

# RAK7204 Quick Start Guide

## Prerequisites

### What do you need?

Before going through each and every step in the installation guide of the RAK7204 WisNode Sense Home, make sure to prepare the necessary items listed below:

1. **RAK7204 WisNode Sense Home**
2. Micro USB Cable
3. Gateway in Range, for Testing
4. Windows PC

 **NOTE:**

This device released by RAKwireless is already pre-loaded with its latest firmware upon manufacturing. If you want to have your device's firmware and bootloader burned, please refer to the sections below:

1. [Burning the Bootloader](#)
2. [Burning the Firmware](#)

### What's Included in the Package?

- 1pc - RAK7204 WisNode Sense Home
- 1pc - LoRa Antenna
- 1pc - 3500mAh Lithium Battery

## Product Configuration

### Interfacing with the RAK7204 WisNode Sense Home

In order for you to be able to interface with the RAK7204 WisNode Sense Home with your Windows Machine, you need to download the RAK Serial Port Tool [here](#).

 **WARNING**

The included battery is **non rechargeable**. Please do note that when configuring the device, you have to connect the battery first in order for it to work.

- Connect your RAK7204 WisNode Sense Home in your Windows Machine using the provided micro-usb cable.

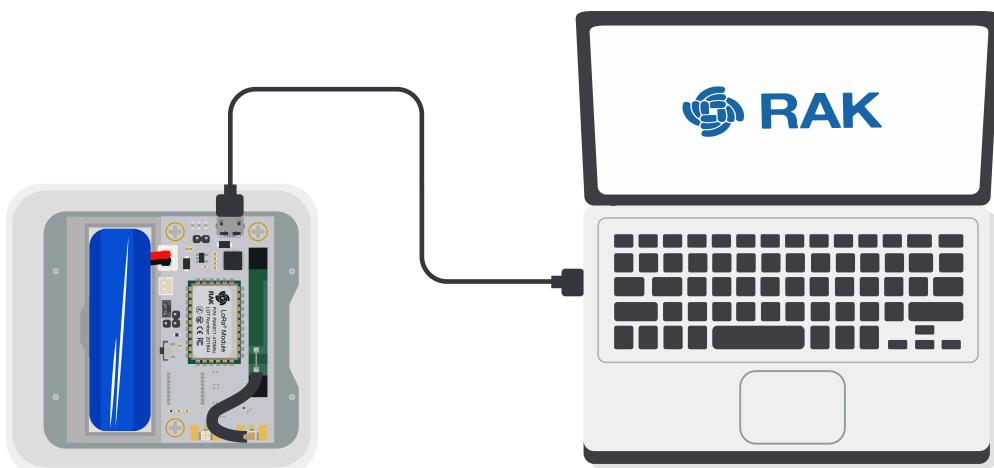


Figure 1: RAK7204 WisNode Sense Home to Laptop Connection

### WARNING

The pin distance of the battery connector is **2.0mm**. Reverse connection or short circuit may damage the device and may cause overheating and combustion of the battery. Therefore, when replacing the battery, it is necessary to strictly confirm whether the positive and negative poles of the connector are correct.

- Open the RAK Serial Port Tool :

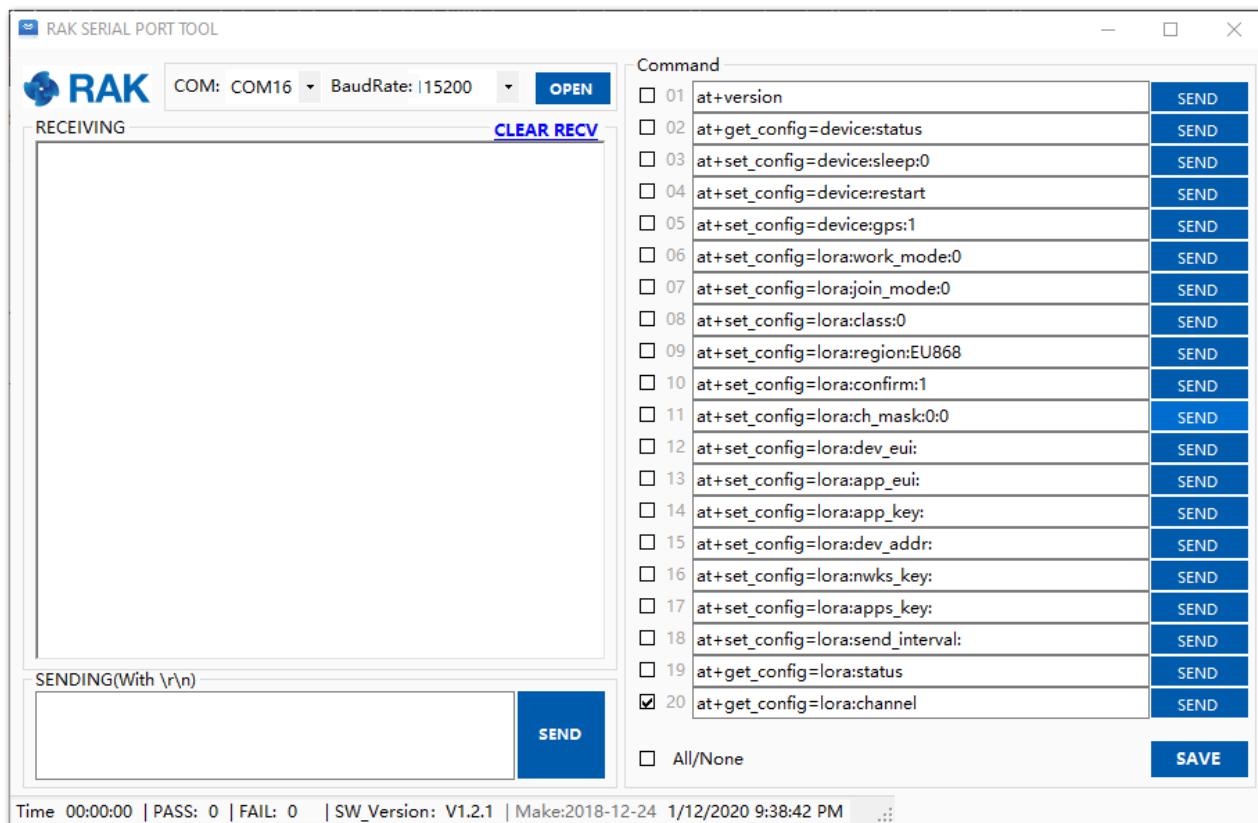


Figure 2: RAK Serial Port Tool

- In choosing the correct COM Port number for your device. Go to your Device Manager by pressing : Windows + R and type `devmgmt.msc` or search in the Start Menu

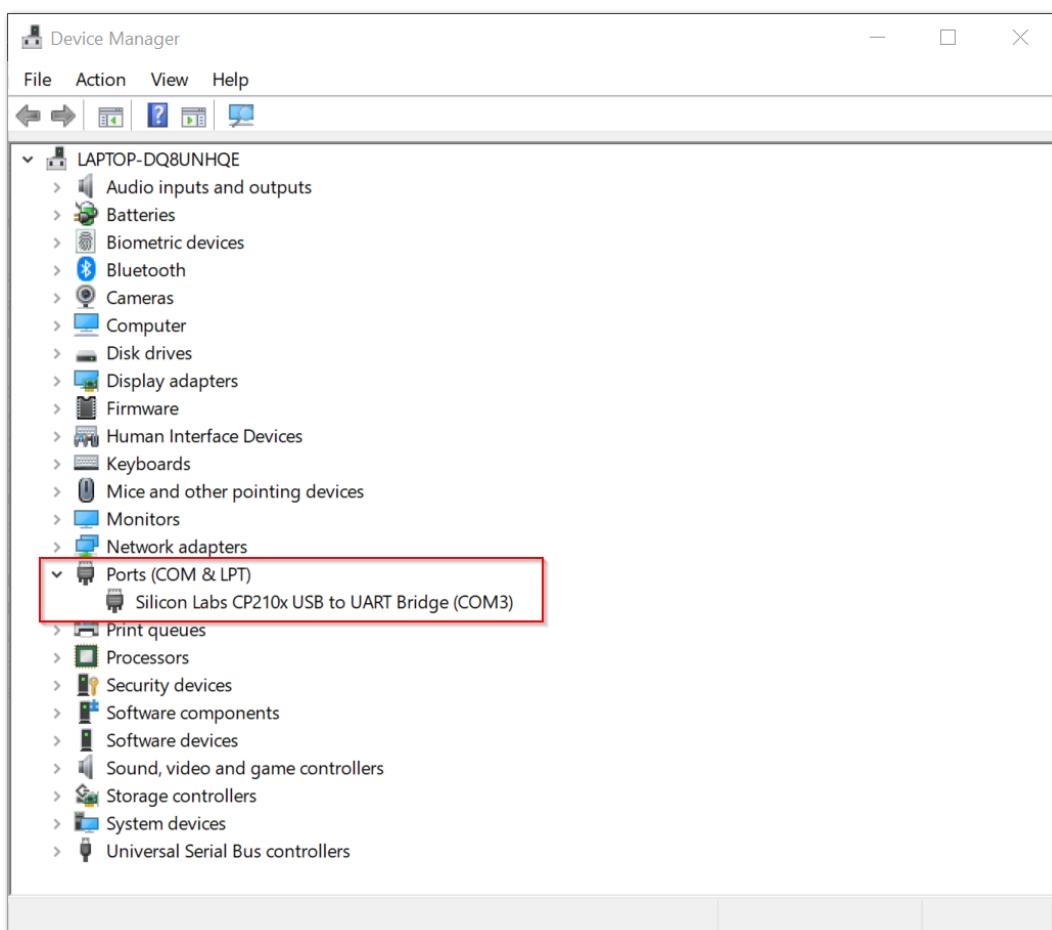


Figure 3: Device Manager

- Look for Ports (COM & LPT) and Find the name Silicon Labs CP210x USB to UART Bridge and take note of the COM Port Number.

 **NOTE:**

If you didn't find any Port with the name Silicon Labs CP210x USB to UART Bridge, make sure you have downloaded the CP210x USB Drivers in your Machine.

- Choose the Correct Port Number from the device manager and the Correct Baudrate then click Open:

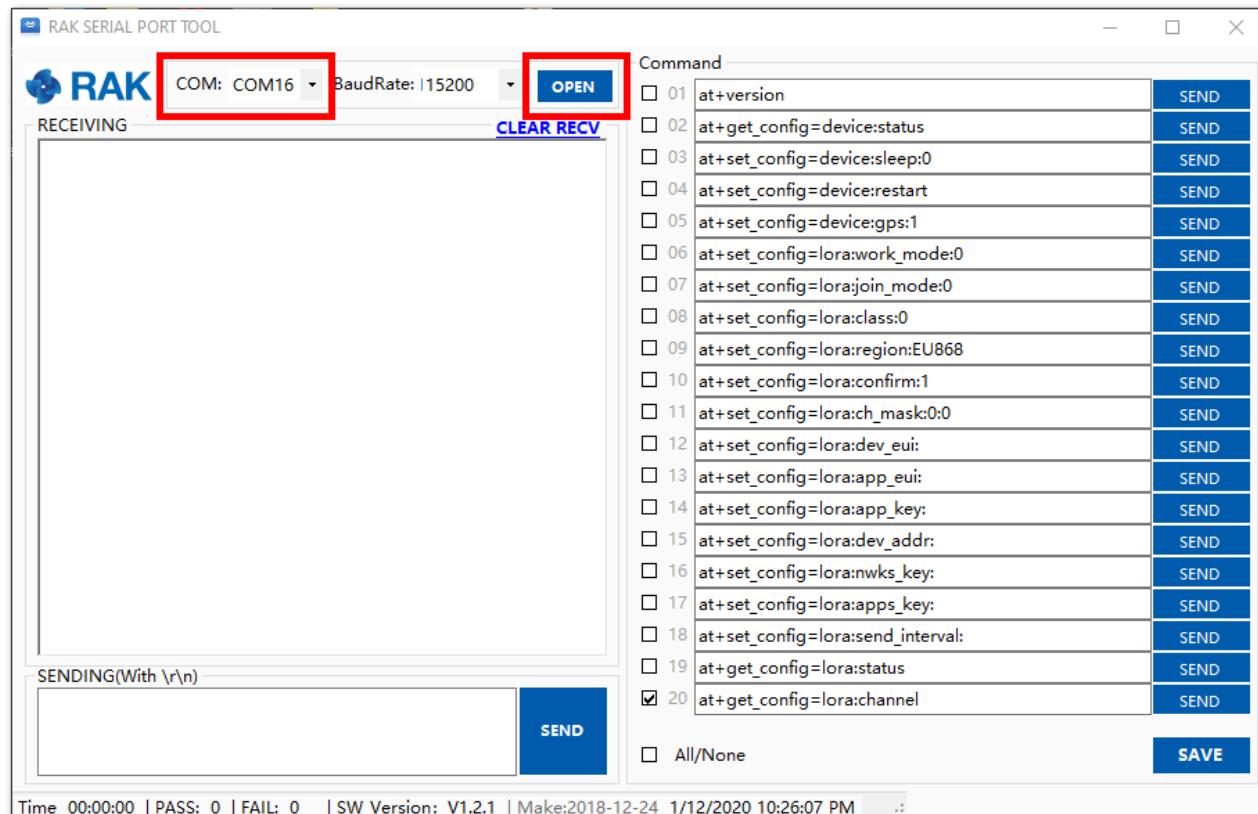
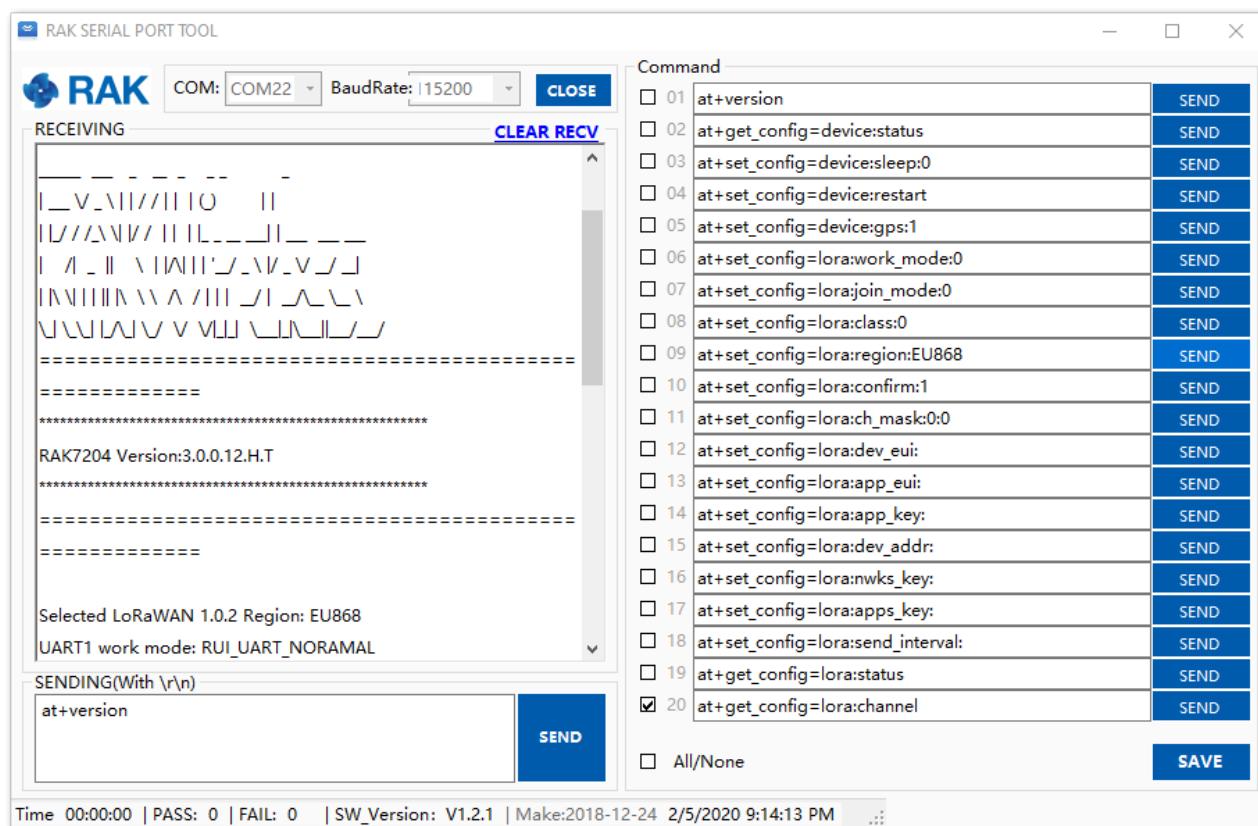


Figure 4: Correct Port Number and Correct Baud rate



**Figure 5:** Connection Success

## Connecting to Helium (The People's Network)

Helium has quickly become the most widespread LPWAN communal network with more than 27,000 devices deployed globally. All our node products are compatible with it and the process of adding a device to the network is intuitive and straightforward. This section will focus on giving a brief guide on how to connect the RAK7204 to the network console, assuming that there is a Helium Hotspot within range.

Log in or create your account [here](#) :

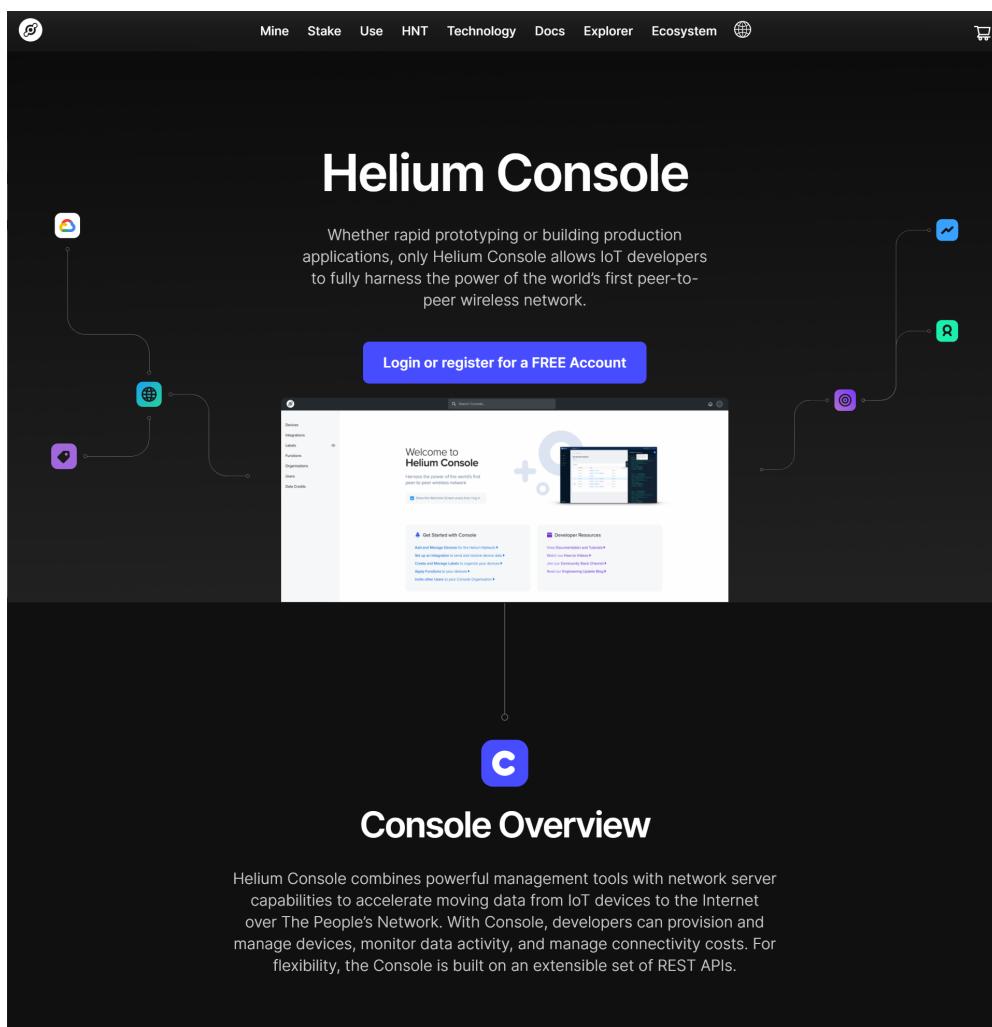


Figure 6: Helium Console

Once registered/logged in you will end up at the home page where you can see your function tree on the left and your DC balance at the tops as well as a number of useful links.

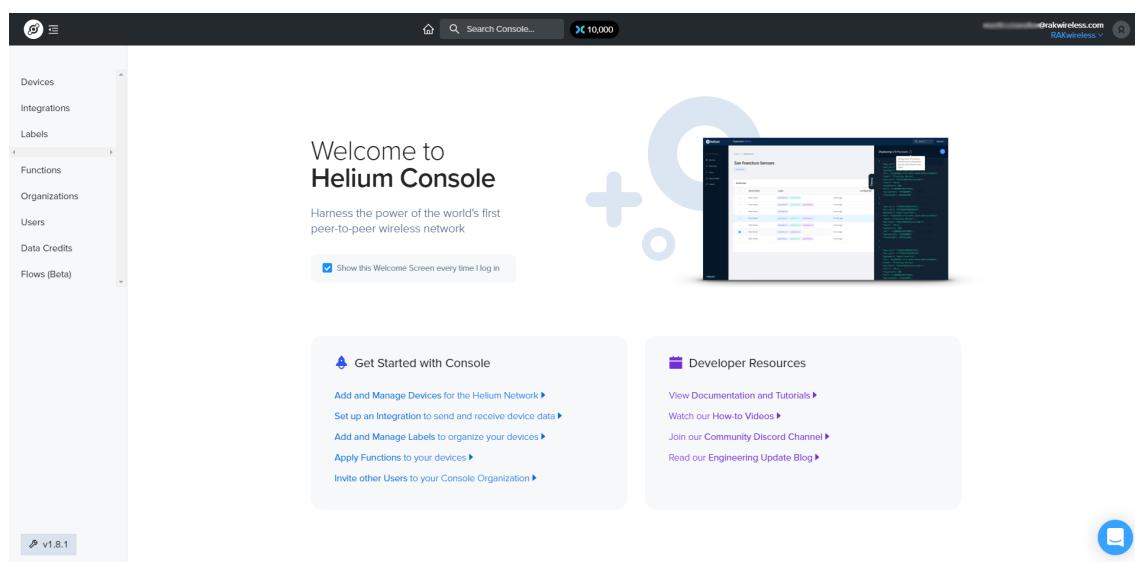
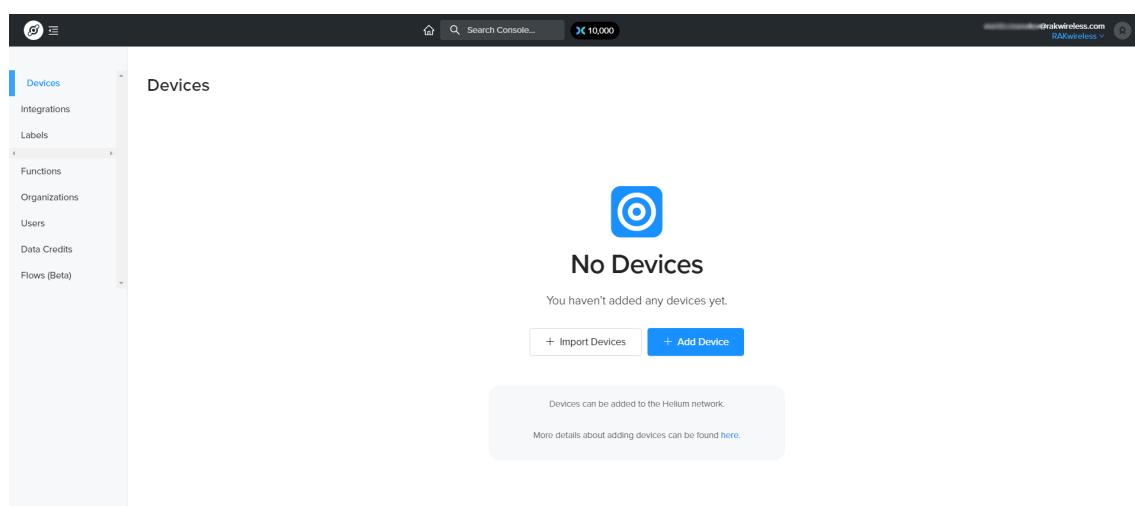


Figure 7: Helium Console home screen

Go to the "Devices" section in the function tree. If this is your first time doing this there will be no devices registered. Click the "+ Add Device" button to get started



**Figure 8:** Helium Devices

A window will pop up with a set of field containing the device parameters required for its registration (Figure 9):

The dialog box is titled 'Add a New Device'. It contains four input fields for device parameters: 'Name' (RAK7204), 'Dev EUI' (2F0C14928D...), 'App EUI' (C9EED4E9E8...), and 'App Key' (D5AB9CE22B961B1C2707472A60...). Below these is a field for 'Attach a Label (Optional)'. At the bottom are 'Cancel' and 'Submit' buttons.

**Figure 9:** Add a New Device

Fill a name of your choosing. The Dev EUI, App EUI and App Key (press the eye icon to reveal the values) will have random values generated for you by default. You can manually replace them with values of your own. For the purpose of this tutorial we will stay with the default. Press the "Submit" button and you are done.

The screenshot shows the 'Devices' page again. The sidebar is identical. The main area now displays a table with one row for the registered device. The columns are: Device Name (RAK7204), Device EUI (2F0C14928D...), Labels (None), Integrations, Frame Up, Frame Down, Packets Transferred, DC Used, and Date Activated (Apr 20, 2021 10:36 AM). There are buttons for 'Edit Columns', '10 results', 'Quick Action', and navigation arrows.

**Figure 10:** Helium Devices

Now your RAK7204 is registered and is awaiting activation. For this to happen we need to import the Dev EUI, App EUI and App Key in the RAK7204 using the RAK Serial Port Tool

Open the tool, select the desired port (default baud rate) and open it. Then start importing your settings.

Configure your LoRa band and activation mode. This tutorial will be using the EU868 band and OTAA (only option available for now with Helium) with device class A (default one, does not need configuring).

- Regional band and activation mode setting

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

```
at+set_config=lora:region:EU868
```

- Enter the Dev UI

Use the command below by replacing the XXXX with your Device EUI from the Helium console:

```
at+set_config=lora:dev_eui:XXXX
```

- Enter the App EUI

Same as with the Device EUI, replace the XXXX with your value:

```
at+set_config=lora:app_eui:XXXX
```

- Enter App Key

Finally fill in the App key with the command:

```
at++set_config=lora:app_key:XXXX
```

- Join Network

Finish executing the join command in order for the node to initiate the join procedure.

Once the procedure is initiated and successfully complete you will have a notification in the serial console

```
at+join
```

Your output should resemble the one in Figure 11:

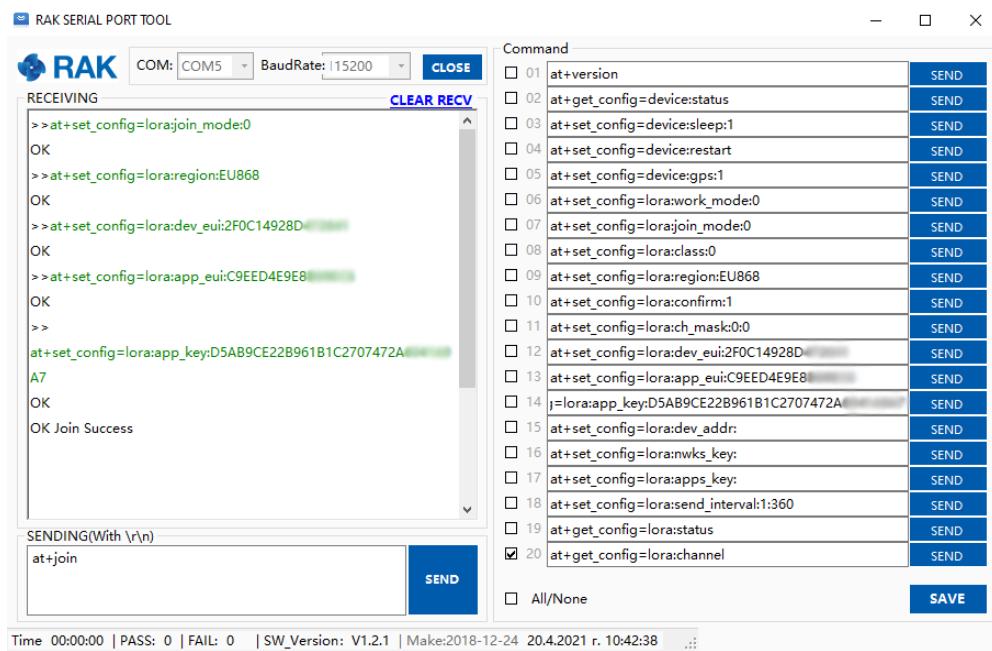


Figure 11: RAK7204 EU868 and key

If you take a look at the Helium console you will also see the join request packets both in the graph and event log. Your node is now a part of the Helium Network.

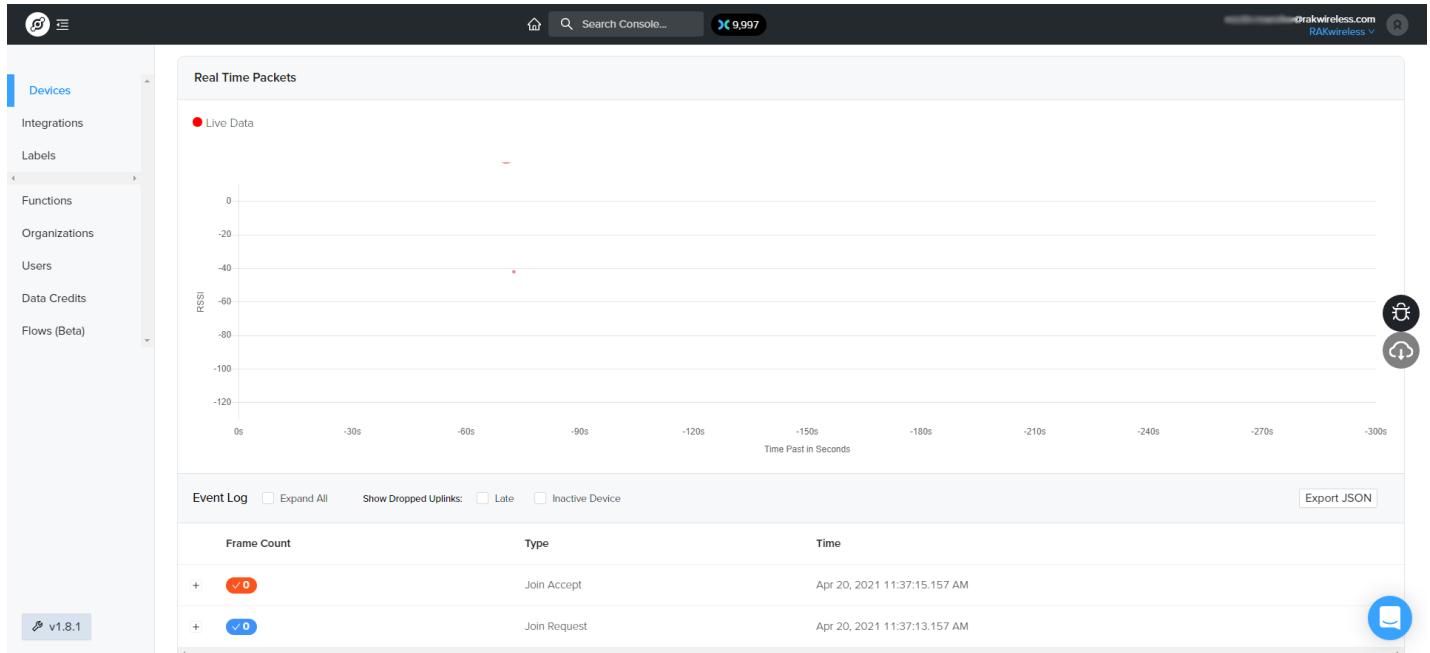
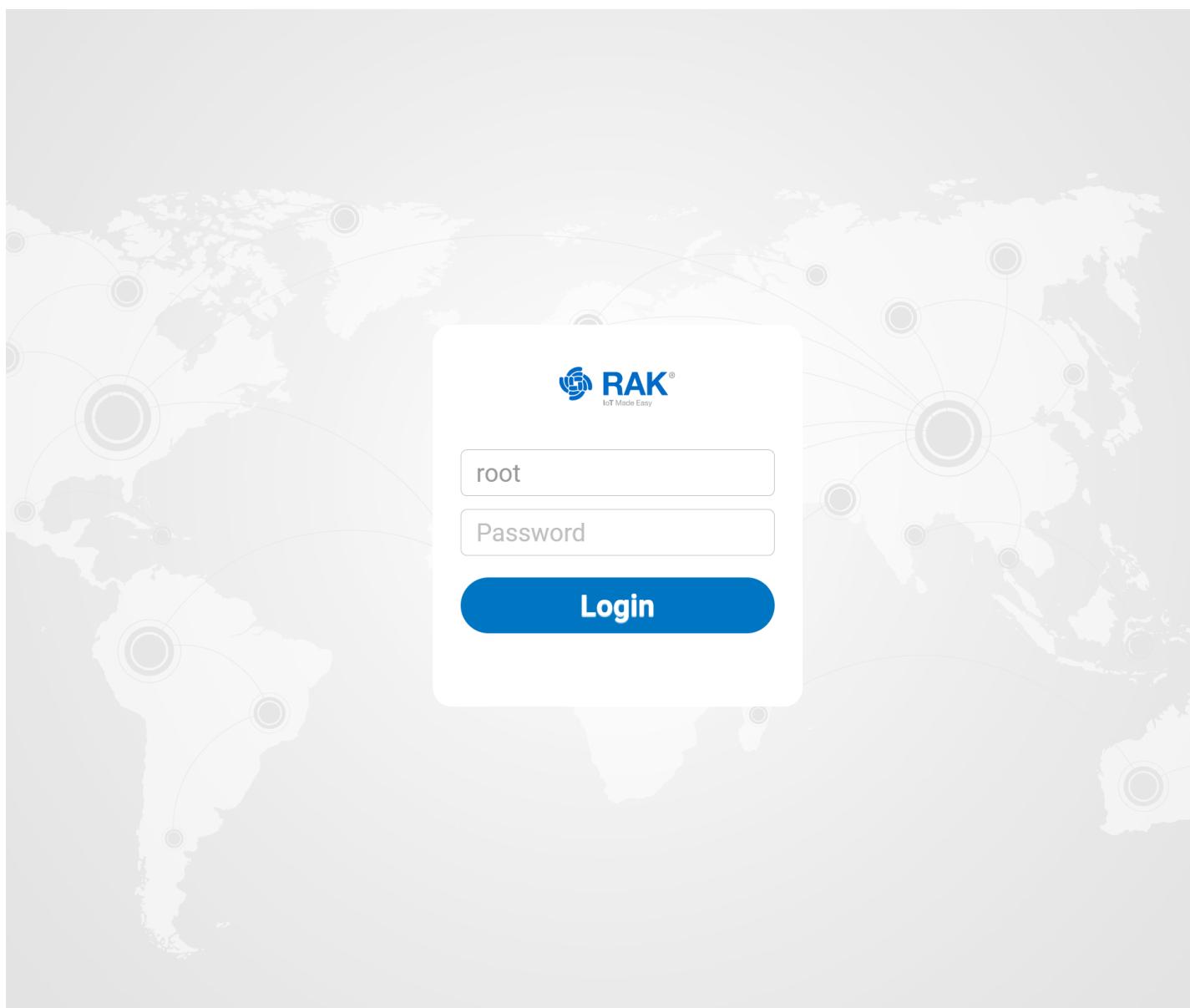


Figure 12: Helium console live device data

## Configuring the Built-in Server

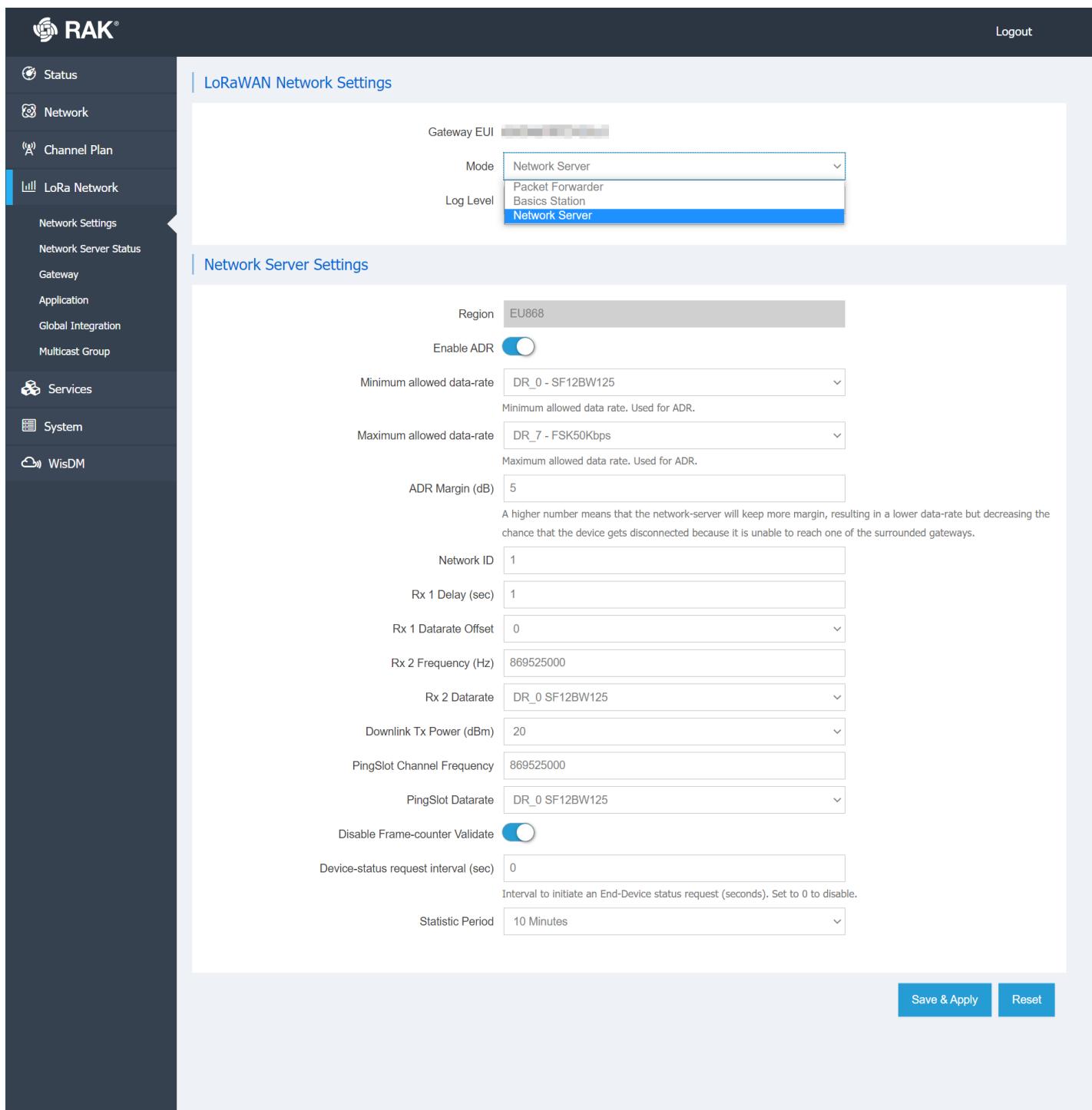
The procedure for connecting the RAK7204 to the Built-in Server is straightforward and includes going through the following steps:

1. Open a browser and access the Web UI of your WisGate Edge Gateway by entering its IP address in the browser address bar. You will be greeted by the Log in screen. The default user name/password are both "root".



**Figure 13:** Built-in Server Log in screen

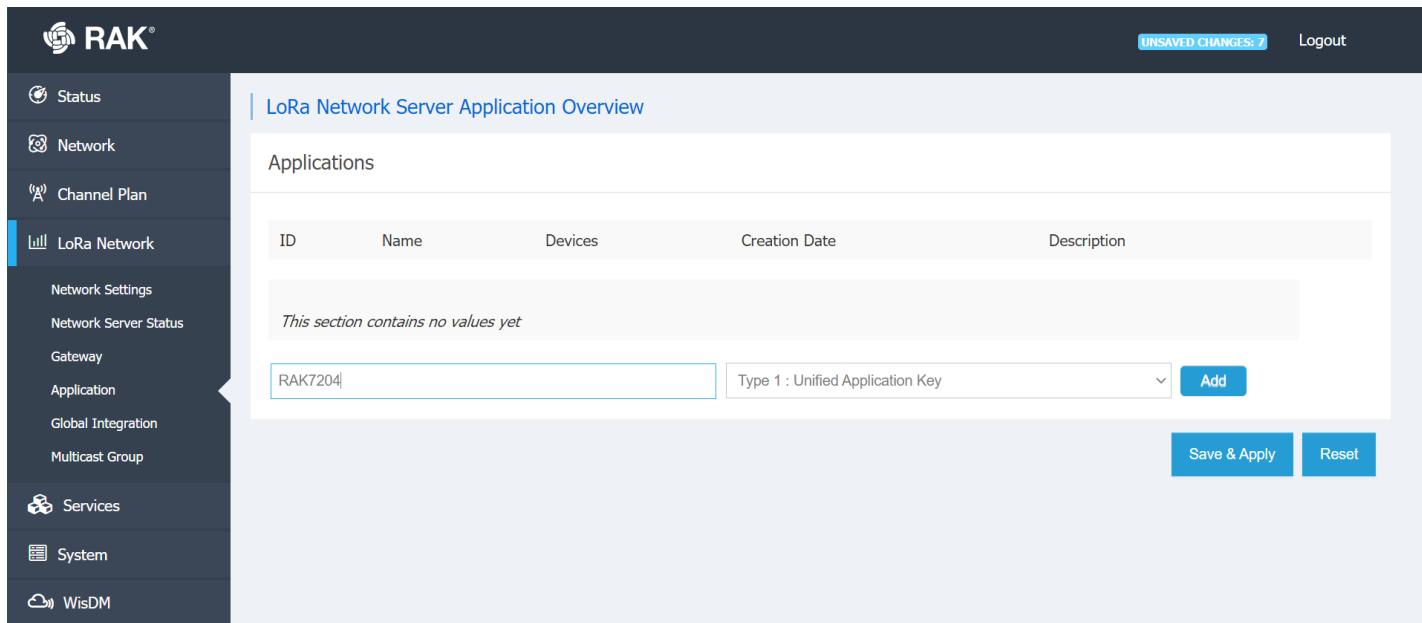
Once you have logged in make sure you are in Network Server mode. By default the gateway is working in this mode, so if this is the first time you are configuring it it should already be set. If this is not the case go to the Network Server menu, the Network Settings sub menu and select the Network Server option in the Mode drop-down menu:



The screenshot shows the RAK Documentation Center interface. On the left, a sidebar menu includes options like Status, Network, Channel Plan, LoRa Network (which is selected), Network Settings, Network Server Status, Gateway, Application, Global Integration, Multicast Group, Services, System, and WisDM. The main content area is titled "LoRaWAN Network Settings". It displays "Gateway EUI" as a redacted string. Under "Mode", "Network Server" is selected from a dropdown menu. The "Log Level" dropdown also lists "Network Server". Below this, the "Network Server Settings" section contains various configuration fields: "Region" set to EU868, "Enable ADR" turned on, "Minimum allowed data-rate" set to DR\_0 - SF12BW125, "Maximum allowed data-rate" set to DR\_7 - FSK50Kbps, and "ADR Margin (dB)" set to 5. Other settings include "Network ID" (1), "Rx 1 Delay (sec)" (1), "Rx 1 Datarate Offset" (0), "Rx 2 Frequency (Hz)" (869525000), "Rx 2 Datarate" (DR\_0 SF12BW125), "Downlink Tx Power (dBm)" (20), "PingSlot Channel Frequency" (869525000), "PingSlot Datarate" (DR\_0 SF12BW125), "Disable Frame-counter Validate" turned off, "Device-status request interval (sec)" (0), and "Statistic Period" (10 Minutes). At the bottom right are "Save & Apply" and "Reset" buttons.

Figure 14: Network Server mode

- Start the Device registration process by going to the Application sub menu in the LoRa Network section and creating an application. Enter a name in the field, leave the default type and press "Save & Apply"

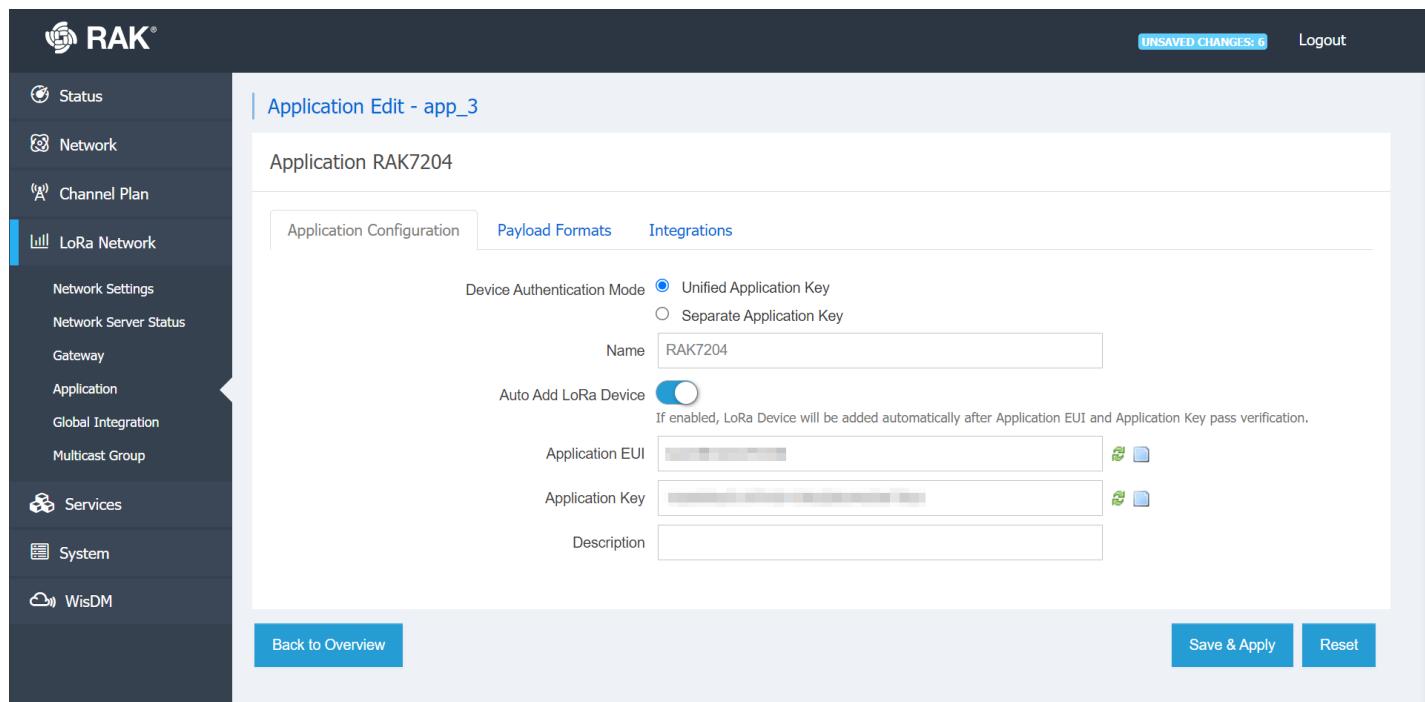


The screenshot shows the RAK Documentation Center interface. The sidebar menu is identical to Figure 14. The main content area is titled "LoRa Network Server Application Overview". It has a "Applications" section with a table header: ID, Name, Devices, Creation Date, and Description. Below the table, a message says "This section contains no values yet". A search bar contains "RAK7204" and a dropdown menu is set to "Type 1 : Unified Application Key". At the bottom right are "Save & Apply" and "Reset" buttons.

Figure 15: Creating your application

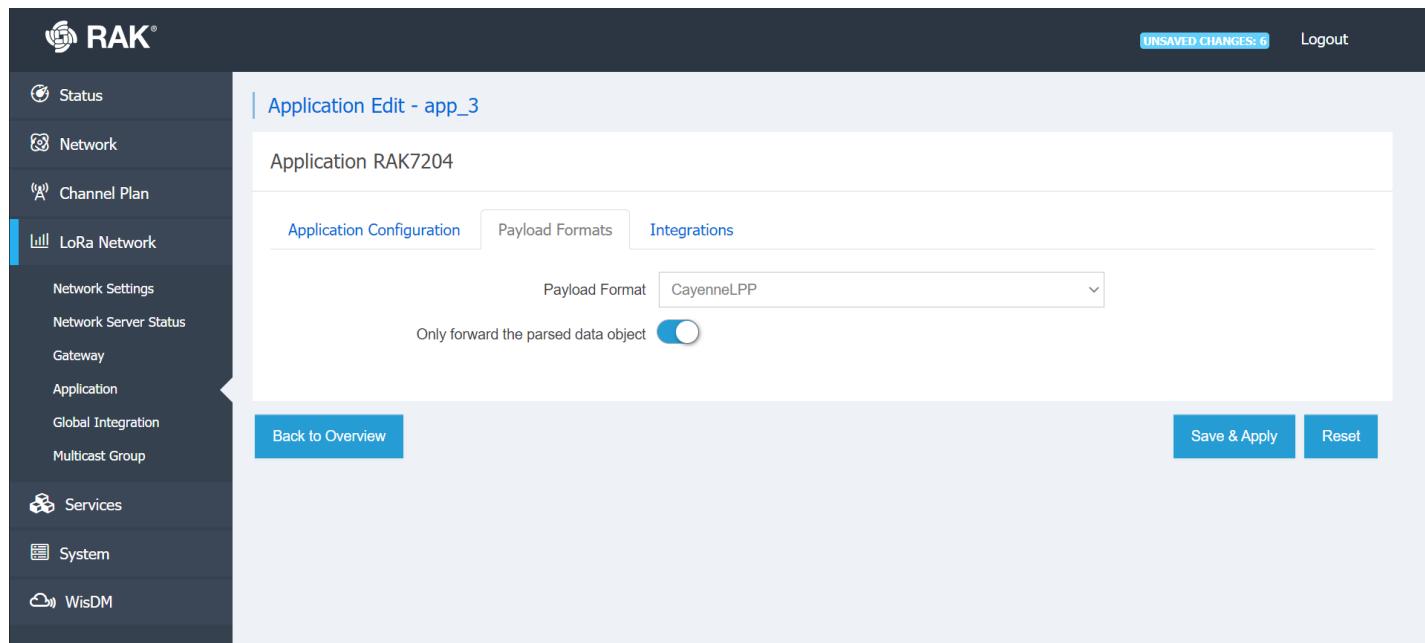
In the following screen you will see fields for the application parameters.

In the Application Configuration sub-window enable the "Auto Add LoRa Device" functionality. Generate a random Application EUI and Application Key via the green arrow button next to the text fields. Note these down as you will need them for the RAK7204 configuration.



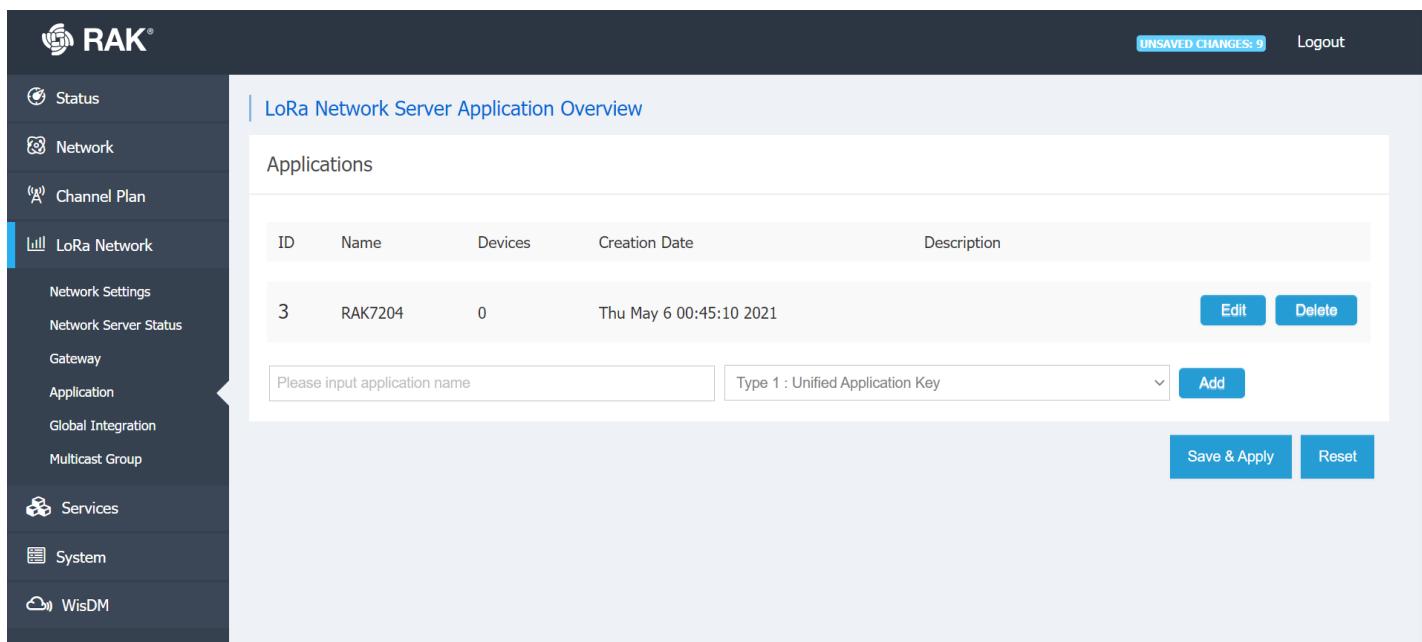
**Figure 16:** Application configuration

Move to the Payload Formats sub-window and set the payload format to "CayenneLPP" via the drop down menu. This is the format that the RAK7204 uses, thus enabling this functionality will allow you to see the parsed data in the Application Server. Finally enable the "Only forward the parsed data object" functionality, press the "Save & Apply" button to finalize the configuration changes.



**Figure 17:** Payload Format

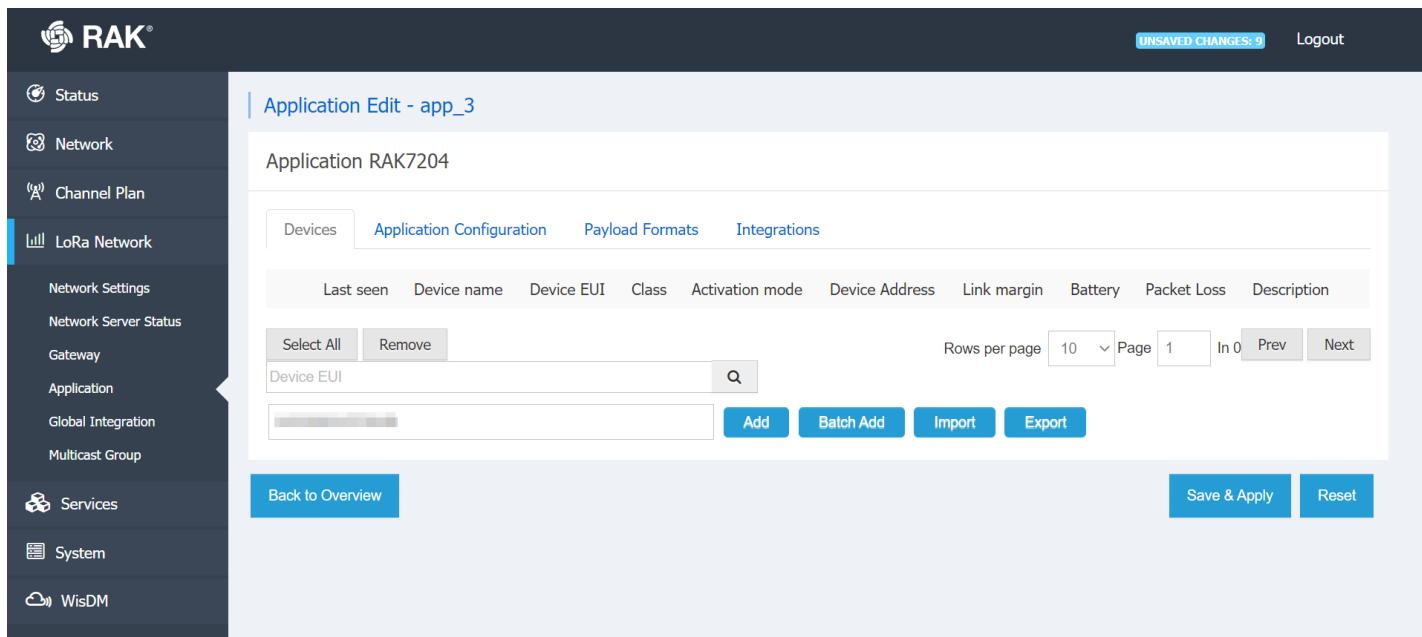
3. Now that your application has been created you need to go to the device section by pressing the "Edit" button.



The screenshot shows the 'LoRa Network Server Application Overview' page. On the left, a sidebar menu includes options like Status, Network, Channel Plan, LoRa Network (which is selected), Network Settings, Network Server Status, Gateway, Application, Global Integration, Multicast Group, Services, System, and WisDM. The main content area displays a table titled 'Applications' with one row for 'RAK7204'. The table columns are ID, Name, Devices, Creation Date, and Description. Below the table are input fields for 'Please input application name' and 'Type 1 : Unified Application Key', followed by 'Add', 'Save & Apply', and 'Reset' buttons.

**Figure 18:** Editing an application

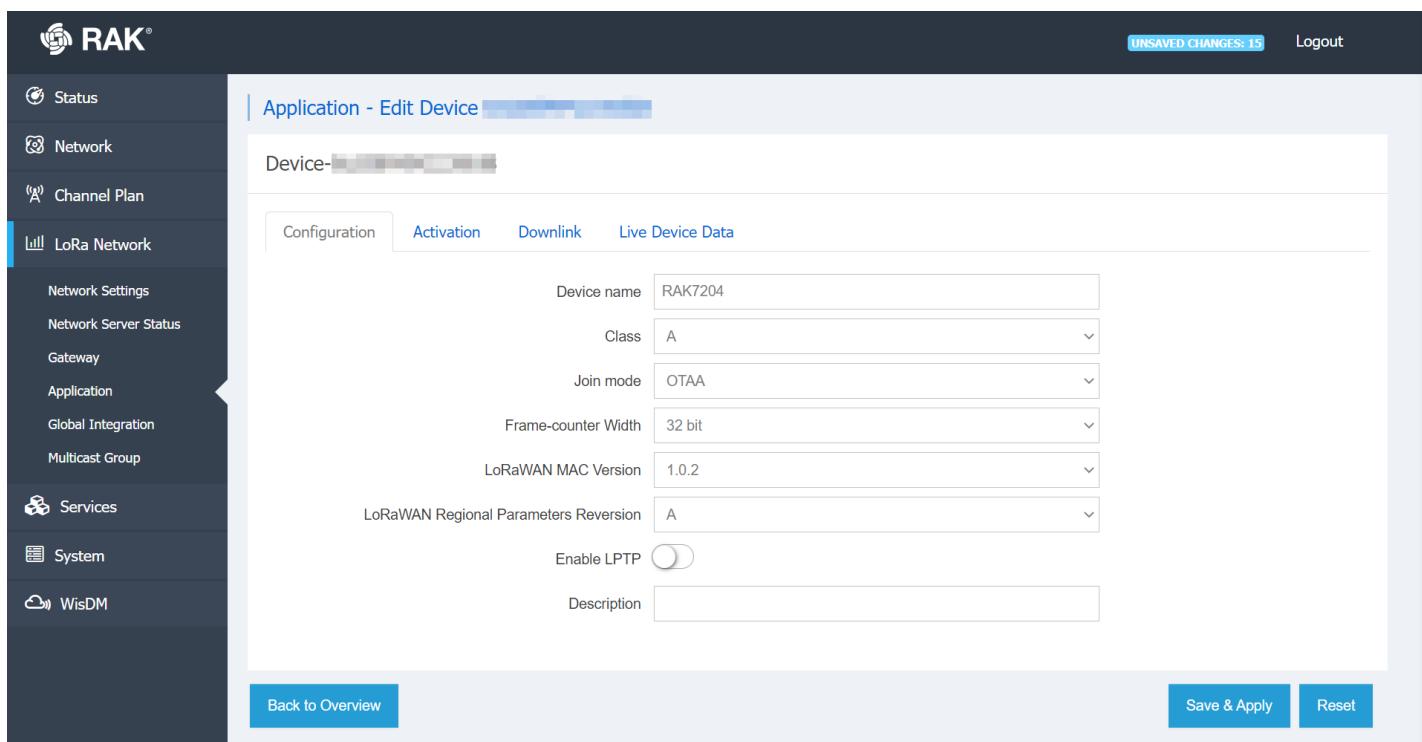
Now you are in the "Devices" section and you can add a device by entering its Device EUI, which you can find on a sticker on the back of the RAK7204. Press the "Add" button to proceed.



The screenshot shows the 'Application Edit - app\_3' page for 'Application RAK7204'. The sidebar is identical to Figure 18. The main content has tabs for Devices, Application Configuration, Payload Formats, and Integrations. The Devices tab is active, showing a table with columns: Last seen, Device name, Device EUI, Class, Activation mode, Device Address, Link margin, Battery, Packet Loss, and Description. There are buttons for Select All, Remove, and a search bar for Device EUI. Below the table are buttons for Add, Batch Add, Import, and Export. At the bottom are 'Back to Overview', 'Save & Apply', and 'Reset' buttons.

**Figure 19:** Adding a device

In the configuration screen enter a name for your device and leave the rest of the parameters with their default values (the Description is optional). Finish by pressing the "Save & Apply" button.



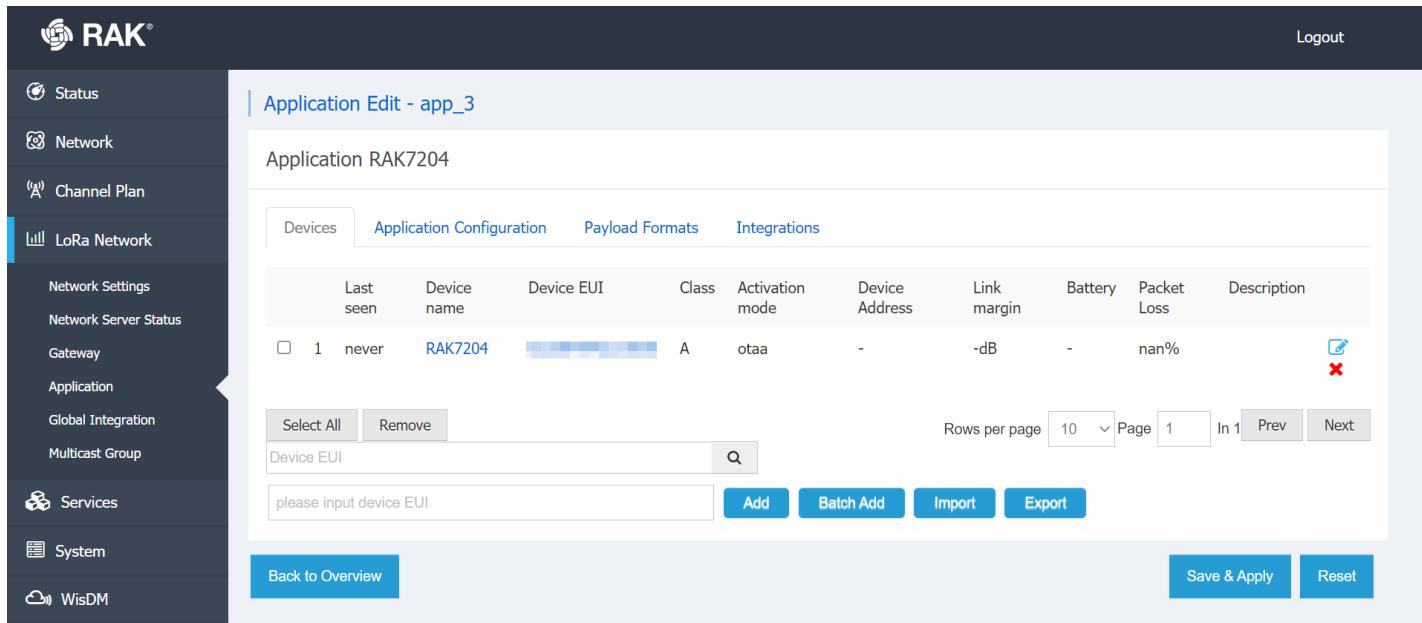
Device-[REDACTED]

Configuration	Activation	Downlink	Live Device Data
Device name: RAK7204	Class: A	Join mode: OTAA	Frame-counter Width: 32 bit
LoRaWAN MAC Version: 1.0.2	LoRaWAN Regional Parameters Reversion: A	Enable LPTP: <input type="checkbox"/>	
Description: <span style="background-color: #e0e0e0; border: 1px solid #ccc; padding: 2px 10px;">[REDACTED]</span>			

[Back to Overview](#) [Save & Apply](#) [Reset](#)

Figure 20: Device parameters

Your Device is now added to the Built-in server and you should see it in the "Devices" section.



Application RAK7204

Devices	Application Configuration	Payload Formats	Integrations
Last seen: 1 never Device name: RAK7204 Device EUI: <span style="background-color: #e0e0e0; border: 1px solid #ccc; padding: 2px 10px;">[REDACTED]</span> Class: A Activation mode: otaa Device Address: - Link margin: -dB Battery: - Packet Loss: nan% <a href="#"></a> <a href="#"></a>	Select All Remove	Rows per page: 10 Page: 1 In 1 Prev Next	Device EUI: <input type="text"/> <a href="#"></a> please input device EUI <a href="#">Add</a> <a href="#">Batch Add</a> <a href="#">Import</a> <a href="#">Export</a>

[Back to Overview](#) [Save & Apply](#) [Reset](#)

Figure 21: Devices section

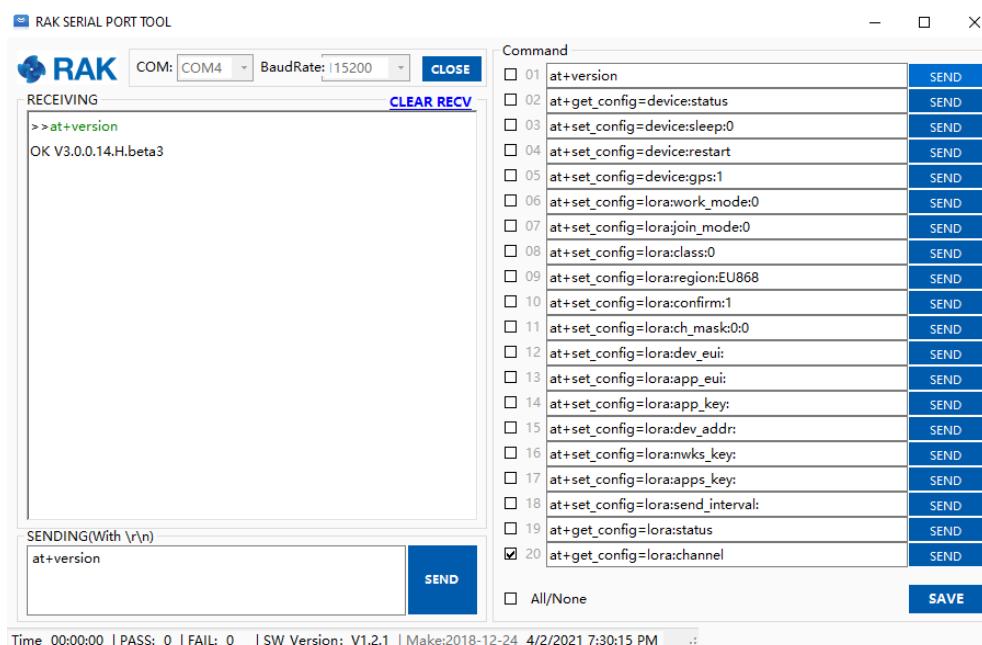
Next you need to import the same configuration in the RAK7204.

## Configuring the RAK7204

Start by connecting to your RAK7204 as described in the [Product Configuration](#) section of the Quick Start Guide.

Open a terminal and check your firmware version using the command:

```
at+version
```



**Figure 22:** Firmware version

If there is a newer firmware version at the [link](#), update using this [procedure](#).

Follow the steps in order to update the RAK7204 configuration and connect it to the Built-in Server

1. Import the **Device EUI** (from the label on the back), **Application EUI** and **Application Key** (you should have noted them down as instructed in the previous section), by executing the following commands in order (replace the "xxxx" with your values).

#### Device EUI

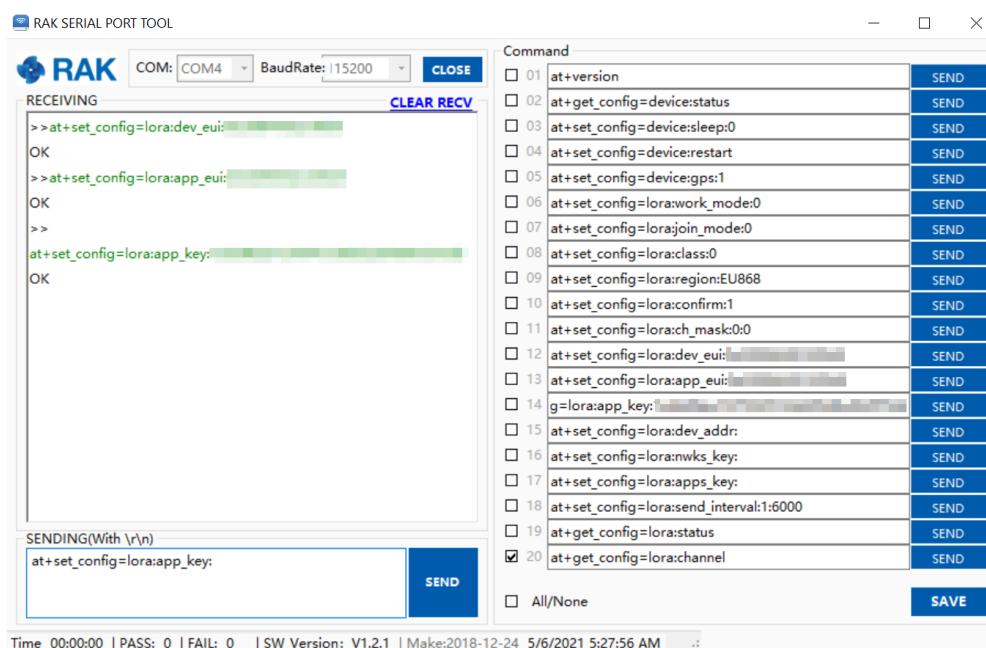
```
at+set_config=lora:dev_eui:xxxx
```

#### Application EUI

```
at+set_config=lora:app_eui:xxxx
```

#### Application Key

```
at+set_config=lora:app_key:xxxx
```



**Figure 23:** Importing EUIs and Key

- Set the activation parameters (**LoRa Region, Device Class, Activation Mode**). This example will use the **EU868** regional band, **Class A, OTAA** activation mode.

#### LoRa Region

```
at+set_config=lora:region:EU868
```

#### Device Class

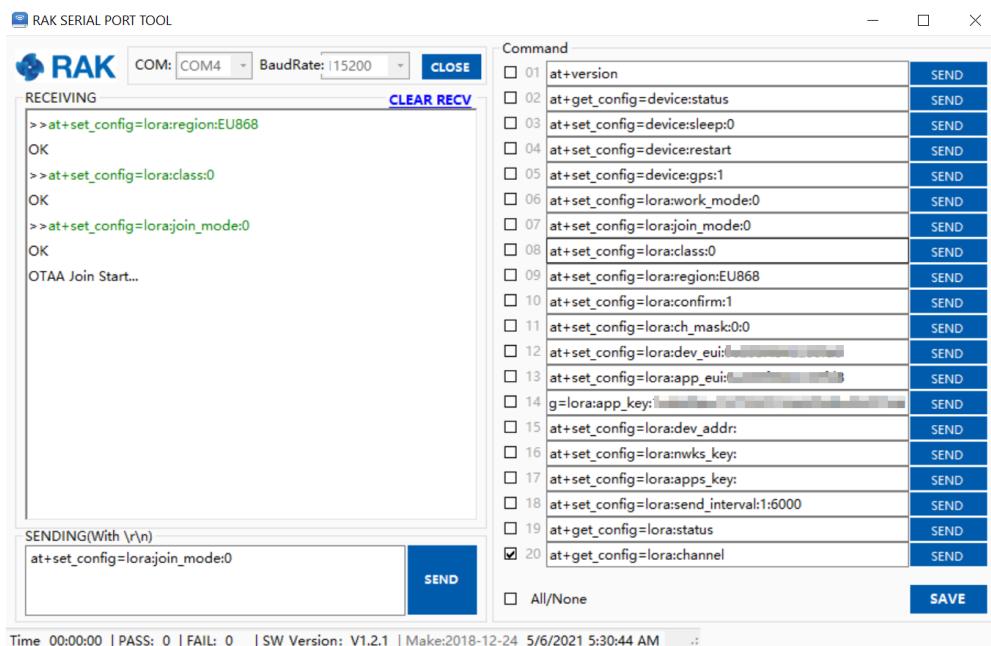
```
at+set_config=lora:class:0
```

#### Activation Mode

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

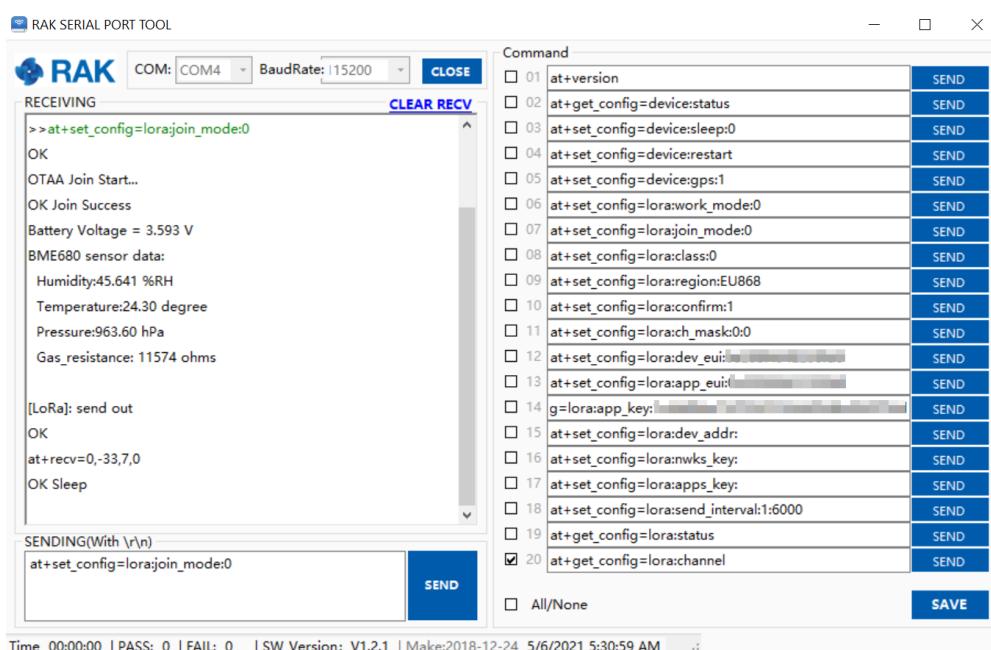
After executing the last command the node will automatically start the join procedure. If for some reason this is not the case, you can always use the command below to start the activation process:

```
at+join
```



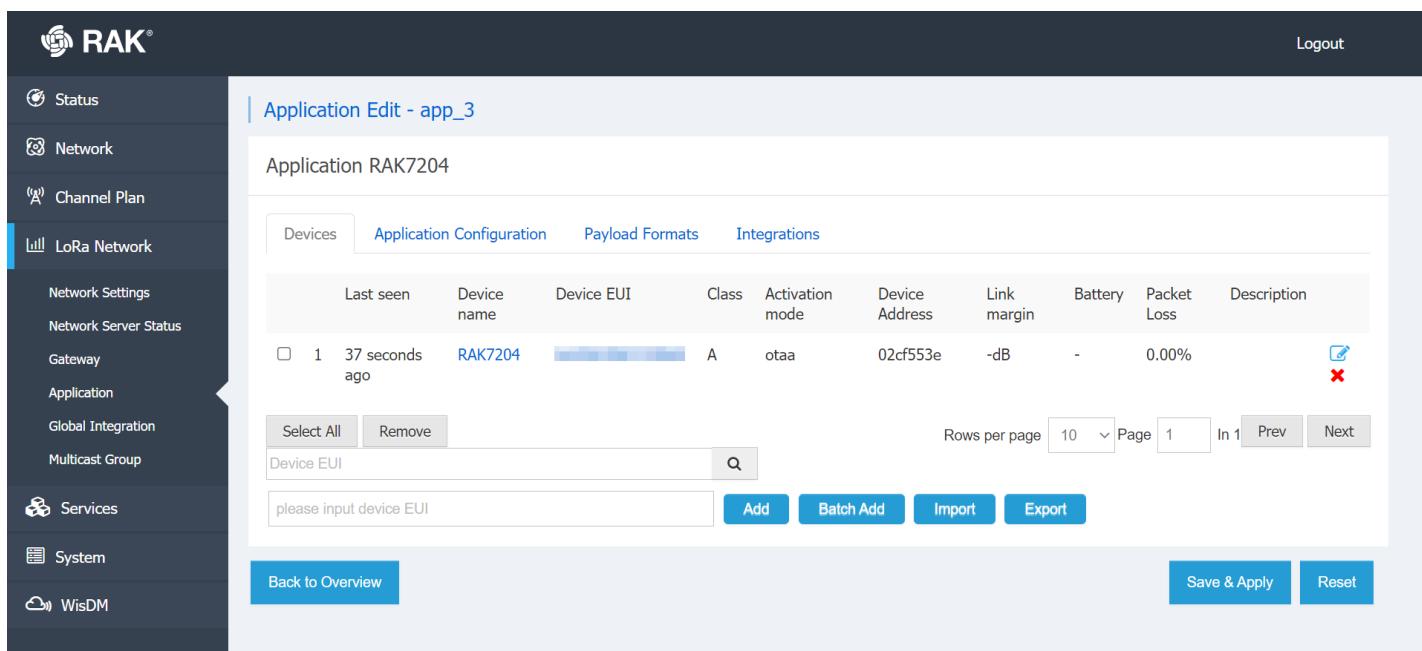
**Figure 24:** Activation parameters

Upon successful registration the following response will be shown in the Serial Tool.



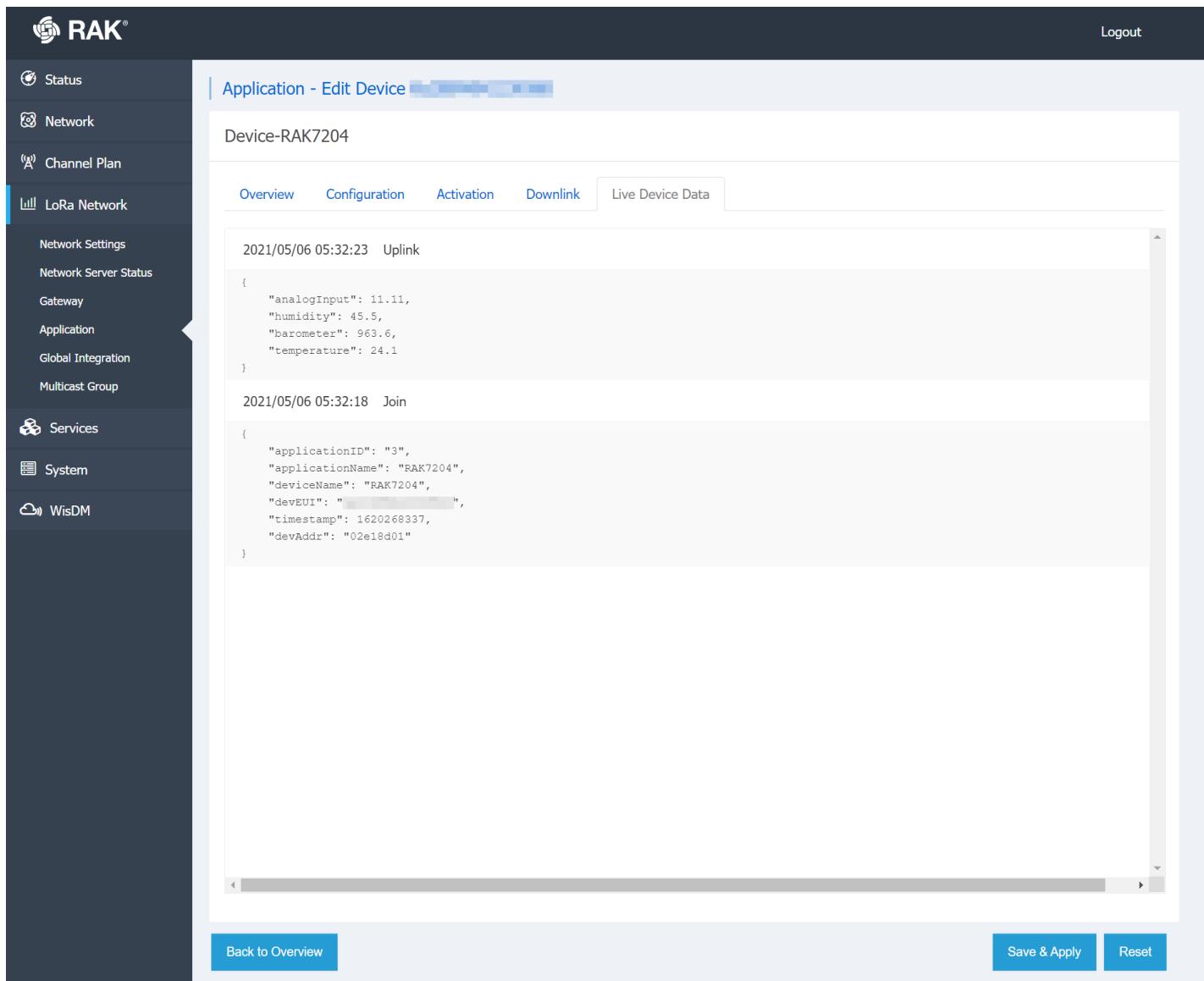
**Figure 25:** Successful Device Join

If you check the Devices Section and the Live Device Data in the Devices section you should see the device being online for some time and also some packets, in this case the Join request and an uplink packet where the data is visible in a human readable format (as we chose the Cayenne payload format)



Last seen	Device name	Device EUI	Class	Activation mode	Device Address	Link margin	Battery	Packet Loss	Description
1 37 seconds ago	RAK7204	[REDACTED]	A	otaa	02cf553e	-dB	-	0.00%	

Figure 26: Device up-time



```

2021/05/06 05:32:23 Uplink
{
  "analogInput": 11.11,
  "humidity": 45.5,
  "barometer": 963.6,
  "temperature": 24.1
}

2021/05/06 05:32:18 Join
{
  "applicationID": "3",
  "applicationName": "RAK7204",
  "deviceName": "RAK7204",
  "devEUI": "[REDACTED]",
  "timestamp": 1620268337,
  "devAddr": "02e18d01"
}

```

Figure 27: Device real time packets

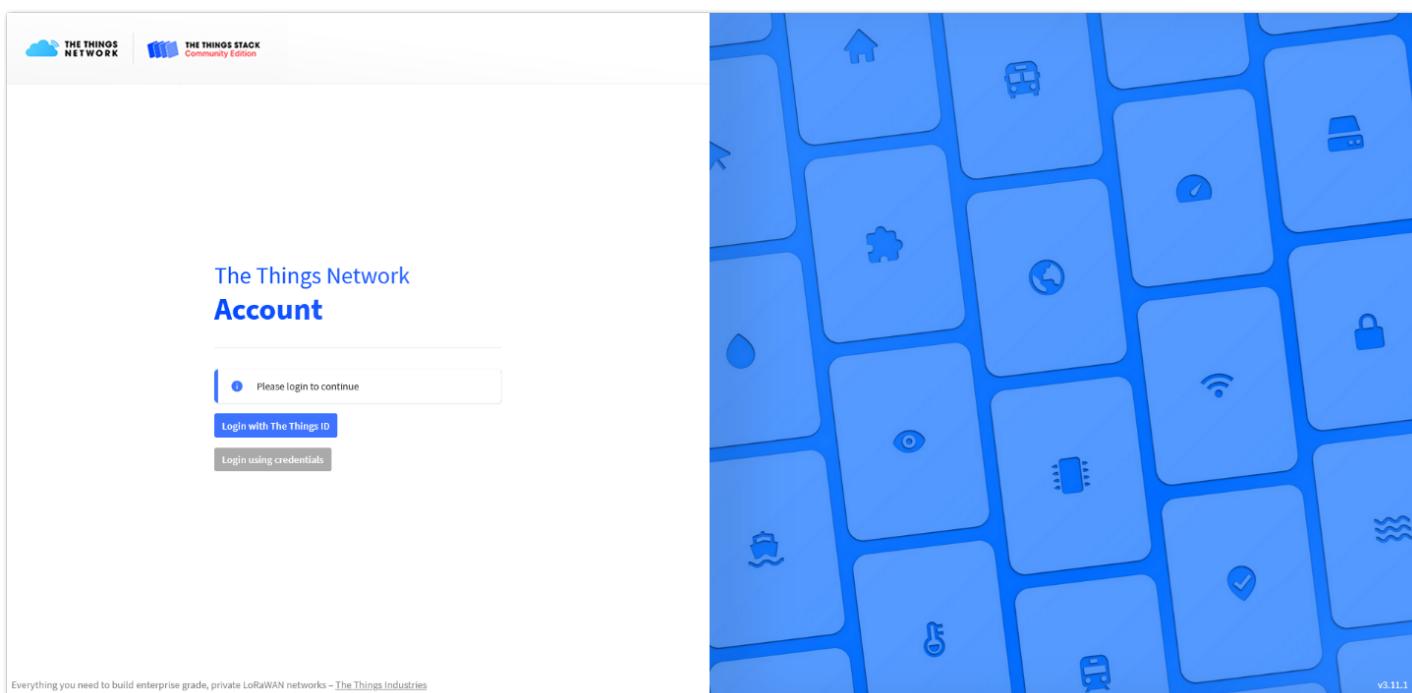
Your RAK7204 is now connected to the Built-in server and transmitting data over regular intervals.

## Connecting to The Things Network V3 (TTNv3)

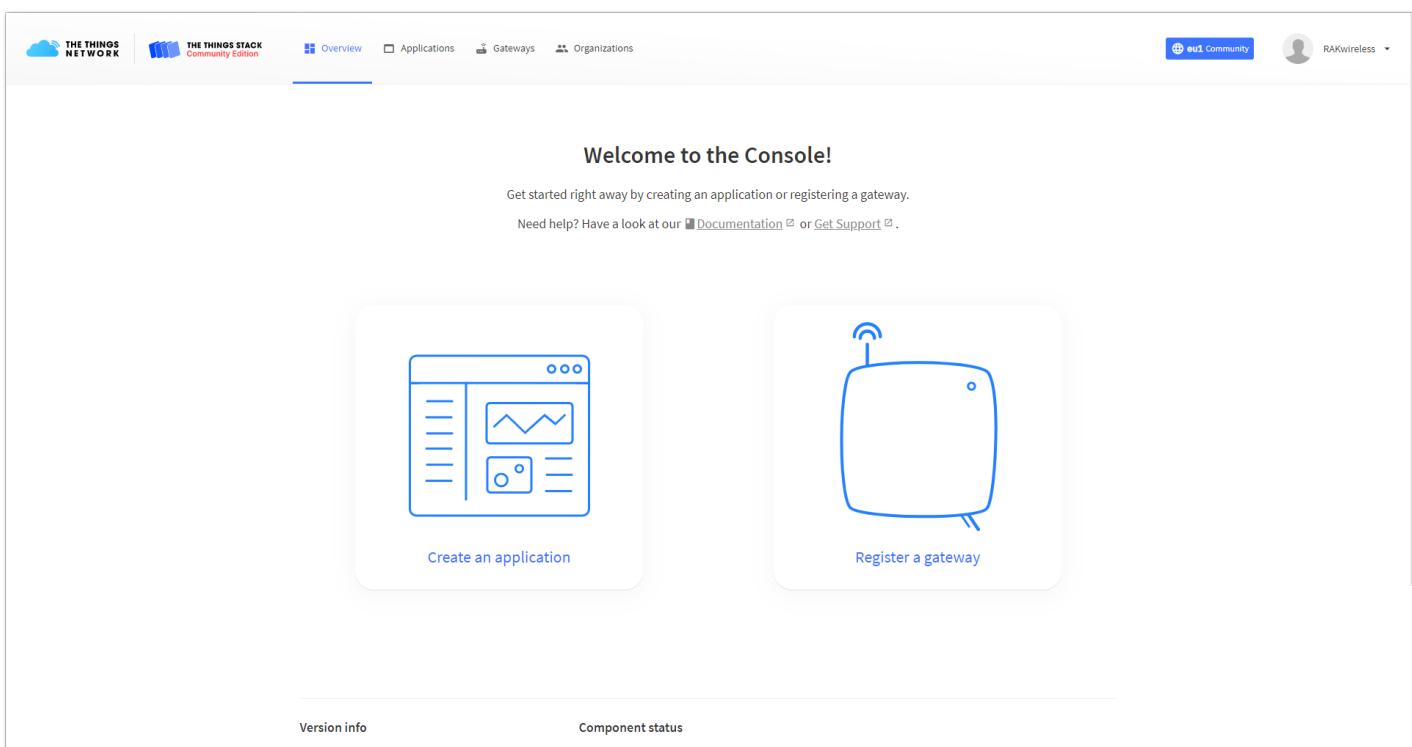
At The Things Conference 2021, it was announced that The Things Network is upgrading to The Things Stack v3.

In this section, it will be shown how to connect RAK7204 WisNode Sense Home to The Things Stack.

First, log in to the TTNv3. To do so, head to the TTNv3 site [site](#). If you already have a TTN account, you can use your The Things ID credentials to log in.



**Figure 28:** The Things Stack Home Page



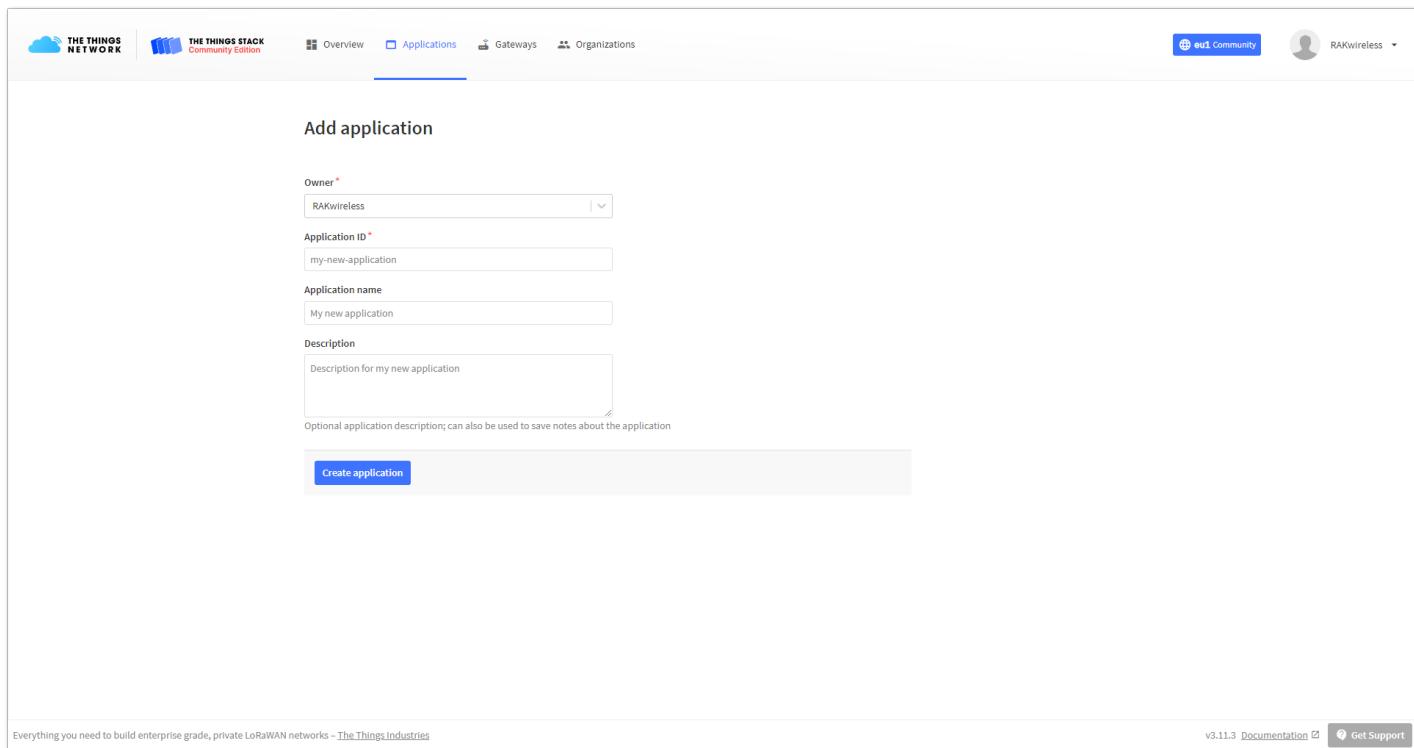
**Figure 29:** Console Page after successful login

 **NOTE:**

- To connect RAK7204 WisNode Sense Home to TTNv3, you should already have connected a gateway in range to TTNv2 or TTNv3, or you have to be sure that you are in the range of a public gateway.
- This tutorial is for EU868 Frequency band.

## Adding an Application

- If you do not have created applications yet, to create an application, choose **Create an application**. If you have created applications before, navigate through **Go to applications > + Add application**.

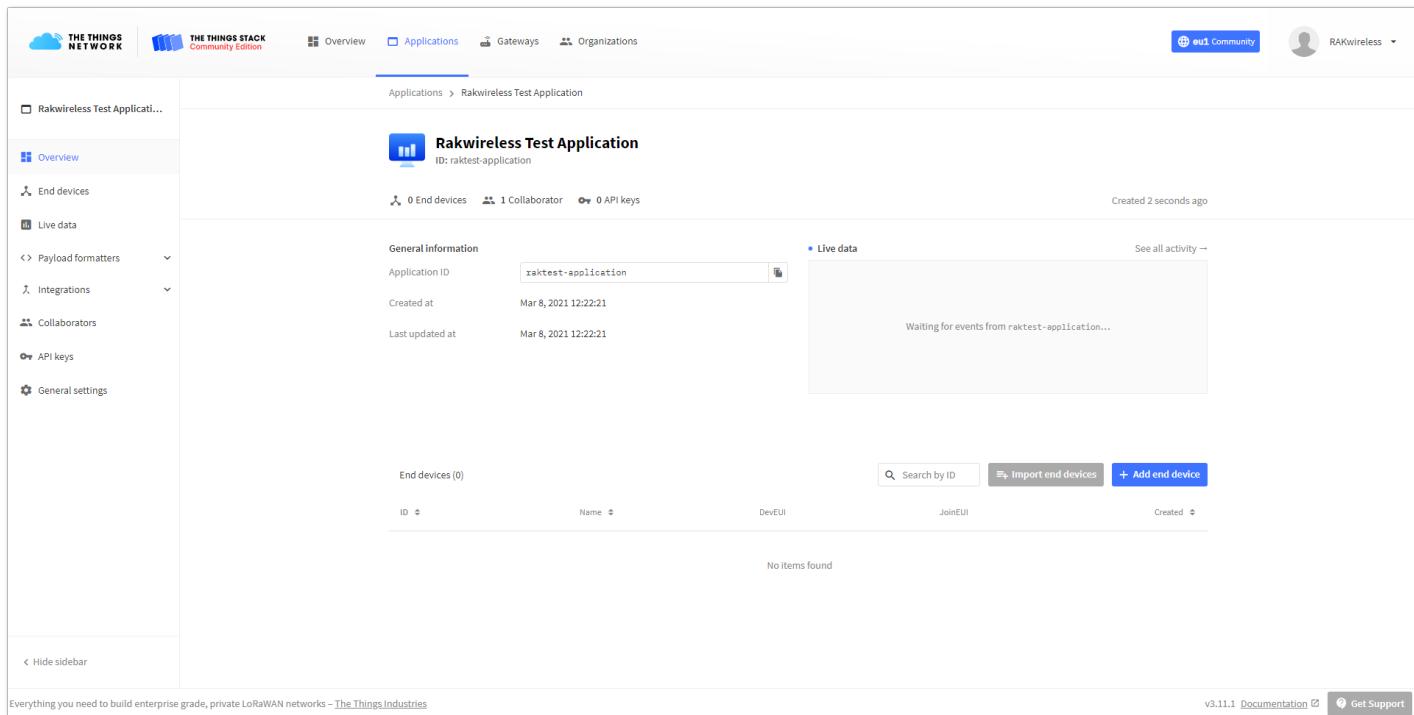


The screenshot shows the 'Add application' page in The Things Stack Community Edition. The top navigation bar includes links for Overview, Applications, Gateways, and Organizations. On the right, there are links for 'eu1 Community' and a user profile for 'RAKwireless'. The main form has fields for 'Owner' (RAKwireless), 'Application ID' (my-new-application), 'Application name' (My new application), and 'Description' (Description for my new application). A note below the description field says 'Optional application description; can also be used to save notes about the application'. A blue 'Create application' button is at the bottom.

**Figure 30:** Create an application page

- Fill in the needed information:

- Owner** - Automatically filled by The Things Stack, based on your account or created organization.
- Application ID** - This will be the unique ID of your application in the Network. ID must contain only lowercase letters, numbers, and dashes (-).
- Application name** (optional) - This is the name of your application.
- Description** (optional) – Description of your application. Optional application description; can also be used to save notes about the application.



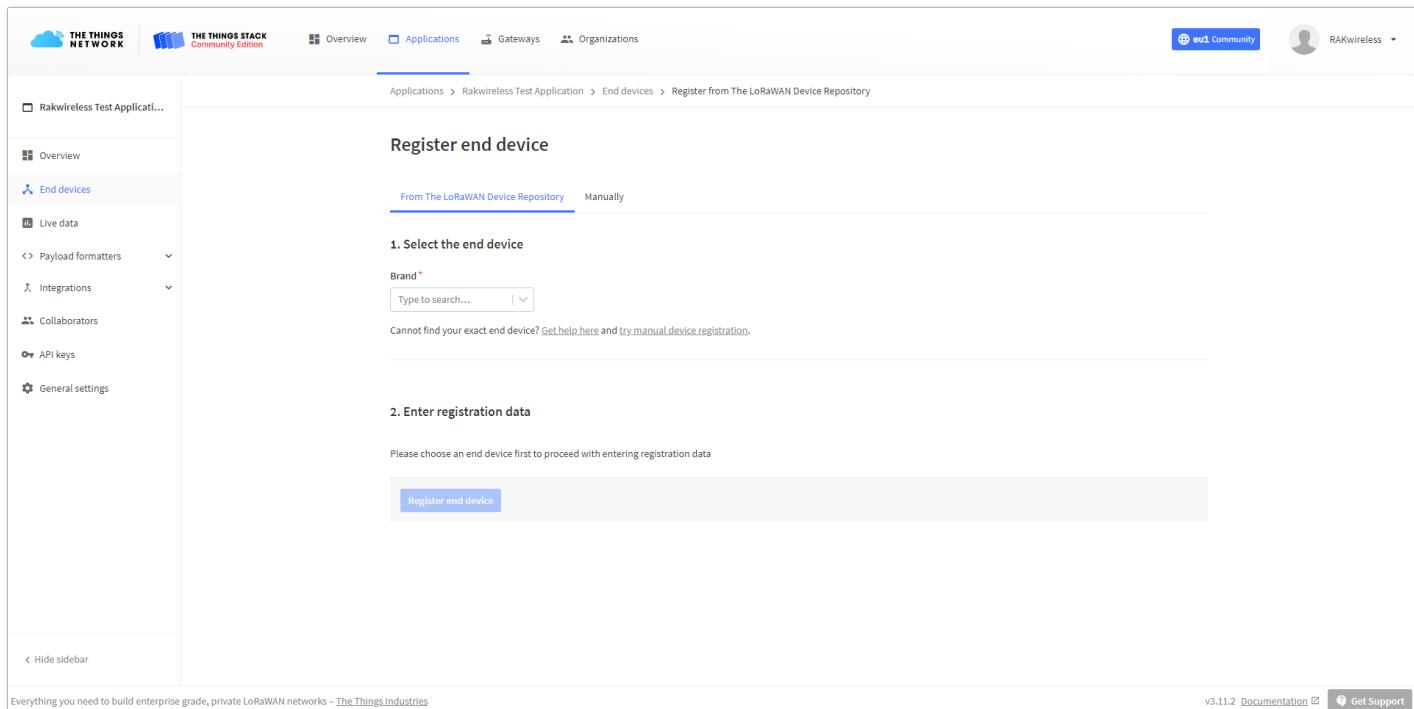
The screenshot shows the 'Rakwireless Test Application' overview page. The sidebar includes 'Overview', 'End devices', 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'. The main content area shows the application's ID (rakttest-application), general information (Application ID: rakttest-application, Created at: Mar 8, 2021 12:22:21, Last updated at: Mar 8, 2021 12:22:21), live data (Waiting for events from rakttest-application...), and end devices (0). There is a search bar and buttons for 'Import end devices' and '+ Add end device'.

**Figure 31:** Application Overview

## OTAA Mode

### Register the Device

1. From the Application Overview page, click on **+ Add end device**.

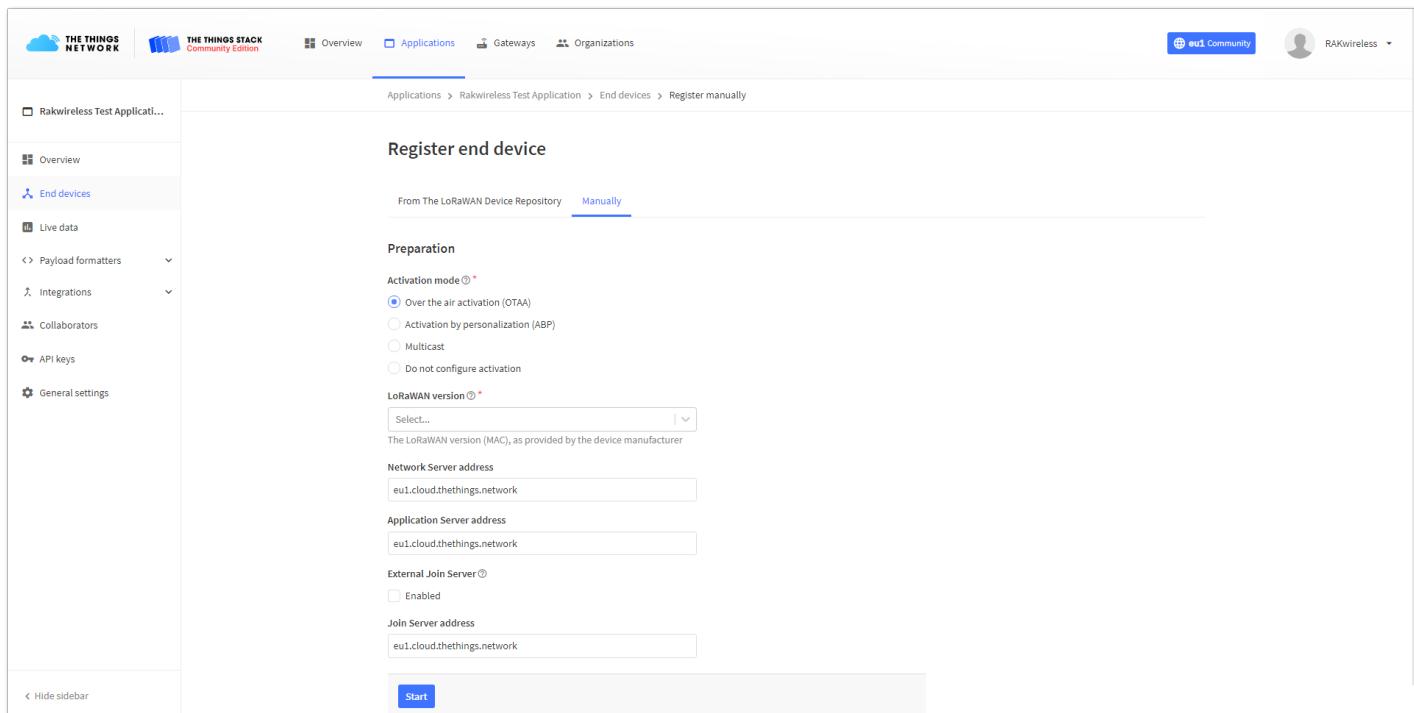


The screenshot shows the 'Register end device' page in The Things Stack Community Edition. The left sidebar is open, showing options like Overview, End devices (which is selected), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main content area has a heading 'Register end device' and two tabs: 'From The LoRaWAN Device Repository' (selected) and 'Manually'. Under the 'From The LoRaWAN Device Repository' tab, there's a section titled '1. Select the end device' with a 'Brand\*' dropdown menu containing 'Type to search...'. Below it is a note: 'Cannot find your exact end device? [Get help here](#) and try manual device registration.' Under the 'Manually' tab, there's a section titled '2. Enter registration data' with a note: 'Please choose an end device first to proceed with entering registration data'. At the bottom right of the main content area is a blue 'Register end device' button.

**Figure 32:** Adding a device in OTAA mode

2. Below the **Register end device** heading, you can find two options for registering a device. Choose **Manually**.

- For Activation mode, choose **Over the air activation (OTAA)**
- For the LoRaWAN version, choose **MAC V1.0.2** (RAK7204 is LoRaWAN 1.0.2 fully compliant).



The screenshot shows the 'Register manually' page in The Things Stack Community Edition. The left sidebar is open, showing the same navigation options as Figure 32. The main content area has a heading 'Register end device' and two tabs: 'From The LoRaWAN Device Repository' (disabled) and 'Manually' (selected). Under the 'Manually' tab, there's a section titled 'Preparation' with a 'Activation mode' dropdown menu where 'Over the air activation (OTAA)' is selected. There are also other options: 'Activation by personalization (ABP)', 'Multicast', and 'Do not configure activation'. Below this is a 'LoRaWAN version' dropdown menu with a note: 'The LoRaWAN version (MAC), as provided by the device manufacturer'. It lists several options like '868 MHz EU868', '915 MHz US915', etc. Further down are fields for 'Network Server address' (set to 'eu1.cloud.thethings.network'), 'Application Server address' (set to 'eu1.cloud.thethings.network'), 'External Join Server' (with an 'Enabled' checkbox), and 'Join Server address' (set to 'eu1.cloud.thethings.network'). At the bottom right of the main content area is a blue 'Start' button.

**Figure 33:** Registering the device in OTAA mode

3. To get to the next step of the registration, click **Start**.

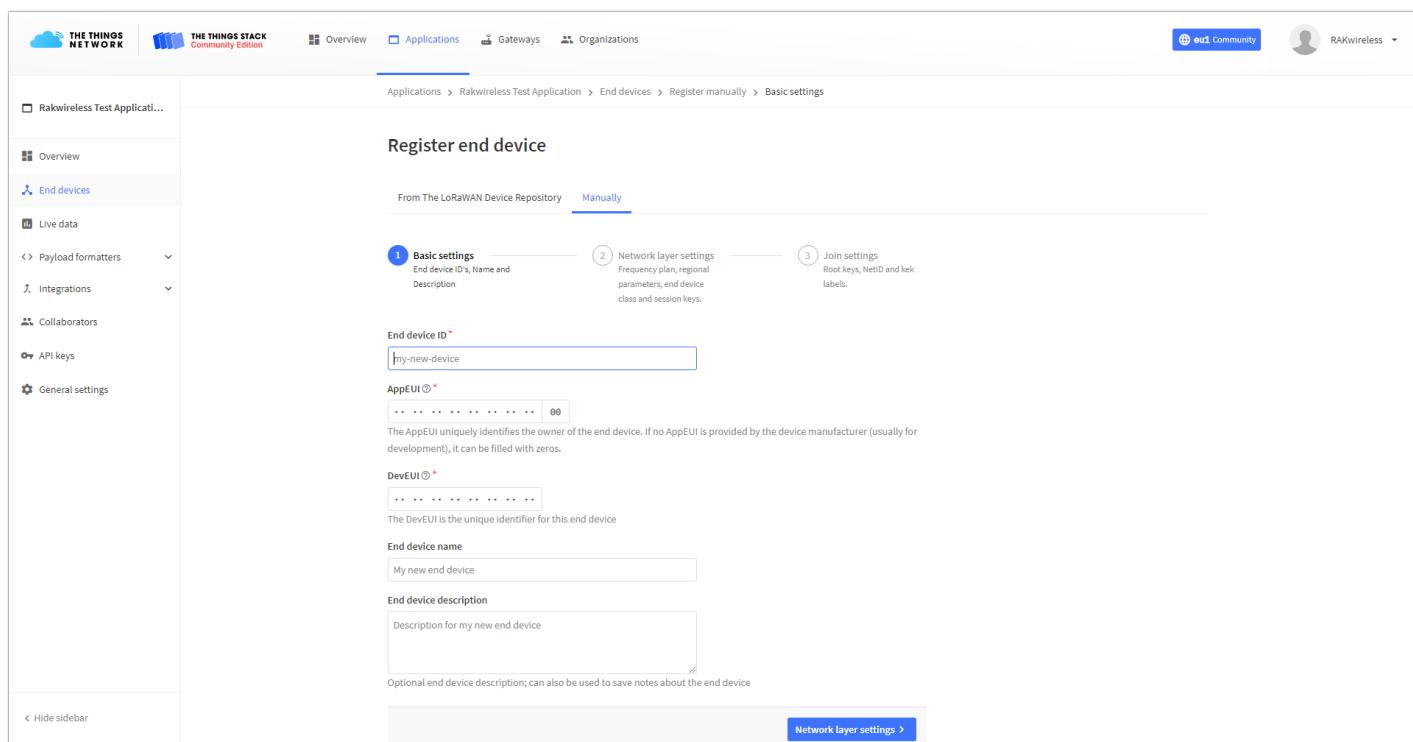


Figure 34: Basic settings for OTAA mode

#### 4. Fill in the Basic settings for the device:

- **End device ID** - This is the unique identifier for your RAK7204 WisNode Sense Home in your application. You need to enter this manually. The End device ID must contain only lowercase letters, numbers, and dashes (-).
- **AppEUI** - The AppEUI uniquely identifies the owner of the end device. It is provided by the device manufacturer. To get the AppEUI, connect your device via USB cable to your computer. Open RAK Serial Port Tool, choose the correct COM port and BaudRate, and run the following command:

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

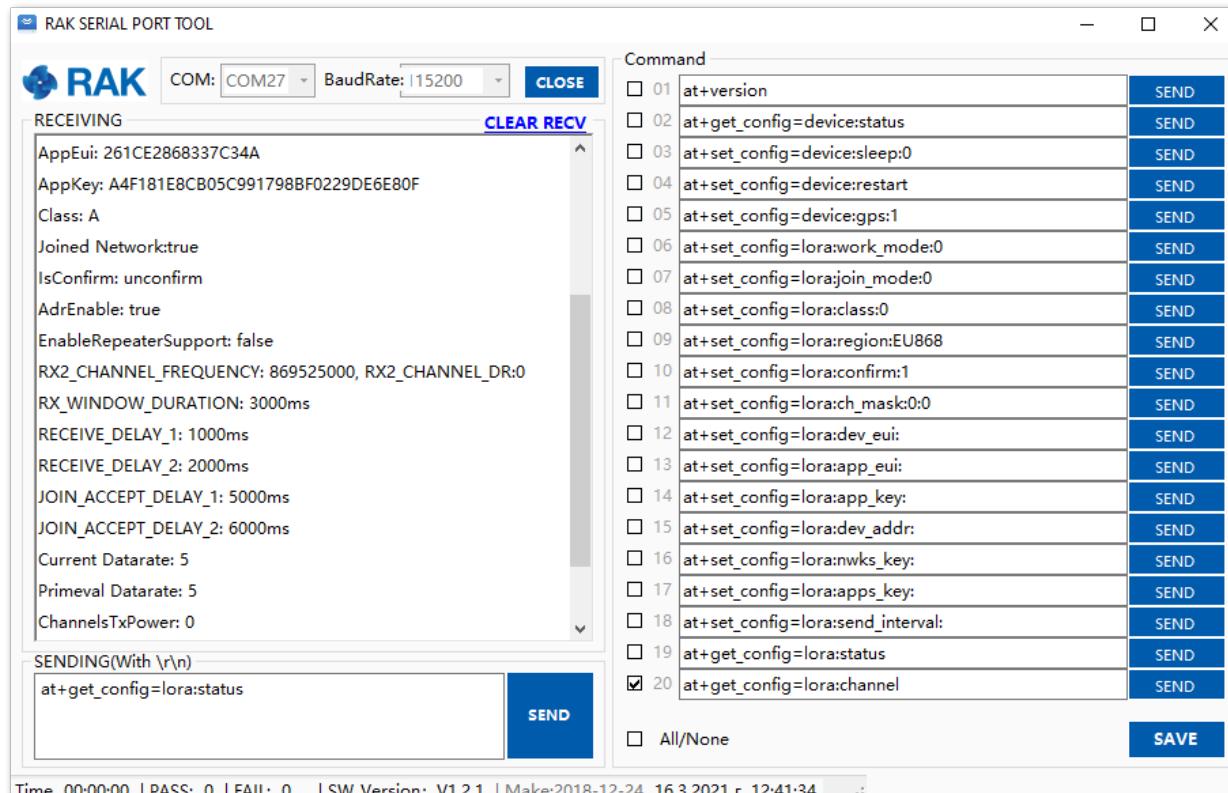


Figure 35: AppEUI of the device

- **DevEUI** - The unique identifier for this end device. It can be found on a sticker on the back of the device.

- **End device name** (optional) - A unique, human-readable identifier for your device. You make it up, so be creative. Device IDs cannot be used by multiple devices within the same application.
- **End device description** (optional) - Optional end device description; can also be used to save notes about the end device.

## 5. Click Network layer setting.

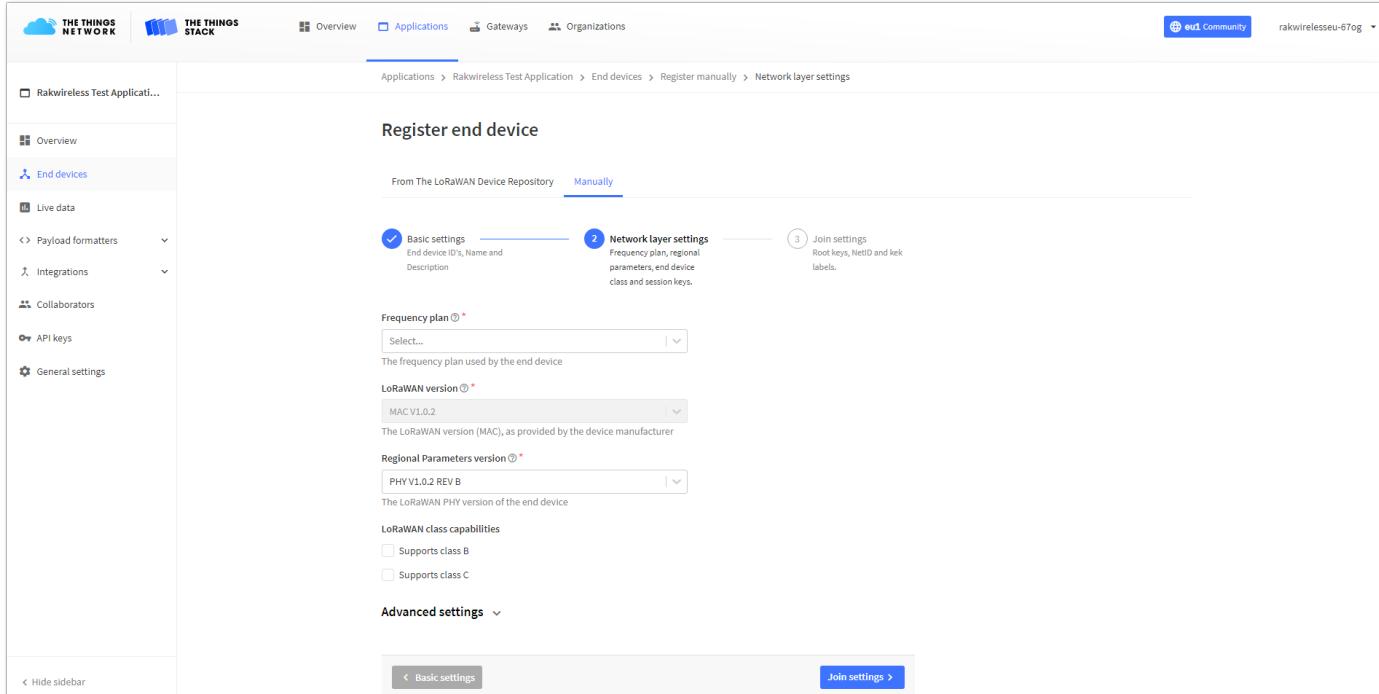


Figure 36: Network layer setting for OTAA mode

## 6. Here you must configure the Network layer settings for the device:

- **Frequency plan** - The frequency plan used by the end device. Note that, for this tutorial, the frequency plan used is Europe 863-870 MHz (SF9 for RX2 – recommended).
- **Regional Parameters version** - The Regional Parameters specify frequency, dwell time, and other communication settings for different geographical areas. The Regional Parameters version is the version of the LoRa Alliance specification which your device supports. This should be provided by the device manufacturer in a datasheet. For this example, **PHY V1.0.2 REV B** is chosen.
- **LoRaWAN class capabilities** – Here you can select if your device supports Class B, Class C, or both.

## 7. In Advanced settings, you can configure additional settings for your device.

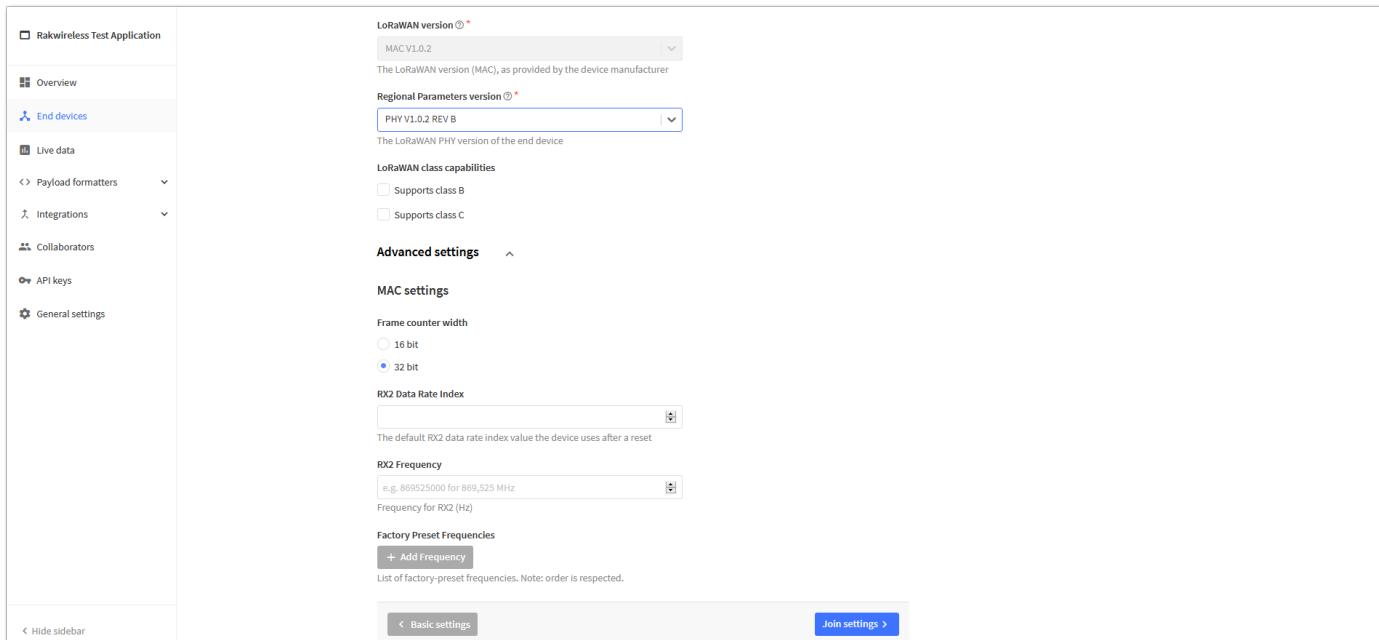
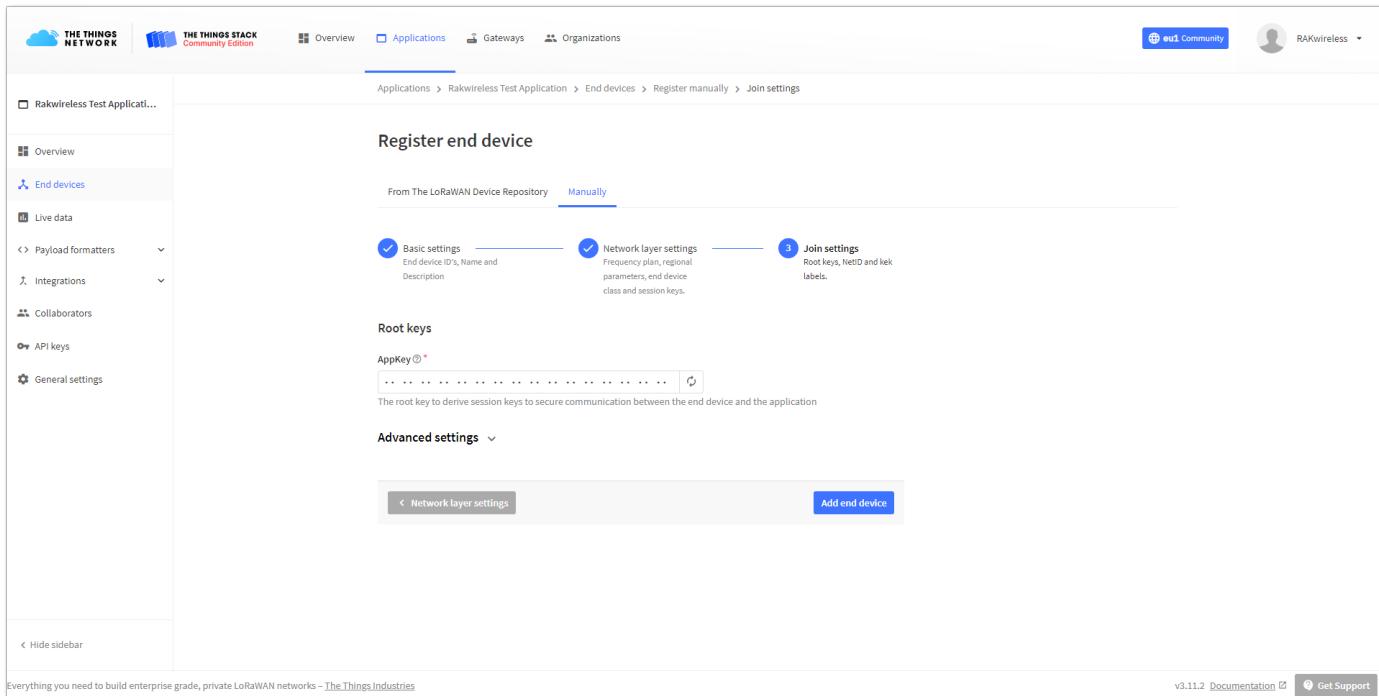


Figure 37: Advanced network layer settings of the device

 **NOTE:**

For this example, these settings will be left as default.

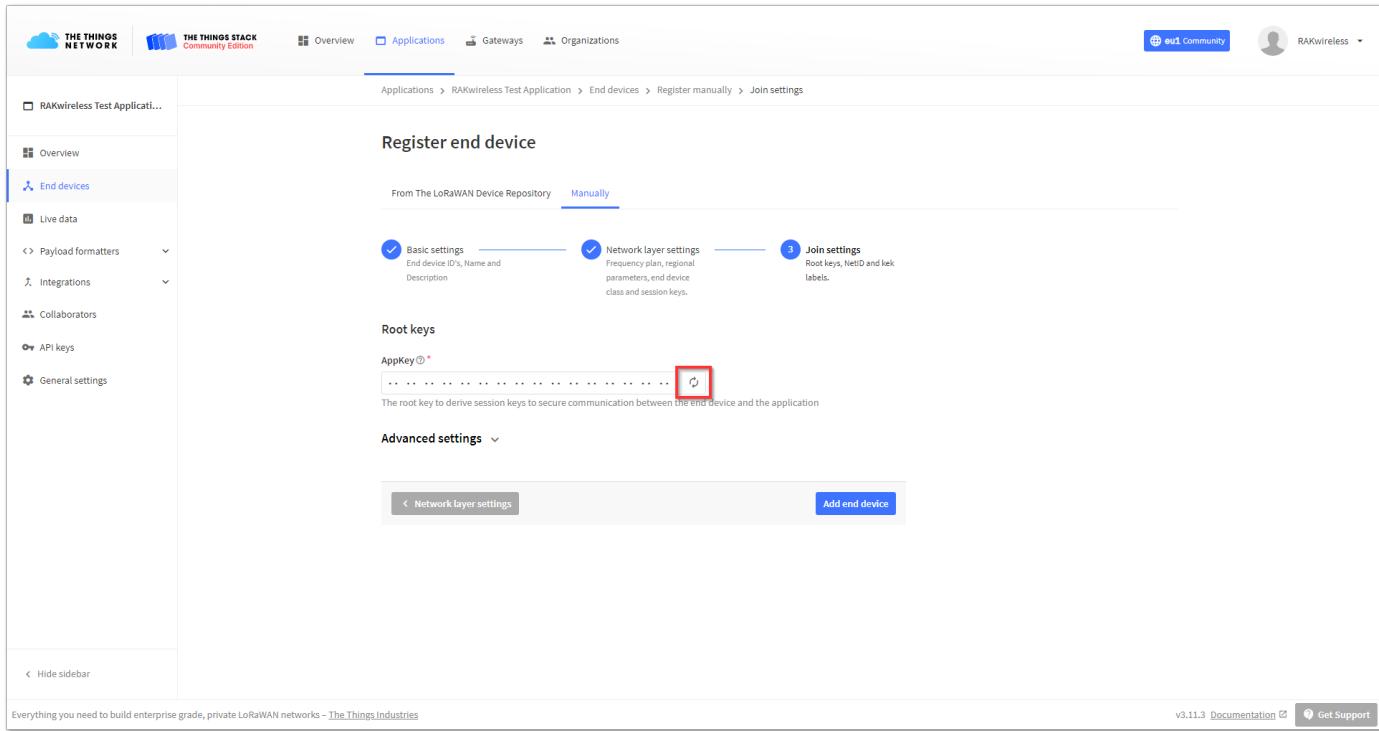
## 8. Click **Join settings**.



The screenshot shows the 'Register end device' page in The Things Stack Community Edition. The sidebar on the left is collapsed. The main navigation path is Applications > Rakwireless Test Application > End devices > Register manually > Join settings. The 'Manually' tab is selected. On the right, there are three tabs: 'Basic settings' (selected), 'Network layer settings', and 'Join settings'. Under 'Basic settings', it says 'From The LoRaWAN Device Repository' and 'End device ID's, Name and Description'. Under 'Join settings', it says 'Root keys, NetID and kek labels.' Below these tabs is a section for 'Root keys' with a field for 'AppKey'. A 'Generate' button is located to the right of the AppKey field. At the bottom of the page are 'Network layer settings' and 'Add end device' buttons.

**Figure 38:** Join settings for OTAA mode

## 9. Fill in the **Application key (AppKey)** to secure communication between the end device and the application. The AppKey can be generated automatically by clicking the **Generate** button next to the **AppKey** field.



This screenshot is identical to Figure 38, but the 'Generate' button next to the 'AppKey' field is highlighted with a red box. All other elements and text are the same as in Figure 38.

**Figure 39:** Generate the AppKey

## 10. In the **Advanced settings**, you can configure more options about your device.

**Advanced settings ^**

**Home NetID**  
  
 ID to identify the LoRaWAN network

**Application Server ID**  
  
 The AS-ID of the Application Server to use

**Application Server KEK label**  
  
 The KEK label of the Application Server to use for wrapping the application session key

**Network Server KEK label**  
  
 The KEK label of the Network Server to use for wrapping the network session key

[◀ Network layer settings](#) [Add end device](#)

**Figure 40:** Advanced join settings for OTAA mode

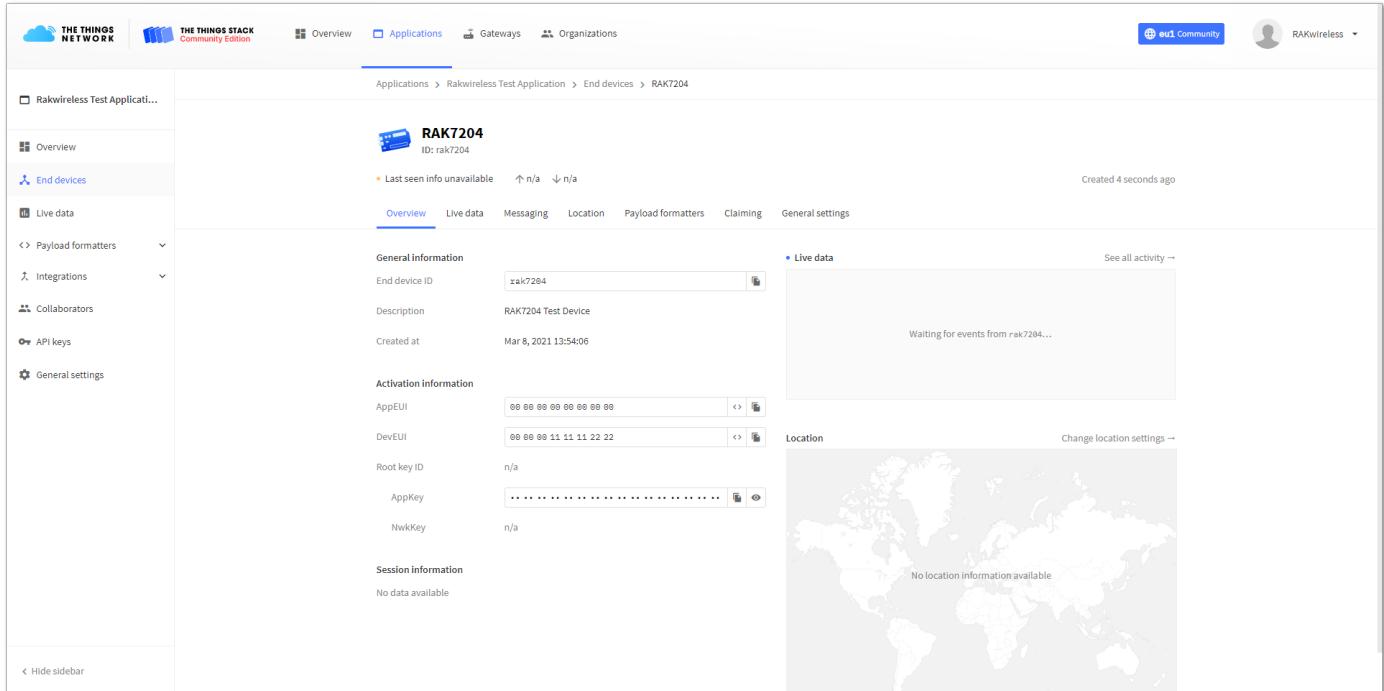
 **NOTE:**

For this example, these settings will be left as default.

11. Finally, to finish registering your device, click **Add end device**.

## Configuring the Device in OTAA Mode

1. For configuring the node, you will need the following three parameters: **Device EUI**, **Application EUI**, and **Application Key**. You can see them all on the **Device Overview** page. But since the two EUI's come with the device, you only need the Application Key from there.



**Figure 41:** OTAA device parameters

2. Using the RAK Serial Port Tool, set the join mode, device class, and your LoRaWAN region to your correct frequency band, with the following set of AT commands:
  - For the join mode (OTAA):

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

- For the class (Class A):

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

- For the region:

```
at+set_config=lora:region:EU868
```

 **NOTE:**

Remember to replace the **frequency band** with the one for your LoRaWAN region. Check first your [frequency plan](#).

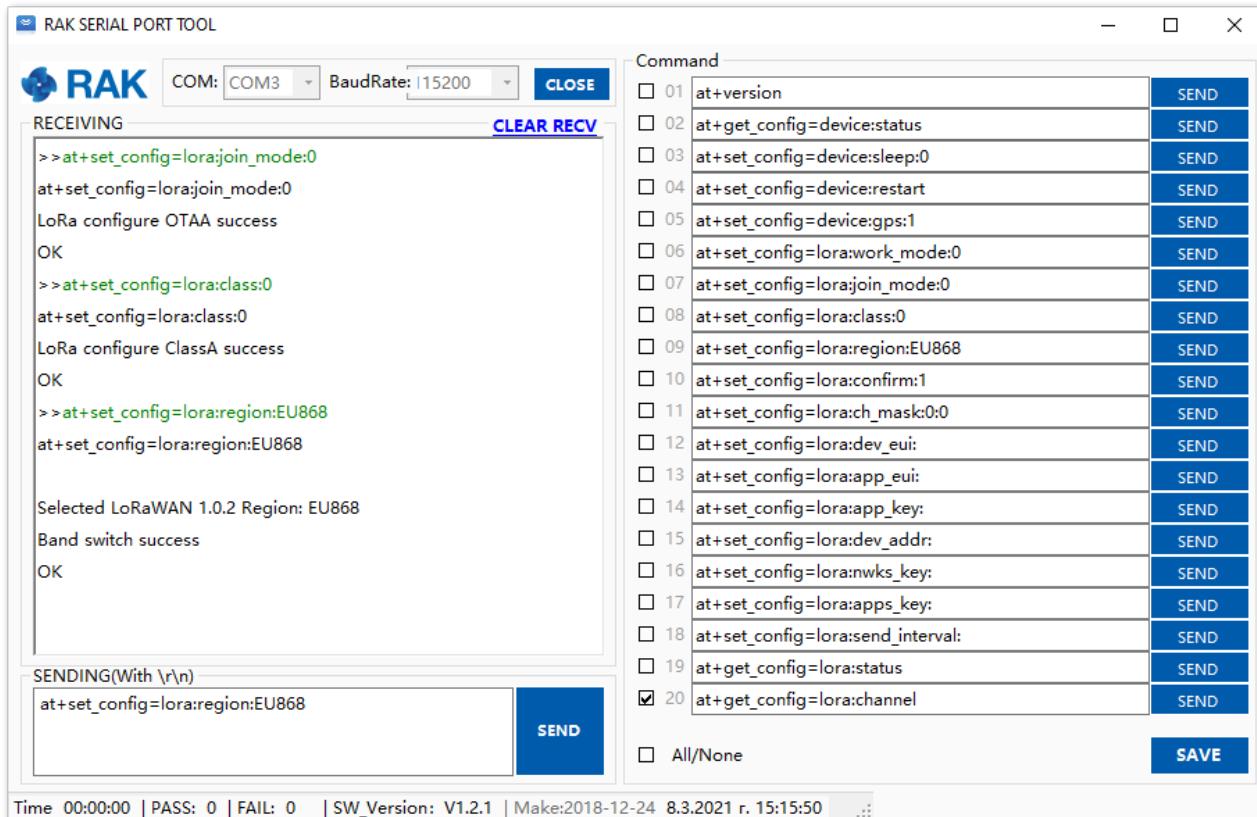


Figure 42: Setting up the RAK7204 WisNode Sense Home operation mode

 **NOTE:**

The following tutorial is based on using the EU868 frequency band.

- Now that those parameters are set, enter the **App Key**, using the command below. Remember to replace the "XXXX" with the corresponding parameter value for your particular case.

```
at+set_config=lora:app_key:XXXX
```

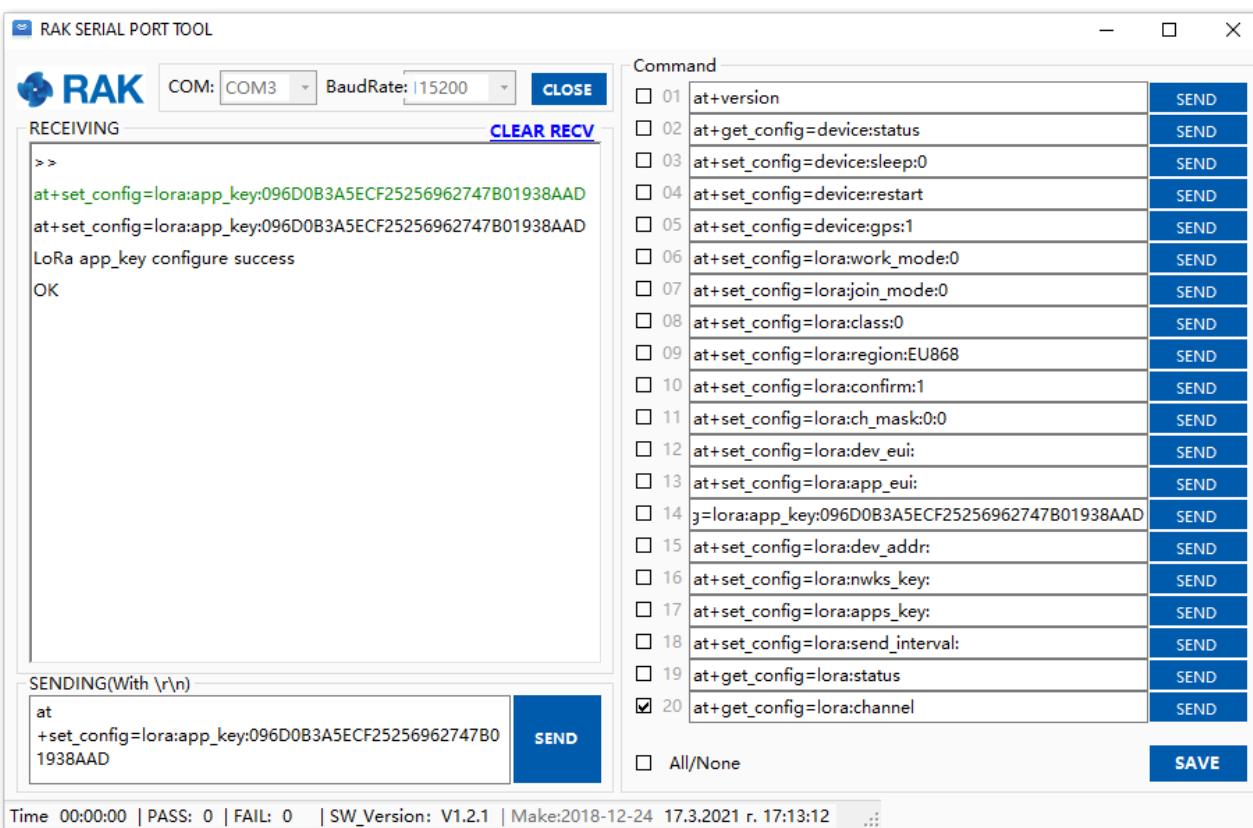


Figure 43: Setting up the RAK7204 WisNode Sense Home OTAA parameters

4. Finally, execute the join command:

```
at+join
```

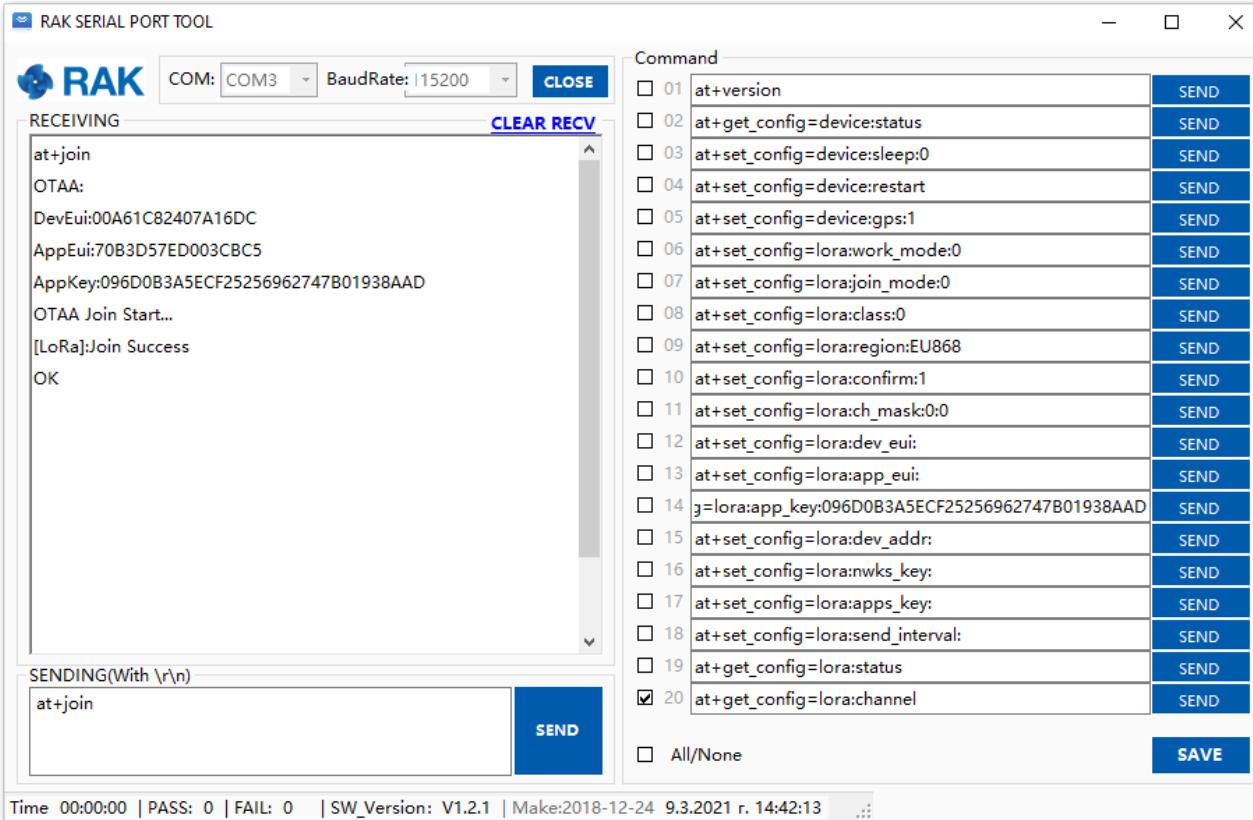
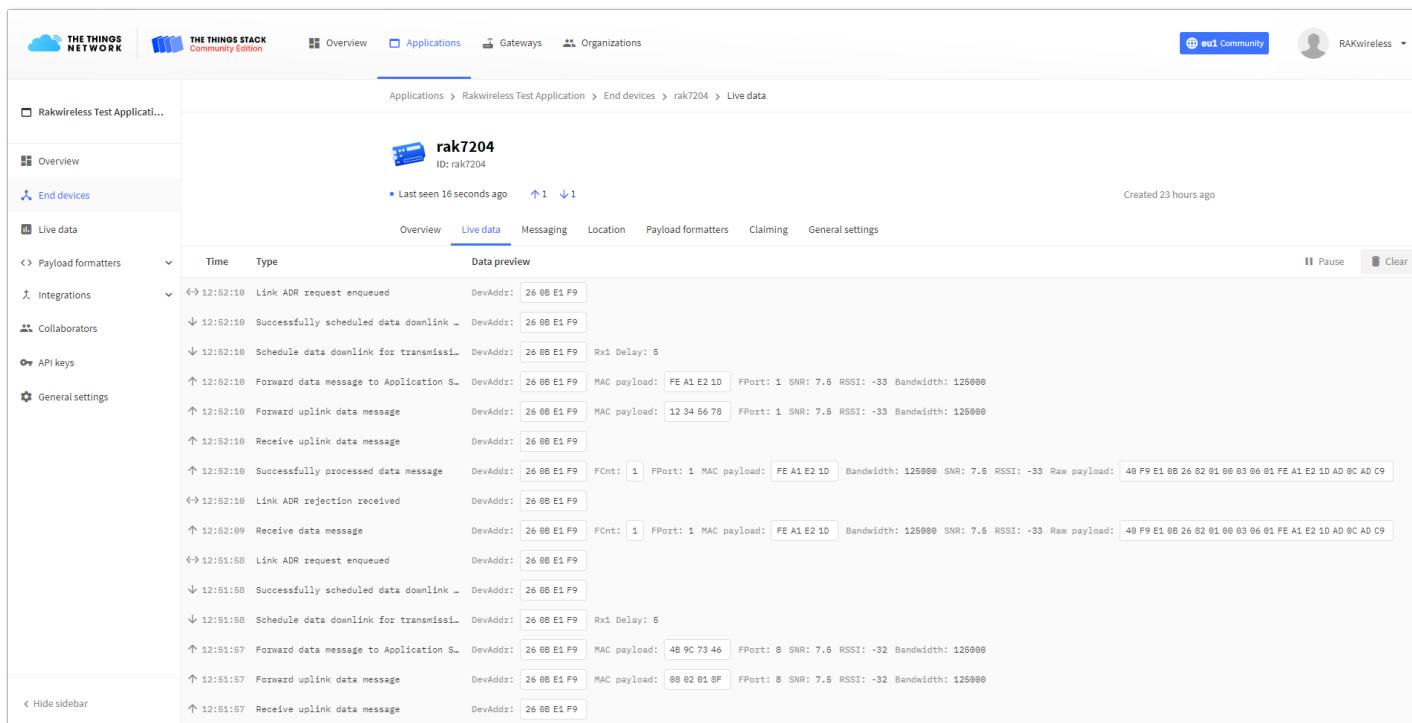


Figure 44: Join command

If you get a response in the **Live data** feed in The Things Stack, it means your RAK7204 is successfully connected!



The screenshot shows the 'Live data' tab for a RAK7204 device. The interface includes a sidebar with 'Overview', 'End devices', and 'Live data'. The main area displays a timeline of events with columns for 'Time', 'Type', and 'Data preview'. Key entries include:

- 12:52:10 Link ADR request enqueued (DevAddr: 26 0B E1 F9)
- 12:52:10 Successfully scheduled data downlink (DevAddr: 26 0B E1 F9)
- 12:52:10 Schedule data downlink for transmission (DevAddr: 26 0B E1 F9, Rx1 Delay: 6)
- 12:52:10 Forward data message to Application S. (DevAddr: 26 0B E1 F9, MAC payload: FE A1 E2 10, FPort: 1, SNR: 7.5, RSSI: -33, Bandwidth: 128000)
- 12:52:10 Forward uplink data message (DevAddr: 26 0B E1 F9, MAC payload: 12 34 66 78, FPort: 1, SNR: 7.5, RSSI: -33, Bandwidth: 128000)
- 12:52:10 Receive uplink data message (DevAddr: 26 0B E1 F9)
- 12:52:10 Successfully processed data message (DevAddr: 26 0B E1 F9, FCnt: 1, FPort: 1, MAC payload: FE A1 E2 10, Bandwidth: 128000, SNR: 7.5, RSSI: -33, Raw payload: 48 F9 E1 0B 26 82 81 00 83 86 01 FE A1 E2 10 AD 8C AD C9)
- 12:52:10 Link ADR rejection received (DevAddr: 26 0B E1 F9)
- 12:52:09 Receive data message (DevAddr: 26 0B E1 F9, FCnt: 1, FPort: 1, MAC payload: FE A1 E2 10, Bandwidth: 128000, SNR: 7.5, RSSI: -33, Raw payload: 48 F9 E1 0B 26 82 01 00 03 06 01 FE A1 E2 10 AD 8C AD C9)
- 12:51:58 Link ADR request enqueued (DevAddr: 26 0B E1 F9)
- 12:51:58 Successfully scheduled data downlink (DevAddr: 26 0B E1 F9)
- 12:51:58 Schedule data downlink for transmission (DevAddr: 26 0B E1 F9, Rx1 Delay: 6)
- 12:51:57 Forward data message to Application S. (DevAddr: 26 0B E1 F9, MAC payload: 4B 9C 73 46, FPort: 8, SNR: 7.5, RSSI: -32, Bandwidth: 128000)
- 12:51:57 Forward uplink data message (DevAddr: 26 0B E1 F9, MAC payload: 68 62 01 8F, FPort: 8, SNR: 7.5, RSSI: -32, Bandwidth: 128000)
- 12:51:57 Receive uplink data message (DevAddr: 26 0B E1 F9)

**Figure 45:** Sending data to The Things Stack from RAK7204 WisNode Sense Home

## Connecting to ChirpStack

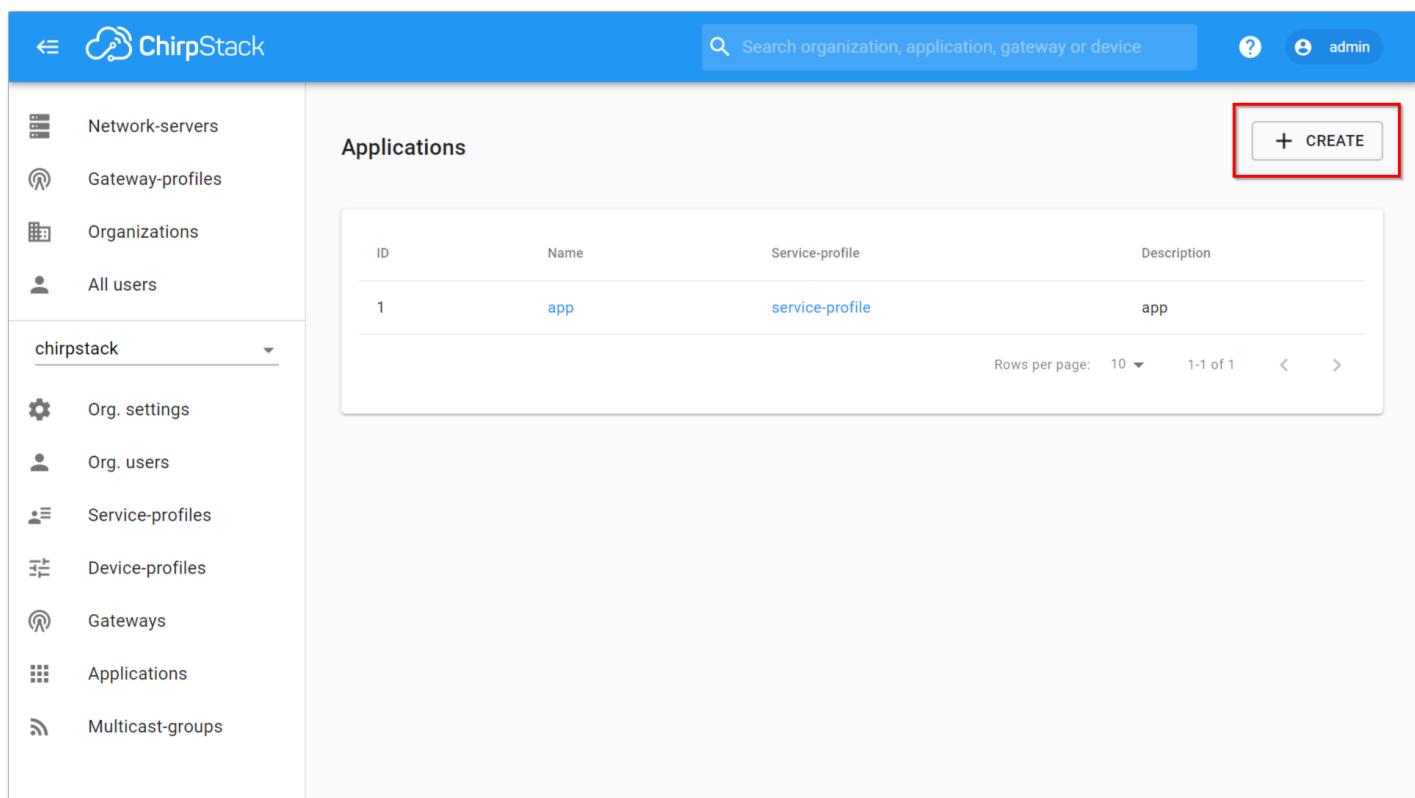
The ChirpStack or previously known as LoRaServer project provides open-source components for building LoRaWAN networks. You can learn more about ChirpStack [here](#).

You can use RAK7204 WisNode Sense Home to connect with ChirpStack according to the following steps:

### NOTE:

In this section, it is assumed that you are using RAK Gateway and its built-in ChirpStack or RAK cloud testing ChirpStack. It is also assumed that a Gateway with the ChirpStack has been configured successfully. If not, please have a look at RAK's documents for more details about RAK LPWAN Gateway and [RAK cloud testing](#).

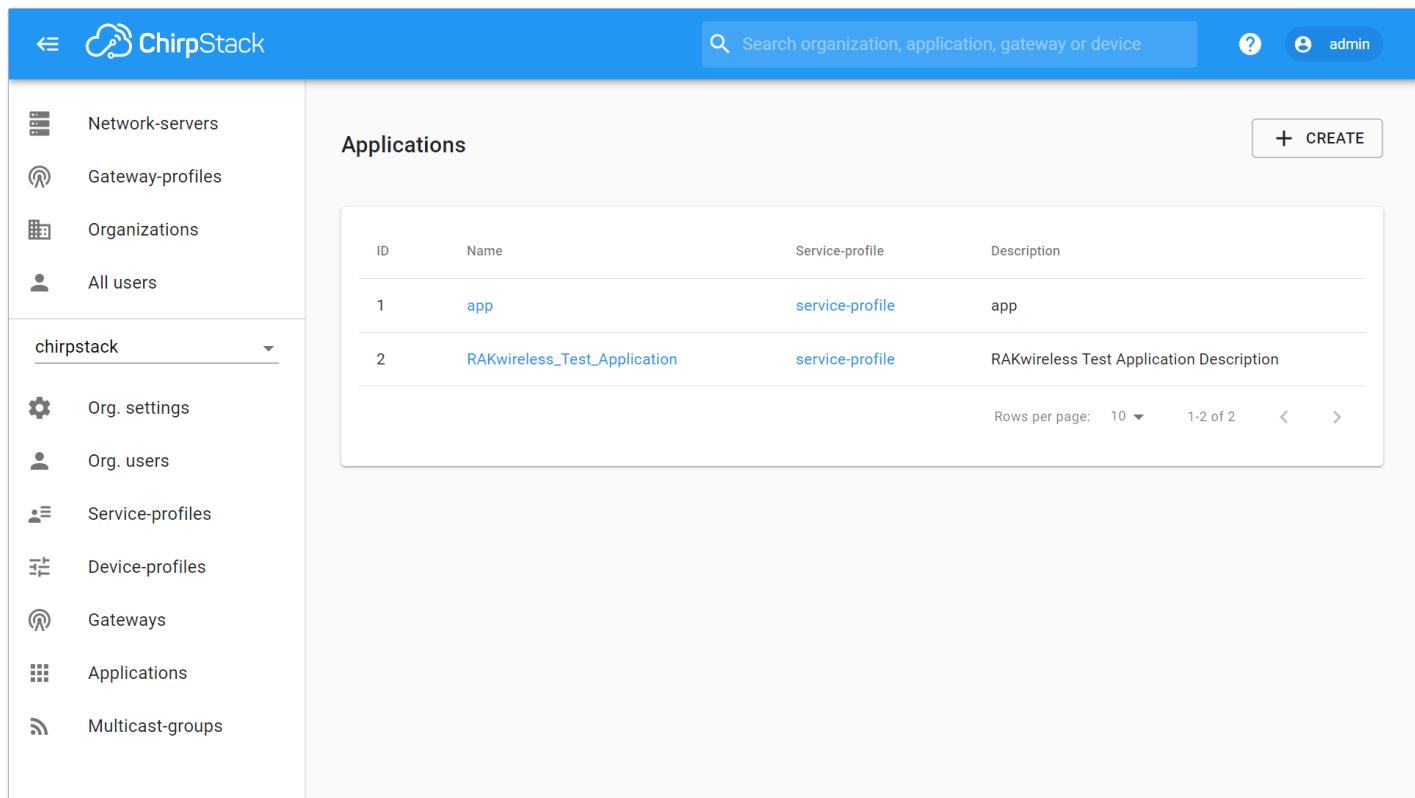
1. Open the web page of the ChirpStack which you want to connect with and login.
2. By default, there is already one or more items in this page, you can use it or create a new item. Now, let's create a new item by clicking the “CREATE” button, then filling them in.



ID	Name	Service-profile	Description
1	app	service-profile	app

**Figure 46:** ChirpStack Applications

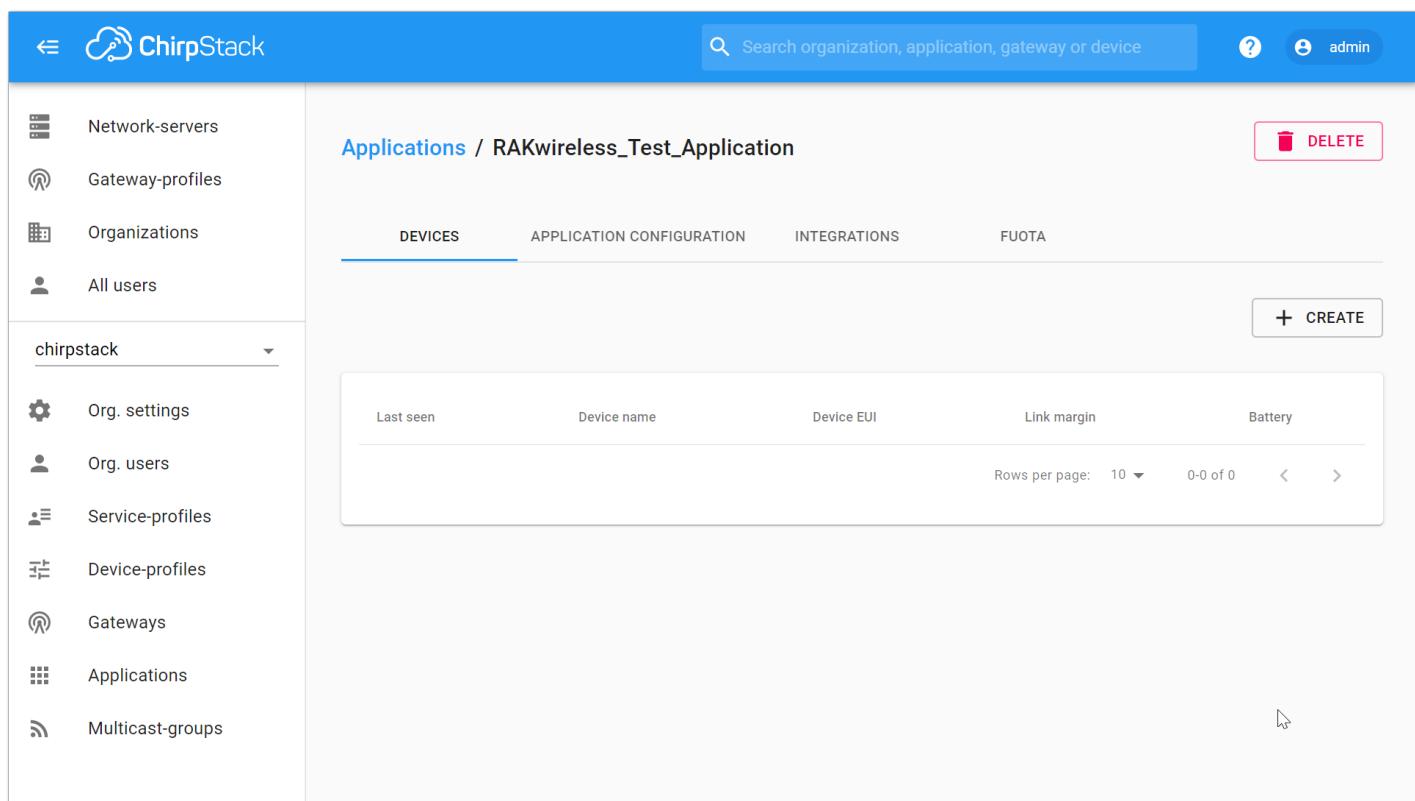
3. Fill up the necessary information then Click "**CREATE APPLICATION**".



ID	Name	Service-profile	Description
1	app	service-profile	app
2	RAKwireless_Test_Application	service-profile	RAKwireless Test Application Description

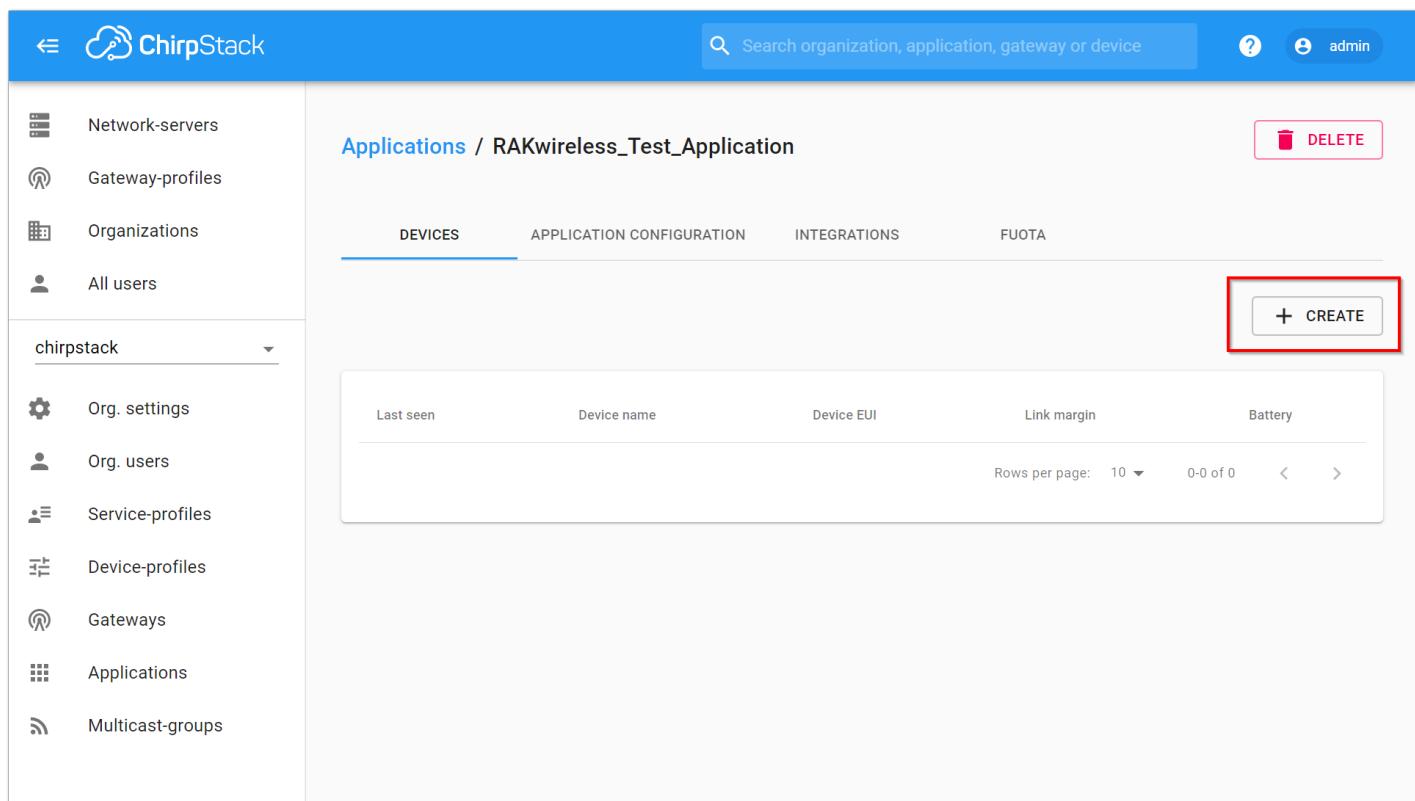
**Figure 47:** Creating the Application

4. Click the new item name "**RAKwireless\_Test\_Application**":



The screenshot shows the ChirpStack interface for managing applications. On the left, a sidebar lists various management options like Network-servers, Gateway-profiles, Organizations, and users. Below this is a dropdown for 'chirpstack'. The main area is titled 'Applications / RAKwireless\_Test\_Application'. It features tabs for 'DEVICES', 'APPLICATION CONFIGURATION', 'INTEGRATIONS', and 'FUOTA'. The 'DEVICES' tab is active. A table below shows columns for Last seen, Device name, Device EUI, Link margin, and Battery. At the bottom of the table, it says 'Rows per page: 10 ▾ 0-0 of 0 < >'. In the top right corner, there is a 'DELETE' button and a 'CREATE' button, which is highlighted with a red box.

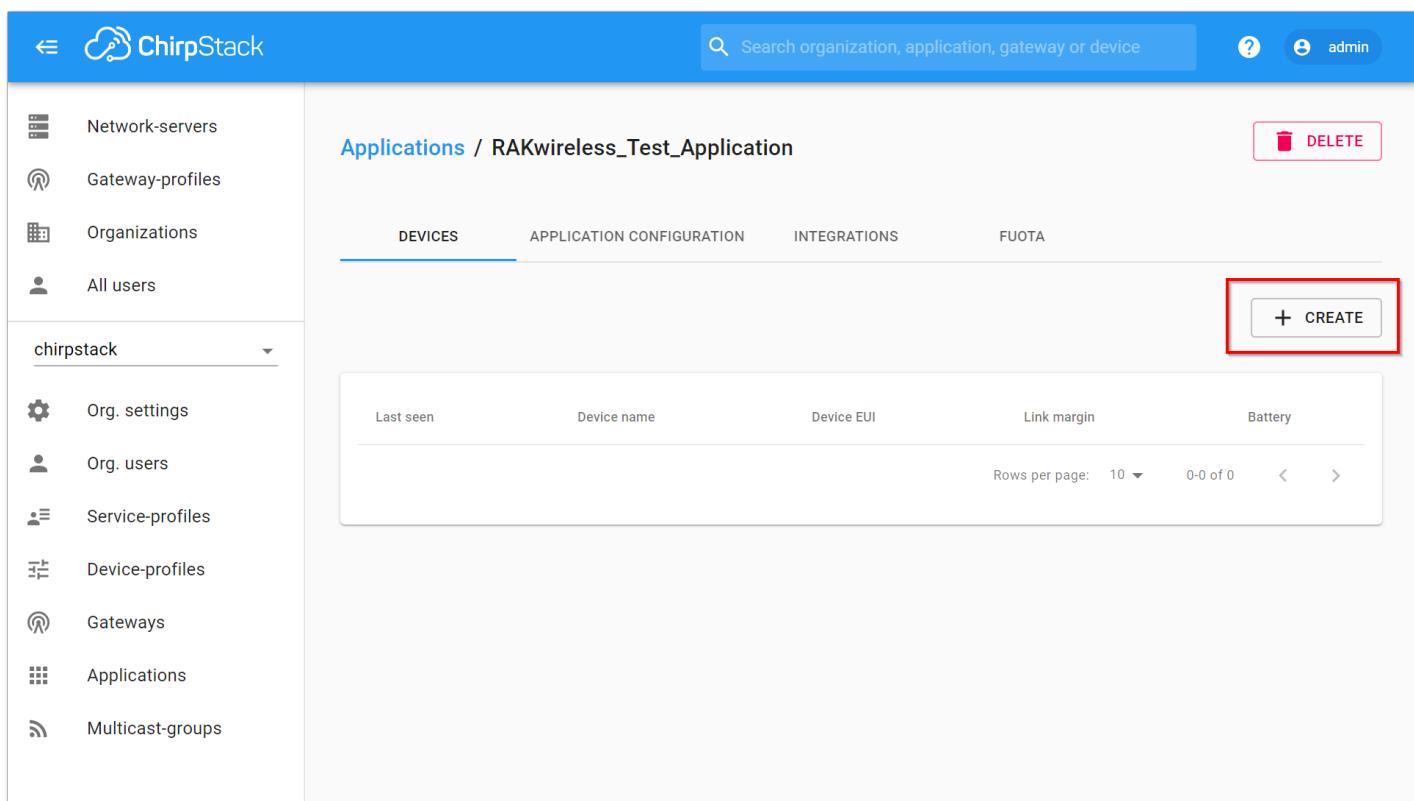
Figure 48: Applications page in ChirpStack



This screenshot is identical to Figure 48, showing the ChirpStack Applications page for the RAKwireless\_Test\_Application. The 'DEVICES' tab is selected, and the 'CREATE' button in the top right is highlighted with a red box. The rest of the interface, including the sidebar and table, is the same as in Figure 48.

Figure 49: RAKwireless Test Application

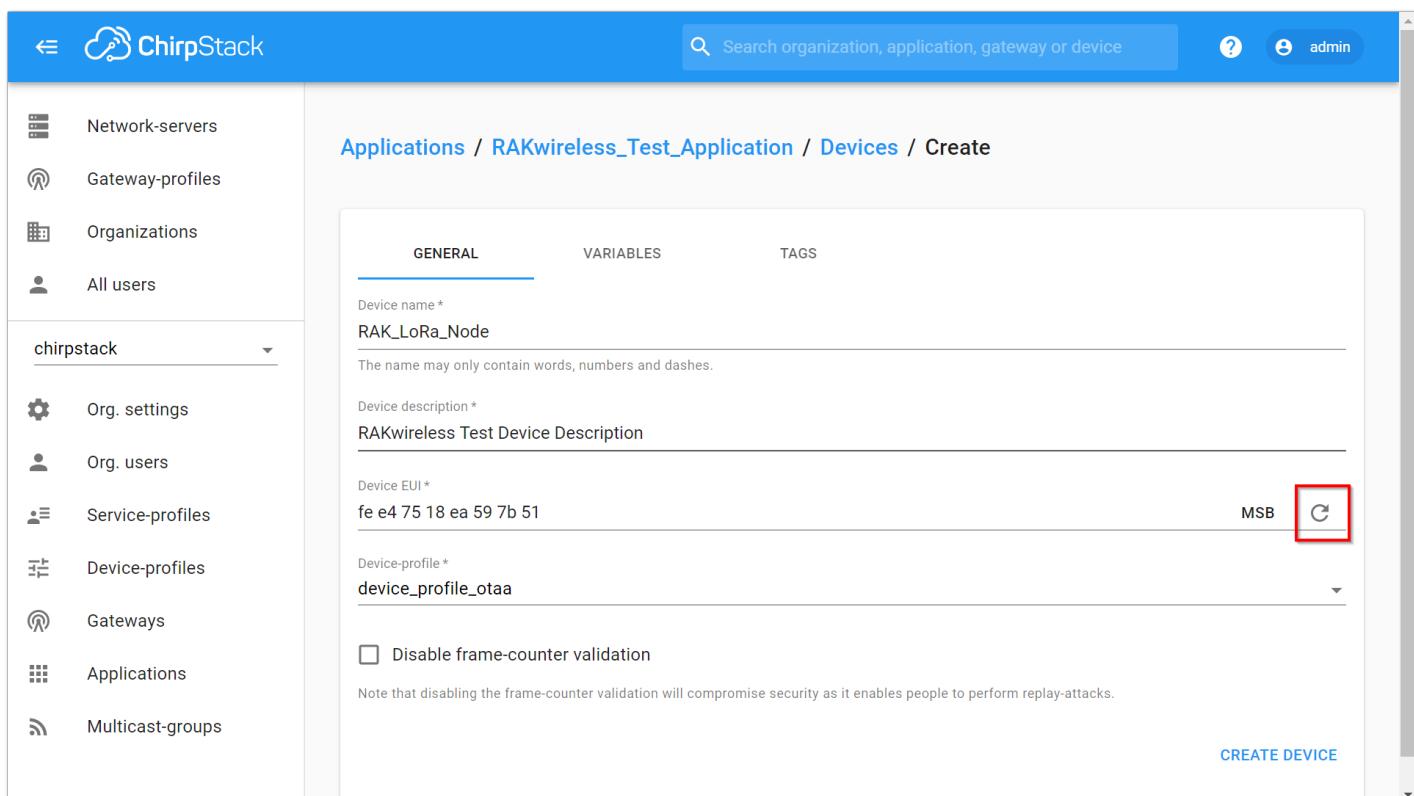
5. Add a Node device into ChirpStack by clicking the “CREATE” button:



The screenshot shows the ChirpStack Documentation Center interface. On the left, there's a sidebar with various navigation options like Network-servers, Gateway-profiles, Organizations, All users, and chirpstack (which is expanded to show Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups). The main content area is titled 'Applications / RAKwireless\_Test\_Application'. Below it, there are tabs for DEVICES (which is selected), APPLICATION CONFIGURATION, INTEGRATIONS, and FUOTA. In the top right corner of the main content area, there's a red-bordered button labeled '+ CREATE'. Below this, there's a table header with columns: Last seen, Device name, Device EUI, Link margin, and Battery. At the bottom of the table area, it says 'Rows per page: 10 ▾ 0-0 of 0 < >'.

**Figure 50:** Adding a Node Device

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



The screenshot shows the 'Create Device' form. It has three tabs: GENERAL (selected), VARIABLES, and TAGS. Under GENERAL, there are fields for 'Device name \*' (RAK\_LoRa\_Node), 'Device description \*' (RAKwireless Test Device Description), 'Device EUI \*' (fe e4 75 18 ea 59 7b 51), 'Device-profile \*' (device\_profile\_otaa), and a checkbox for 'Disable frame-counter validation'. A note below the checkbox states: 'Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.' At the bottom right of the form is a 'CREATE DEVICE' button. A red box highlights the 'C' icon next to the Device EUI field.

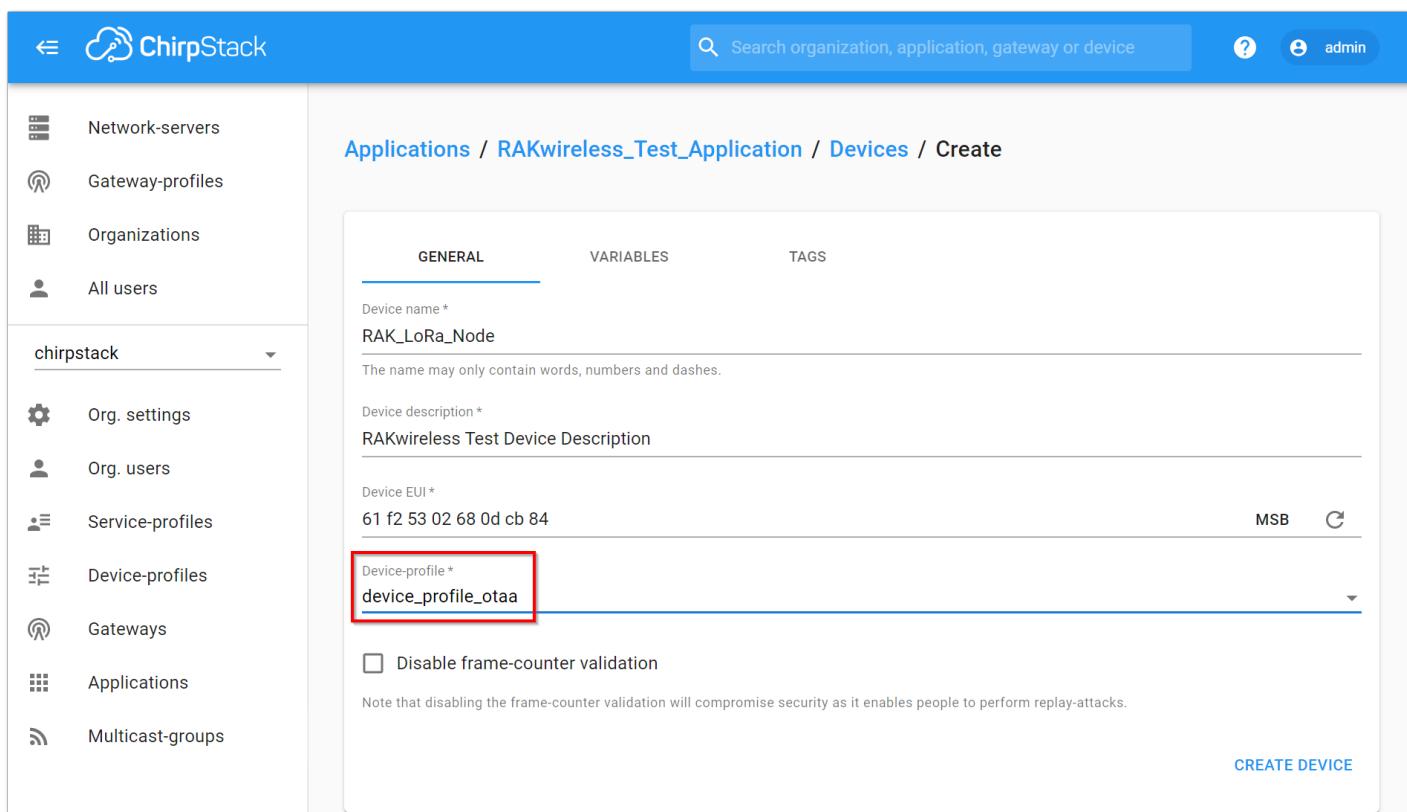
**Figure 51:** Filling the Device Parameters

#### NOTE:

If you want to join in OTAA mode, select “DeviceProfile\_OTAA” in the “Device-profile” item. If you want to join in ABP mode and CN470 frequency, then, 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.

## OTAA Mode

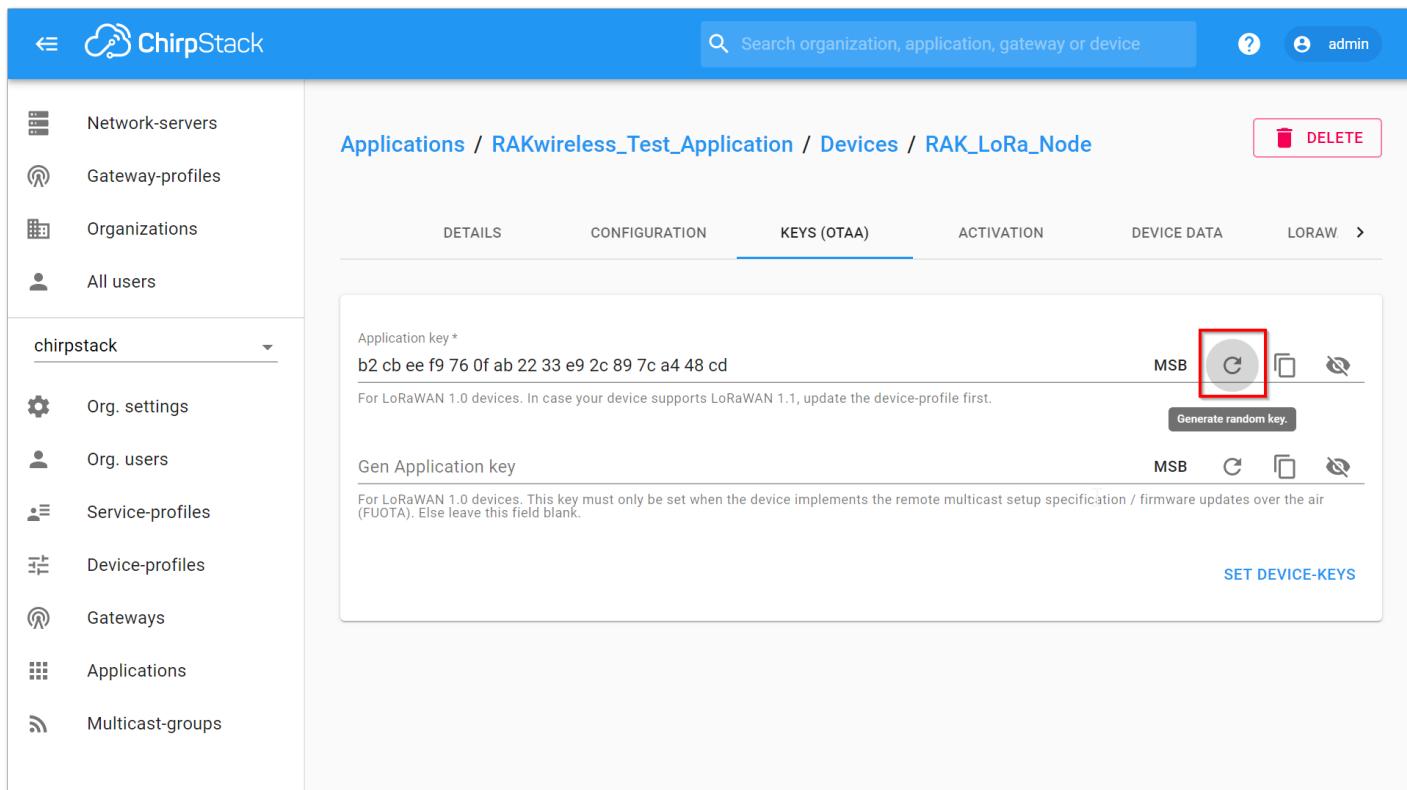
- To join ChirpStack in OTAA mode, select “**DeviceProfile\_OTAA**”.



The screenshot shows the ChirpStack Documentation Center interface. On the left, there's a sidebar with various navigation options like Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'chirpstack' which includes Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main area shows the path 'Applications / RAKwireless\_Test\_Application / Devices / Create'. There are three tabs: GENERAL (selected), VARIABLES, and TAGS. Under GENERAL, there are fields for 'Device name \*' (RAK\_LoRa\_Node), 'Device description \*' (RAKwireless Test Device Description), 'Device EUI \*' (61 f2 53 02 68 0d cb 84), and 'Device-profile\*' (device\_profile\_otaa). A checkbox for 'Disable frame-counter validation' is present with a note about security. At the bottom right is a 'CREATE DEVICE' button.

**Figure 52:** Selecting OTAA Activation Mode in ChirpStack

- Press “CREATE DEVICE” button. You may write the application key by yourself or generate it automatically by clicking the icon highlighted in the image.

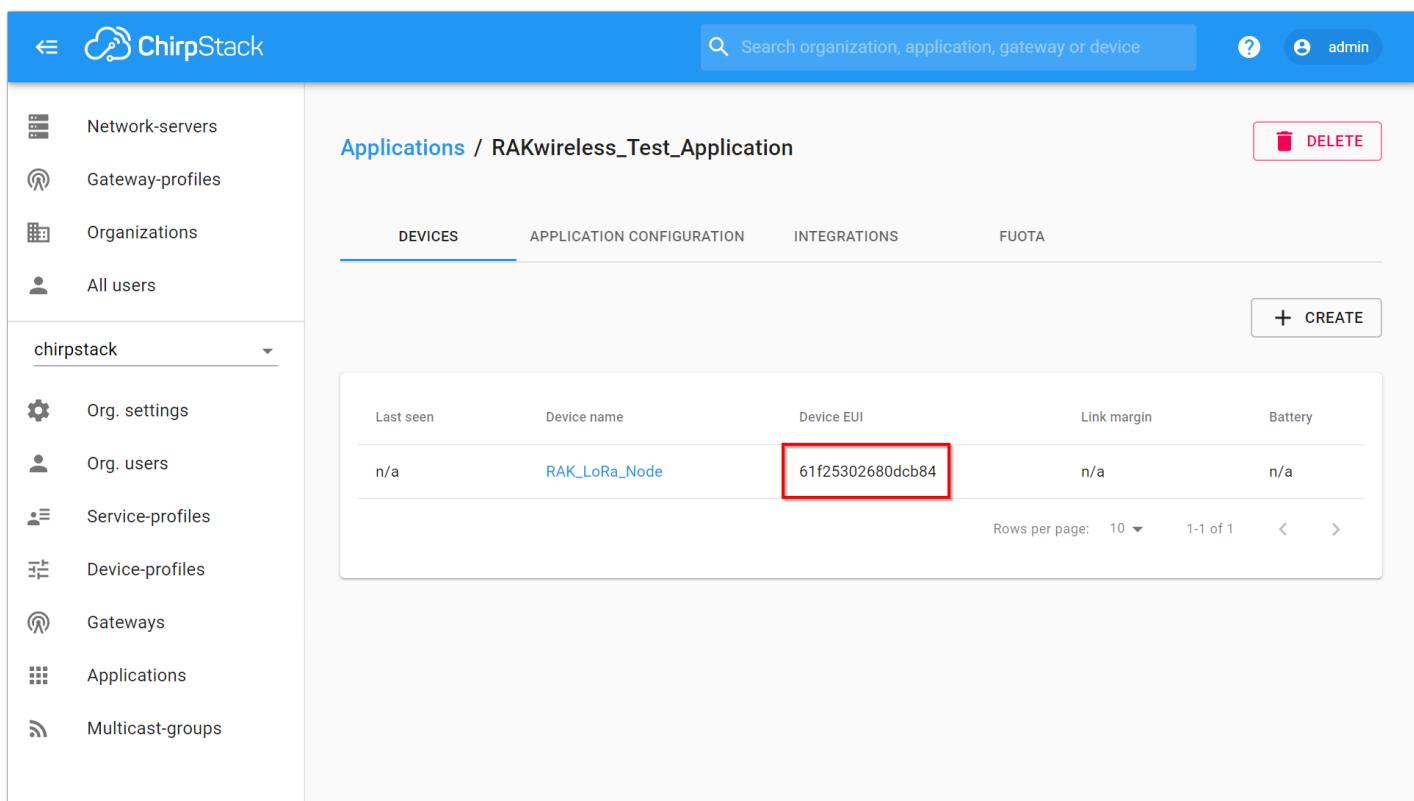


The screenshot shows the ChirpStack Documentation Center interface for the 'RAK\_LoRa\_Node' device. The sidebar is the same as in Figure 52. The main area shows the path 'Applications / RAKwireless\_Test\_Application / Devices / RAK\_LoRa\_Node'. There are several tabs: DETAILS, CONFIGURATION, KEYS (OTAA) (selected), ACTIVATION, DEVICE DATA, and LORAWAN. Under KEYS (OTAA), there are two fields: 'Application key\*' containing the hex value 'b2 cb ee f9 76 0f ab 22 33 e9 2c 89 7c a4 48 cd' and 'Gen Application key' which is empty. Below these fields are buttons for 'MSB' (radioed), a copy icon (highlighted with a red box), a delete icon, and a 'Generate random key' button. At the bottom right is a 'SET DEVICE-KEYS' button.

**Figure 53:** Application Key Generation

- Click "SET DEVICE KEYS" button. Now, you've completed the configuration on ChirpStack.

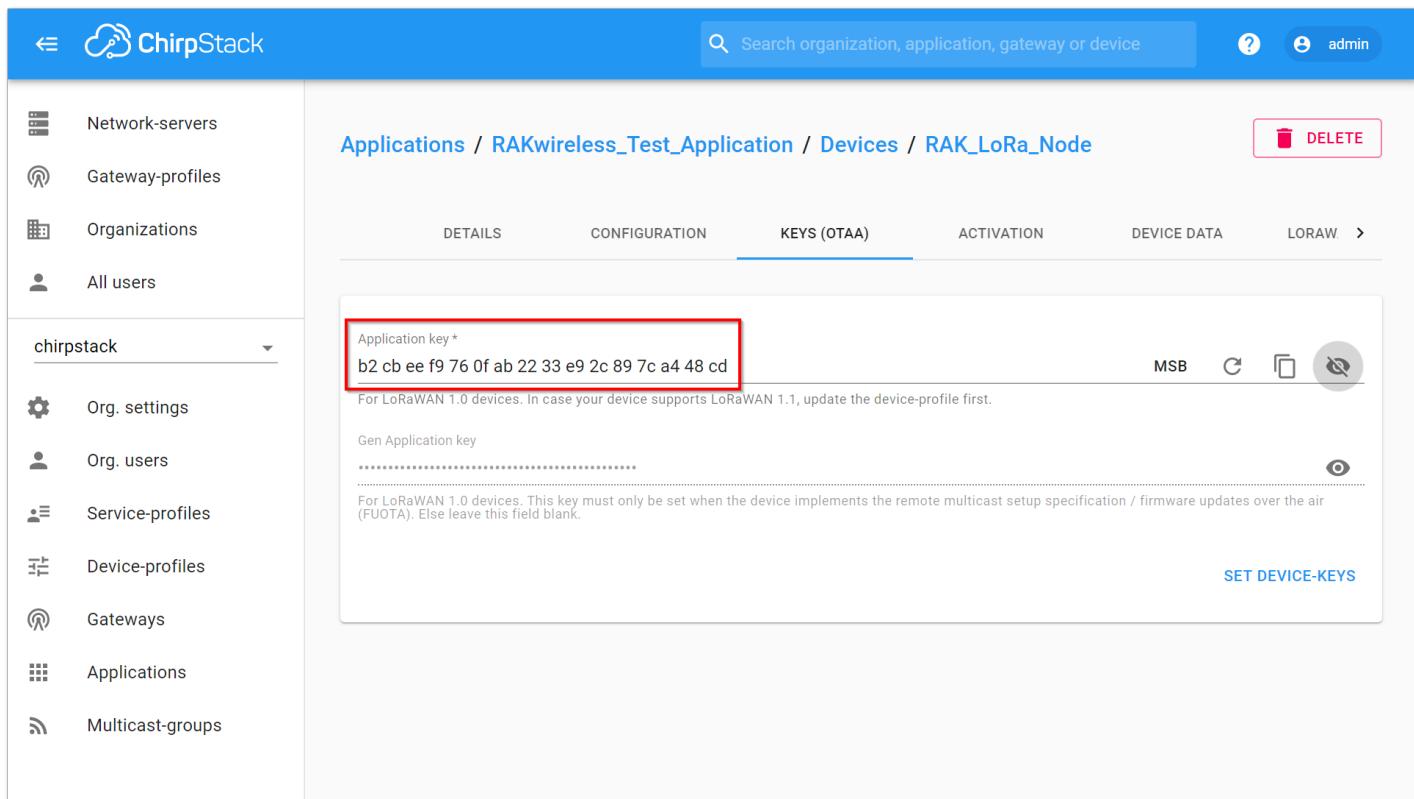
- The Device EUI which was set in the previous section to your RAK7204 WisNode Sense Home as "dev\_eui" is the same in the image highlighted below.



The screenshot shows the ChirpStack Documentation Center interface. On the left, there is a sidebar with various navigation options like Network-servers, Gateway-profiles, Organizations, All users, and chirpstack. The chirpstack section is expanded, showing Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled "Applications / RAKwireless\_Test\_Application". It has tabs for DEVICES, APPLICATION CONFIGURATION, INTEGRATIONS, and FUOTA. Below the tabs is a table with columns: Last seen, Device name, Device EUI, Link margin, and Battery. A single row is present with "n/a" in the first three columns and "n/a" in the last two. The "Device name" column contains "RAK\_LoRa\_Node" and the "Device EUI" column contains "61f25302680dcb84", which is highlighted with a red box. At the bottom of the table, it says "Rows per page: 10 ▾ 1-1 of 1 < >".

**Figure 54:** Device EUI Code

- Same with the Application Key, which was set in the previous section as "app\_key" is the same with the image highlighted.



The screenshot shows the ChirpStack Documentation Center interface, similar to Figure 54 but with more detail. The sidebar and main navigation are identical. The main content area is titled "Applications / RAKwireless\_Test\_Application / Devices / RAK\_LoRa\_Node". It has tabs for DETAILS, CONFIGURATION, KEYS (OTAA), ACTIVATION, DEVICE DATA, and LORAWAN. The KEYS (OTAA) tab is active. Under this tab, there is a form field labeled "Application key \*" containing the hex value "b2 cb ee f9 76 0f ab 22 33 e9 2c 89 7c a4 48 cd", which is highlighted with a red box. Below the field, a note says "For LoRaWAN 1.0 devices. In case your device supports LoRaWAN 1.1, update the device-profile first." There are also sections for "Gen Application key" and "For LoRaWAN 1.0 devices. This key must only be set when the device implements the remote multicast setup specification / firmware updates over the air (FUOTA). Else leave this field blank." At the bottom right of the form, there is a "SET DEVICE-KEYS" button.

**Figure 55:** Application Key LoRaWAN

 **NOTE:**

The Application EUI which was into RAK7204 WisNode Sense Home as “app\_eui” is not needed for ChirpStack.

4. Next, let's **configure** RAK7204 WisNode Sense Home by using **AT commands**. To do this, connect your RAK7204 WisNode Sense Home to a PC, power it on and open **RAK Serial Port Tool** on your computer.

```
at+version
```

sh

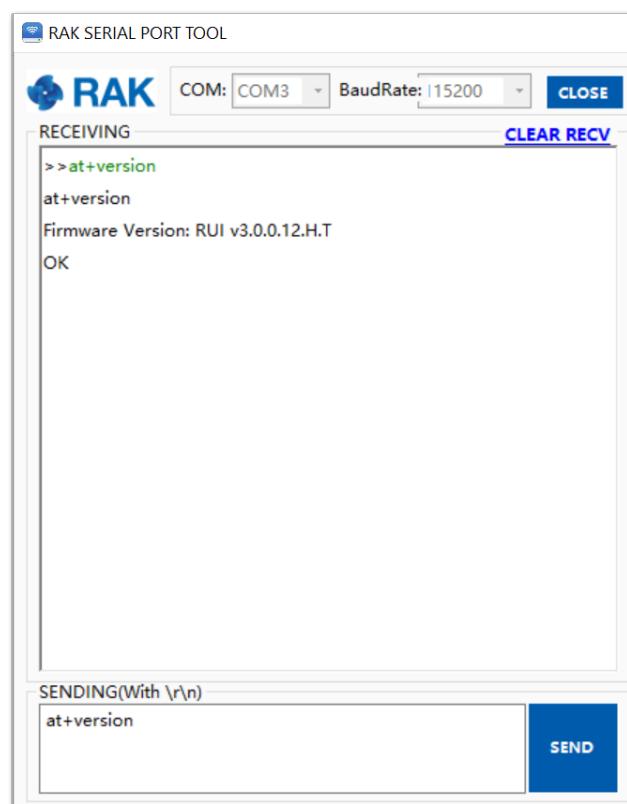


Figure 56: RAK Serial Port Tool

- Now, let us join our RAK7204 WisNode Sense Home using the OTAA activation mode.
5. If the join mode is not in OTAA, just set the LoRa join mode to **OTAA** and LoRa class to **Class A** by typing the AT commands shown in the picture below.

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

sh

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

sh

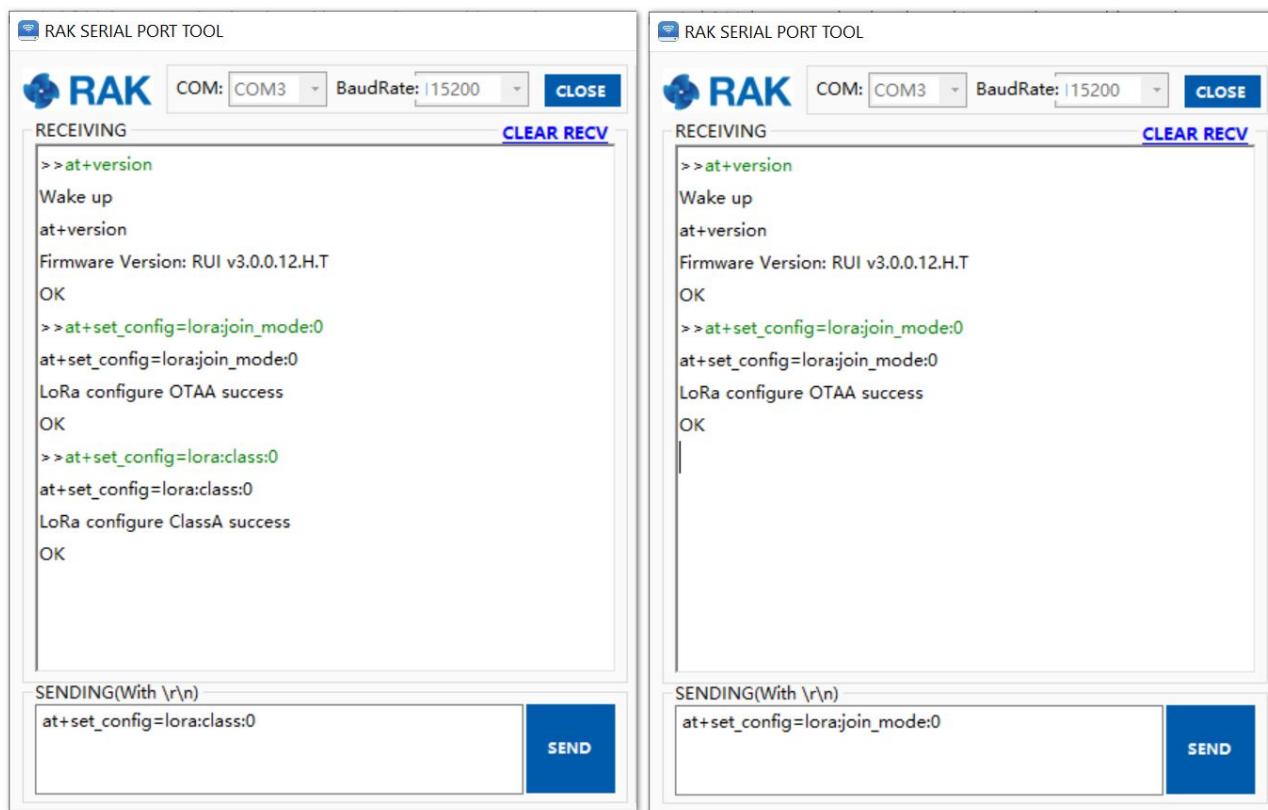


Figure 57: Setting of LoRaWAN mode and class

6. Type the following AT command to set the: **Frequency/Region**, **Device EUI**, **Application EUI** and **Application Key**. Remember to replace the "XXXX" with the corresponding parameter value for your particular case:

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

```
at+set_config=lora:dev_eui:XXXX
```

```
at+set_config=lora:app_eui:XXXX
```

```
at+set_config=lora:app_key:XXXX
```

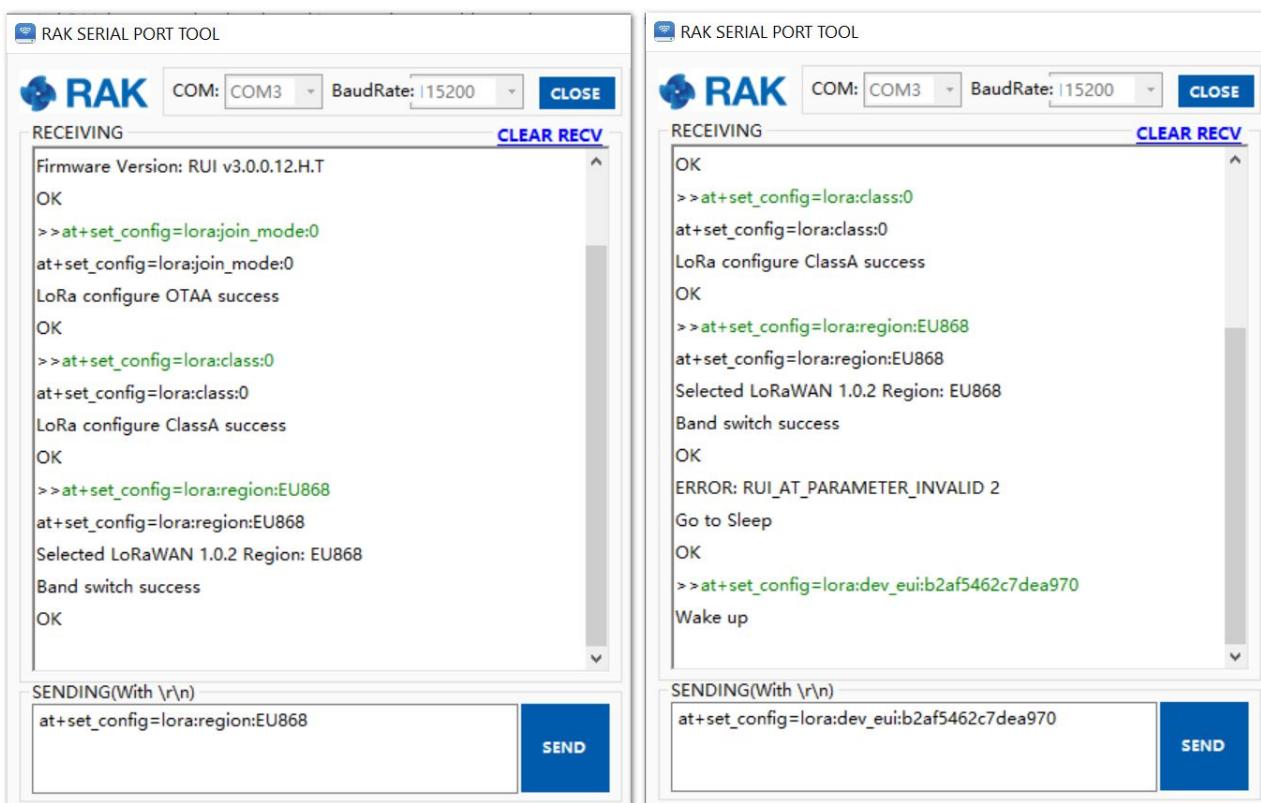


Figure 58: Setting of Frequency and Device EUI

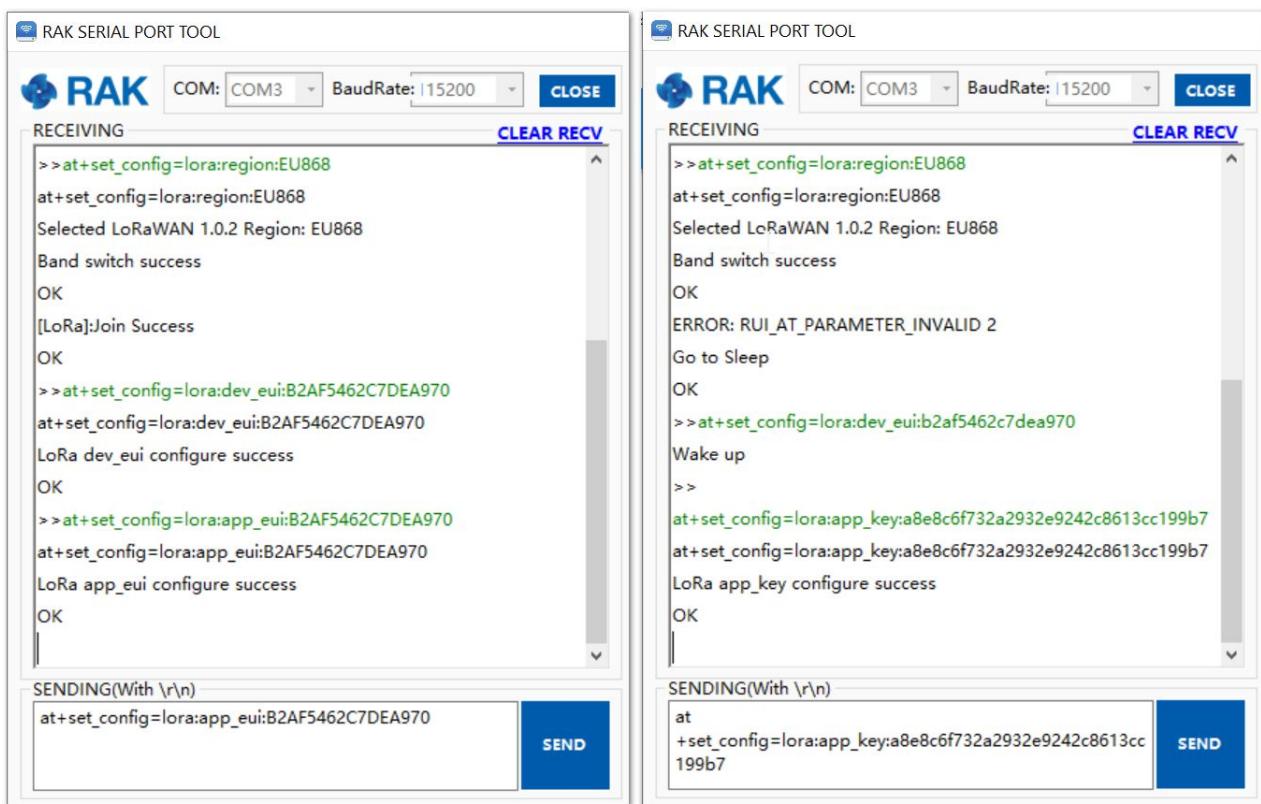


Figure 59: Setting of Application EUI and Key

7. Then, **join** in OTAA mode.

at+join

sh

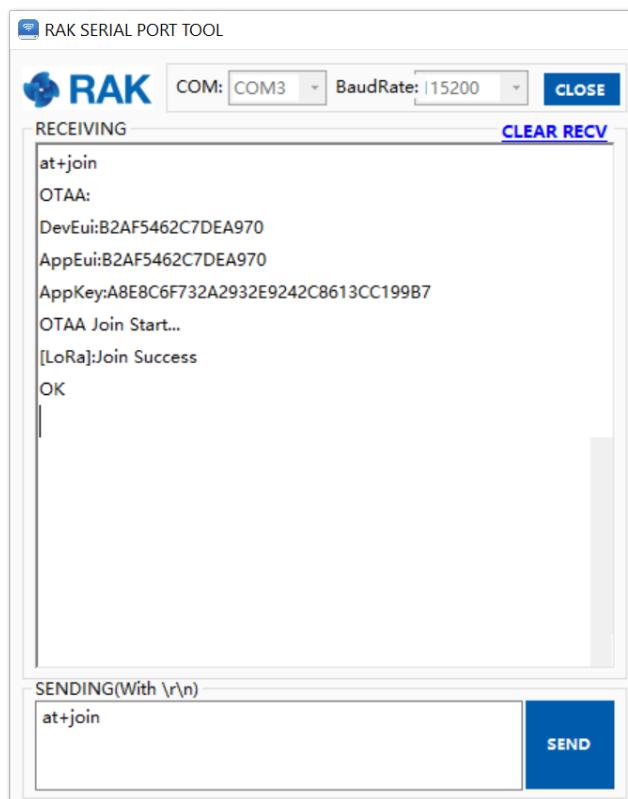


Figure 60: Joining in OTAA

- Joined Successfully! 🎉

8. You can view the "JoinRequest" and "JoinAccept" on ChirpStack page:

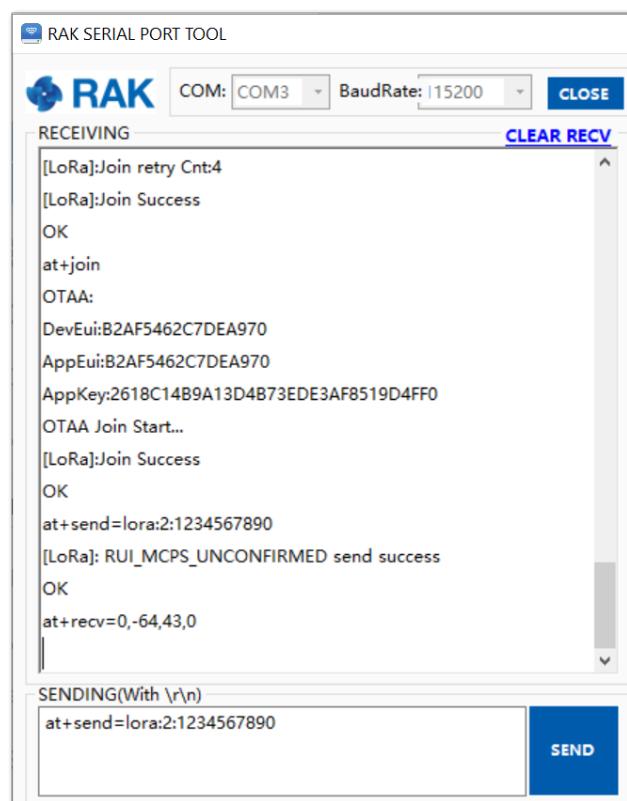
Frame Type	Time	Message Type	MAC Address
Downlink	5:42:29 PM	JoinAccept	5e9d1e0857cf25f1
Uplink	5:42:29 PM	JoinRequest	5e9d1e0857cf25f1

Figure 61: Join Request of the Device in the ChirpStack

9. Let's try sending data from our RAK7204 WisNode Sense Home to the ChirpStack by typing the command below in the serial port.

```
at+send=lora:2:1234567890
```

sh



**Figure 62:** Sending Data to ChirpStack

- You can see the message on ChirpStack page:

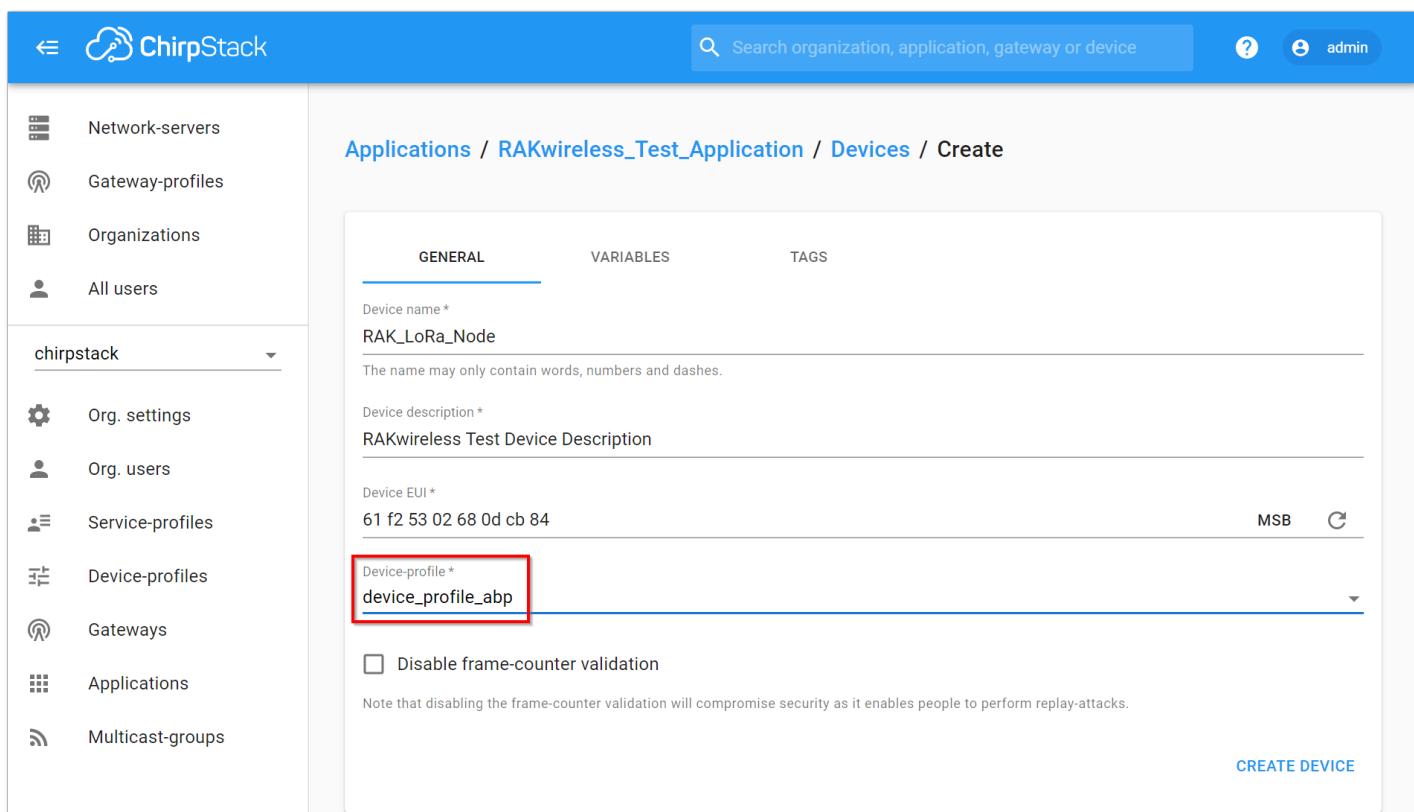
**Figure 63:** Message Received in ChirpStack

## ABP Mode

1. If you select “Device Profile ABP” or “DeviceProfile\_ABP\_CN470”, it means you want to join ChirpStack in **ABP mode**.

### WARNING

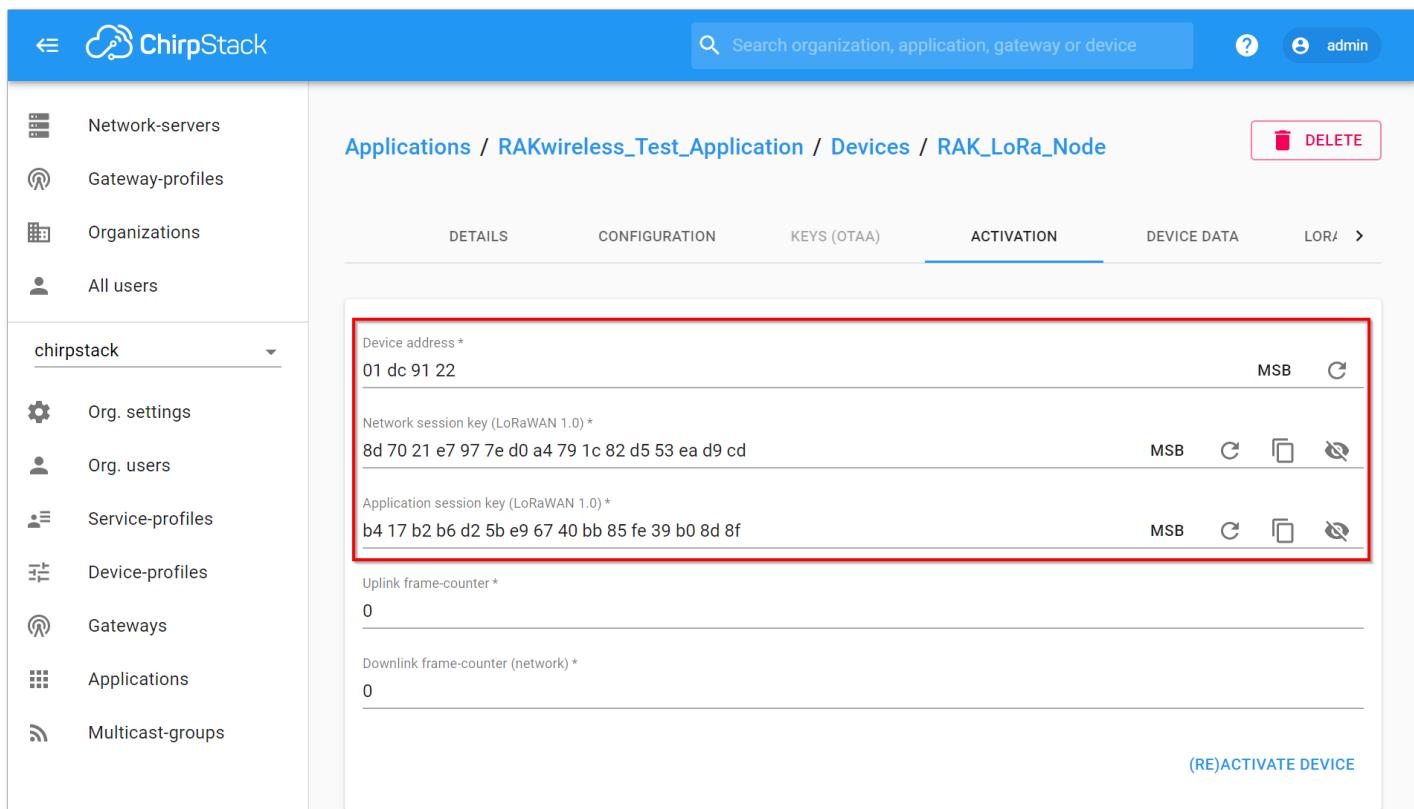
Frequency AS923 in ABP Mode is not supported in ChirpStack



The screenshot shows the ChirpStack Documentation Center interface. On the left, a sidebar menu under 'chirpstack' includes 'Network-servers', 'Gateway-profiles', 'Organizations', 'All users', 'Org. settings', 'Org. users', 'Service-profiles', 'Device-profiles', 'Gateways', 'Applications', and 'Multicast-groups'. The main area displays the path 'Applications / RAKwireless\_Test\_Application / Devices / Create'. The 'GENERAL' tab is selected, showing fields for 'Device name \*' (RAK\_LoRa\_Node), 'Device description \*' (RAKwireless Test Device Description), 'Device EUI \*' (61 f2 53 02 68 0d cb 84), and 'Device-profile \*' (device\_profile\_abp). A checkbox for 'Disable frame-counter validation' is present with a note about replay-attacks. A 'CREATE DEVICE' button is at the bottom right.

**Figure 64:** Switching to ABP Mode

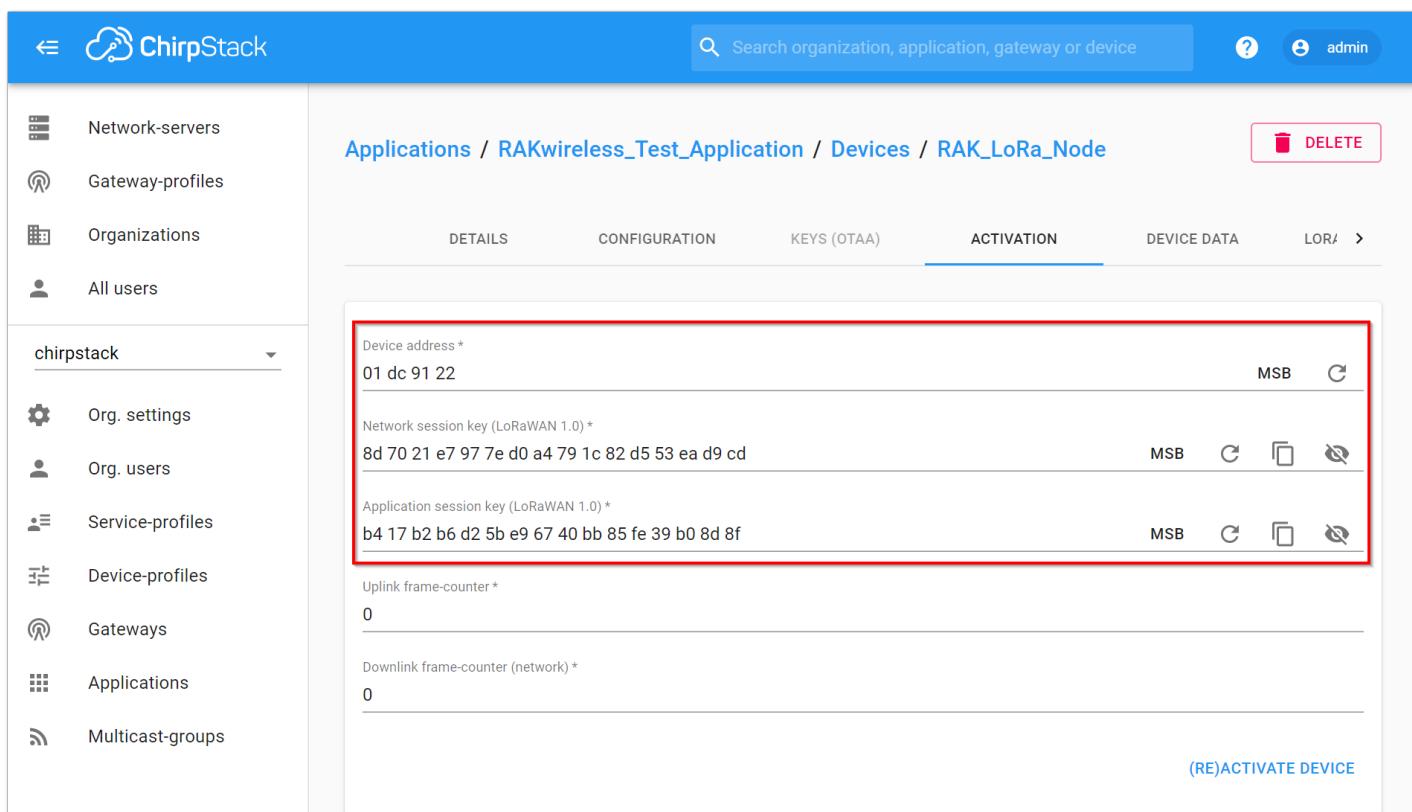
2. Then you can see that there are some parameters for ABP in the “**ACTIVATION**” item:



The screenshot shows the ChirpStack Documentation Center interface for the device 'RAK\_LoRa\_Node'. The sidebar is identical to Figure 64. The main area shows the device details with tabs for 'DETAILS', 'CONFIGURATION', 'KEYS (OTAA)', 'ACTIVATION' (which is selected), 'DEVICE DATA', and 'LORA'. In the 'ACTIVATION' tab, several parameters are listed: 'Device address \*' (01 dc 91 22), 'Network session key (LoRaWAN 1.0) \*' (8d 70 21 e7 97 7e d0 a4 79 1c 82 d5 53 ea d9 cd), and 'Application session key (LoRaWAN 1.0) \*' (b4 17 b2 b6 d2 5b e9 67 40 bb 85 fe 39 b0 8d 8f). These three parameters are highlighted with a red box. Below them are 'Uplink frame-counter \*' (0) and 'Downlink frame-counter (network)' (0). A '(RE)ACTIVATE DEVICE' button is at the bottom right.

**Figure 65:** ABP Parameters

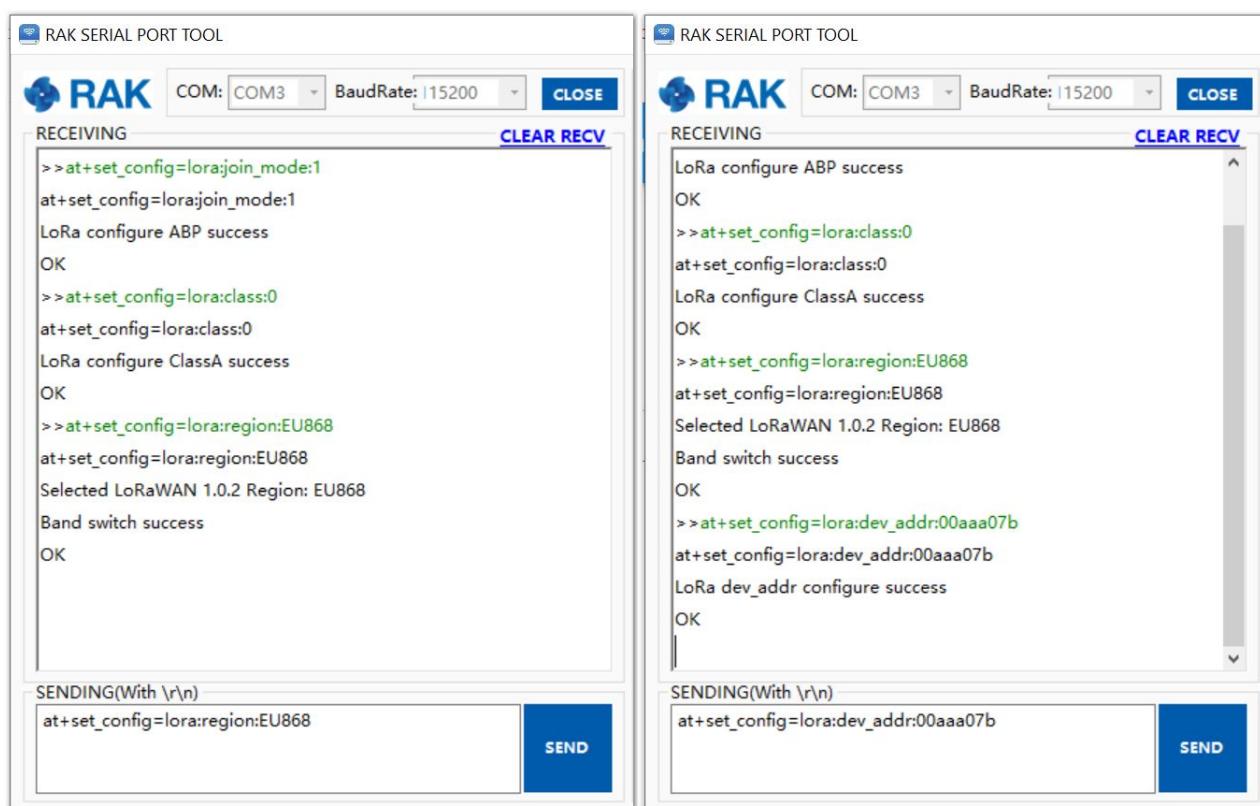
3. Next, let's use these parameters to set WisNode LoRa by using **AT command**. Let's join in **ABP** mode and set **EU868** frequency as an example.
4. If the join mode is not in ABP, just set the LoRa join mode to **ABP** and LoRa class to **Class A** by typing the following commands in RAK Serial Port Tool



The screenshot shows the ChirpStack Documentation Center interface. On the left, there's a sidebar with categories like Network-servers, Gateway-profiles, Organizations, All users, chirpstack, Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main area shows the path: Applications / RAKwireless\_Test\_Application / Devices / RAK\_LoRa\_Node. The 'ACTIVATION' tab is selected. A red box highlights the 'Network session key (LoRaWAN 1.0)' and 'Application session key (LoRaWAN 1.0)' fields, both of which have 'MSB' and circular icons next to them.

Figure 66: Setting of LoRaWAN Mode and Class

- Type the following AT command to set your respective: **Frequency/Region**, **Device Address**, **Network Session Key** and **App Session Key**.



The screenshot shows two instances of the RAK SERIAL PORT TOOL. The left instance has 'COM: COM3' and 'BaudRate: 115200'. It shows the following AT command session:

```

>>at+set_config=lora:join_mode:1
at+set_config=lora:join_mode:1
LoRa configure ABP success
OK
>>at+set_config=lora:class:0
at+set_config=lora:class:0
LoRa configure ClassA success
OK
>>at+set_config=lora:region:EU868
at+set_config=lora:region:EU868
Selected LoRaWAN 1.0.2 Region: EU868
Band switch success
OK

```

The right instance also has 'COM: COM3' and 'BaudRate: 115200'. It shows the following AT command session:

```

LoRa configure ABP success
OK
>>at+set_config=lora:class:0
at+set_config=lora:class:0
LoRa configure ClassA success
OK
>>at+set_config=lora:region:EU868
at+set_config=lora:region:EU868
Selected LoRaWAN 1.0.2 Region: EU868
Band switch success
OK
>>at+set_config=lora:dev_addr:00aaa07b
at+set_config=lora:dev_addr:00aaa07b
LoRa dev_addr configure success
OK

```

Figure 67: Setting of Frequency and Device Address

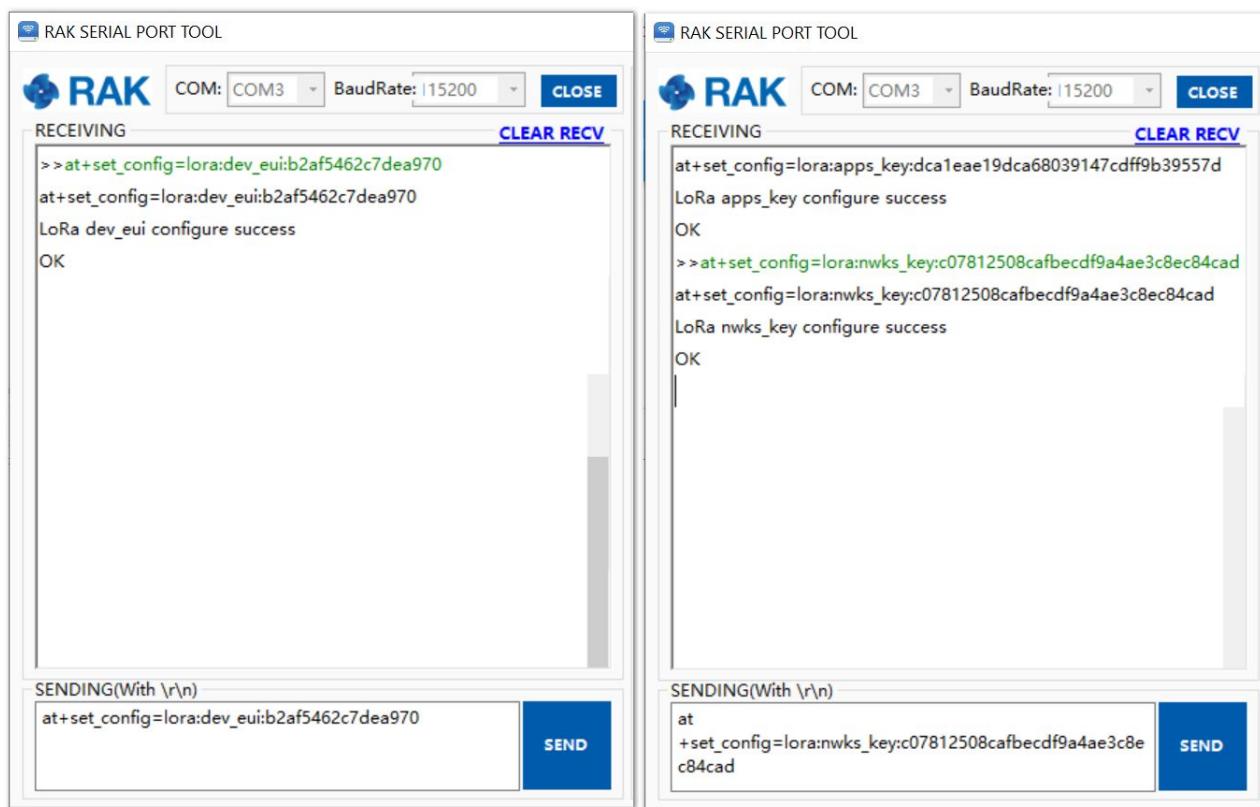


Figure 68: Setting of Device EUI and Network Session Key

#### 6. Then, join in ABP mode.

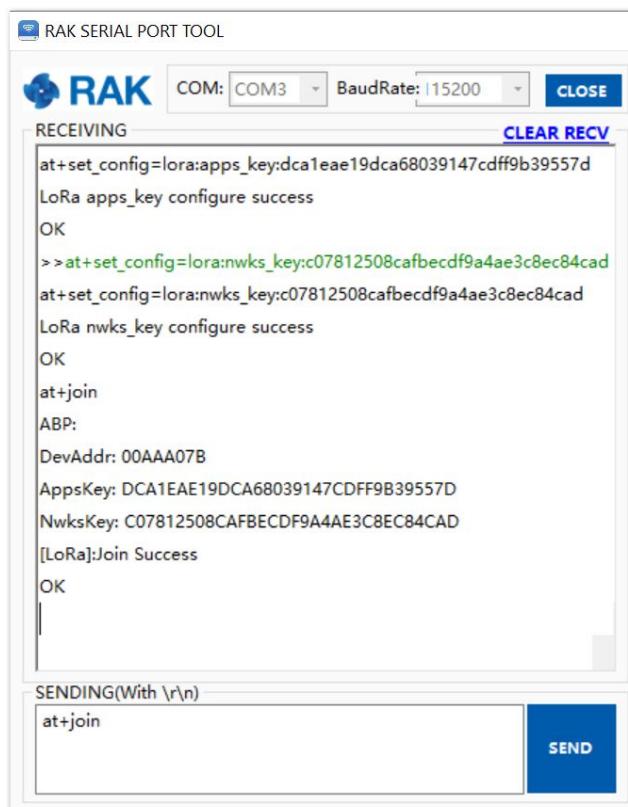


Figure 69: Joining of ABP

- Now, try sending data from our WisNode LoRa to the Chirpstack

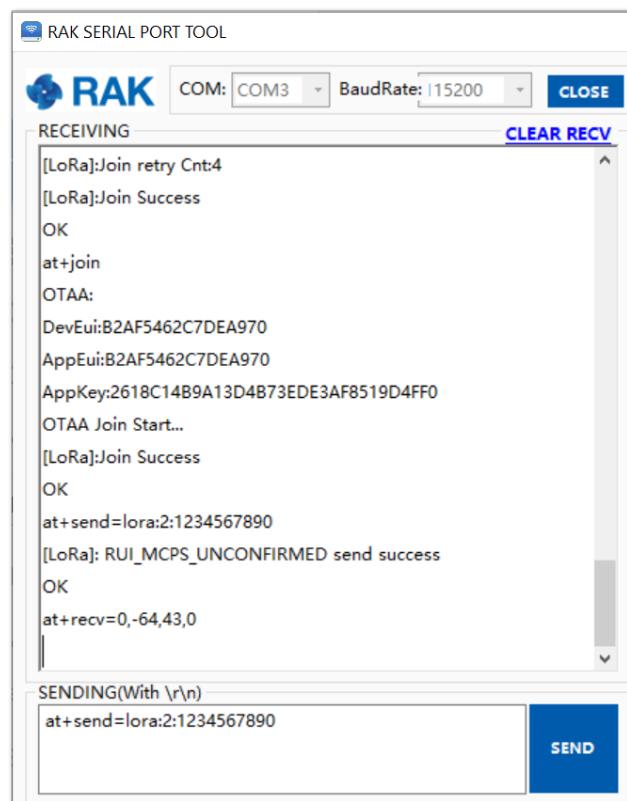
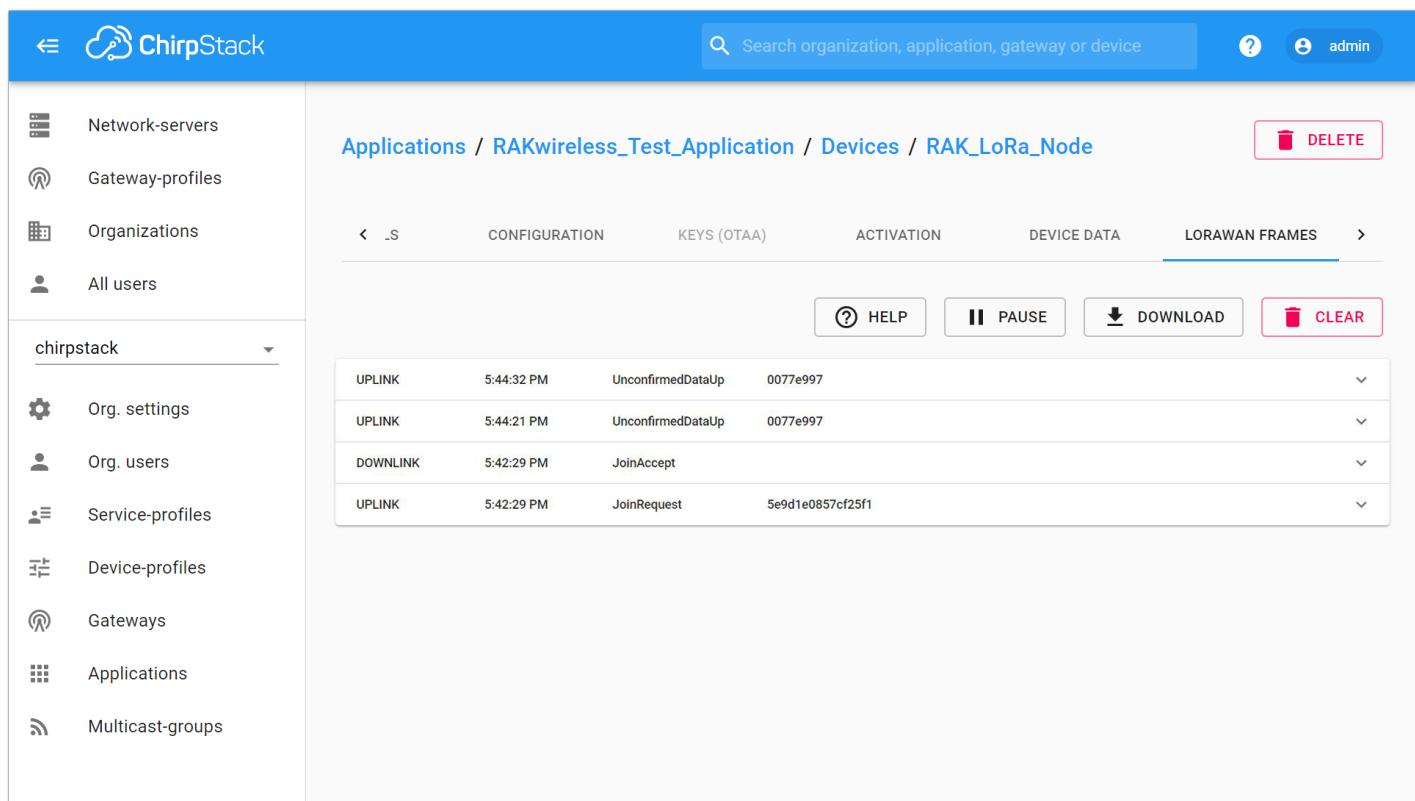


Figure 70: Sending Data to ChirpStack

- You can see the data which is just sent from RAK7204 WisNode Sense Home on ChirpStack page:



The screenshot shows the ChirpStack web interface. The left sidebar has a navigation menu with items like Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for "chirpstack". The main content area shows the path "Applications / RAKwireless\_Test\_Application / Devices / RAK\_LoRa\_Node". There are tabs for CONFIGURATION, KEYS (OTAA), ACTIVATION, DEVICE DATA, and LORAWAN FRAMES, with LORAWAN FRAMES selected. Below the tabs are buttons for HELP, PAUSE, DOWNLOAD, and CLEAR. A table lists LORAWAN FRAMES:

	UPLINK	5:44:32 PM	UnconfirmedDataUp	0077e997
	UPLINK	5:44:21 PM	UnconfirmedDataUp	0077e997
	DOWNLINK	5:42:29 PM	JoinAccept	
	UPLINK	5:42:29 PM	JoinRequest	5e9d1e0857cf25f1

Figure 71: Message Status in ChirpStack

## Decoding Sensor Data on ChirpStack and TTN Analyzing Sensor Data from RAK7204 WisNode Sense Home

In the previous section, we have successfully sent some raw data from our RAK7204 WisNode Sense Home to The Things Network, but the problem is that you can't really see the actual sensor data from the payload. In this section, we will solve that and understand what each payload means.

Let's take this data for example:



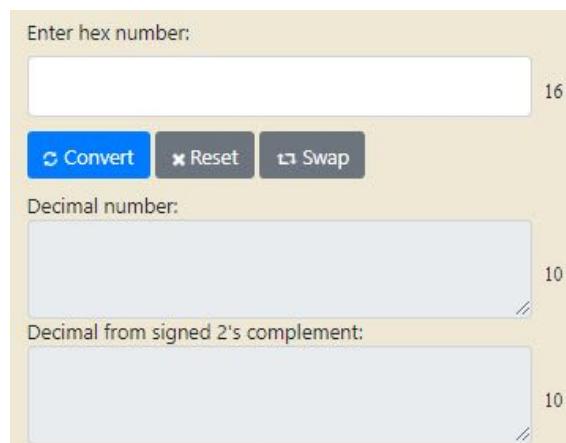
**Figure 72:** Received Raw Data in TTN



**Figure 73:** Actual Data sent to Cayenne

For this example, the payload is : **08 02 01 63 07 68 4B 06 73 25 9E 02 67 01 15 04 02 22 72 04 02 22 72**

Now lets analyze each data , which is in Hexadecimal Format. We will be using the data mentioned above as an example. We will convert the Hexadecimal Data into Decimal Data using this [converter](#) in order to be able to understand it.



**Figure 74:** Hexadecimal to Decimal converter

## 1. Battery Voltage

Parameter	Hex Data	Decimal Equivalent	Multiplier	True Value
Data flag	08 02			
Battery Voltage	01 67	355	0.01 Signed	3.55 V

## 2. Humidity Data

Example Data: **07 68 4B**

Parameter	Hex Data	Decimal Equivalent	Multiplier	True Value
Data flag	07 68			
Humidity	4B	75	0.5 % Unsigned	37.5 % RH

### 3. Pressure Data

Example Data: **06 73 25 9E**

Parameter	Hex Data	Decimal Equivalent	Multiplier	True Value
Data flag	06 73			
Pressure	25 9E	9630	0.1 hPa Unsigned MSB	963.0 hPa

### 4. Temperature Data

Example Data: **02 67 01 15**

Parameter	Hex Data	Decimal Equivalent	Multiplier	True Value
Data flag	02 67			
Temperature	01 15	277	0.1 °C Signed MSB	27.7°C

### 5. Gas Resistance Data

Example Data: **04 02 22 72**

Parameter	Hex Data	Decimal Equivalent	Multiplier	True Value
Data flag	04 02			
Gas Resistance	22 72	8818	0.01 kΩ Signed	88.18 kΩ

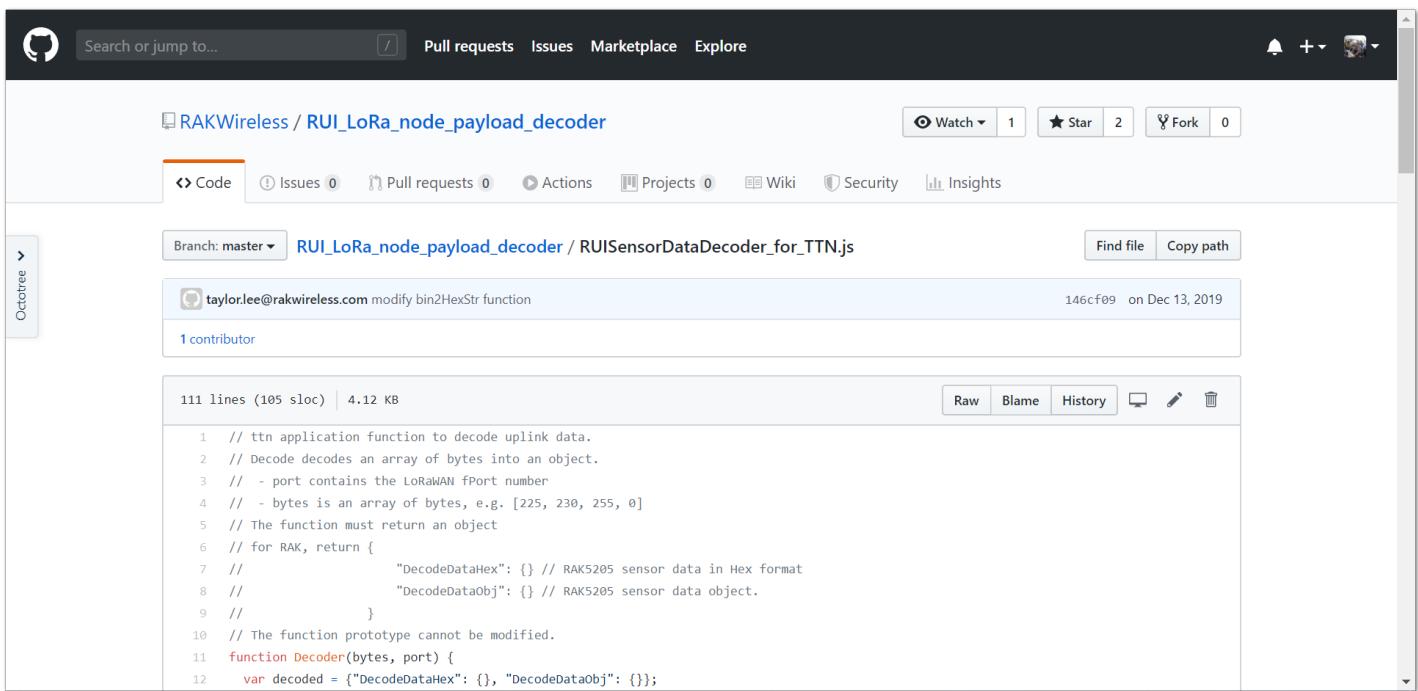
For further details about the LPP format, you can take a look at this [link](#).

## Decoding Sensor Data in TTN

### Input Decoding Function in TTN

1. To start with, download the decoding function through this [link](#).

► Click me to view the code



```

1 // ttn application function to decode uplink data.
2 // Decode decodes an array of bytes into an object.
3 // - port contains the LoRaWAN fPort number
4 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
5 // The function must return an object
6 // for RAK, return {
7 //           "DecodeDataHex": {} // RAK5205 sensor data in Hex format
8 //           "DecodeDataObj": {} // RAK5205 sensor data object.
9 //         }
10 // The function prototype cannot be modified.
11 function Decoder(bytes, port) {
12   var decoded = {"DecodeDataHex": {}, "DecodeDataObj": {}};

```

Figure 75: Github Page for the Decoding Function

- From your TTN console, go to application page and click the "**Payload Formats**" tab as shown in the image below.

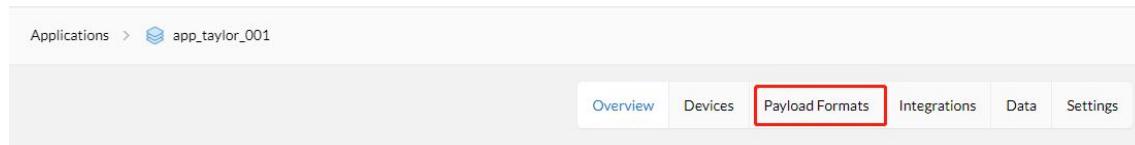
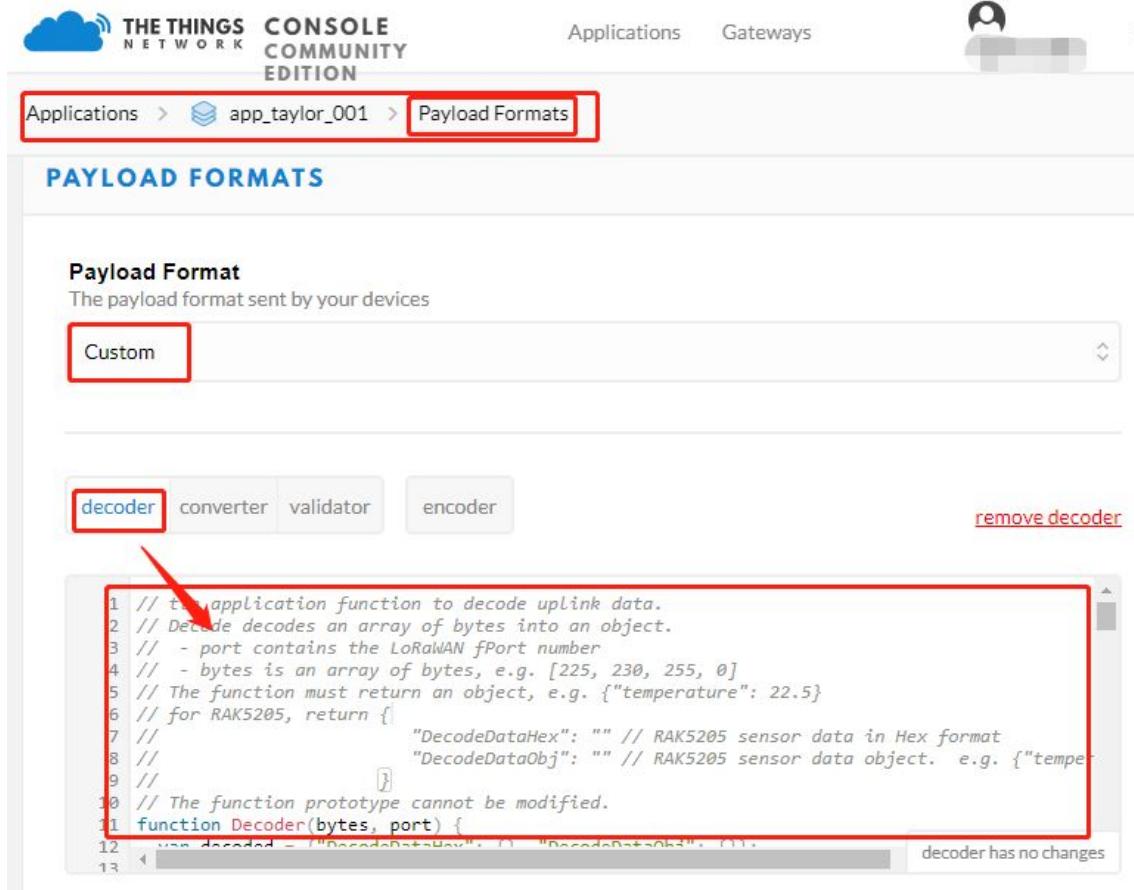


Figure 76: Payload Format at TTN Application Page

- Next, select "**Payload Format**" as "**Custom**". Then, from the decoder tab, copy and paste the decoder function from step 1.



```

1 // ttn application function to decode uplink data.
2 // Decode decodes an array of bytes into an object.
3 // - port contains the LoRaWAN fPort number
4 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
5 // The function must return an object, e.g. {"temperature": 22.5}
6 // for RAK5205, return {
7 //           "DecodeDataHex": "" // RAK5205 sensor data in Hex format
8 //           "DecodeDataObj": "" // RAK5205 sensor data object. e.g. {"tempe
9 //         }
10 // The function prototype cannot be modified.
11 function Decoder(bytes, port) {
12   var decoded = {"DecodeDataHex": {}, "DecodeDataObj": {}};

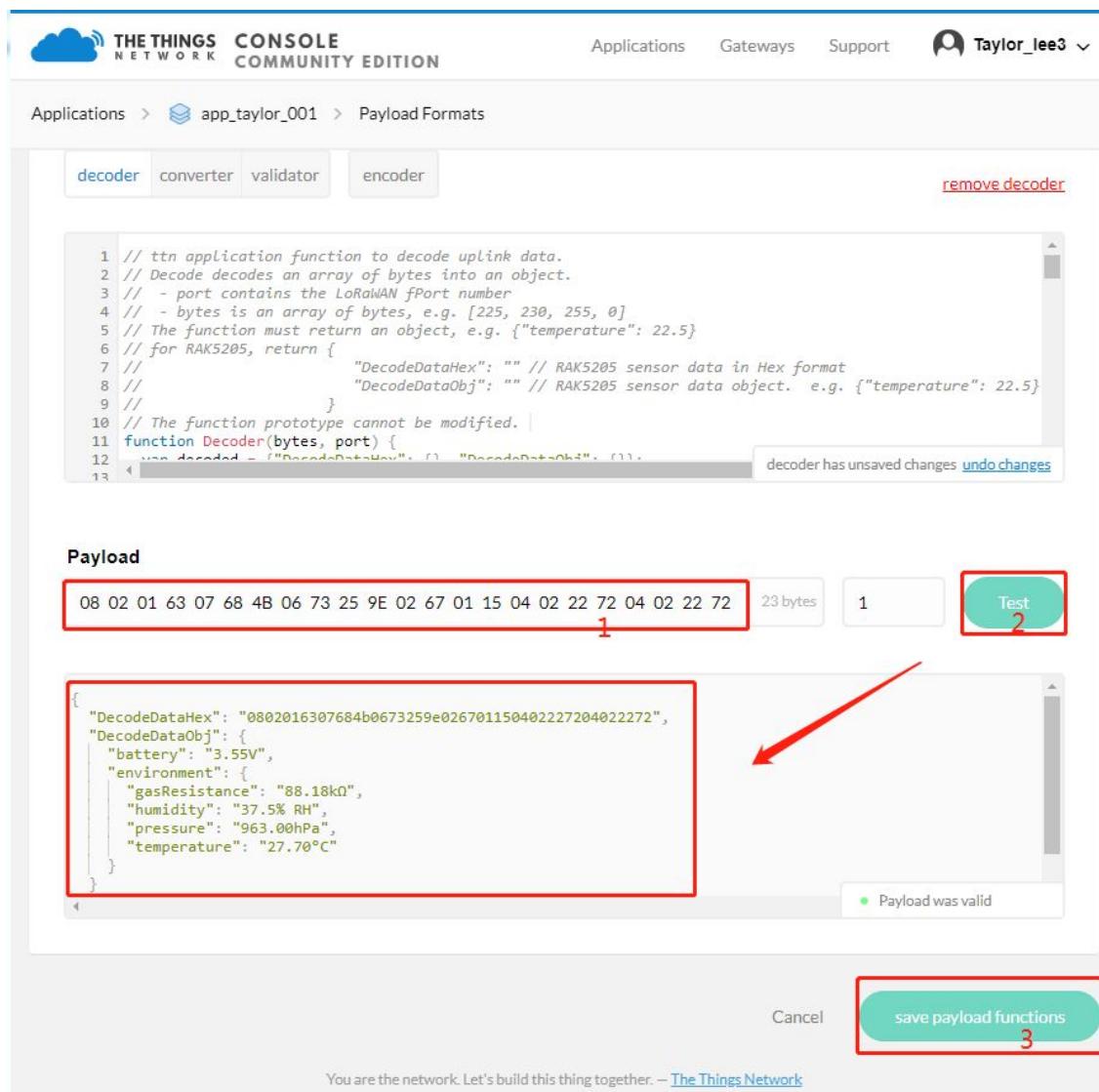
```

Figure 77: Decoder Function

## Testing the Validity of Decoding Sensor Data in TTN

- Input the data below in the "**Payload**" box as shown in the image below.

08 02 01 63 07 68 4B 06 73 25 9E 02 67 01 15 04 02 22 72 04 02 22 72



The screenshot shows the The Things Network Console interface. At the top, there's a navigation bar with 'THE THINGS NETWORK CONSOLE COMMUNITY EDITION', 'Applications', 'Gateways', 'Support', and a user profile 'Taylor\_lee3'. Below the navigation, the path 'Applications > app\_taylor\_001 > Payload Formats' is visible. A tabs bar at the top of the main area includes 'decoder' (which is selected), 'converter', 'validator', and 'encoder'. To the right of the tabs is a red link 'remove decoder'. The main content area contains a code editor with the following content:

```

1 // ttn application function to decode uplink data.
2 // Decode decodes an array of bytes into an object.
3 // - port contains the LoRaWAN fPort number
4 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
5 // The function must return an object, e.g. {"temperature": 22.5}
6 // for RAK5205, return {
7 //     "DecodeDataHex": "", // RAK5205 sensor data in Hex format
8 //     "DecodeDataObj": "" // RAK5205 sensor data object. e.g. {"temperature": 22.5}
9 // }
10 // The function prototype cannot be modified.
11 function Decoder(bytes, port) {
12     var decoded = {"DecodeDataHex": "", "DecodeDataObj": null};
13

```

A message at the bottom right says 'decoder has unsaved changes [undo changes](#)'. Below the code editor, there's a section titled 'Payload' containing a hex string '08 02 01 63 07 68 4B 06 73 25 9E 02 67 01 15 04 02 22 72 04 02 22 72' with a red border around it. To the right of the hex string are '23 bytes' and a button labeled '1'. Further to the right is a green button labeled 'Test' with a red border and a red arrow pointing to it from below. Below the hex string, a JSON object is shown:

```

{
    "DecodeDataHex": "0802016307684b0673259e026701150402227204022272",
    "DecodeDataObj": {
        "battery": "3.55V",
        "environment": {
            "gasResistance": "88.18kΩ",
            "humidity": "37.5% RH",
            "pressure": "963.00hPa",
            "temperature": "27.70°C"
        }
    }

```

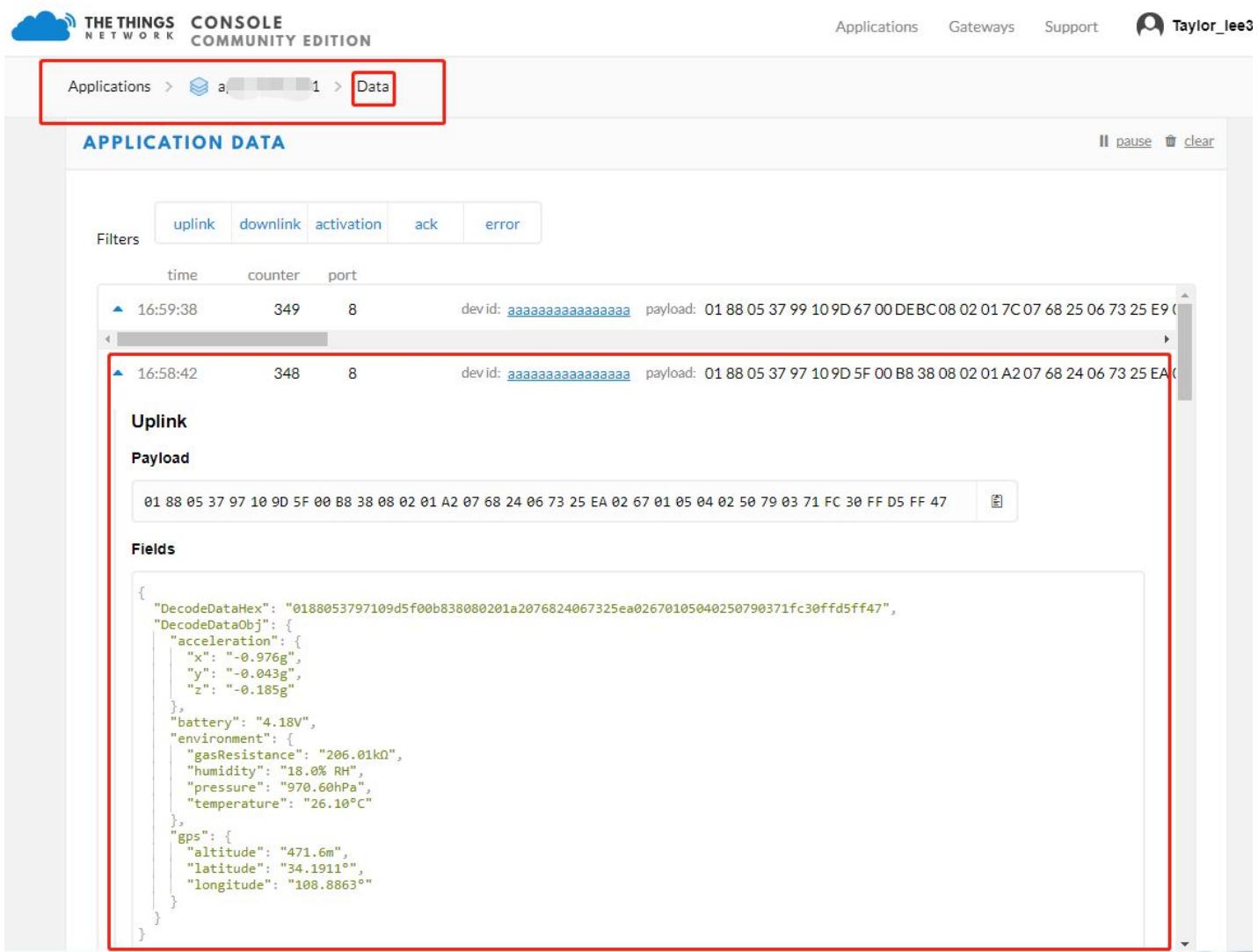
To the right of this JSON object is a green message 'Payload was valid'. At the bottom right of the payload section is a red button labeled 'save payload functions' with a red border and a red arrow pointing to it from above. A note at the bottom says 'You are the network. Let's build this thing together. — [The Things Network](#)'.

**Figure 78:** Testing Payload Data

2. Then, click "**Test**" and it will generate a code with the decoded data as shown in the image above.

## Testing in Real System in TTN

After gateway and node go online, click the uplink data record from the application data tab to check the decode status. From the image below, we can see the data decoded successfully in TTN.



The screenshot shows the 'APPLICATION DATA' section of the The Things Network Console. The 'Data' tab is selected in the breadcrumb navigation. The main area displays two uplink entries. The first entry at 16:59:38 has a dev id of 'aaaaaaaaaaaaaaaaaa' and a payload of '01 88 05 37 99 10 9D 67 00 DE BC 08 02 01 7C 07 68 25 06 73 25 E9 00'. The second entry at 16:58:42 has a dev id of 'aaaaaaaaaaaaaaaaaa' and a payload of '01 88 05 37 97 10 9D 5F 00 B8 38 08 02 01 A2 07 68 24 06 73 25 EA 02 67 01 05 04 02 50 79 03 71 FC 30 FF D5 FF 47'. Below these entries, the 'Uplink' and 'Payload' sections are visible, along with a large red box highlighting the 'Fields' section which contains the decoded JSON payload:

```
{
  "DecodeDataHex": "0188053797109d5f00b838080201a2076824067325ea02670105040250790371fc30ffd5ff47",
  "DecodeDataObj": {
    "acceleration": {
      "x": "-0.976g",
      "y": "-0.045g",
      "z": "-0.185g"
    },
    "battery": "4.18V",
    "environment": {
      "gasResistance": "206.01kΩ",
      "humidity": "18.0% RH",
      "pressure": "970.60hPa",
      "temperature": "26.10°C"
    },
    "gps": {
      "altitude": "471.6m",
      "latitude": "34.1911°",
      "longitude": "108.8863°"
    }
  }
}
```

Figure 79: Uplink Decoded Data

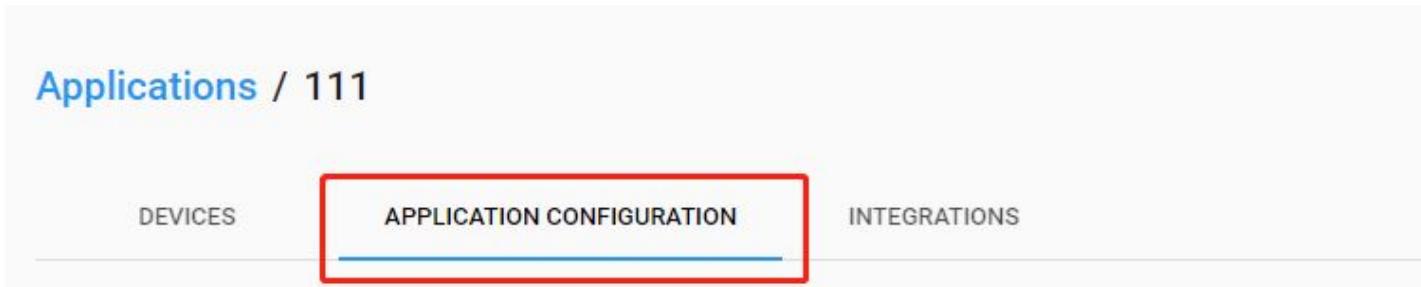
## Decoding Sensor Data in ChirpStack

### Input Decoding Function in ChirpStack

1. To start with, download the decoding function through this [link](#).

► Click me to view the code

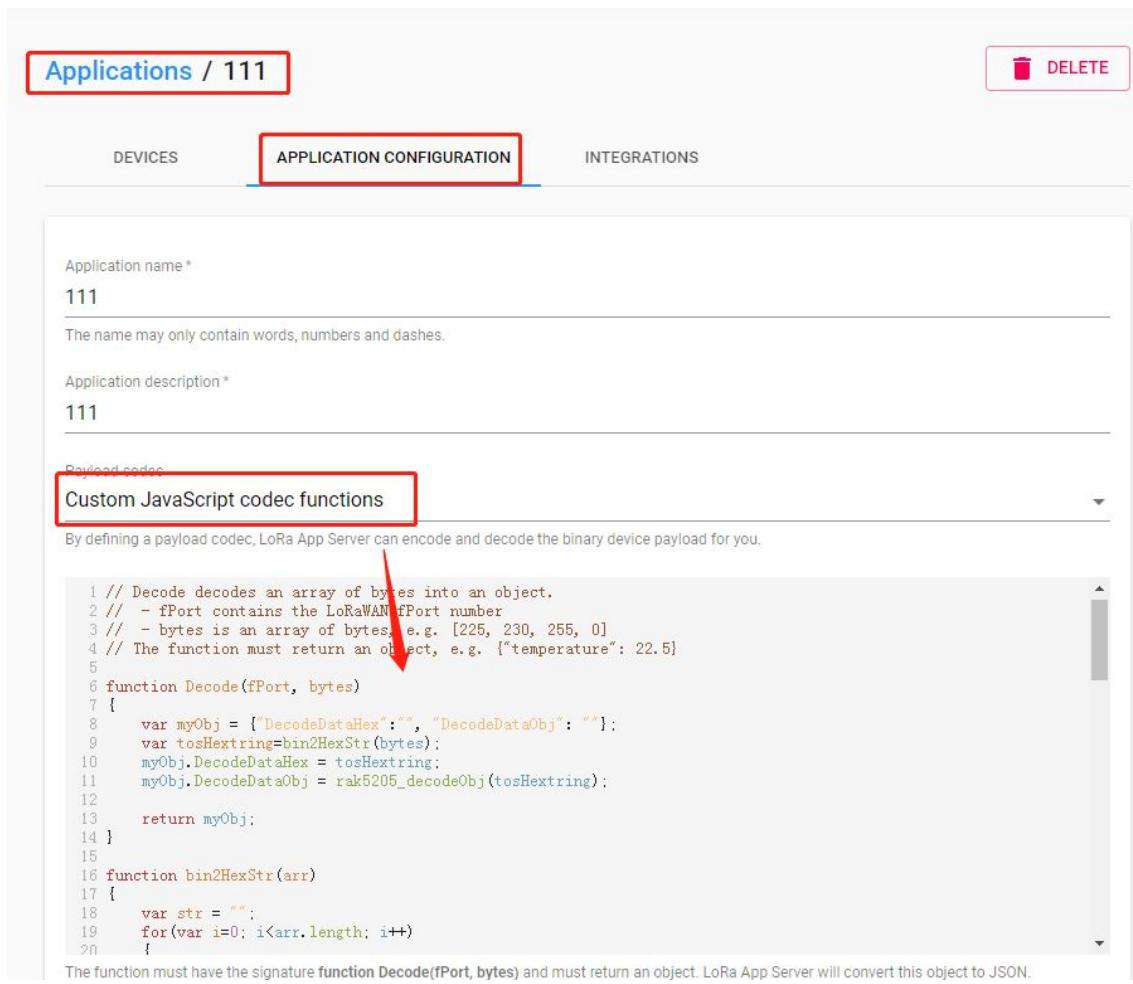
2. From your ChirpStack, go to application page and click the "APPLICATION CONFIGURATION" tab as shown in the image below.



The screenshot shows the ChirpStack application configuration interface. The top navigation bar includes tabs for 'DEVICES', 'APPLICATION CONFIGURATION' (which is highlighted with a red box), and 'INTEGRATIONS'. The main content area displays the text 'Applications / 111'.

Figure 80: Application Configuration Tab

3. Next, select "Payload codec" as "Custom JavaScript codec functions". Then, from the decoder tab, copy and paste the decoder function from step 1.



Applications / 111

**APPLICATION CONFIGURATION**

DEVICES INTEGRATIONS

Application name\*  
111  
The name may only contain words, numbers and dashes.

Application description\*  
111

Payload codec  
**Custom JavaScript codec functions**

By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you.

```

1 // Decode decodes an array of bytes into an object.
2 // - fPort contains the LoRaWAN fPort number
3 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
4 // The function must return an object, e.g. {"temperature": 22.5}
5
6 function Decode(fPort, bytes)
7 {
8     var myObj = {"DecodeDataHex": "", "DecodeDataObj": ""};
9     var tosHexString=bin2HexStr(bytes);
10    myObj.DecodeDataHex = tosHexString;
11    myObj.DecodeDataObj = rak5205_decodeObj(tosHexString);
12
13    return myObj;
14 }
15
16 function bin2HexStr(arr)
17 {
18     var str = "";
19     for(var i=0; i<arr.length; i++)
20     {

```

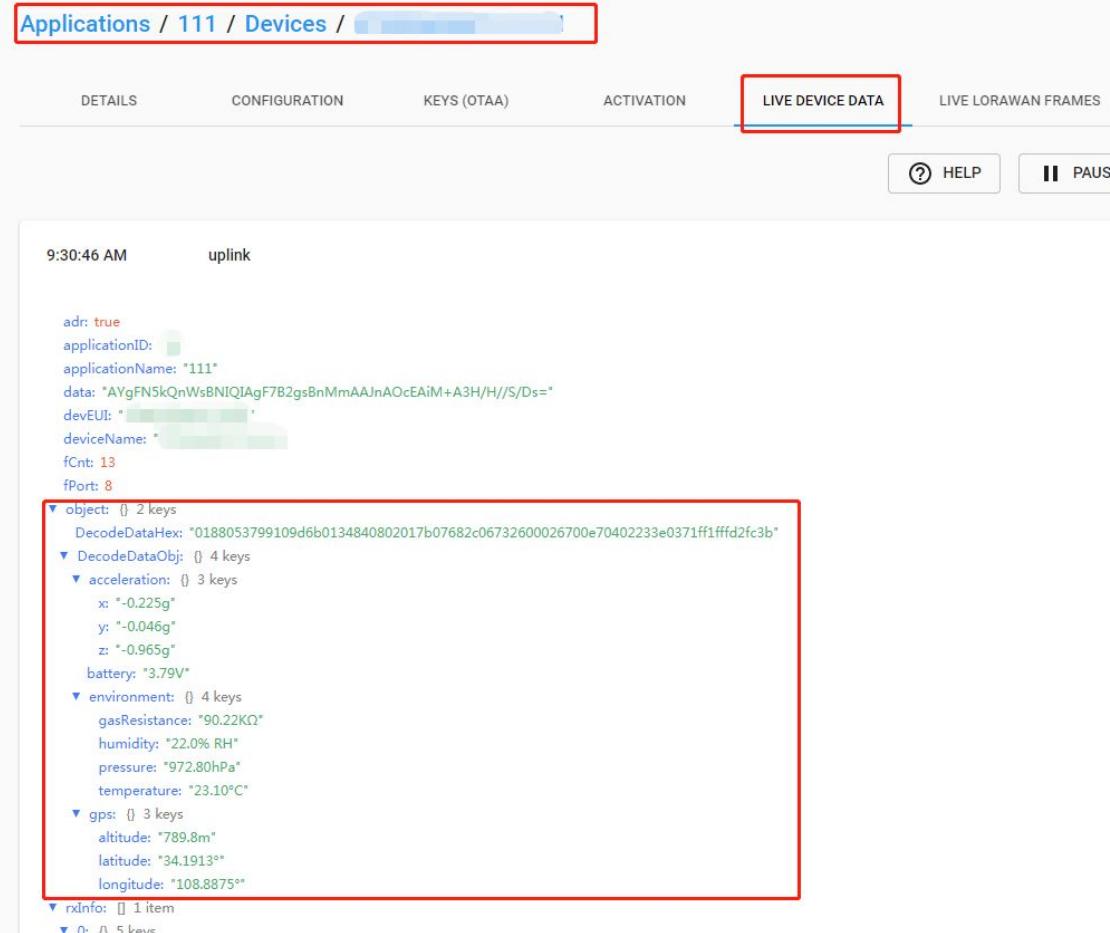
The function must have the signature function Decode(fPort, bytes) and must return an object. LoRa App Server will convert this object to JSON.

Figure 81: Decoded Function in Chirpstack

4. Then, click '**UPDATE APPLICATION**' button to save decoding function.

## Testing in Real System in ChirpStack

After gateway and node go online, click the uplink data record from the application data at "**LIVE DEVICE DATA**" tab to check the decode status. From the image below, we can see the data decoded successfully in ChirpStack.



Applications / 111 / Devices / [redacted]

DETAILS CONFIGURATION KEYS (OTAA) ACTIVATION **LIVE DEVICE DATA** LIVE LORAWAN FRAMES

9:30:46 AM uplink

```

        adr: true
        applicationID: [redacted]
        applicationName: "111"
        data: "AYgFN5kQnWsBNIQIAgF7B2gsBnMmAAJnAOcEAiM+A3H/H//S/Ds="
        devEUI: "[redacted]"
        deviceName: ""
        fCnt: 13
        fPort: 8
        object: {} 2 keys
            DecodeDataHex: "0188053799109d6b0134840802017b07682c06732600026700e70402233e0371ff1ffffd2fc3b"
            DecodeDataObj: {} 4 keys
                acceleration: {} 3 keys
                    x: "-0.225g"
                    y: "-0.046g"
                    z: "-0.965g"
                    battery: "3.79V"
                environment: {} 4 keys
                    gasResistance: "90.22KΩ"
                    humidity: "22.0% RH"
                    pressure: "972.80hPa"
                    temperature: "23.10°C"
                gps: {} 3 keys
                    altitude: "789.8m"
                    latitude: "34.1913°"
                    longitude: "108.8875°"
            rxInfo: {} 1 item
                0: {} 5 keys

```

Figure 82: Decode Status in ChirpStack

## Lora® P2P Mode

The setup process for the RAK7204 WisNode Sense Home in LoRaP2P Mode is just the same with the process with the RAK811 Wisnode. These are the steps that you need to follow for this mode:

1. First, find two RAK7204 WisNode Sense Home which can work on EU868 frequency and make sure their firmware version isn't less than V3.0.0.1.
2. Next, connect these two RAK7204 WisNode Sense Home with PC through USB cable, and open two serial port tool on PC.
3. Now, configure them to both work in LoRaP2P mode as follow:

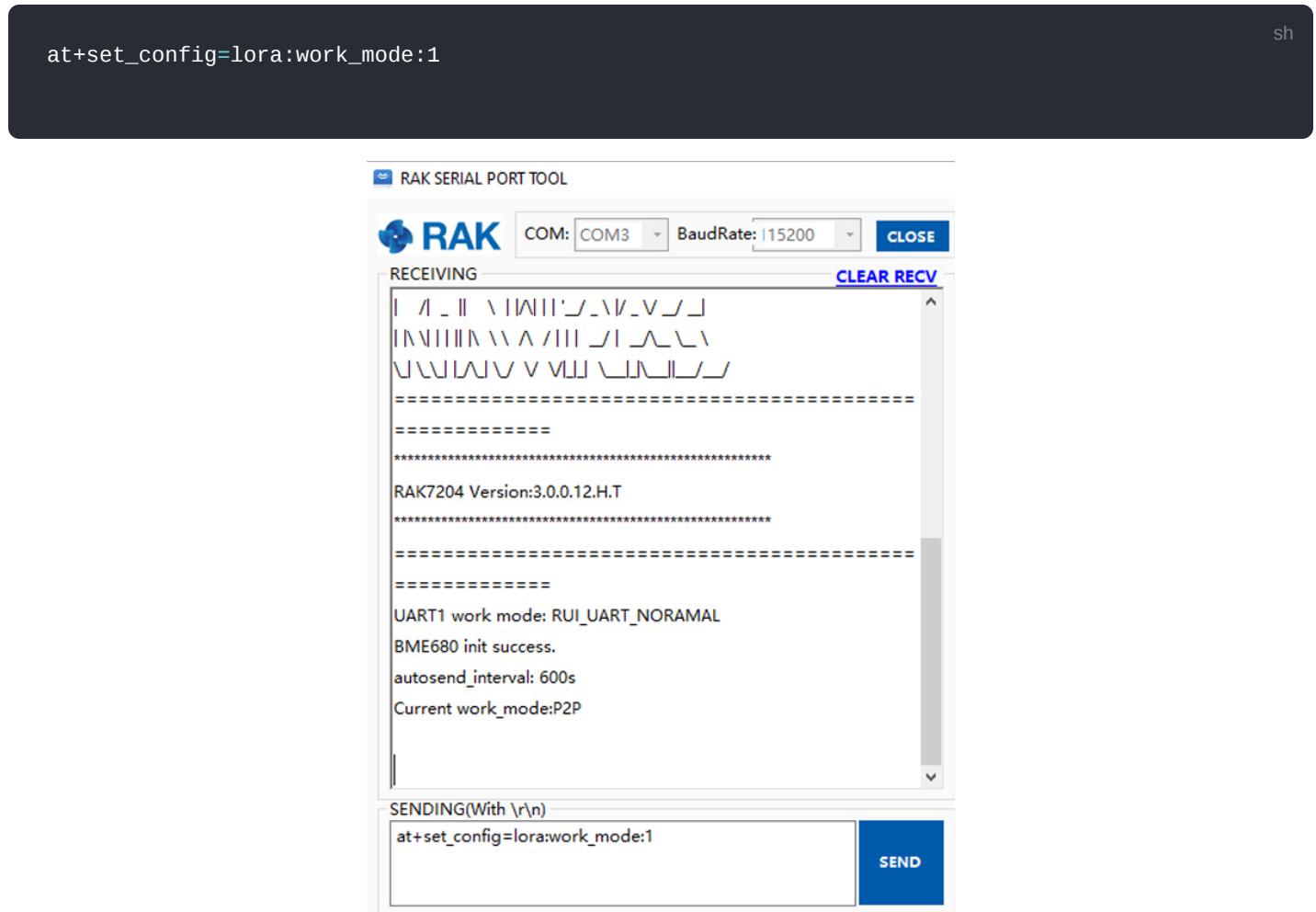
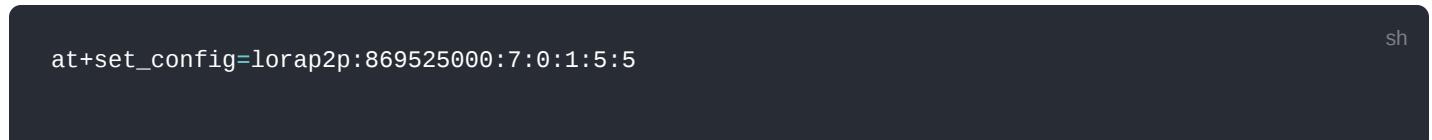


Figure 83: LoRaP2P Mode

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



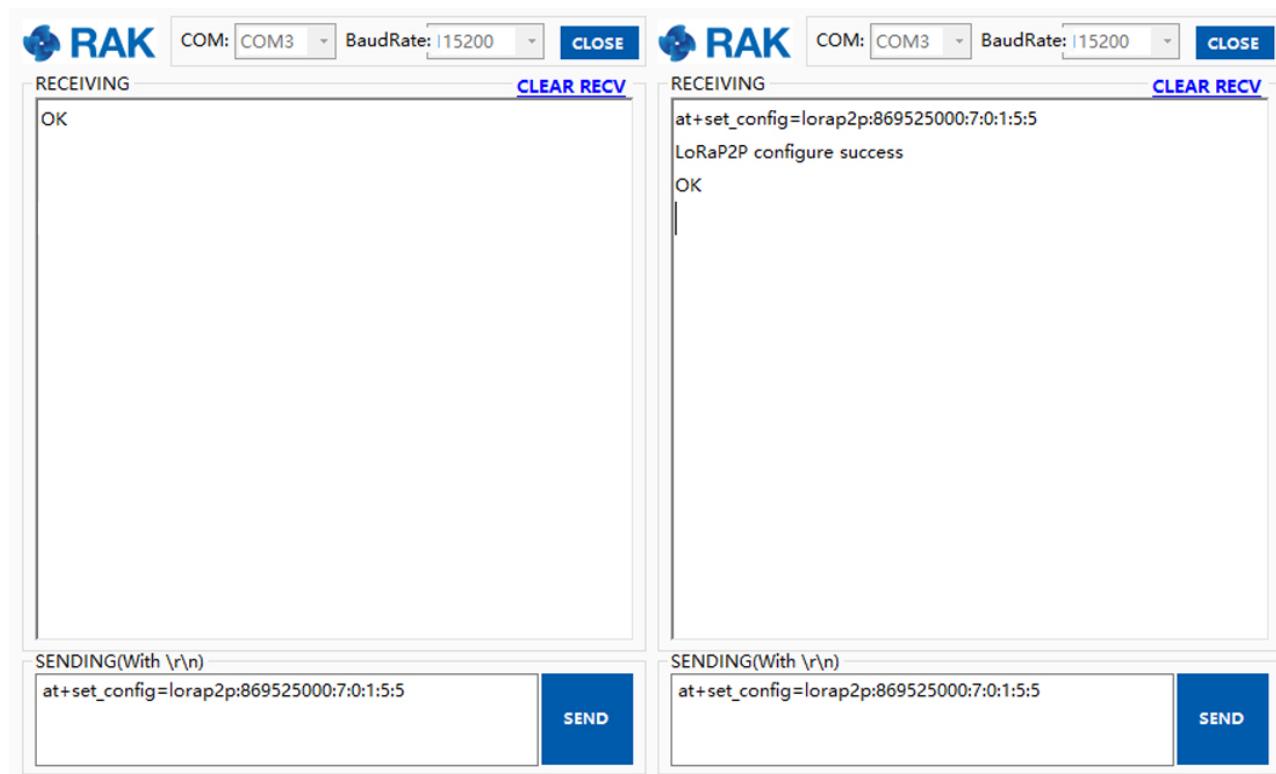


Figure 84: LoRaP2P Configuration

OK! Try to send a message from RAK7204 WisNode Sense Home - 2 (the right one) to RAK7204 WisNode Sense Home - 1 (the left one):

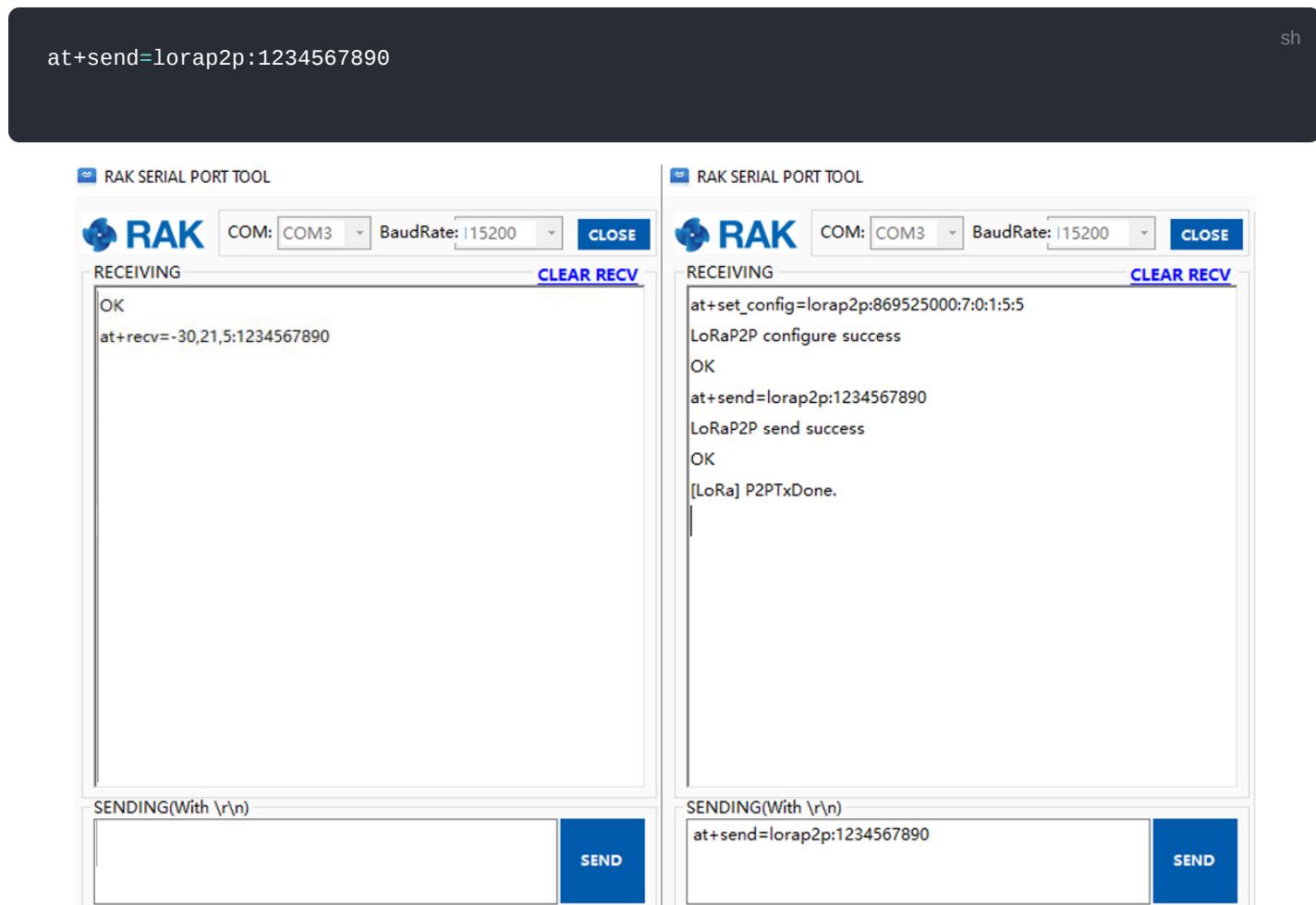


Figure 85: Test Message Sent

**Success!** You can send more messages:

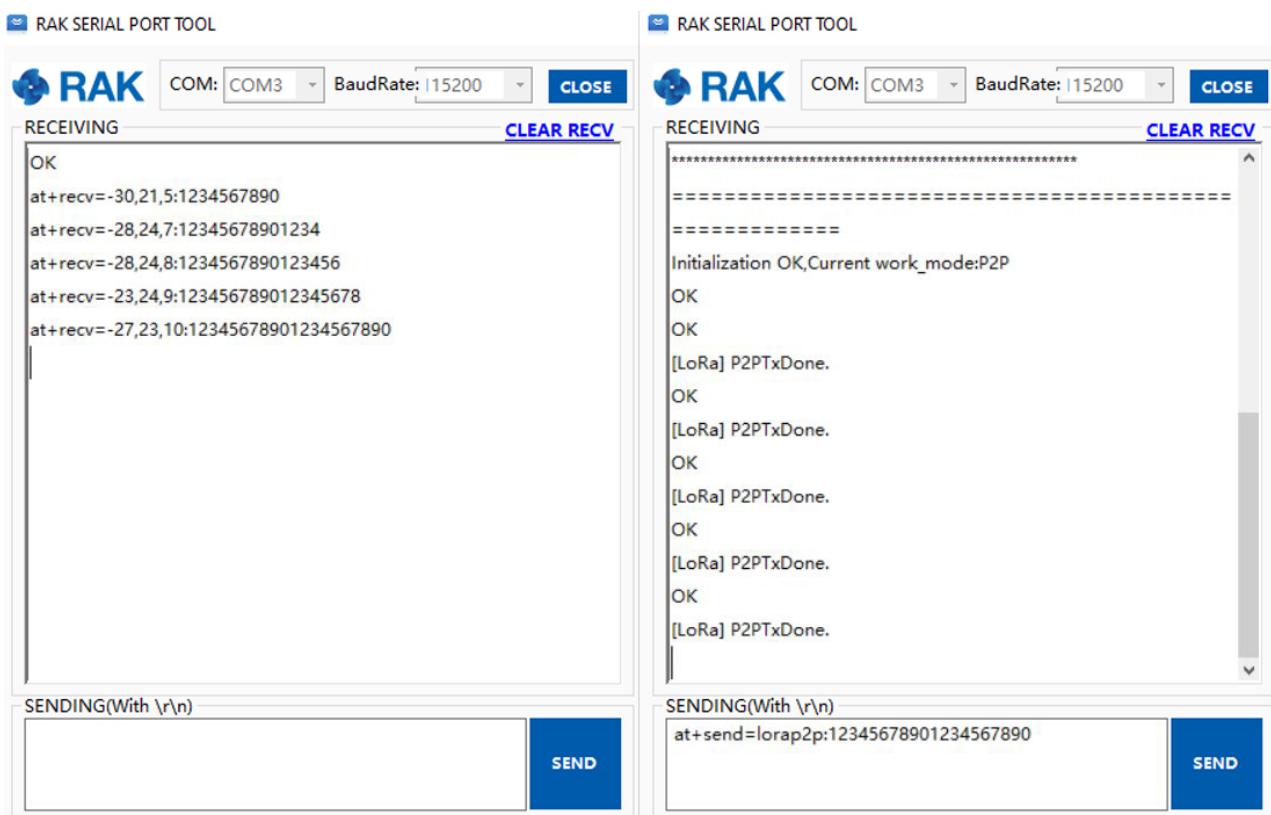


Figure 86: Test Message Sent

Similarly, you can send message from RAK7204 WisNode Sense Home-1 to RAK7204 WisNode Sense Home-2 surely. Just try it freely. Great! We've done it, and that's all about how to use LoRaP2P on RAK811 WisNode.

You can use RAK7204 WisNode Sense Home LoRaP2P mode according to it.

## ADR and DR

You can open the ADR feature of RAK7204 WisNode Sense Home by using the following AT command:

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

or you can close the ADR feature of RAK7204 WisNode Sense Home by using this AT command:

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

There is also an AT command which is used to set the DataRate(DR):

AT Command	Description
at+set_config=lora:dr:X	<p>Set the DR of LoRa Node.</p> <ul style="list-style-type: none"> <li>• X : the number of DR. Generally, the value of X can be 0~5. More details, please check the LoRaWAN 1.0.2 specification.</li> </ul>

For example, if you want to set the current DR to DR0, you just do as follow:

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

## Miscellaneous

# Burning the Bootloader

 **NOTE:**

Usually you don't need to burn the bootloader since there is a bootloader already in RAK7204 WisNode Sense Home from V3.0.0.0 firmware and so on. If the firmware of your RAK7204 WisNode Sense Home is V3.0.0.0 or a newer one, **Skip this section.**

You can burn the bootloader in your RAK7204 WisNode Sense Home by following the steps below:

1. Download and Install the [STM32CubeProgrammer](#) Software from STMicroelectronics on your Windows PC.

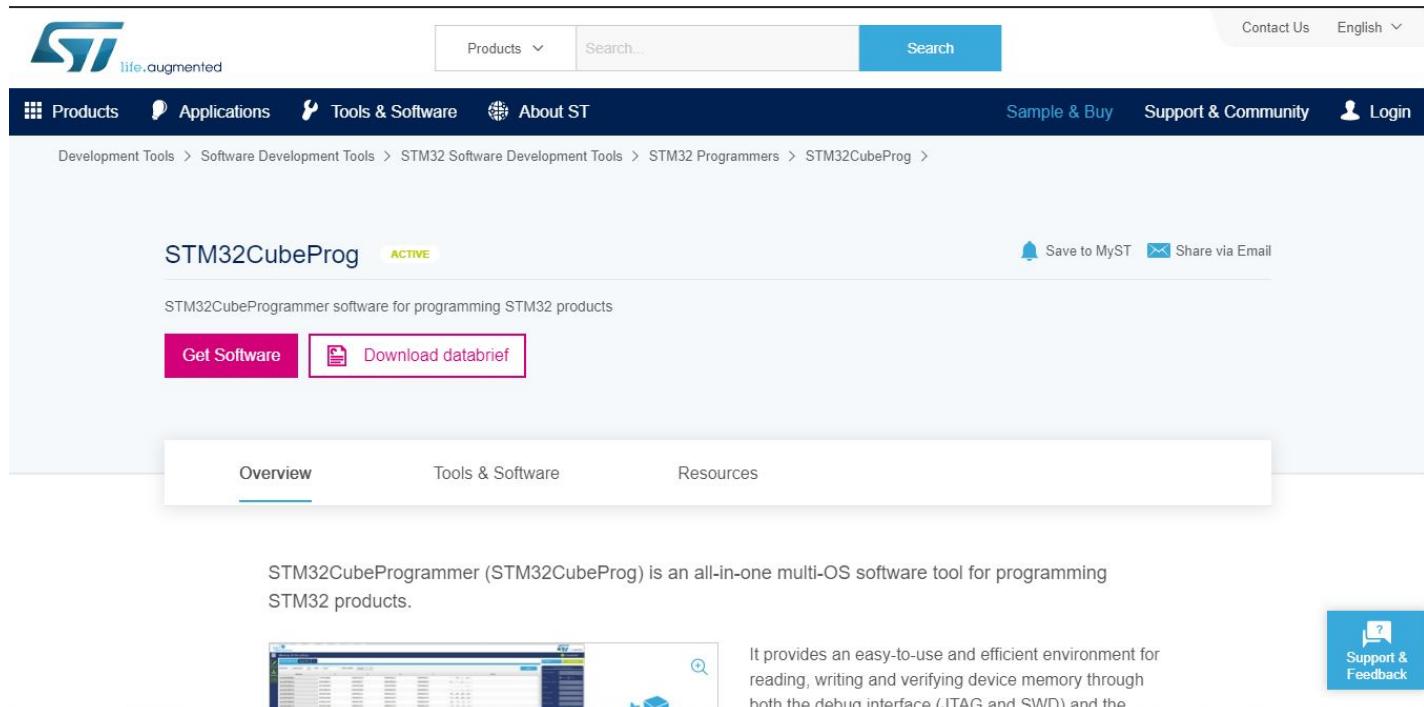


Figure 87: STM32CubeProg Download Page

2. Insert the provided jumper on the Boot line pins ("BOOT" pin and "VDD"), shorting them . Also, make sure that the RX pin of J25 is connected to the RXCP pin.



Figure 88: Boot Line shorted using the Jumper Pins

3. Connect the RAK7204 WisNode Sense Home on your Windows PC's USB Interface and press the RST Button or power it on again. Open the STM32CubeProgrammer Software and Select UART type.

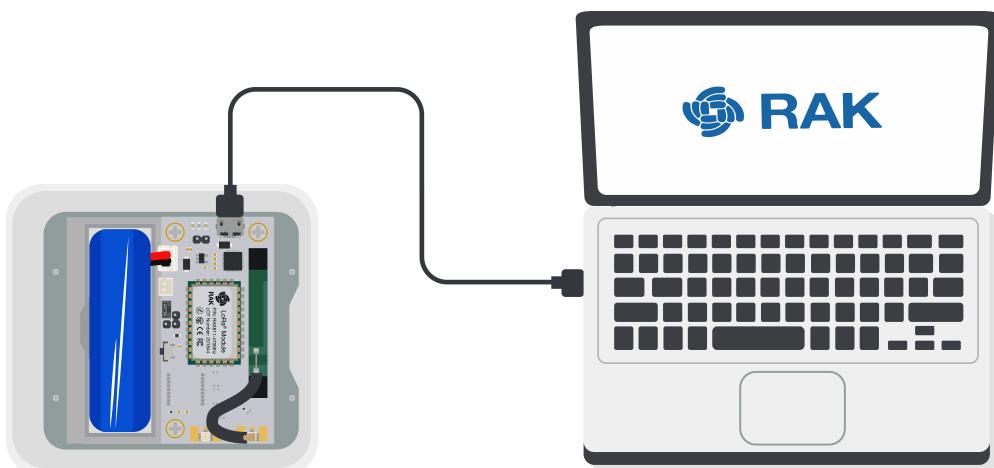


Figure 89: USB Interface

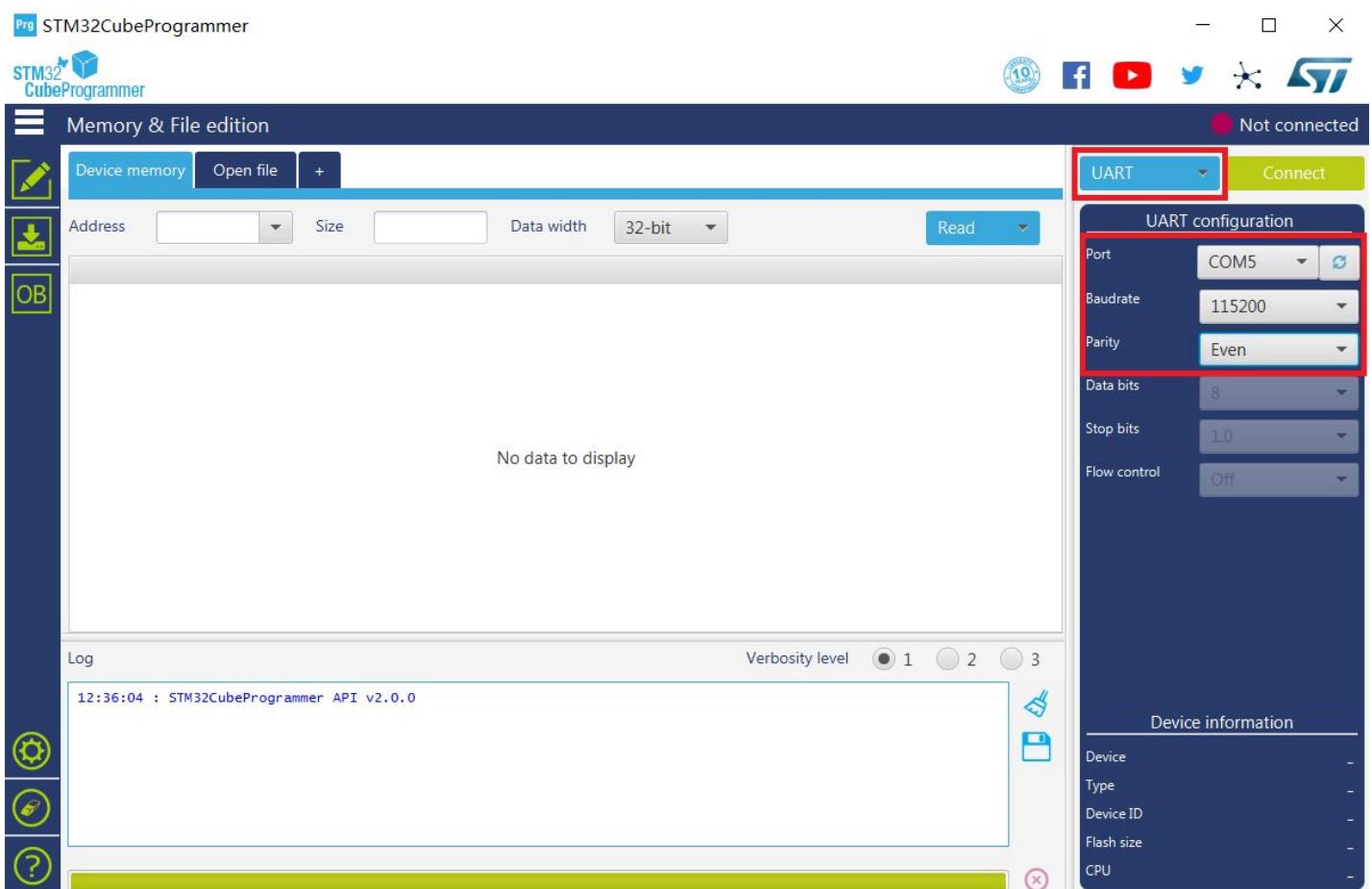


Figure 90: UART Settings in STM32CubeProgrammer

4. Choose the appropriate port number in the **COM Port** field.
  5. Set the Baud Rate to 115200, and the Parity to Even as seen in the figure above then Press **Connect**.
- If you didn't properly set your RAK7204 WisNode Sense Home to work in BOOT Mode, you will see the following information in the Log Section of the Software:

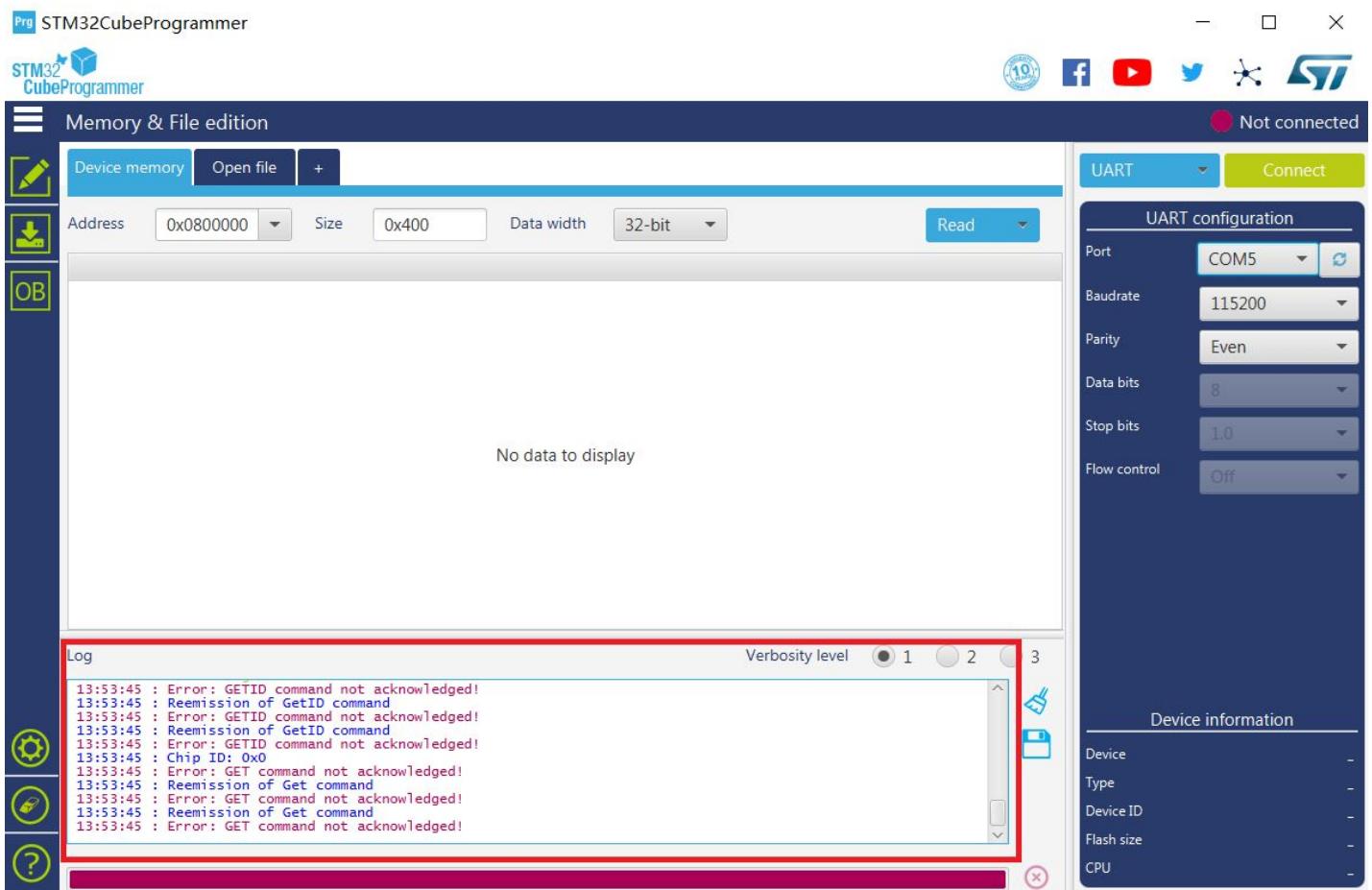


Figure 91: Errors Occurred During Connecting

- If this happens, Close the STM32CubeProgrammer and go back to the section above and set your RAK7204 WisNode Sense Home to work in **Boot Mode** again.
- If all works well, You will then see the following log:

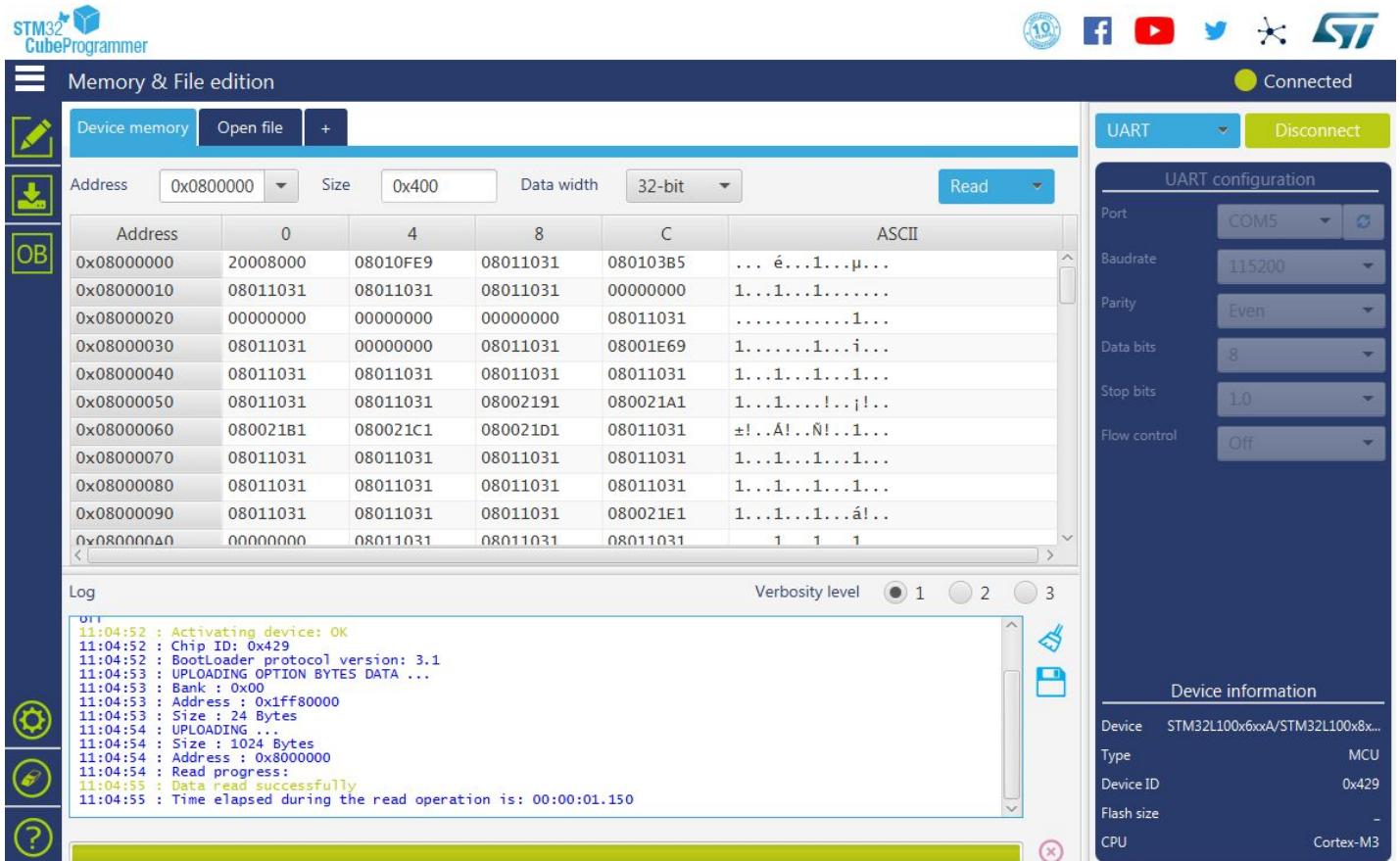


Figure 92: Successful Connection Log to your Device

Now that you have successfully connected your RAK7204 WisNode Sense Home to the STM32CubeProgrammer Tool, let's burn the Bootloader into the RAK7204 WisNode Sense Home.

6. Download the Bootloader for the RAK7204 WisNode Sense Home [here](#).

7. In the STM32CubeProgrammer, Click the "Erase Chip" button to erase all the data on RAK7204 WisNode Sense Home:

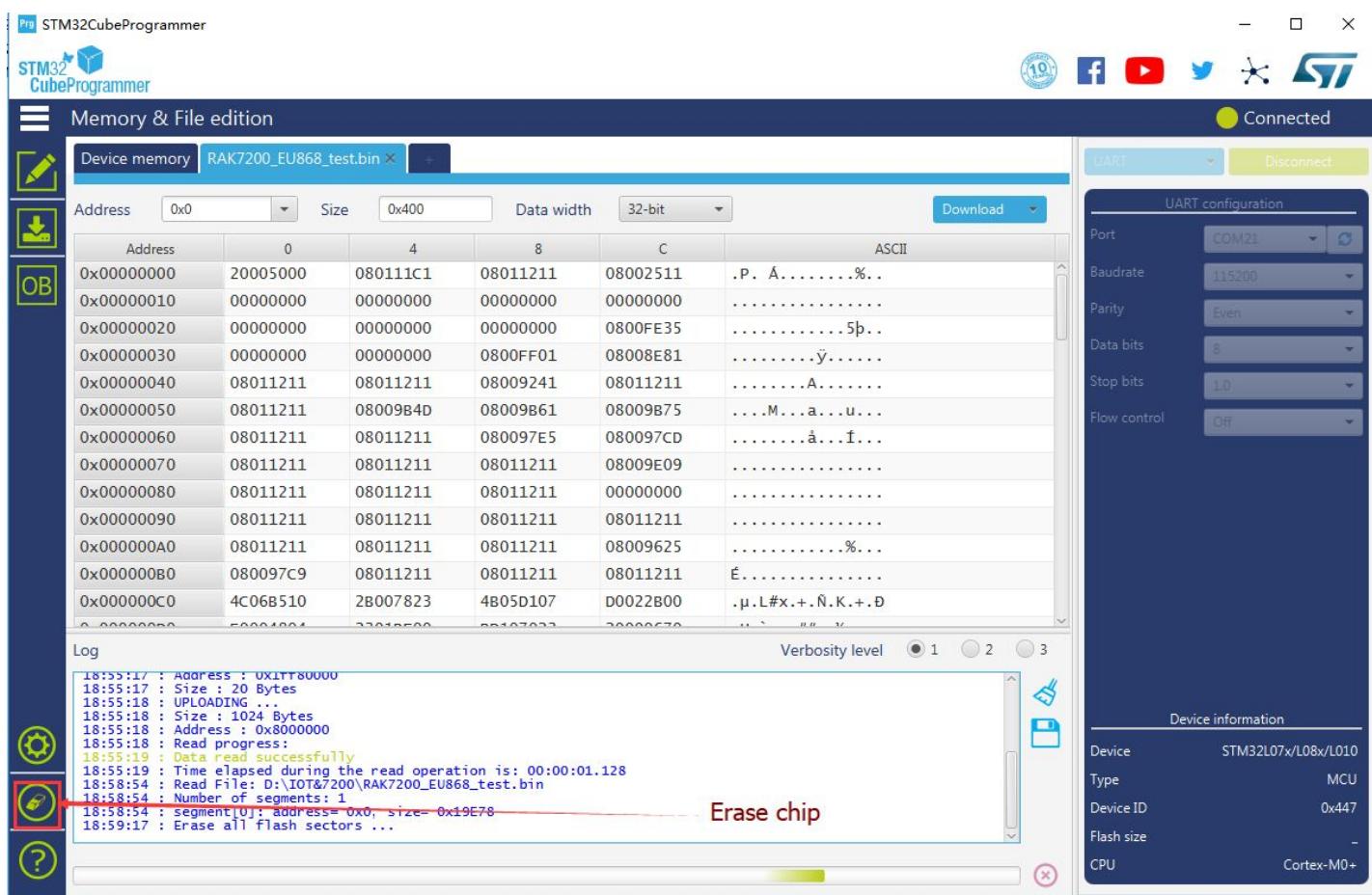


Figure 93: Erasing the Data in the Chip

8. Click "Open File" and select the correct Bootloader file that you have just downloaded.

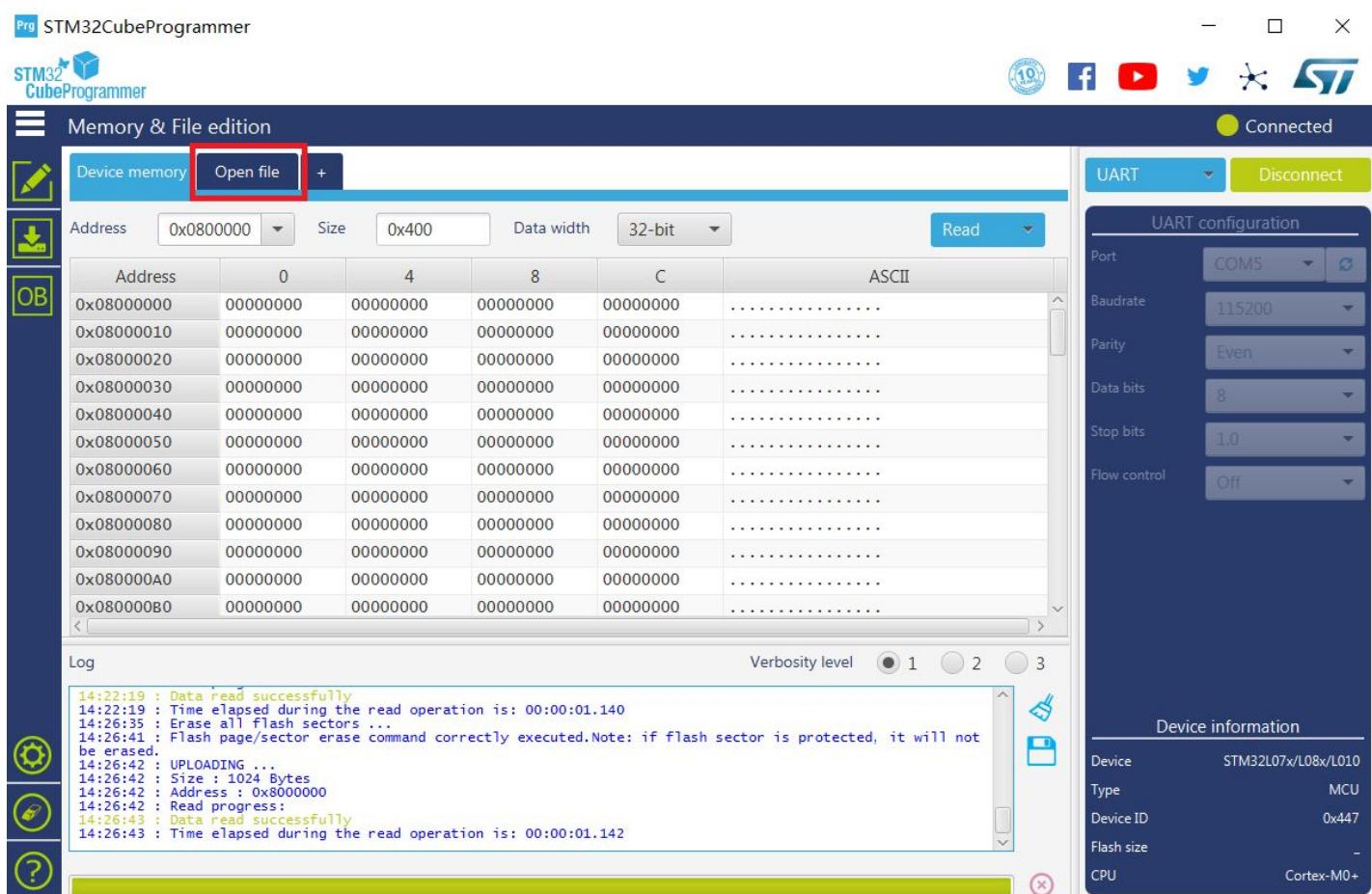


Figure 94: Opening the Bootloader file

9. Click the "Download" Button to start the burning process:

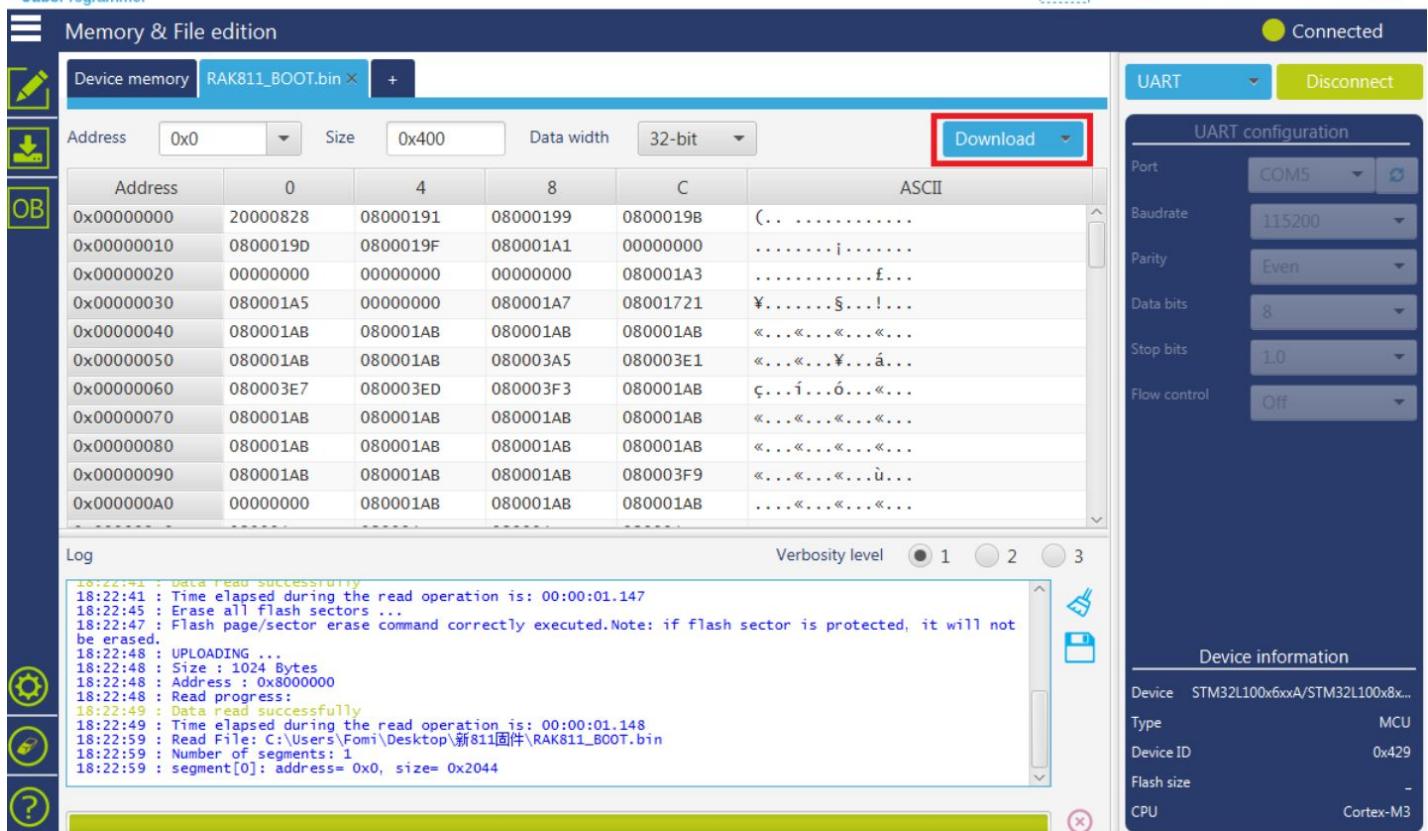


Figure 95: Downloading of Bootloader to the device

- After a couple of seconds, you will see the following window telling that you have successfully burned the Bootloader to your RAK7204 WisNode Sense Home!

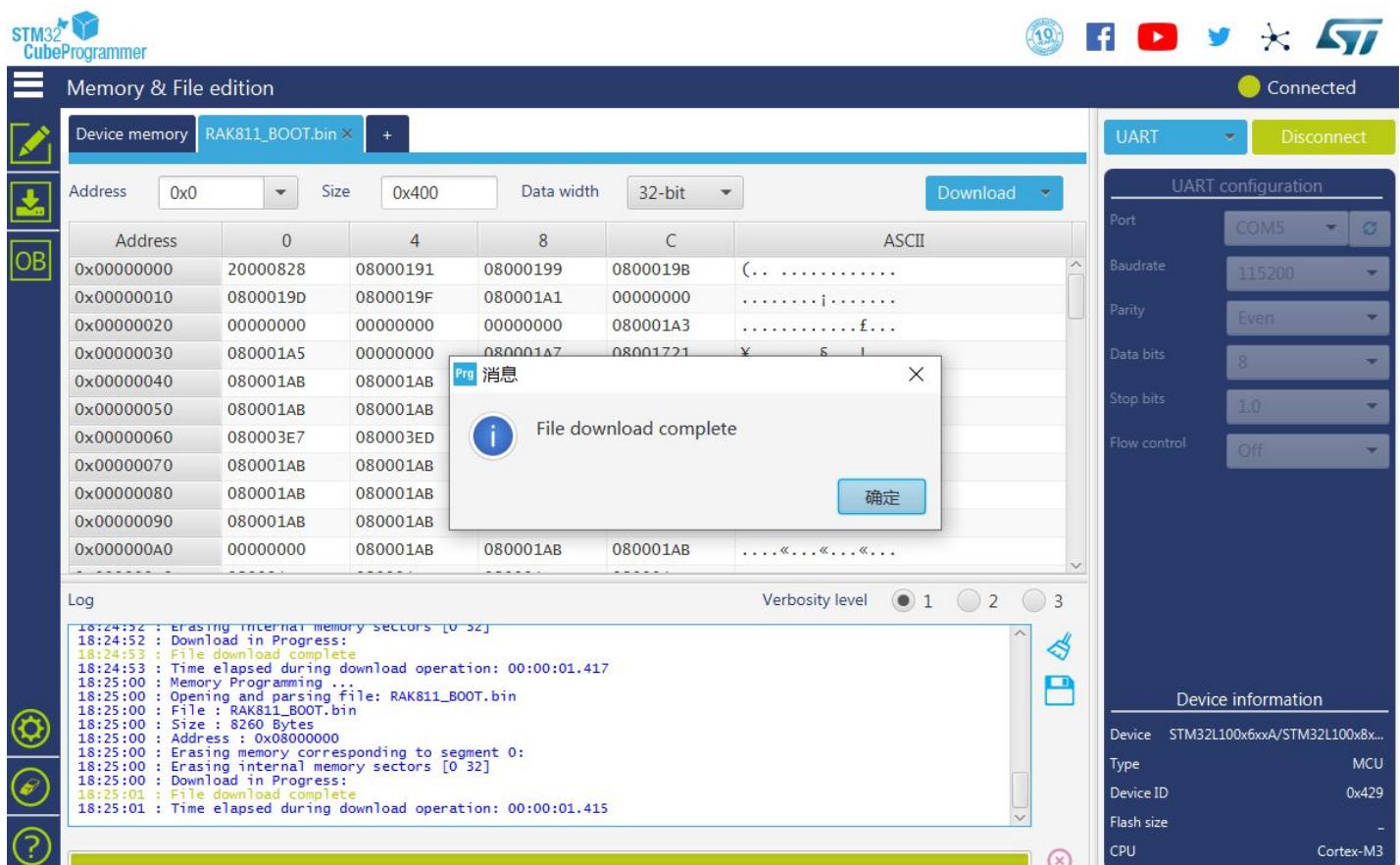


Figure 96: Successfully Burned the Bootloader to the device

- “Disconnect” and close the “STM32CubeProgrammer” tool.

### WARNING

Disconnect your RAK7204 WisNode Sense Home in your Windows PC and do not forget to remove the Jumper on the Boot Line Pins to work in Normal Mode.

# Burning the Firmware

If the firmware version of your **RAK7204 WisNode Sense Home** is newer than V3.0.0.0 or you have just burned the bootloader into the board according to the **Burning the Bootloader** section, follow the steps below

- Make sure you have set your RAK7204 WisNode Sense Home to work in boot mode. If you have just burned the bootloader according to the previous section, it works in boot mode now.
- Open and download the RAK Serial Port Tool [Here](#) and Connect your board via the USB interface and enter the following **AT command** to let it work in boot mode.

## ⚠️ WARNING

Before configuring your RAK7204 WisNode Sense Home, make sure you already connected the Battery provided on your device in order for you to communicate with the device successfully.

```
at+set_config=device:boot
```

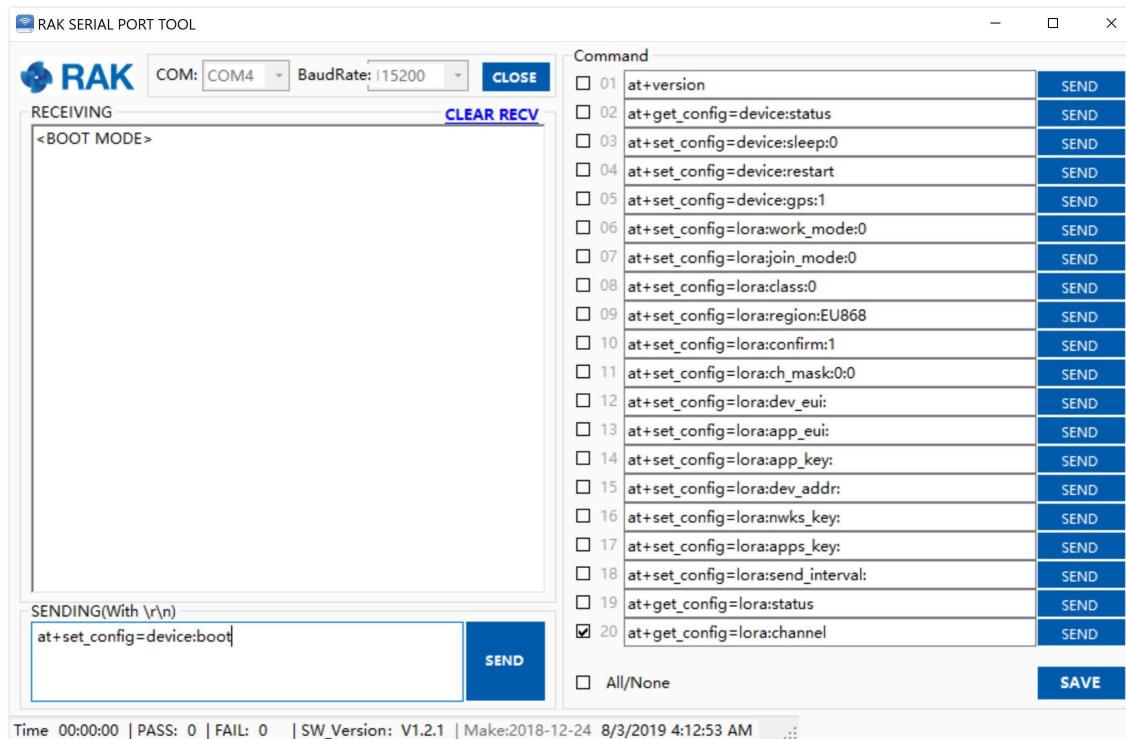


Figure 97: Entering Boot Mode

- Download the [RAK Device Firmware Upgrade \(DFU\) Tool](#) from the RAKwireless website.

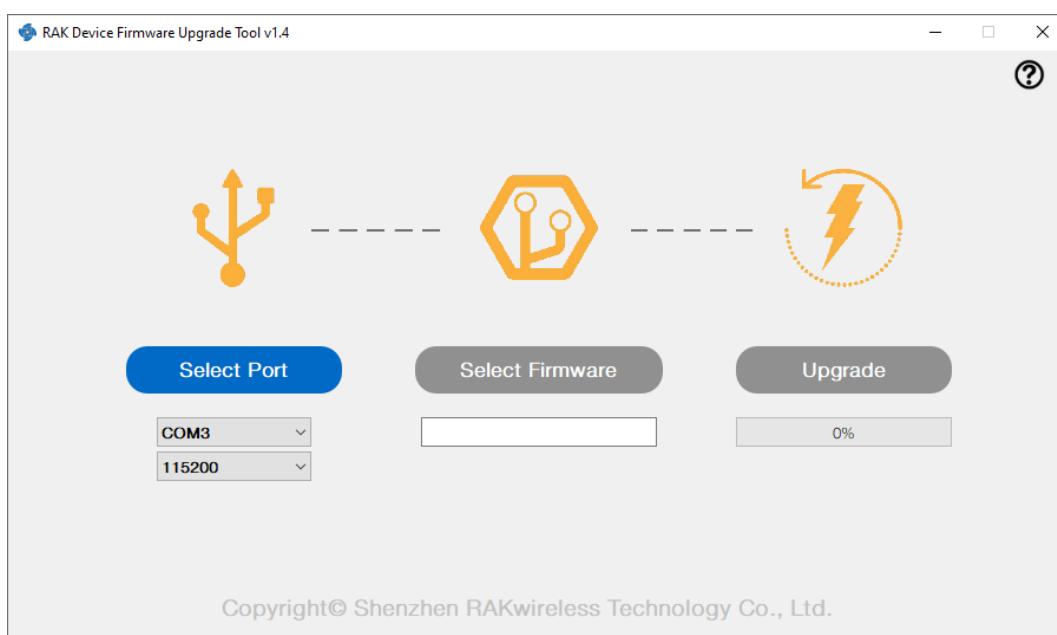


Figure 98: Device Firmware Upgrade Tool

- Download the latest firmware [here](#) for the RAK7204 WisNode Sense Home

 **NOTE:**

Make sure to pick the appropriate bin file depending on the region you are in.

- "RUI\_RAK7204\_V3.x.x.x.H" supported regions are: IN865, EU868, US915, AU915, KR920, AS923
- "RUI\_RAK7204\_V3.x.x.x.L" supported regions are: EU433, CN470 Visit this [article](#) for more information on your local TTN frequency plan.

- Select the application firmware file of the module with the suffix ".bin".

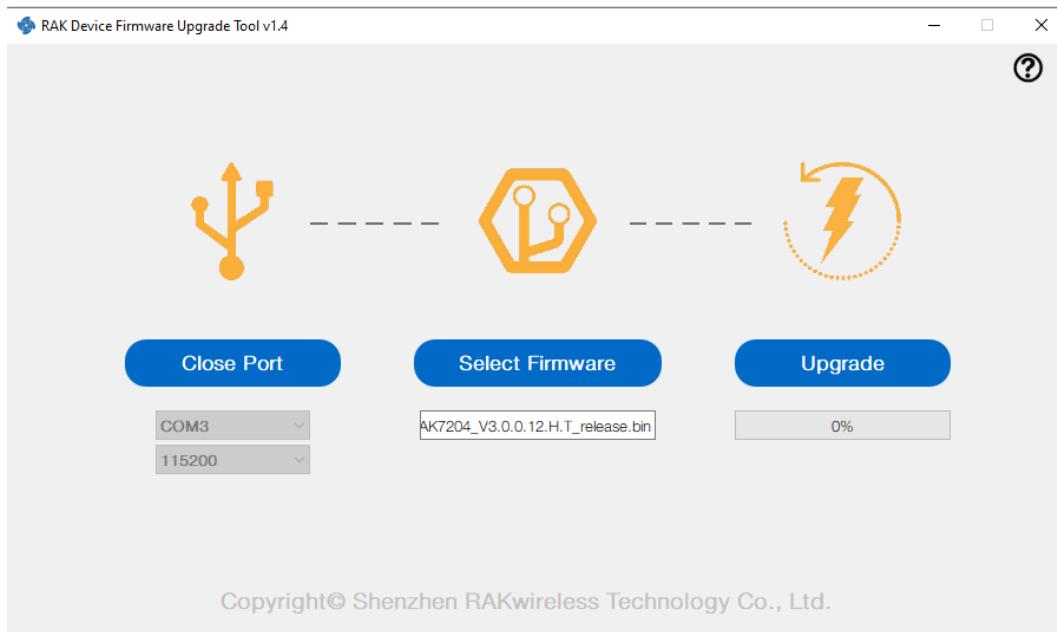
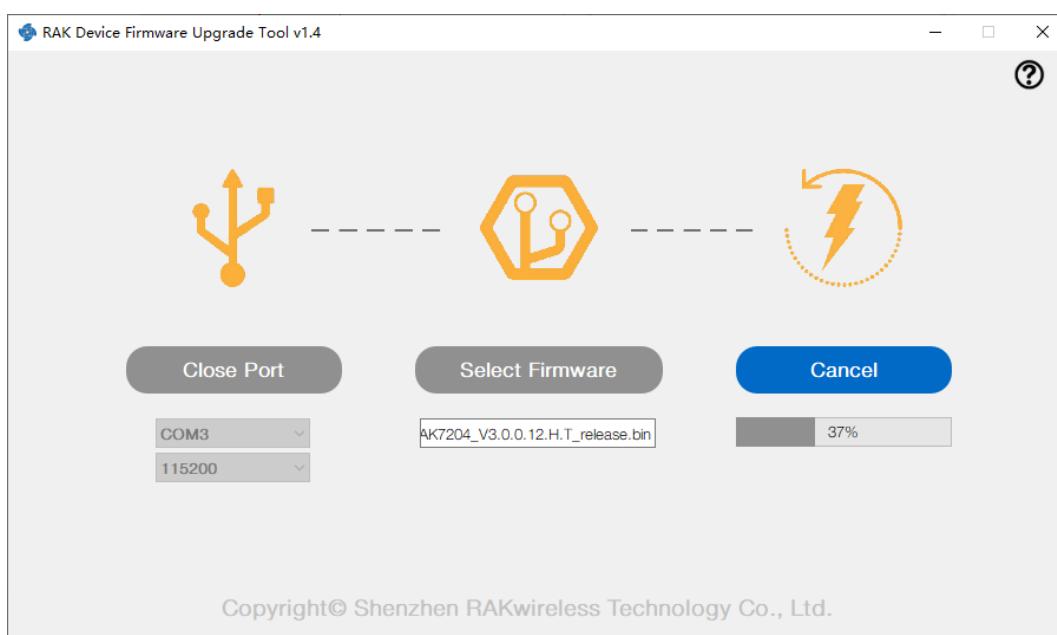
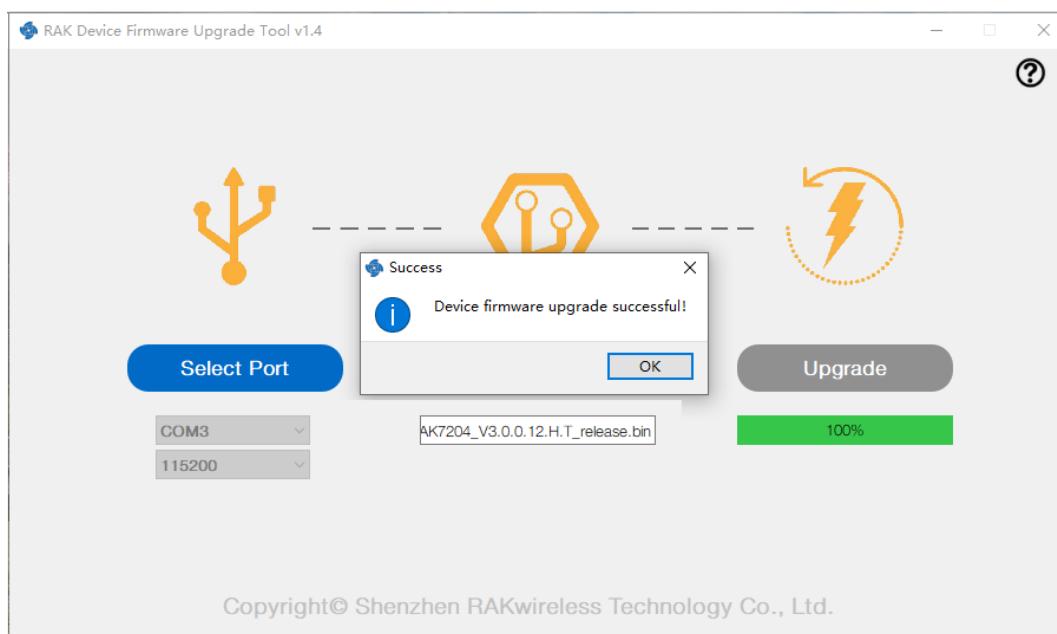


Figure 99: Select firmware

- Click the "upgrade" button to upgrade the device. After the upgrade is complete, the RAK7204 is now ready to work with the new firmware.



**Figure 100:** Firmware upgrading



**Figure 101:** Upgrade successful

- Now, **CLOSE** the upgrade tool and proceed to the next section.

## Testing the Installed Firmware

In order for you to check if you have successfully installed the firmware on your RAK7204 WisNode Sense Home, open the Serial Port tool again. Press the "Reset button" or type the command below. If everything works perfectly, you should see the following message below:

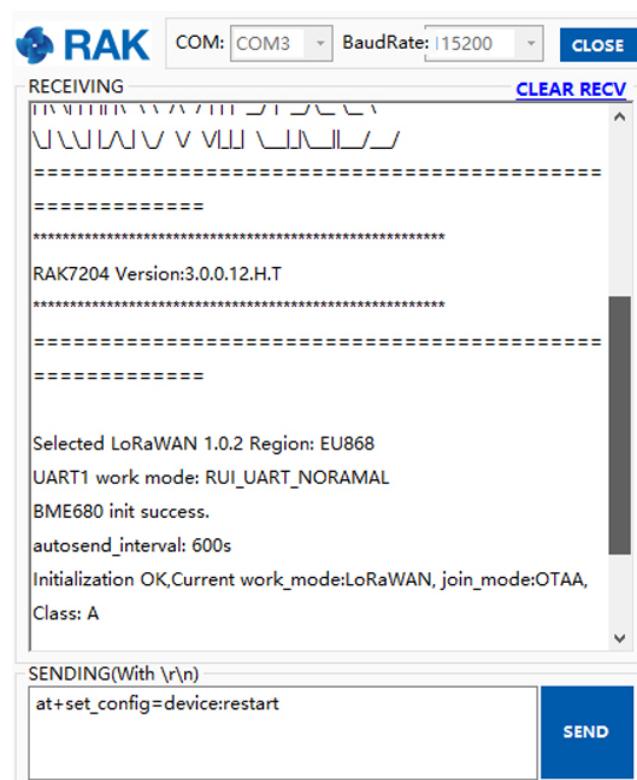


Figure 102: Restarting Your Device

This information means that you have uploaded the Firmware successfully!

Last Updated: 4/15/2022, 8:50:31 AM