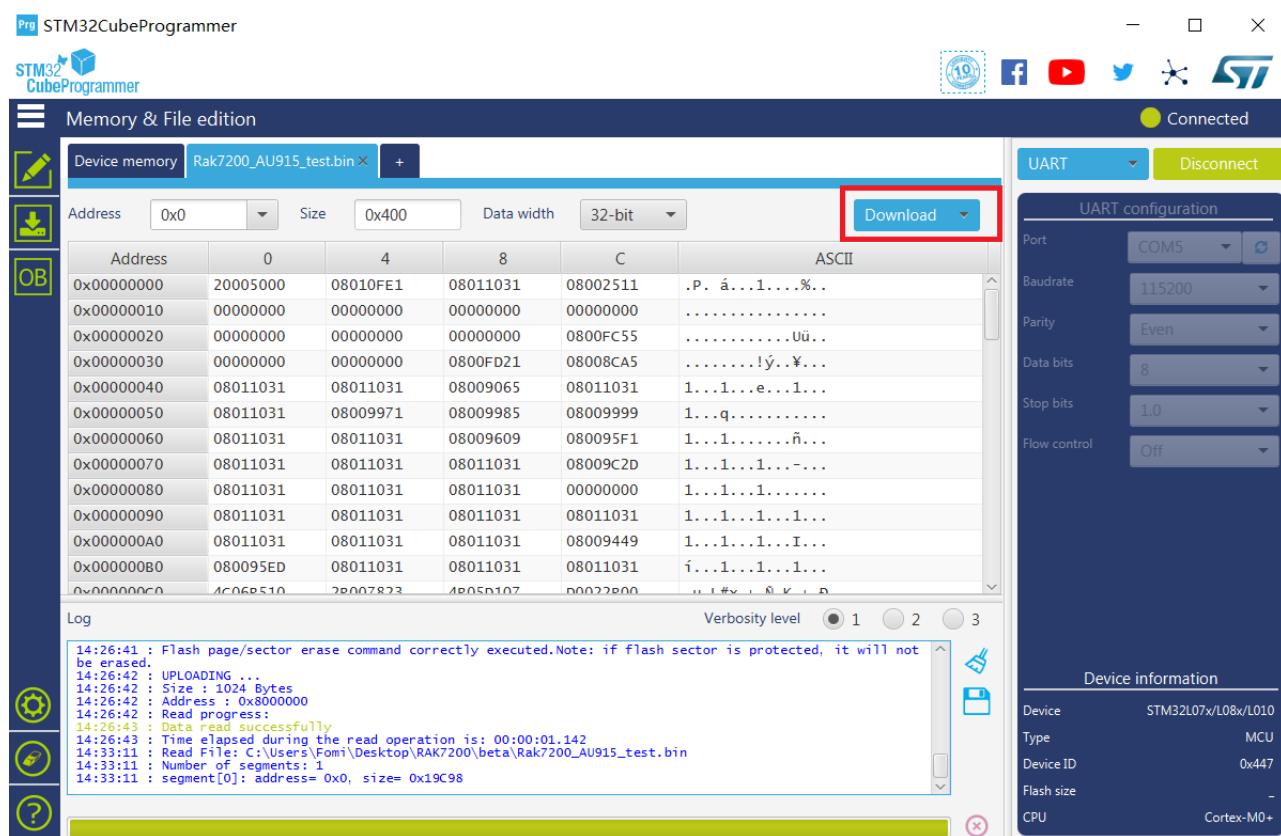


Figure 72: Download or save the new firmware

- To finalize the firmware upgrade process, Disconnect and close the STM32CubeProgrammer tool.

 NOTE:

Don't forget to reset the module in order to exit from the boot mode.