

RAK4630 Module Quick Start Guide

This guide covers the following topics:

- [RAK4630 as a stand-alone device using RUI3](#)
- [RAK4630 as a LoRa/LoRaWAN modem via AT command](#)

Prerequisites

What Do You Need?

Before going through the steps in the installation guide of the RAK4630 WisDuo LPWAN Module, make sure to prepare the necessary items listed below:

Hardware

- [RAK4630 WisDuo LPWAN Module](#) 
- Computer
- USB to UART TTL adapter

Software

- Download and install the [Arduino IDE](#)  .
- Add [RAK4630 as a supported board in Arduino IDE](#) by updating Board Manager URLs in **Preferences** settings of Arduino IDE with this JSON URL `https://raw.githubusercontent.com/RAKwireless/RAKwireless-Arduino-BSP-Index/main/package_rakwireless.com_rui_index.json` . After that, you can then add **RAKwireless RUI nRF Boards** via Arduino board manager.
- [RAK Serial Port Tool](#) 

Product Configuration

Rak4630 as a Stand-Alone Device Using RUI3

Hardware Setup

The RAK4630 requires a few hardware connections before you can make it work. The bare minimum requirement is to have the power section properly configured, reset button, antenna, and USB connection.

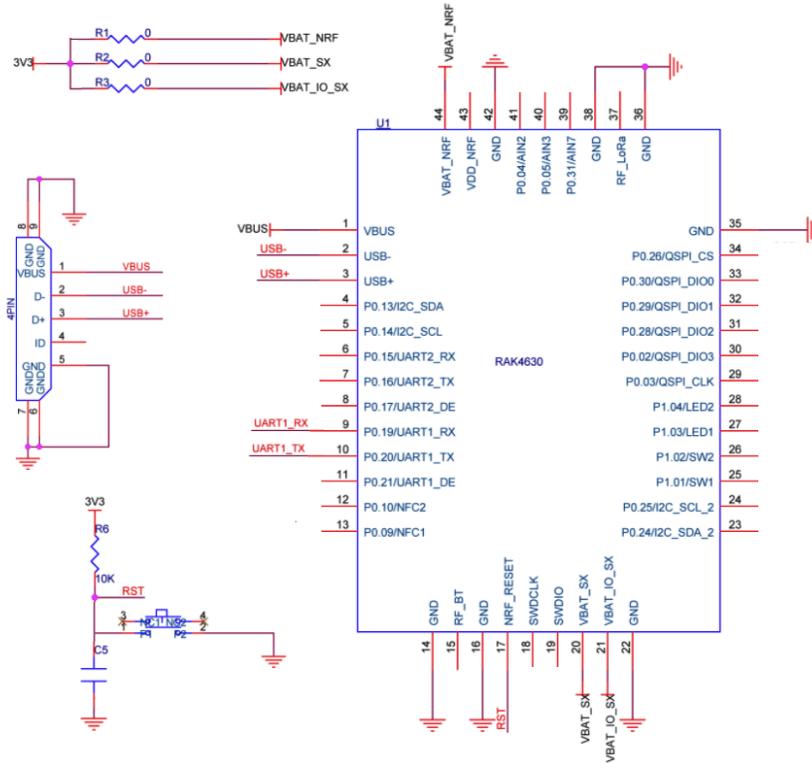


Figure 1: RAK4630 minimum schematic

Ensure that the antennas are properly connected to have a good LoRa and BLE signal. Also, note that you can damage the RF section of the chip if you power the module without an antenna connected to the IPEX connectors.



Figure 2: LoRa antenna

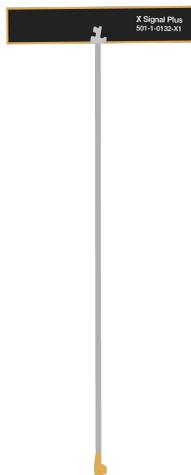


Figure 3: BLE antenna

RAK4630 has a label on its sticker on where to connect the antennas, as shown in **Figure 4**.



Figure 4: RAK4630 antenna label

NOTE

Detailed information about the RAK4630 BLE and LoRa antenna can be found on the [antenna datasheet](#)



WARNING

When using the LoRa or Bluetooth Low Energy transceivers, make sure that an antenna is always connected. Using these transceivers without an antenna can damage the module.

Software Setup

The default firmware of RAK4630 is based on RUI3, which allows you to develop your custom firmware to connect sensors and other peripherals to it. To develop your custom firmware using Arduino IDE, you need first to add **RAKwireless RUI nRF Boards** in the Arduino board manager, which will be discussed in this guide. You can then use RUI3 APIs for your intended application. You can upload the custom firmware via USB or DFU OTA. The AT commands of RAK4630 is still available even if you compile custom firmware via RUI3. You can send AT commands via a USB connection, UART1, or wirelessly via BLE connection.

RAK4630 RUI3 Board Support Package in Arduino IDE

If you don't have an Arduino IDE yet, you can download it on the [Arduino official website](#) and follow the installation procedure in the [miscellaneous section](#) of this document.

NOTE

For Windows 10 and up users:

If your Arduino IDE is installed from the Microsoft App Store, you need to reinstall your Arduino IDE by getting it from the Arduino official website. The Arduino app from the Microsoft App Store has problems using third-party Board Support Packages.

Once the Arduino IDE has been installed successfully, you can now configure the IDE to add the RAK4630 in its board selection by following these steps.

1. Open Arduino IDE and go to **File > Preferences**.

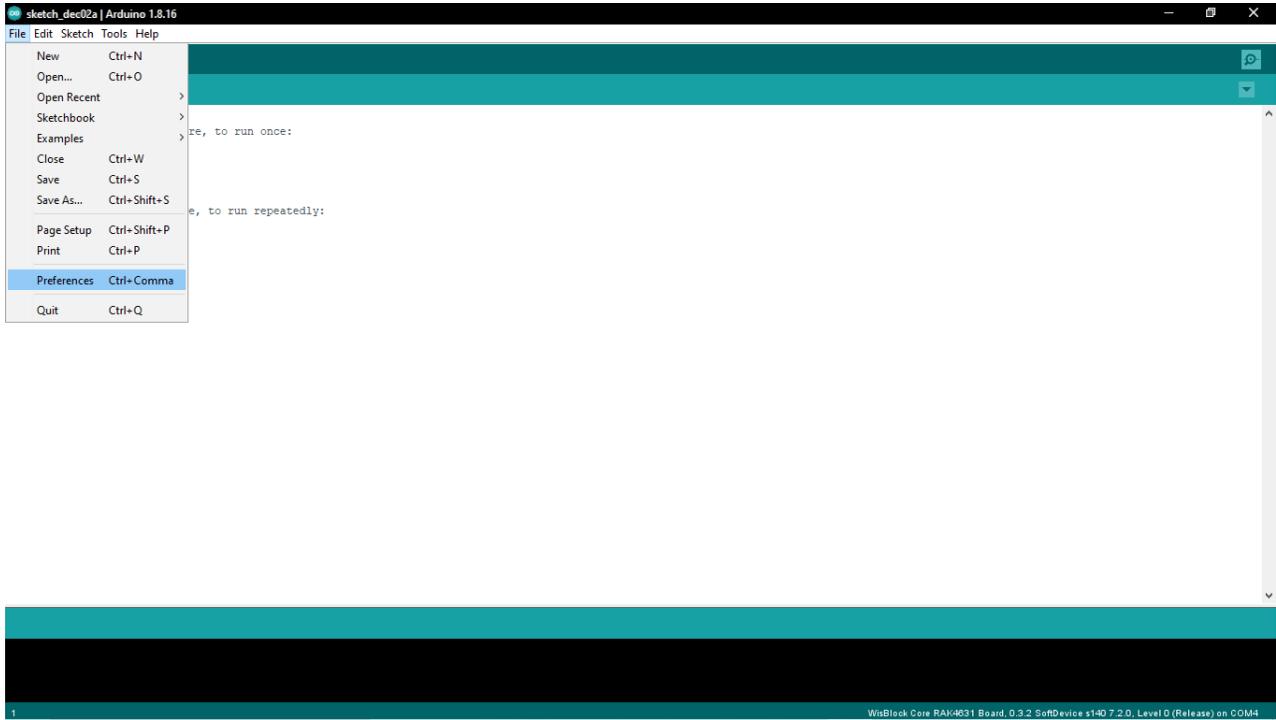


Figure 5: Arduino preferences

2. To add the RAK4630 to your Arduino Boards list, edit the **Additional Board Manager URLs**. Click the icon, as shown in **Figure 6**.

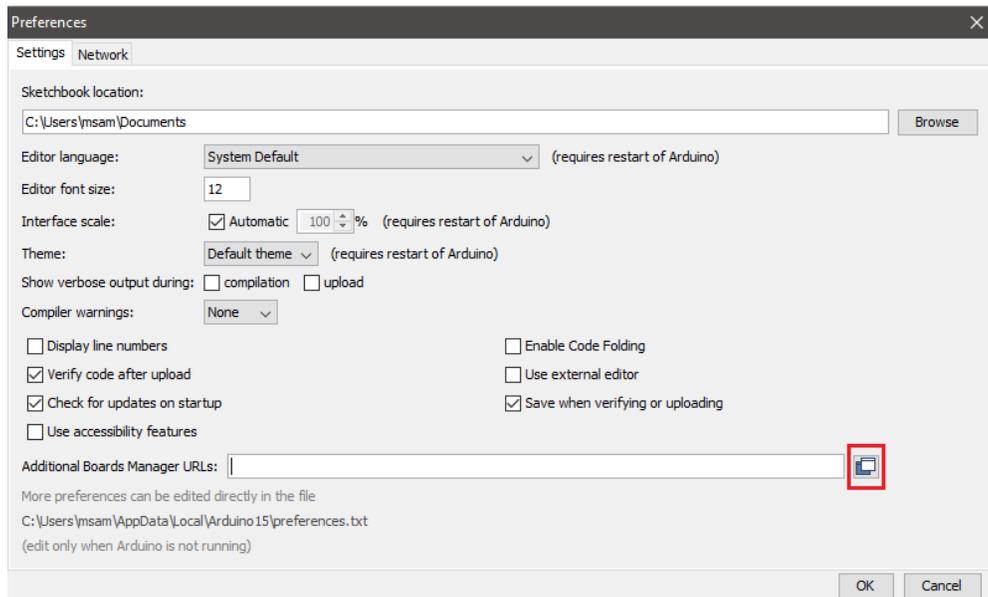


Figure 6: Modifying Additional Board Manager URLs

3. Copy the URL https://raw.githubusercontent.com/RAKwireless/RAKwireless-Arduino-BSP-Index/main/package_rakwireless.com_rui_index.json and paste it on the field, as shown in **Figure 7**. If there are other URLs already there, just add them on the next line. After adding the URL, click **OK**.

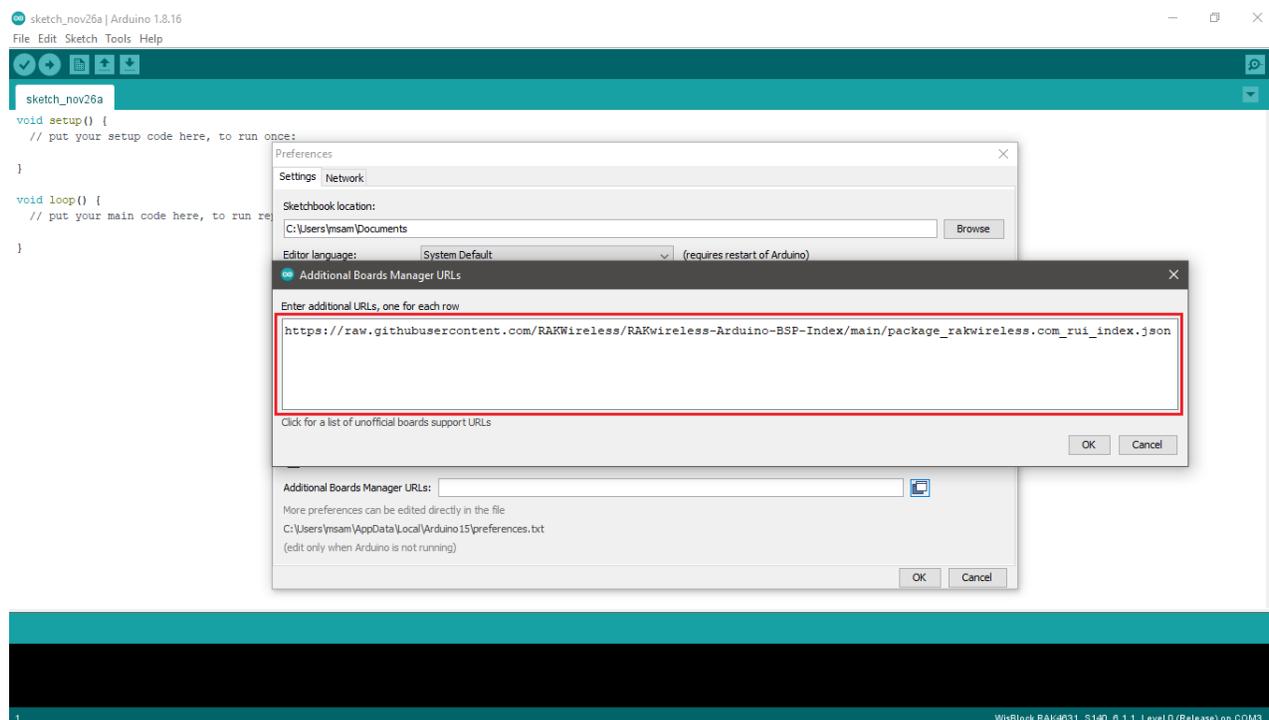


Figure 7: Add additional board manager URLs

4. Restart the Arduino IDE.

5. Open the Boards Manager from Tools Menu.

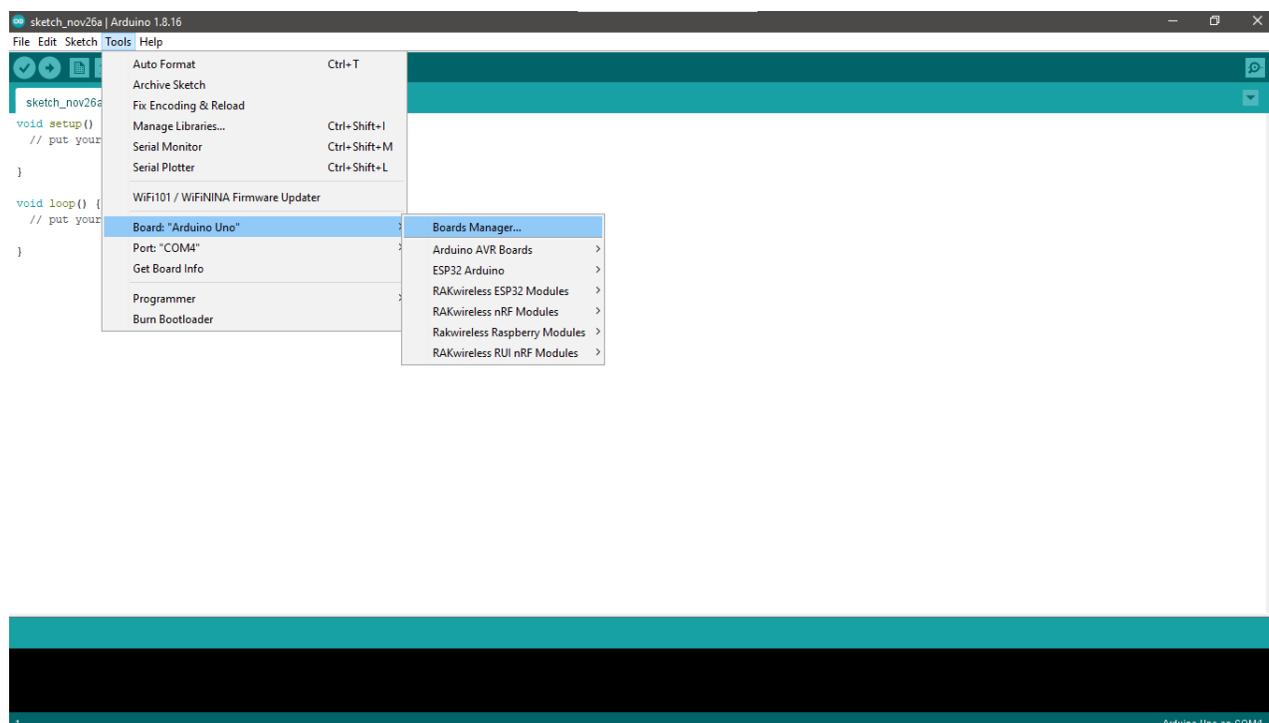


Figure 8: Opening Arduino boards manager

6. Write **RAK** in the search bar, as shown in **Figure 9**. This will show the available RAKwireless module boards that you can add to your Arduino Board list. Select and install the latest version of the **RAKwireless RUI nRF Boards**.

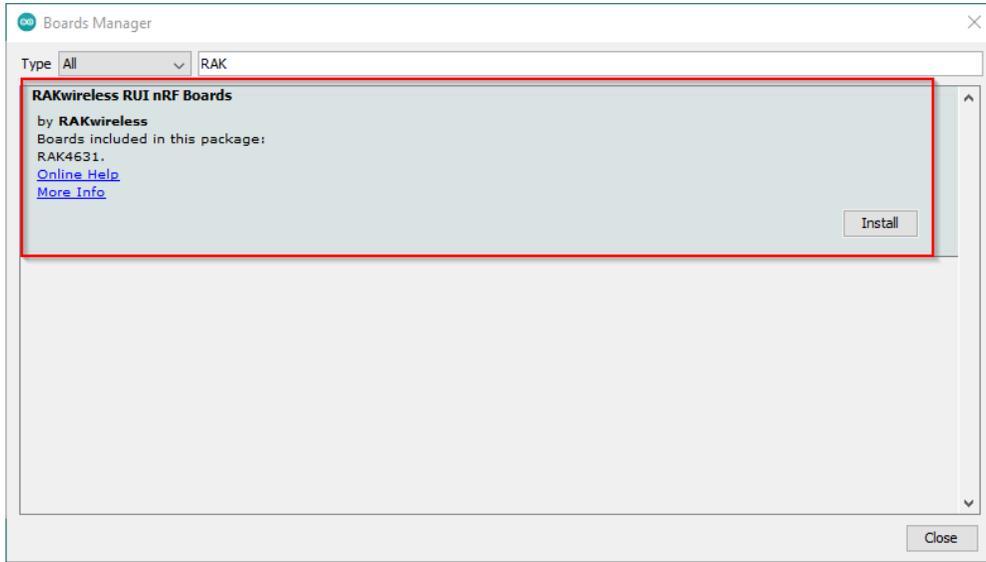


Figure 9: Installing RAKwireless RUI nRF boards

7. Once the BSP is installed, select **Tools > Boards Manager > RAKWireless RUI nRF Modules > WisBlock Core RAK4631 Board**. The RAK4631 board uses RAK4630 WisDuo module.

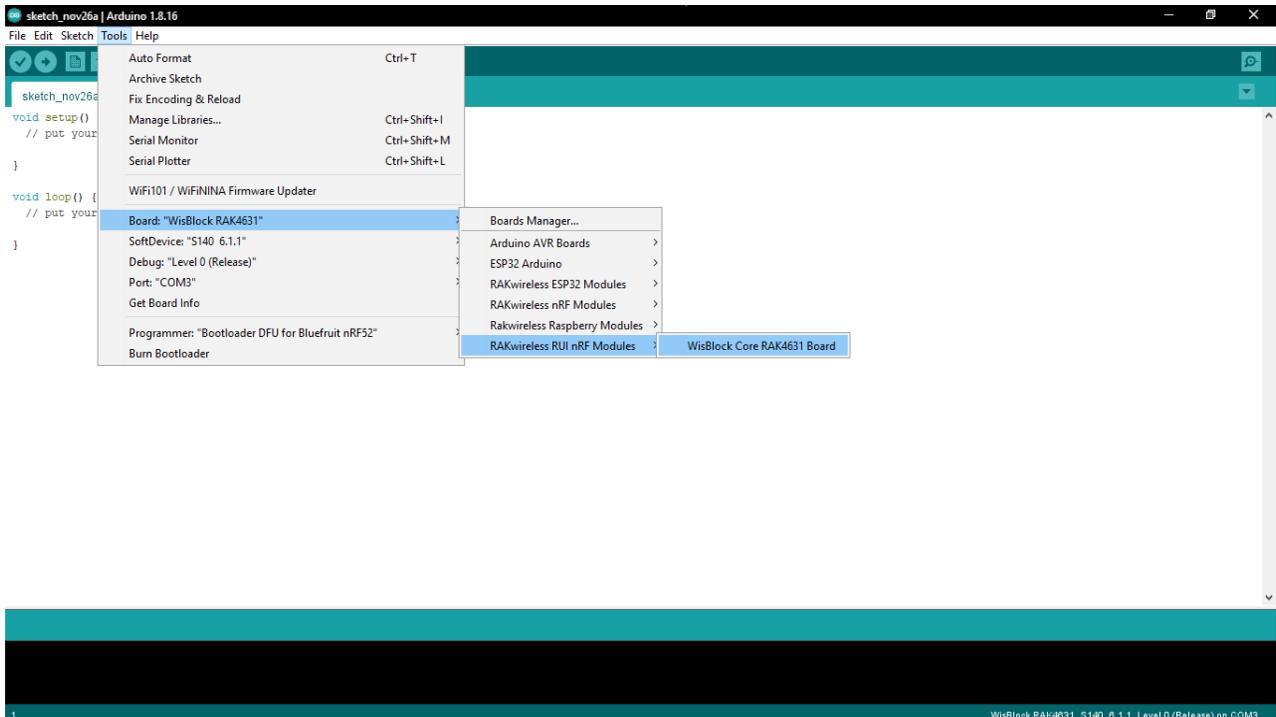


Figure 10: Selecting RAK4630 Module

Compile an Example with Arduino LED Breathing

1. After completing the steps on adding your RAK4630 to the Arduino IDE, you can now try to run a simple program to test your setup. You need to add two LEDs to the bare minimum schematic of the RAK4630 module, as shown in **Figure 11**.

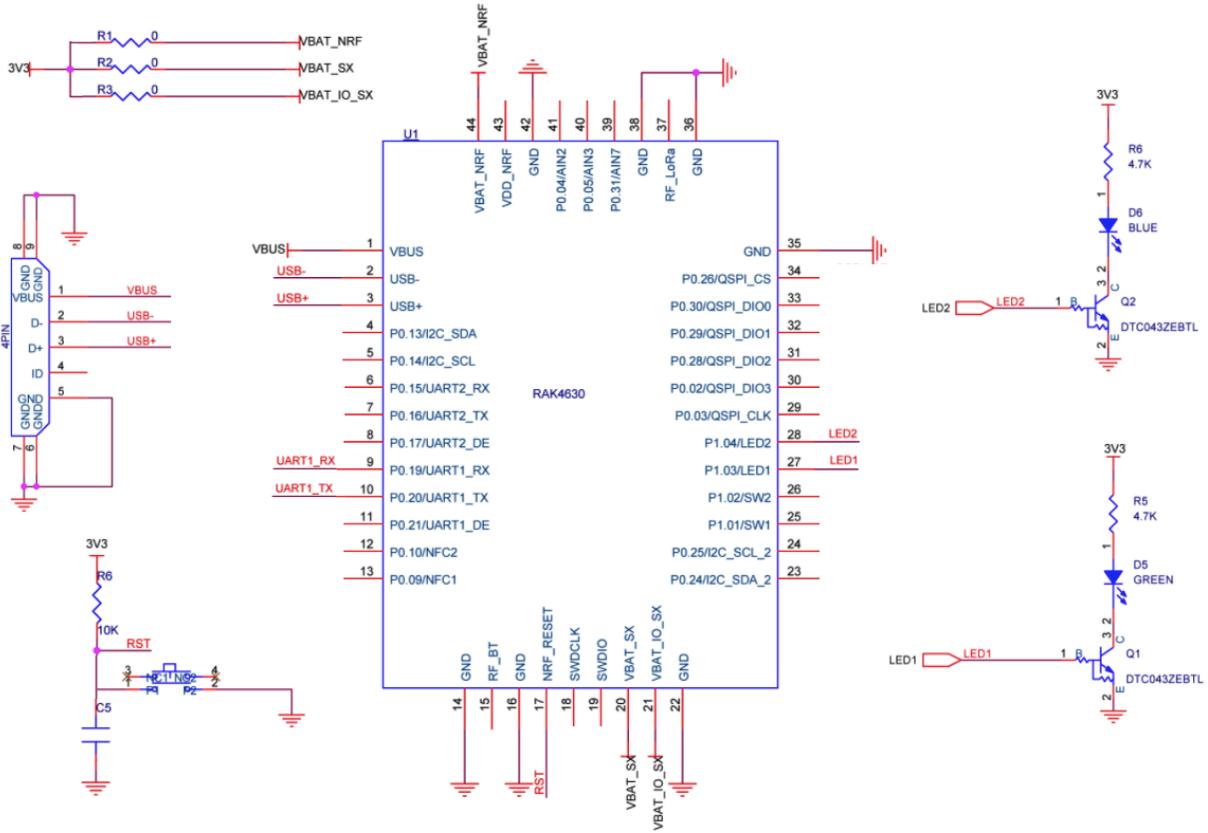


Figure 11: RAK4630 with two LEDs

2. Connect the RAK4630 via USB and check RAK4630 COM Port using Windows **Device Manager**. Double-click the reset button if the module is not detected.

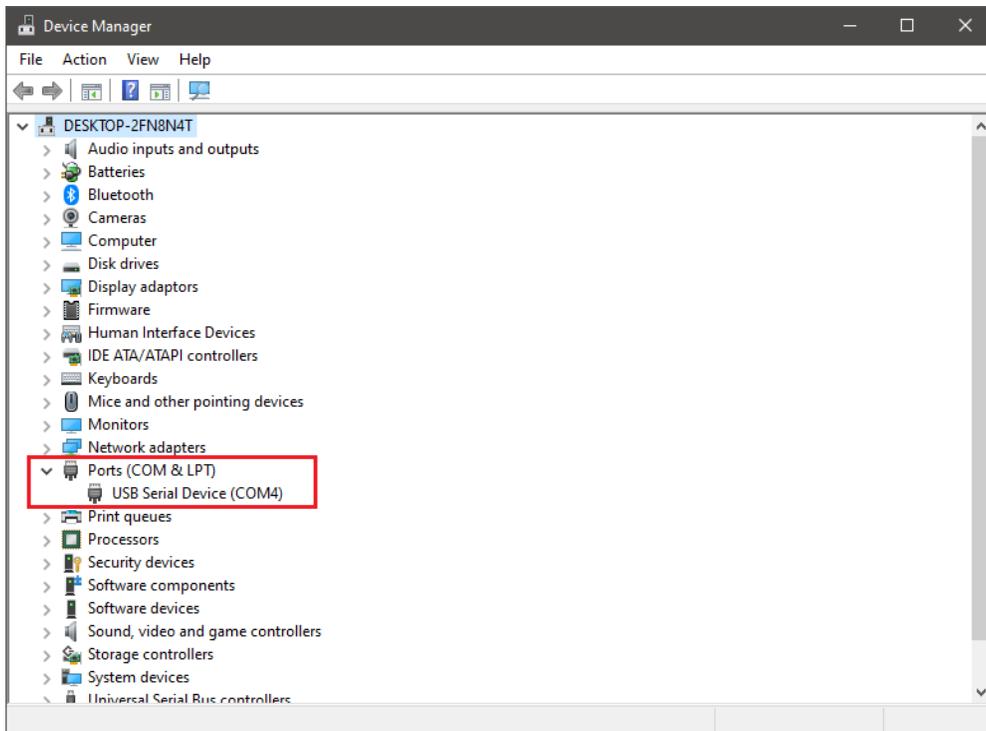


Figure 12: Device manager ports (COM & LPT)

3. Choose RAK4631 on board selection select via **Tools > Boards Manager > RAKWireless RUI nRF Modules > WisBlock Core RAK4631 Board**.

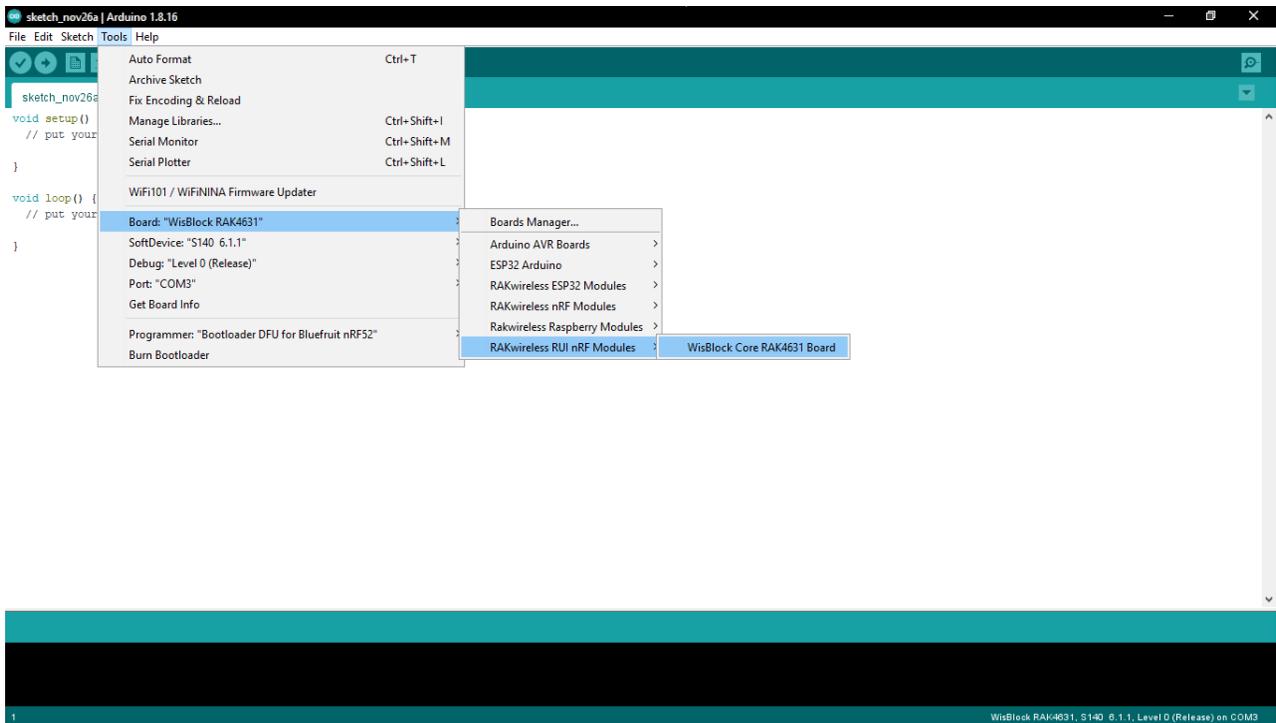


Figure 13: Selecting RAK4630 Module

4. Open **Tools** Menu and select a COM port. **COM4** is currently used.

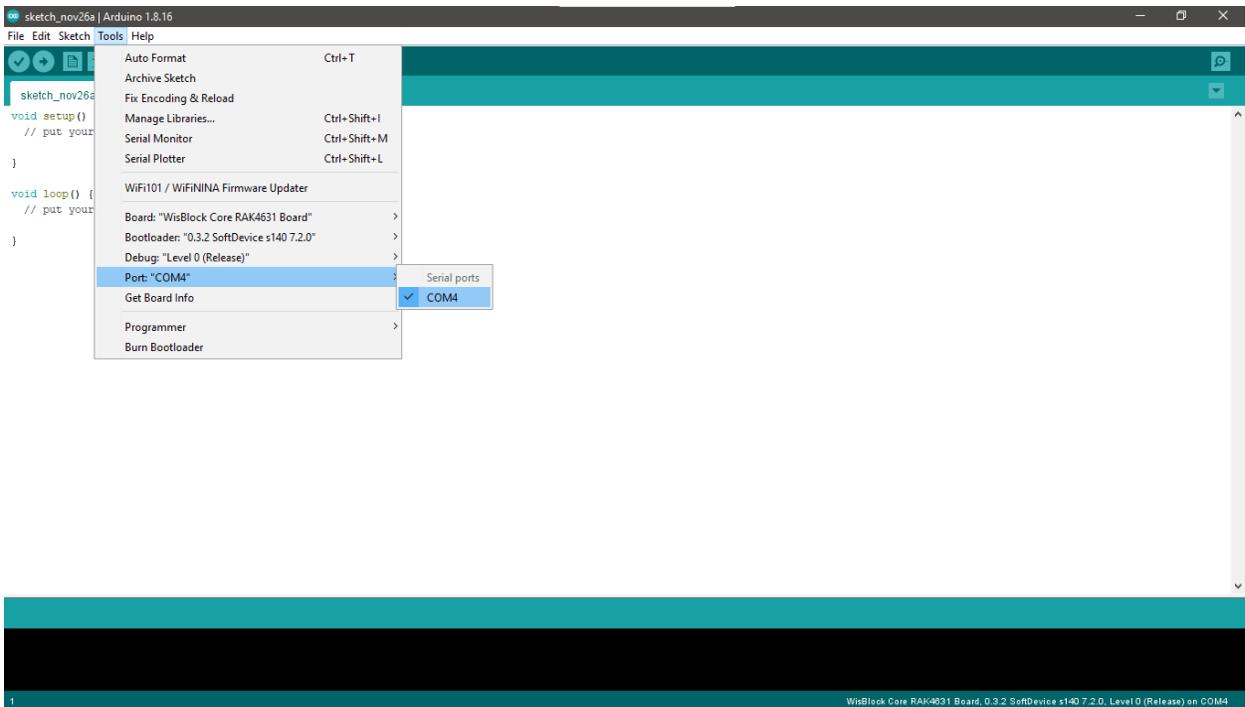


Figure 14: Select COM port

5. You can see the serial monitor icon and click it to connect COM port.

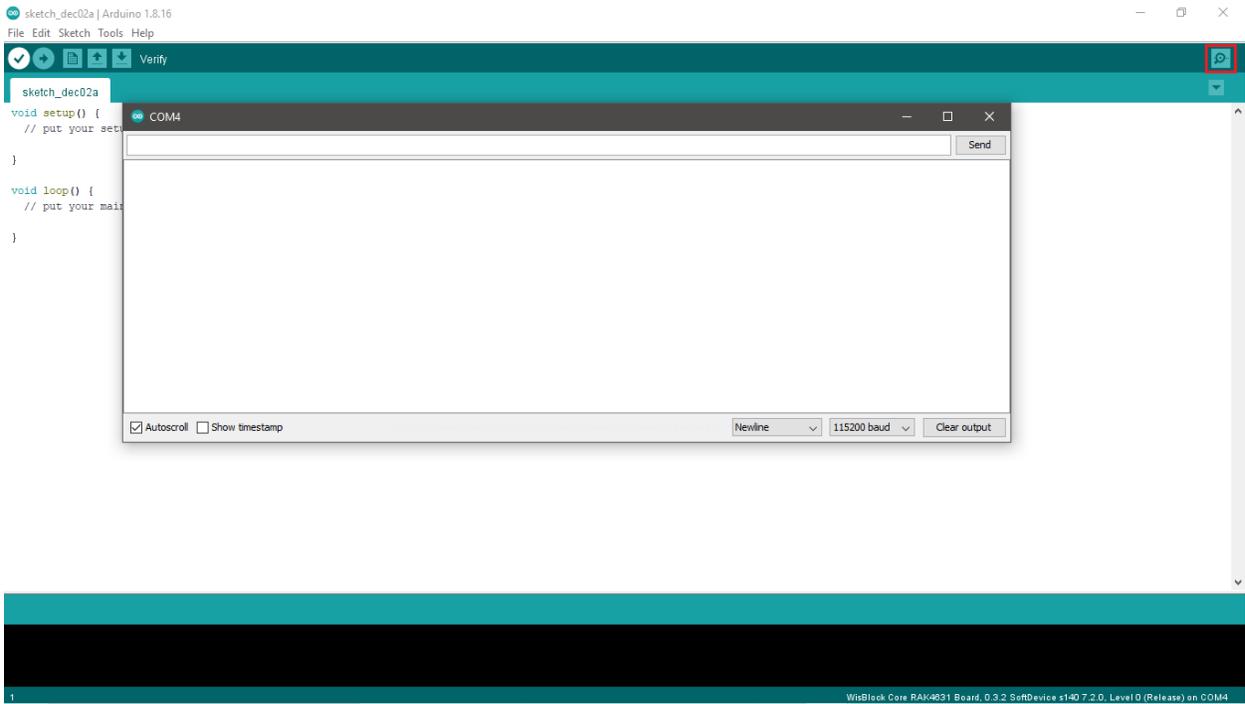


Figure 15: Open Arduino serial monitor

6. If the connection is successful, you can send AT Commands to RAK4630. For example: To check the RUI version, type `AT+VER=?` on the text area, then click on the **Send** button, as shown in **Figure 16**.

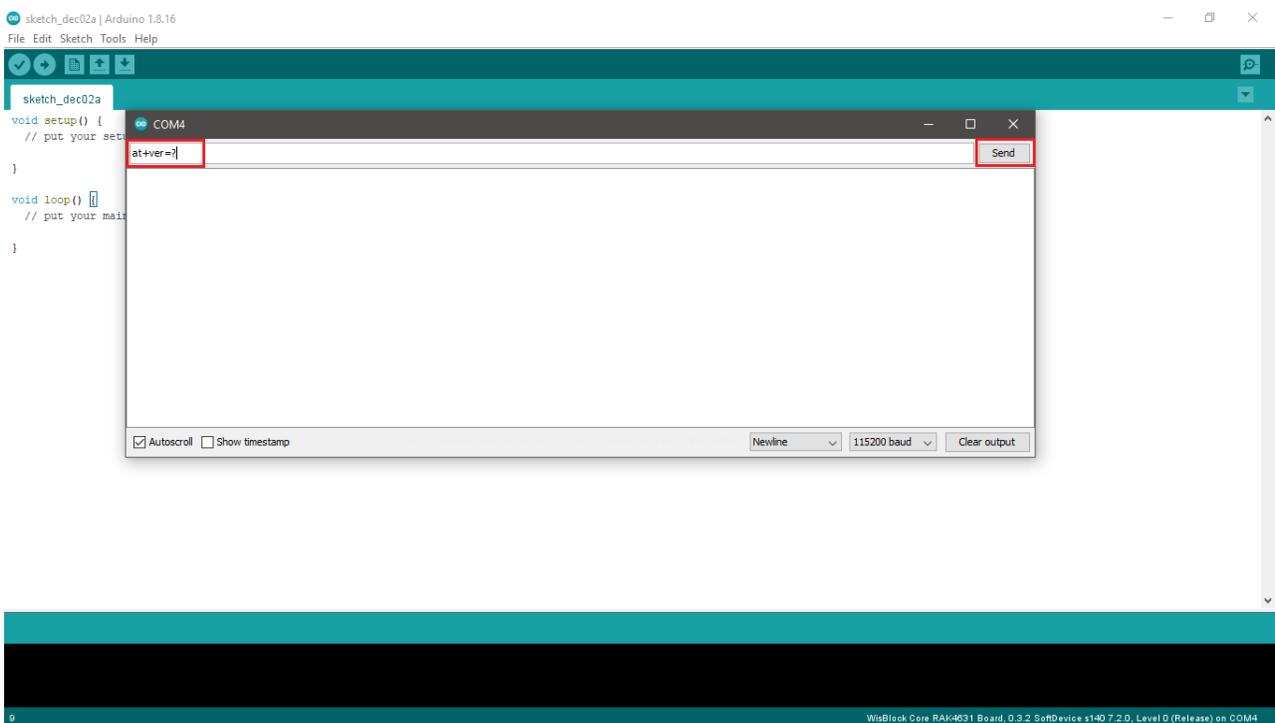


Figure 16: Send AT command

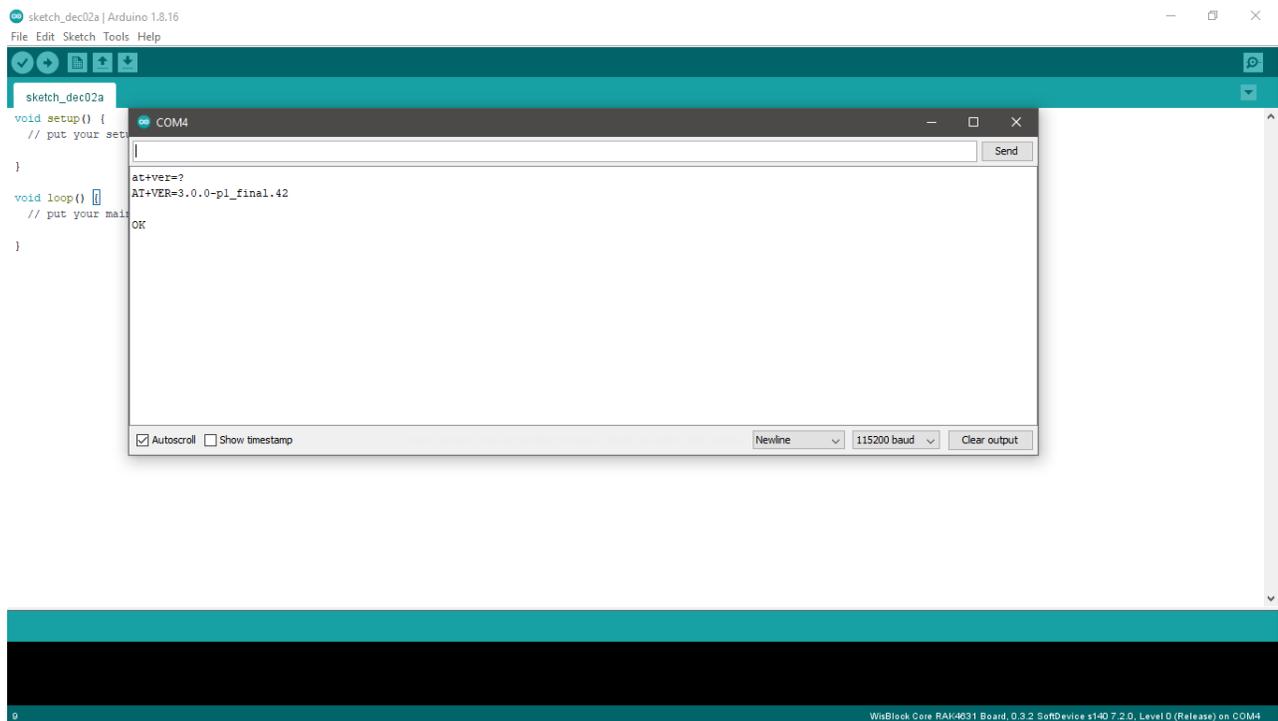


Figure 17: Arduino serial monitor COM4

7. Open `Arduino_Led_Breathing` example code.

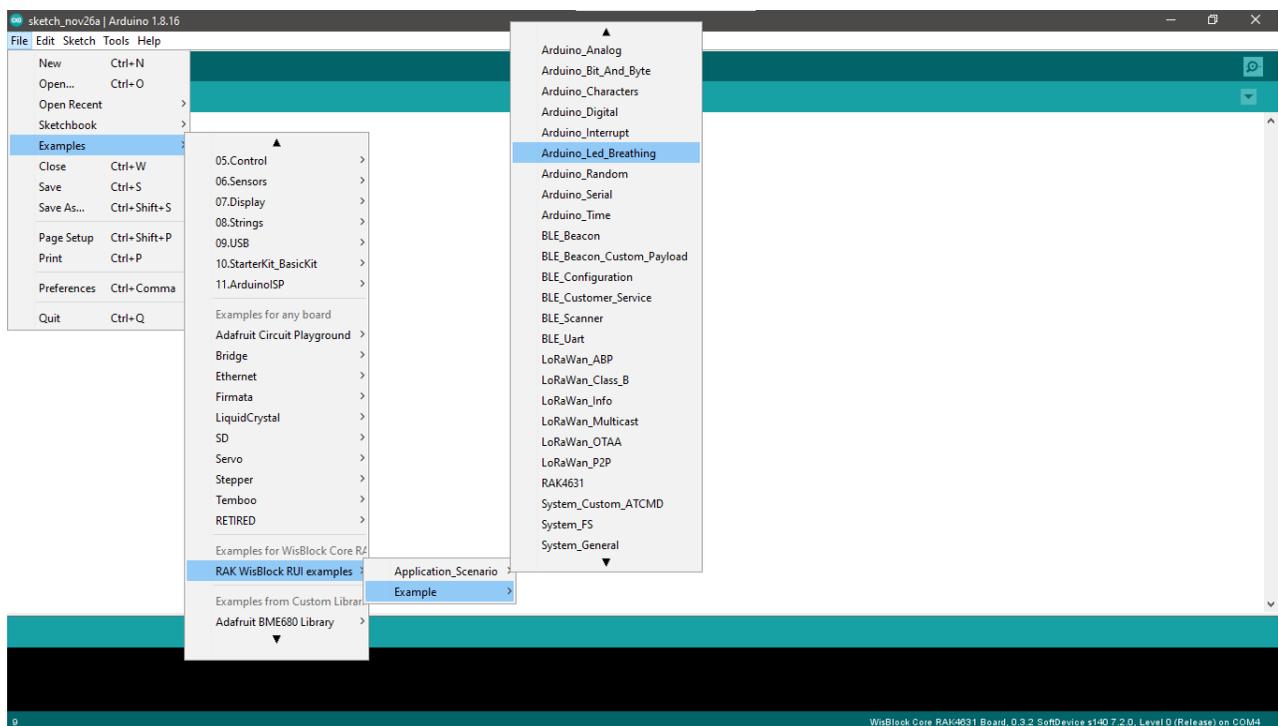


Figure 18: Led Breathing example

8. Click on the **Verify** icon to check if you have successfully compiled the example code.

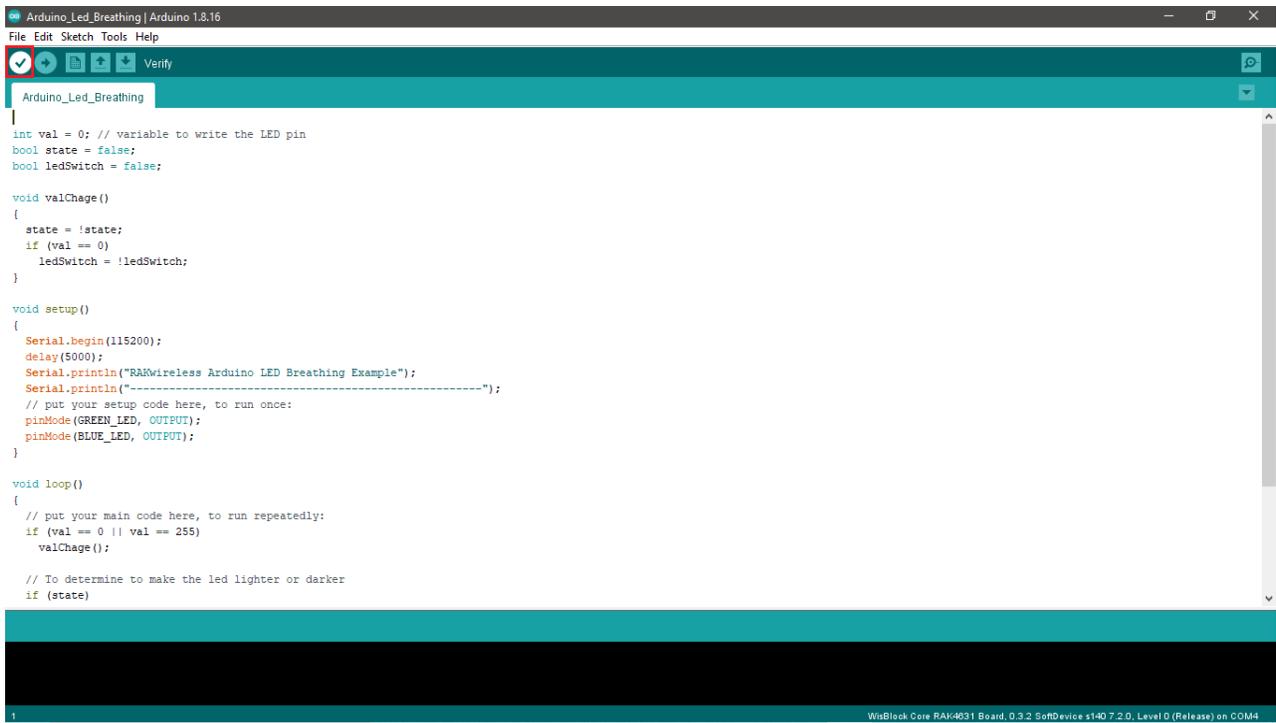


Figure 19: Verify the example code

9. Click the **Upload** icon to send the compiled firmware to your RAK4630.

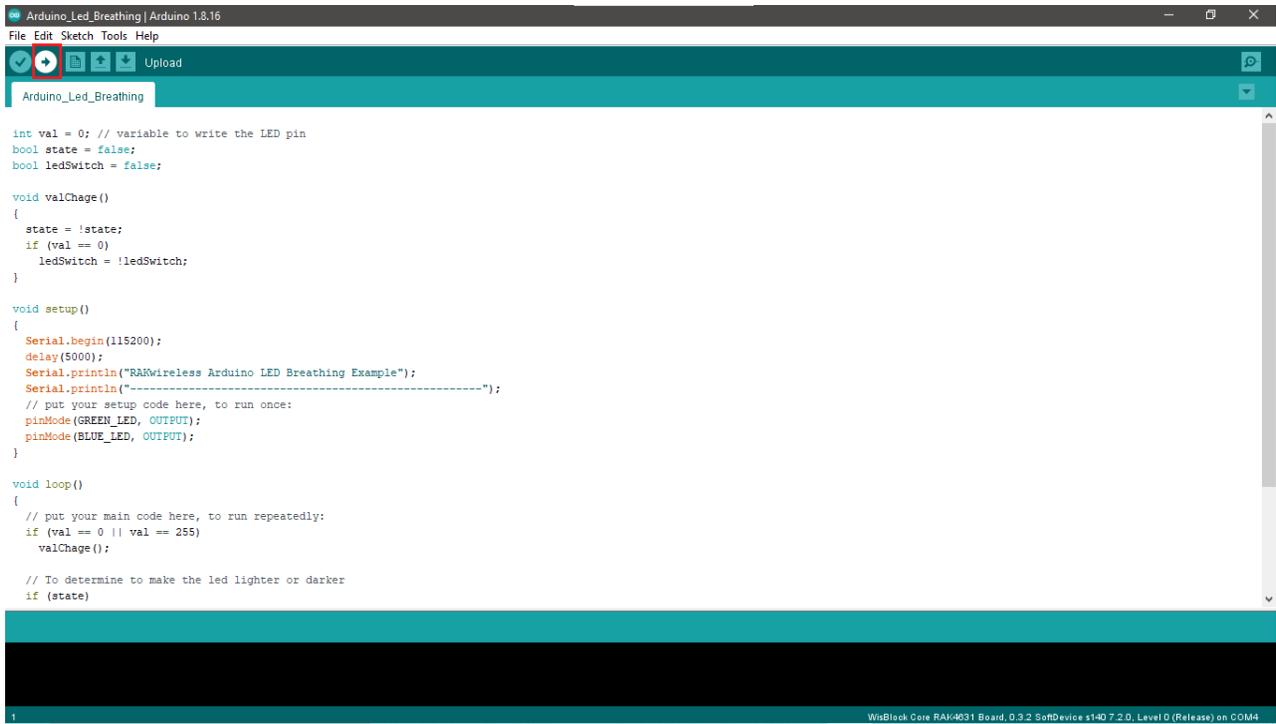
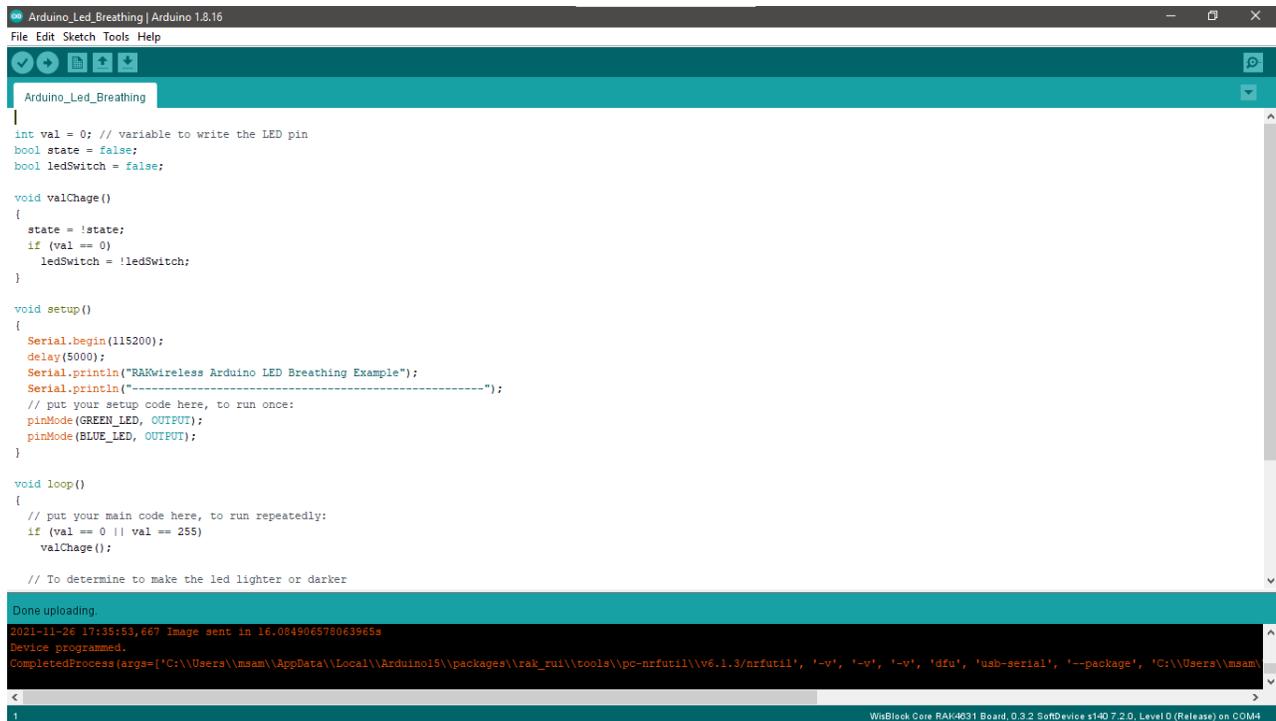


Figure 20: Upload the example code

10. If the upload is successful, you will see the **Device programmed** message.



```

Arduino_Led_Breathing | Arduino 1.8.16
File Edit Sketch Tools Help
Arduino_Led_Breathing
|
int val = 0; // variable to write the LED pin
bool state = false;
bool ledSwitch = false;

void valChange()
{
  state = !state;
  if (val == 0)
    ledSwitch = !ledSwitch;
}

void setup()
{
  Serial.begin(115200);
  delay(5000);
  Serial.println("RAKWireless Arduino LED Breathing Example");
  Serial.println("-----");
  // put your setup code here, to run once:
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BLUE_LED, OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:
  if (val == 0 || val == 255)
    valChange();

  // To determine to make the led lighter or darker
}

Done uploading.
2021-11-26 17:35:53,667 Image sent in 16.084906578063965s
Device programmed.
CompletedProcess(args=['C:\\Users\\msam\\AppData\\Local\\Arduino15\\packages\\rak_rui\\tools\\pc-nrfutil\\v6.1.3\\nrfutil', '-v', '-v', '-v', 'dfu', 'usb-serial', '--package', 'C:\\Users\\msam

```

Figure 21: Device programmed successfully

11. After the Device Programmed is completed, you will see that LEDs are blinking.

LoRaWAN Example

This example illustrates how to program RAK4630 as a stand-alone LoRaWAN end-device via RUI3 Arduino APIs. To use RAK4630 as a LoRaWAN end-device, it needs to be within reach of a working **LoRaWAN gateway** registered to a **LoRaWAN network server(LNS)** or with a built-in network server.

Moreover, this guide will use the internal network server of the [WisGate Edge RAK7268](#) LoRawan gateway. RAK offers various models of WisGate LoRaWAN gateways that you can buy from the [RAK store](#) . This guide is focused on the configuration of RAK4630 as a LoRaWAN end-device.

As a LoRaWAN refresher, these are three main things you need to have a working LoRaWAN application:

- LoRaWAN Gateway (this guide uses RAK7268 WisGate Edge gateway)
- LoRaWAN Network Server (this guide uses RAK7268 WisGate Edge built-in LoRaWAN network server)
- LoRaWAN End-Device (RAK4630 in this guide)

NOTE:

If you are new to LoRaWAN, here are a few good references about LoRaWAN and gateways:

- [LoRaWAN 101](#)
- [What is a LoRaWAN Gateway](#)
- [How do LoRaWAN® Gateways work?](#)
- [Things to Consider When Picking A LoRaWAN® Gateway](#)

LoRaWAN gateway models like WisGate Edge have built-in network servers. It is also common that the LoRaWAN network server is external or in the cloud. The popular LoRaWAN network server in the cloud that you can use for free (but offers enterprise service, too) is TTN.

Adding the RAK4630 as an End-Device to the LoRaWAN Network Server

1. You need to have a functional RAK7268 to proceed with the next steps. For a complete guide, refer to [RAK7268 WisGate Edge Quick Start Guide](#) . Other WisGate Edge configurations can be found in the detailed

documentation of the [WisGate OS](#) .

This section will show you how to add a LoRaWAN application to the built-in network server of RAK7268 and how to add a device to it. You need to have [access on the RAK7268 console portal](#) to perform the necessary configurations. The IP address you need to use to access the portal will depend on your type of connection to RAK7268. For example, if your RAK7268 is configured as Access Point, you can connect to it via

```
http://192.168.230.1/ .
```

NOTE:

This initial guide uses the LoRaWAN built-in network server of RAK7268, but you can use another LoRaWAN network server, as well, like TTN, Chirpstack, Helium, AWS, etc. Those have different user interfaces, but the parameters to be configured should be the same as long as it follows the LoRaWAN specification standard.

You can find the [TTN guide](#) and [Chirpstack guide](#) on the Quick Start Guide section of this document as well but on the section of RAK4630 as LoRa/LoRaWAN Modem.

- Once you have a working RAK7268, you need to ensure that you use the built-in LoRaWAN network server (LNS). You can do this by configuring the gateway to network server mode. Click **LoRa Network** then select **Network Settings**.

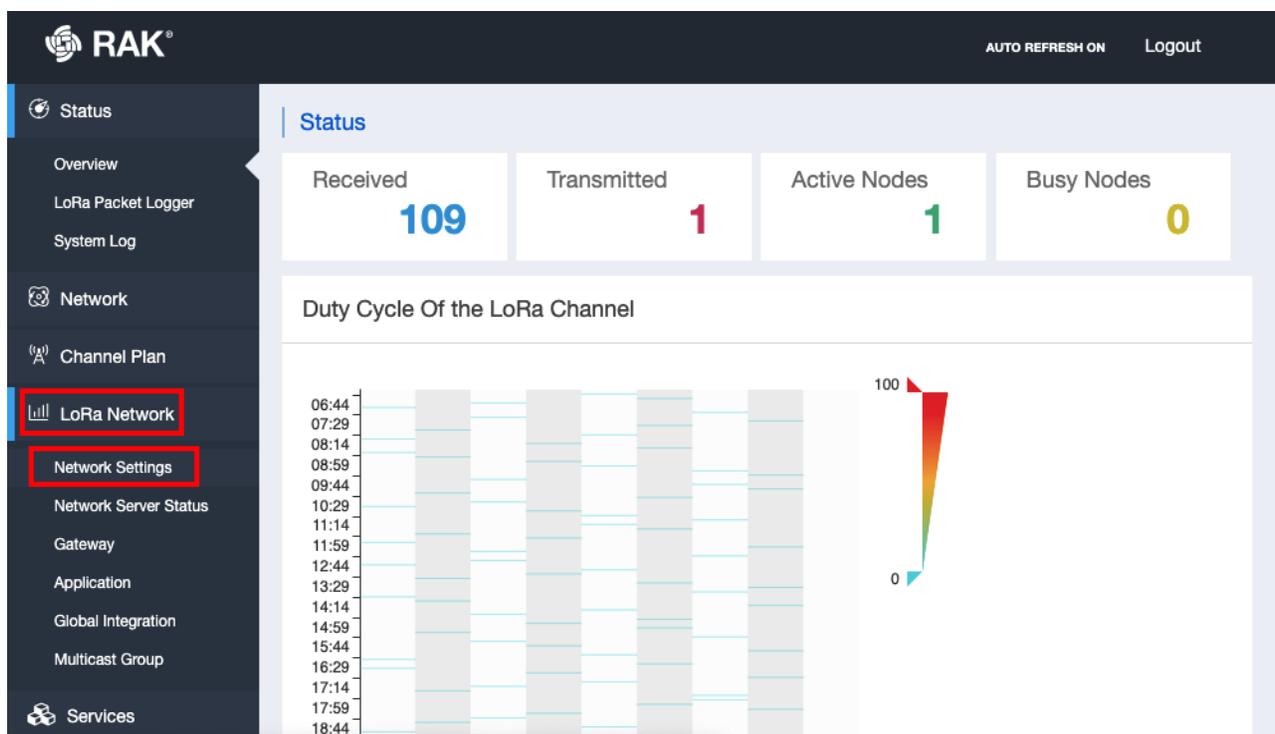


Figure 22: RAK7268 LoRa network settings

- Select the **mode** dropdown list **Network Server** and take note of the **Region**. The **Region** should be the same as the RAK4630 code that you will compile in the later section of this guide. The region of the LoRaWAN gateway and LNS used in this guide is US915.

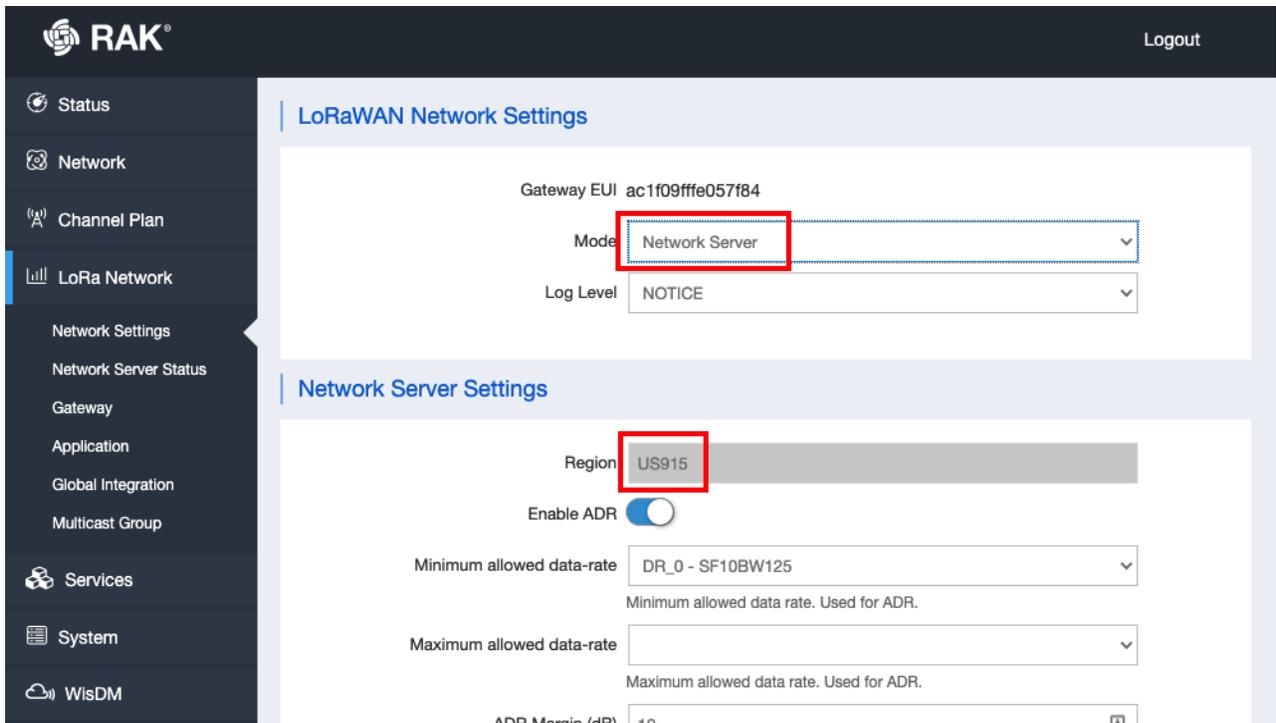


Figure 23: RAK7268 network server mode

4. After configuring the gateway to use a built-in LoRaWAN network server (LNS), you need to add an application to the network server and register a device. Click **Application**, which will show all the LoRaWAN applications you have if there are any.

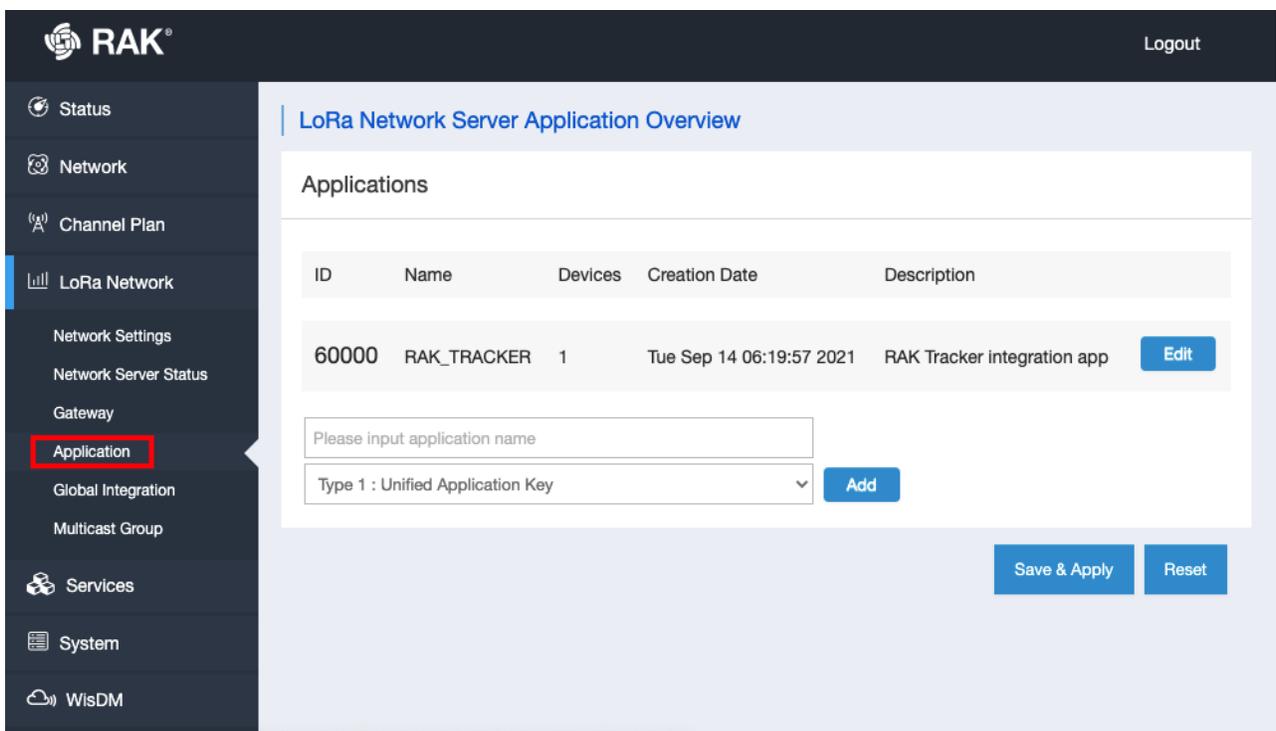


Figure 24: Accessing Application Console in the network server

5. You need to put a meaningful name for your application. For this illustration, the **RAK4630** is used, and then, you should select **Unified Application Key**. You can click **Add** and **Save & Apply** to proceed with the configuration of the application.

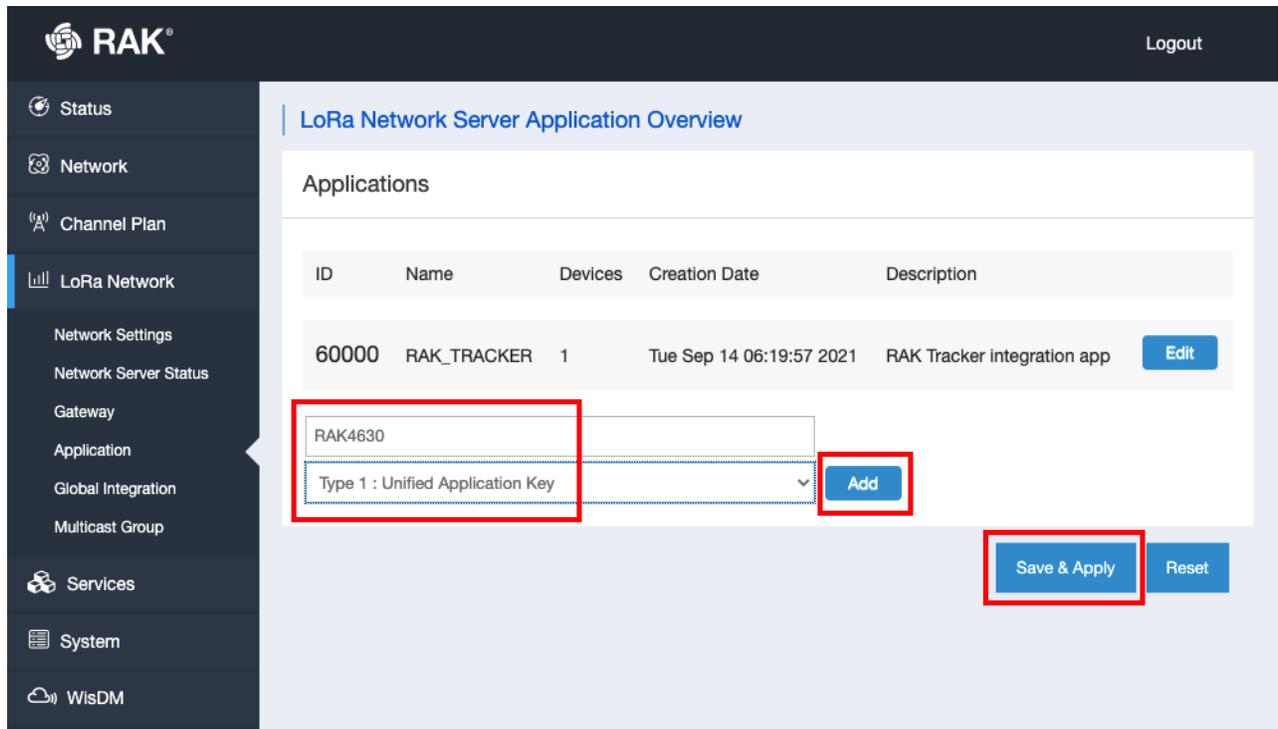


Figure 25: Adding Application to the LoRaWAN network server

6. Another important parameter in a LoRaWAN application is the **Application Key**, which is used in **OTAA** activation. You can manually add one, but it is advisable to let the network server generate it by clicking the generate icon button. You can also add a description to your application. After having the **Application Key**, you can now click **Save & Apply**.

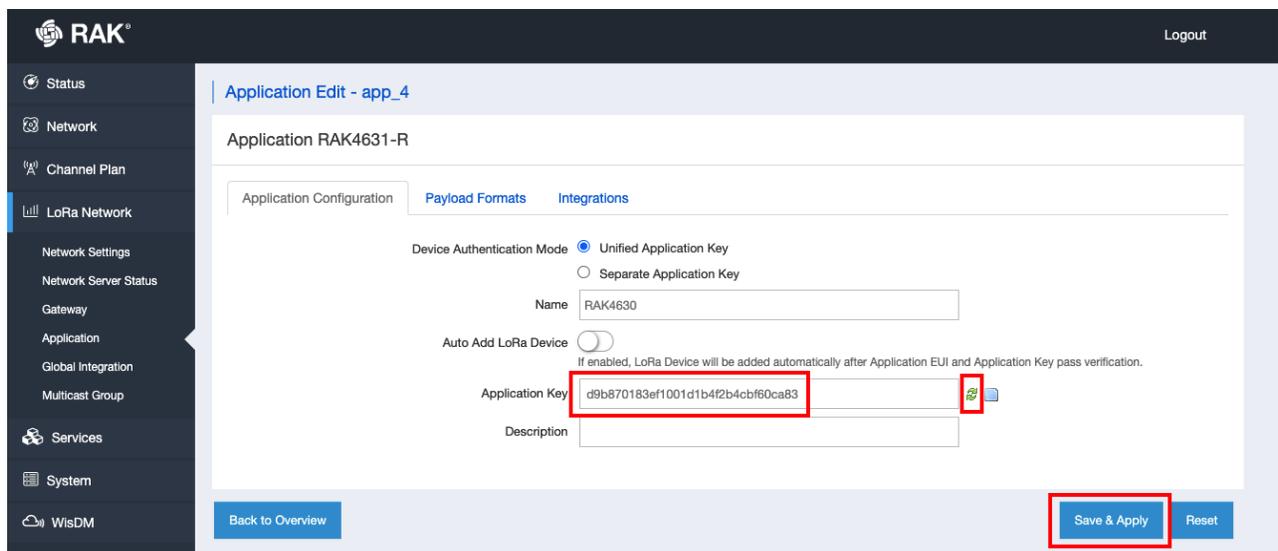


Figure 26: Configure Application Key

7. The new application will now be added to the application list of your network server. To register your RAK4630 WisDuo module as a device in that application, click edit to access the application console.

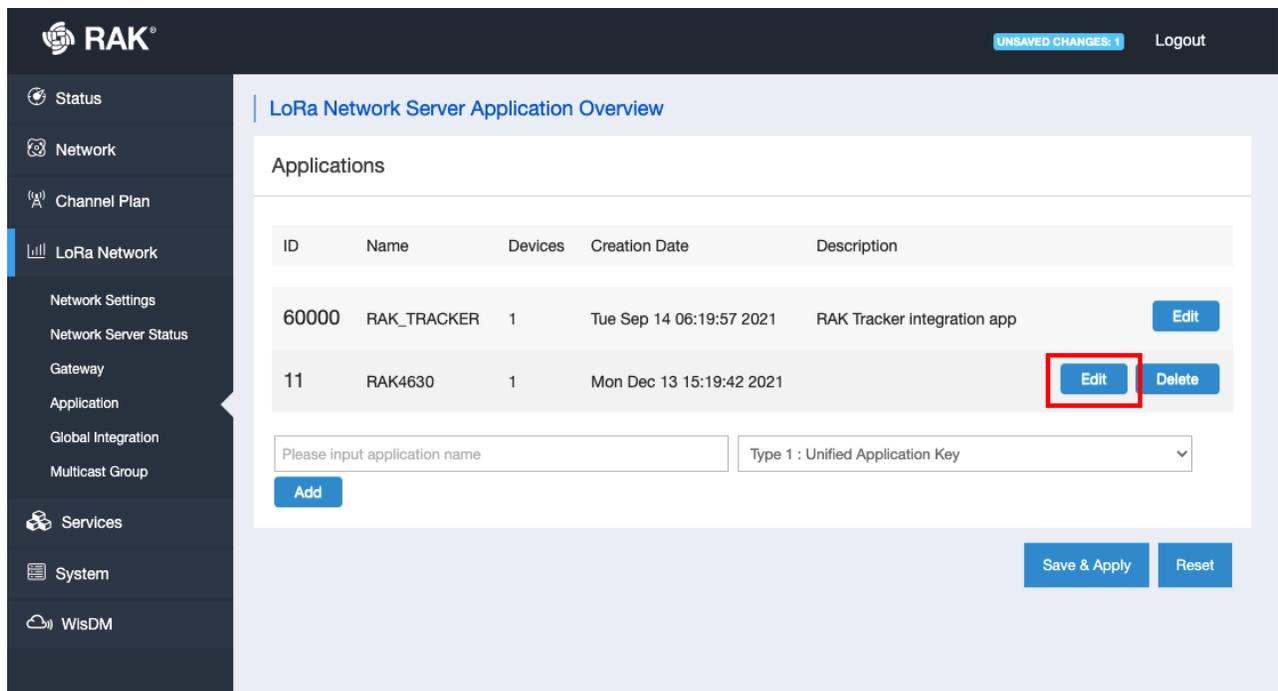


Figure 27: Access the LoRaWAN Application Console

8. To register your RAK4630 WisDuo module, add its device EUI (DEVEUI) on the application console. This must be unique, and it is highly encouraged that you use the **DEVEUI** printed on the sticker of your RAK4630 module. Then click the **Add** button to proceed with the next configurations.

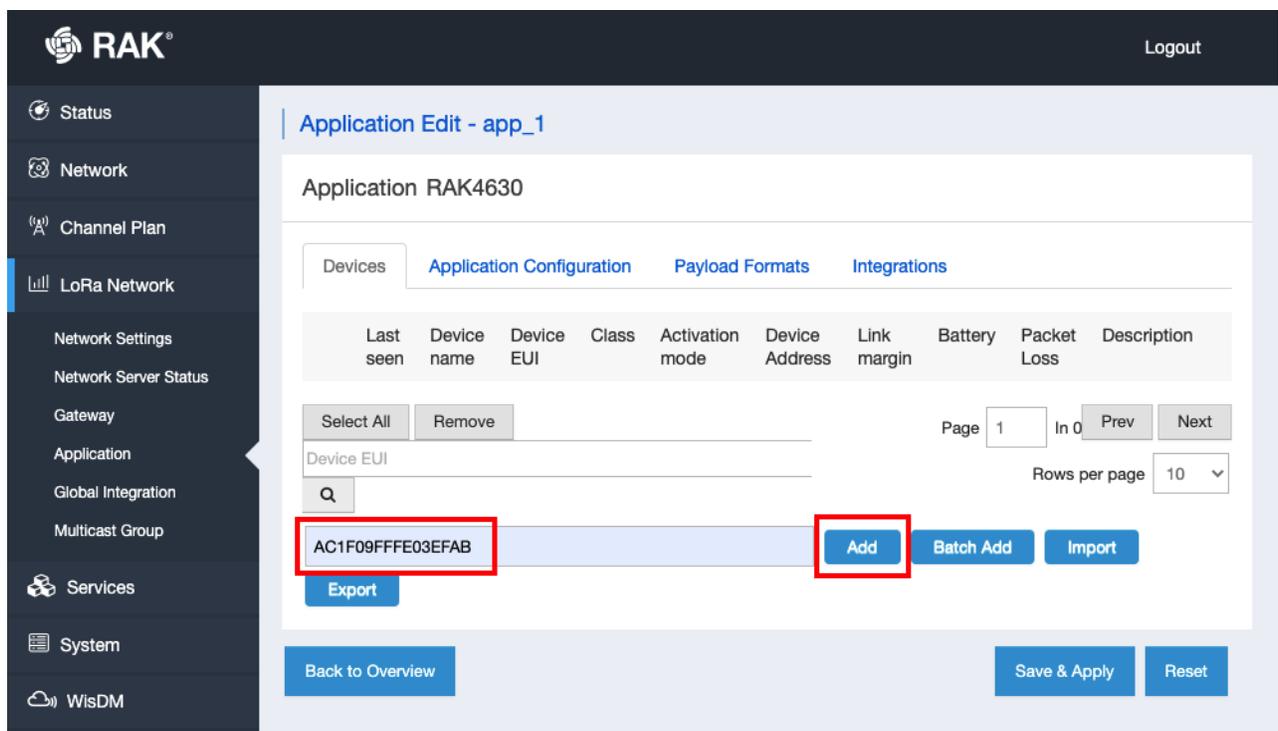


Figure 28: Adding device to the application

9. The next step is to configure the other important parameters related to LoRaWAN. You need to put a name on your device and in this case, it is **RAK4630**. Three parameters in this list must be the same as your RAK4630 code: **Class**, **Join Mode**, and **Application Key**.

On the example Arduino code in the latter part of this guide, the Class A and Join Mode OTAA are default. With that, you will only need to modify the code of the application key. Lastly, the **LoRaWAN MAC Version** must be **1.0.3**. After setting up all these parameters, click **Save & Apply**.

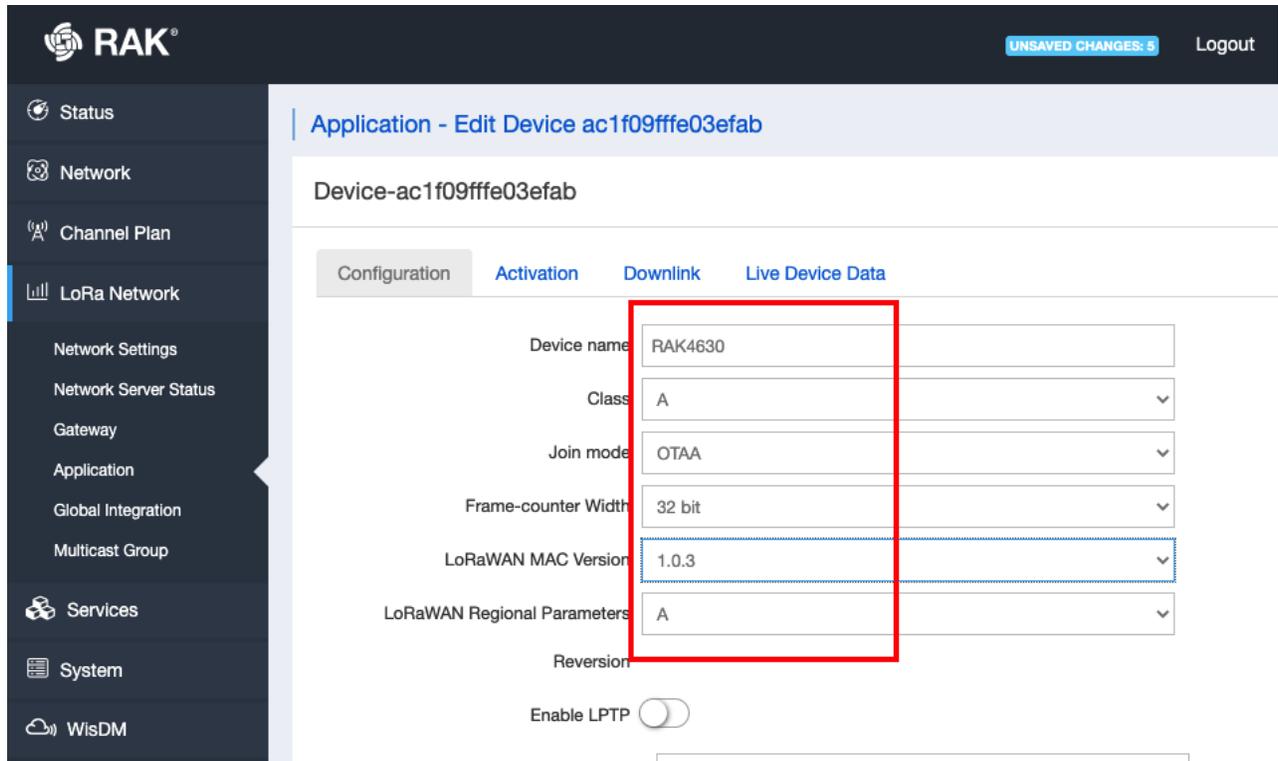


Figure 29: Configuration of the RAK4630 as OTAA device

10. The device will now be shown as a device in your application. The next step is to prepare the RAK4630 WisDuo module so that it can match all the parameters in the network server.

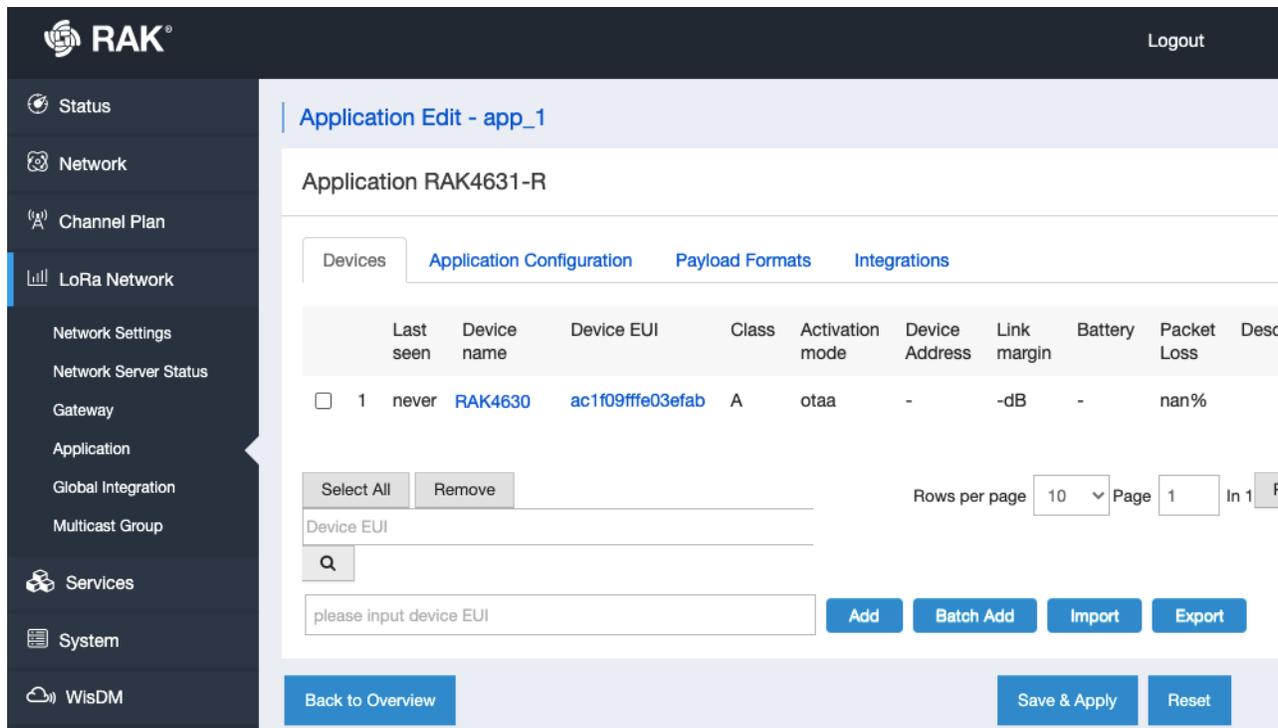


Figure 30: RAK4630 registered on the network server

Uploading LoRaWAN Example to RAK4630

After a successful registration of the RAK4630 device to the LoRaWAN Network Server, you can now proceed with running the LoRaWAN OTAA demo application example.

1. Open the example code under **RAK WisBlock RUI examples**.

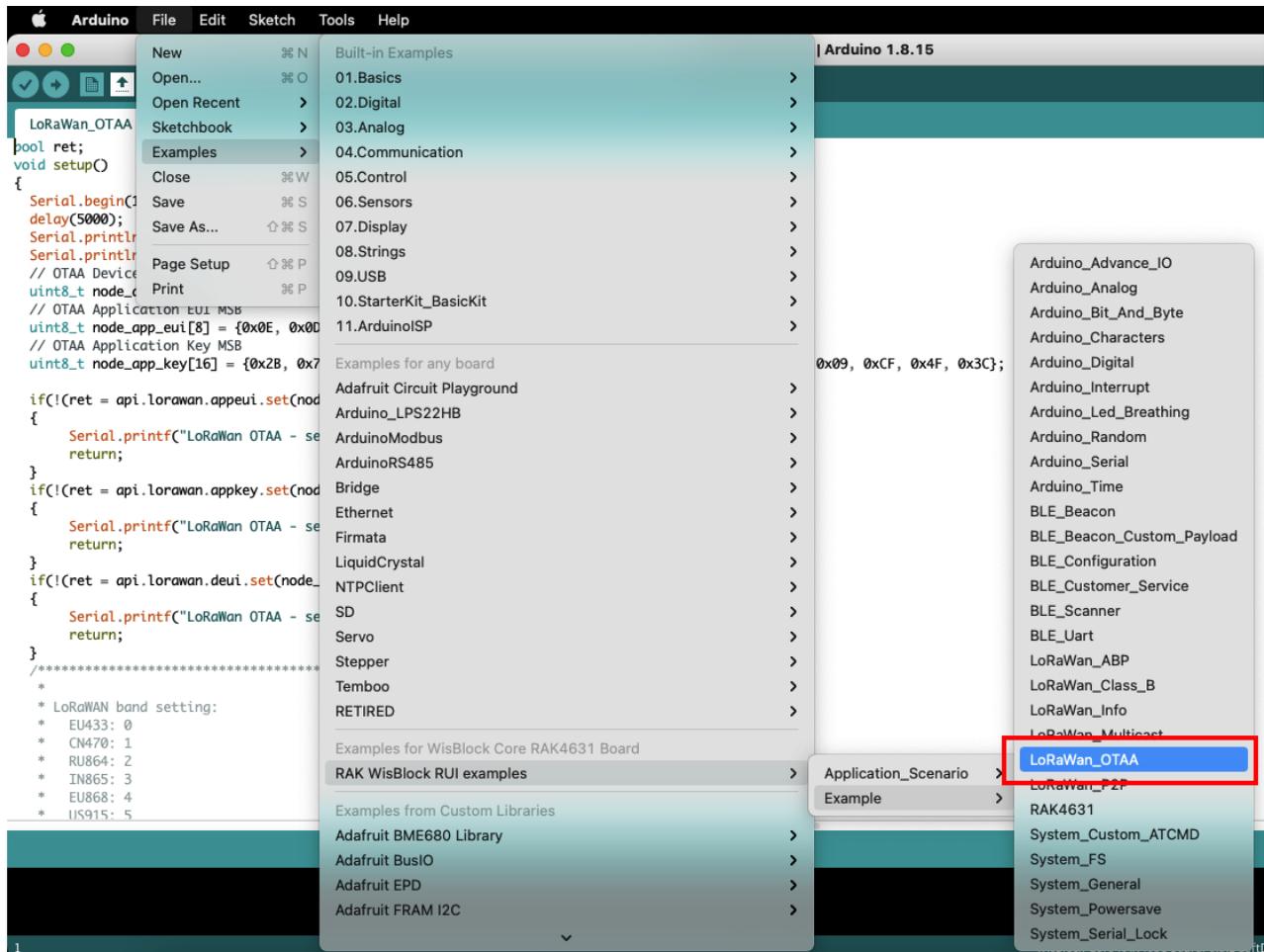


Figure 31: OTAA LoRaWAN application example

2. In the example code, you need to modify the device EUI (DEVEUI) and application key (APPKEY).

- The **device EUI (DEVEUI)** should match the device EUI registered in your network server. This is the same DEVEUI in the RAK4630 sticker if this is the one you used in the previous section [device registration in the network server](#). DEVEUI is MSB.

```

// OTAA Device EUI MSB
uint8_t node_device_eui[8] = {0xAC, 0x1F, 0x09, 0xFF, 0xFE, 0x03, 0xEF, 0xAB};
  
```

- Another important parameter to change is the **application key (APPKEY)**. This parameter should also be the same as the APPKEY in the network server you configured in the previous section [adding RAK4630 in the network server](#) section. APPKEY is MSB.

```

// OTAA Application Key MSB
uint8_t node_app_key[16] = {0xD9, 0xB8, 0x70, 0x18, 0x3E, 0xF1, 0x00, 0x1D, 0x1B, 0x4F, 0x2B, 0x09, 0xCF, 0x4F, 0x3C};
  
```

```

LoRaWAN_OTAA §
bool ret;
void setup()
{
  Serial.begin(115200);
  delay(5000);
  Serial.println("RAKwireless LoRaWAN OTAA Example");
  Serial.println("-----");
  // OTAA Device EUI MSB
  uint8_t node_device_eui[8] = {0xAC, 0x1F, 0x09, 0xFF, 0xFE, 0x03, 0xEF, 0xAB};
  // OTAA Application EUI MSB
  uint8_t node_app_eui[8] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
  // OTAA Application Key MSB
  uint8_t node_app_key[16] = {0xD9, 0xB8, 0x70, 0x18, 0x3E, 0xF1, 0x00, 0x1D, 0x1B, 0x4F, 0x2B, 0x4C, 0xBF, 0x60, 0xCA, 0x83};

  if (!(ret = api.lorawan.appeui.set(node_app_eui, 8)))
  {
    Serial.printf("LoRaWAN OTAA - set device EUI is incorrect! \r\n");
    return;
  }
  if (!(ret = api.lorawan.appkey.set(node_app_key, 16)))
  {
    Serial.printf("LoRaWAN OTAA - set application EUI is incorrect! \r\n");
    return;
  }
  if (!(ret = api.lorawan.deui.set(node_device_eui, 8)))
  {
    Serial.printf("LoRaWAN OTAA - set application key is incorrect! \r\n");
    return;
  }
}
/*****

LoRaWAN band setting:
EU433: 0
CN470: 1
RU864: 2
IN865: 3
EU868: 4
US915: 5

Auto Format finished.
54 WisBlock Core RAK4631 Board, 0.3.2 SoftDevice s140 7.2.0, Level 0 (Release) on /dev/cu.usbmodemC0D048F604F1
  
```

Figure 32: Updating DEVEUI and APPKEY

3. This guide uses US915 regional band, so you need to update the band in the code as well. In addition to that, you also need to set up the channel mask (channels 8 to 15 are the most commonly used channels in the US915 band).

```

if(!(ret = api.lorawan.band.set(5)))
{
  Serial.printf("LoRaWAN OTAA - set band is incorrect! \r\n");
  return;
}

uint16_t maskBuff = 0x0002;
api.lorawan.mask.set(&maskBuff);
  
```

NOTE:

RAK4630 supports the following regions:

- RAK_REGION_EU433 = 0
- RAK_REGION_CN470 = 1
- RAK_REGION_RU864 = 2
- RAK_REGION_IN865 = 3
- RAK_REGION_EU868 = 4
- RAK_REGION_US915 = 5
- RAK_REGION_AU915 = 6
- RAK_REGION_KR920 = 7
- RAK_REGION_AS923 = 8

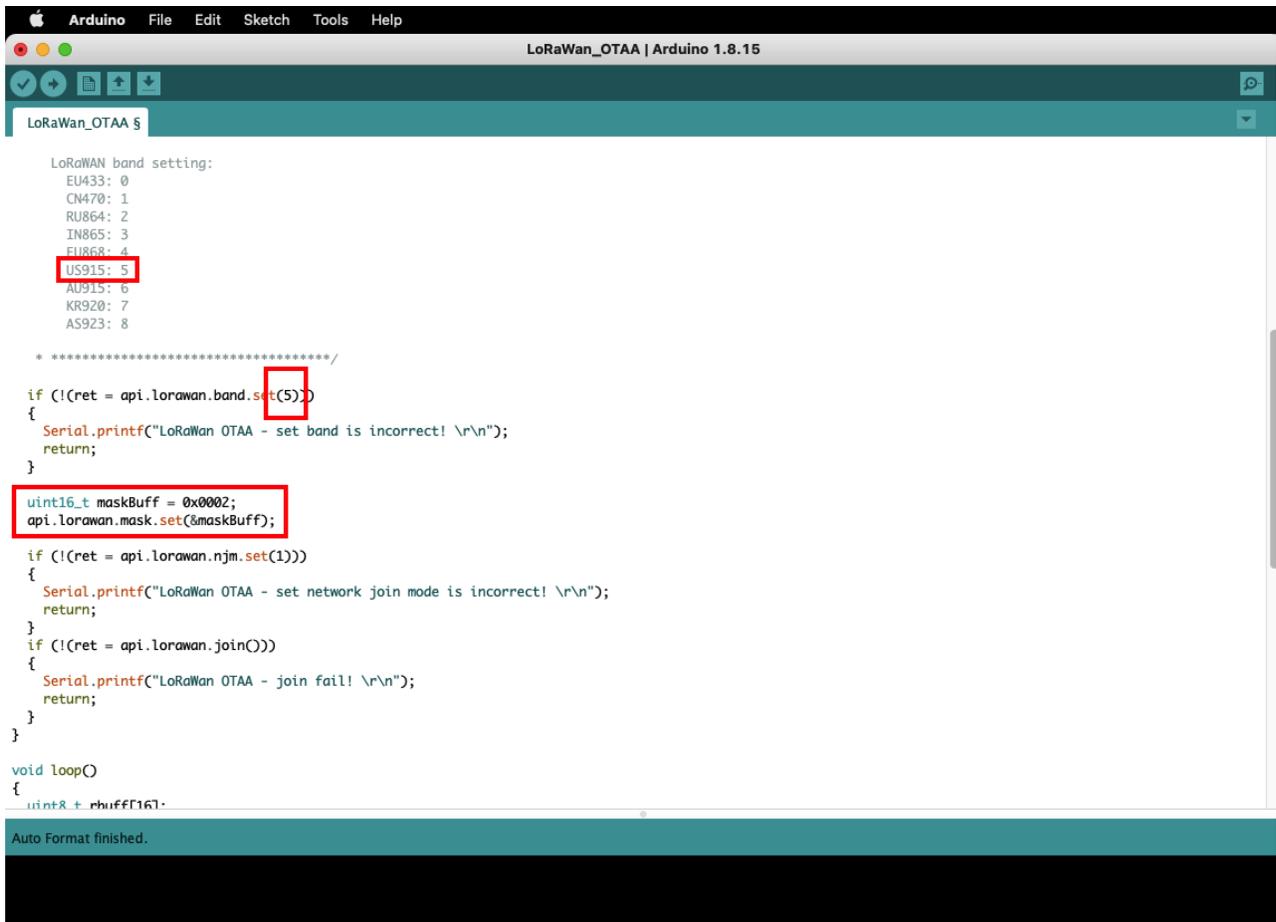


Figure 33: Updating to US915 and setting up channel mask

4. The last step is to upload the code by clicking the **Upload** icon. Take note that you should select the right board and port, as shown in the previous example [LED Blinking](#).



Figure 34: Uploading the code

5. You should now be able to see the terminal logs in the Serial Monitor of Arduino IDE. Sometimes COM port will be disconnected so you won't be able to see the terminal output immediately. You can reconnect the module or try to push the reset button to see the terminal output.

```

23:24:44.202 -> Network Session Key=0x0000000000000000
23:24:44.272 -> 24-bit Network Identifier(NetId)=0x000
23:24:44.308 -> Device Address=0x0000
23:24:45.348 -> Network Join Status=0
23:24:45.348 -> Application Session Key=0x0000000000000000
23:24:45.456 -> Network Session Key=0x0000000000000000
23:24:45.530 -> 24-bit Network Identifier(NetId)=0x000
23:24:45.567 -> Device Address=0x0000
23:24:46.601 -> Network Join Status=0
23:24:46.608 -> Application Session Key=0x0000000000000000
23:24:46.712 -> Network Session Key=0x0000000000000000
23:24:46.788 -> 24-bit Network Identifier(NetId)=0x000
23:24:46.826 -> Device Address=0x0000
23:24:47.083 -> +EVT:JOINED
23:24:47.868 -> Network Join Status=1
23:24:47.868 -> Application Session Key=0x3a8ecba25fcab7535715d293730a6c2
23:24:47.976 -> Network Session Key=0x64afe6f8781d89246f305687276ac
23:24:48.051 -> 24-bit Network Identifier(NetId)=0x100
23:24:48.088 -> Device Address=0x226d46b
23:24:49.125 -> Network Join Status=1
23:24:49.125 -> Application Session Key=0x3a8ecba25fcab7535715d293730a6c2
23:24:49.233 -> Network Session Key=0x64afe6f8781d89246f305687276ac
23:24:49.305 -> 24-bit Network Identifier(NetId)=0x100
23:24:49.342 -> Device Address=0x226d46b
23:24:50.367 -> Network Join Status=1
    
```

Figure 35: Output logs

6. You can also check on the LoRaWAN network server logs of RAK7268 if your device has been successfully joined with the `join request` and `join accept` logs.

Time	Freq.	RSSI	SNR	TxPwr	CRC	mod.	CR	DataRate	FCnt	AirTime	DevAd
23:24:44	925.1	-	-	20	CRC	LORA	4/5	SF10BW500	-	113	-
23:24:42	904.5	-37	15	-	CRC_OK	LORA	4/5	SF10BW125	-	371	-

Figure 36: WisGate LoRa packet logger

Rak4630 as a Lora/Lorawan Modem via AT Command

This section covers the guide on how to use RAK4630 as a LoRa/LoRaWAN modem via AT command.

- [AT command via UART1](#)
- [AT command via BLE](#)

AT Command via UART1

RAK4630 module can be configured using AT commands via the UART1 interface. You need a USB to UART TTL adapter to connect the RAK4630 to your computer's USB port and a serial terminal tool. It is highly recommended to use the [RAK Serial Port Tool](#) so you can easily send AT commands and view the replies from the console output. You can configure the RAK4630 in two ways:

- [LoRaWAN End-Device](#) - RAK4630 as LoRaWAN IoT device.
- [LoRa P2P](#) - Point-to-point communication between two RAK4630 modules.

Connect to the RAK4630

1. Connect the RAK4630 to the serial port of a general-purpose computer (USB port) using a USB to UART TTL adapter like [RAKDAP1](#) , as shown in **Figure 37**.

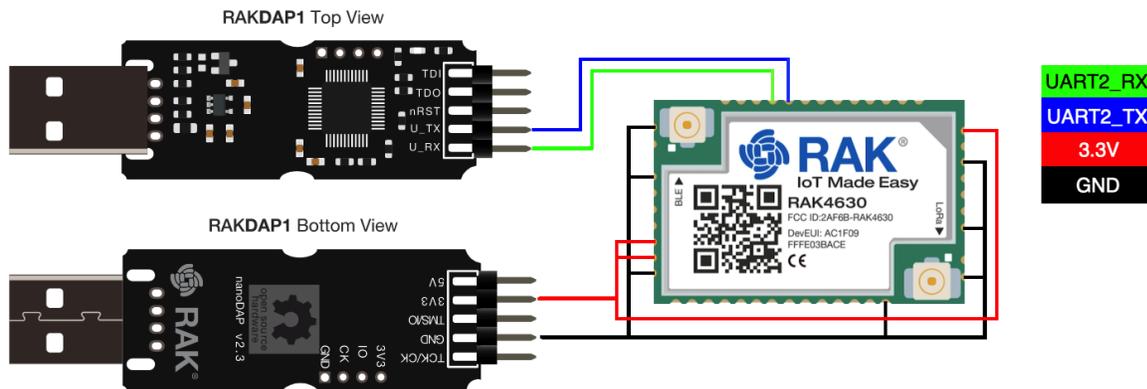


Figure 37: RAK4630 module connection

NOTE:

You can also send AT commands via direct USB connection to the module, as shown in [hardware setup](#)

2. Any serial communication tool can be used, but it is recommended to use the [RAK Serial Port Tool](#) .
3. Configure the serial communication tool by selecting the proper port detected by the computer and configure the link as follows:
 - Baud Rate: **115200 baud**
 - Data Bits: **8 bits**
 - Stop Bits: **1 stop bit**
 - Parity: **NONE**

Configuring RAK4630 as LoRaWAN End-device

To enable the RAK4630 module as a LoRaWAN end-device, a device must be registered first in the LoRaWAN network server. This guide will cover both TTN and Chirpstack LoRaWAN network servers and the associate AT commands for the RAK4630.

This guide covers the following topics:

- [TheThingsNetwork Guide](#) - How to login, register new accounts, and create new applications on TTN.
- [RAK4630 TTN OTAA Guide](#) - How to add OTAA device on TTN and what AT commands to use on RAK4630 OTAA activation.
- [RAK4630 TTN ABP Guide](#) - How to add ABP device on TTN and what AT commands to use on RAK4630 ABP activation.
- [Chirpstack Guide](#) - How to create new applications on Chirpstack.
- [RAK4630 Chirpstack OTAA Guide](#) - How to add OTAA device to Chirpstack and what AT commands to use on RAK4630 OTAA activation.
- [RAK4630 Chirpstack ABP Guide](#) - How to add ABP device on Chirpstack and what AT commands to use on RAK4630 ABP activation.

Connecting to The Things Network (TTN)

In this section, it shows how to connect the RAK4630 module to the TTN platform.

NOTE:

You need to have a working gateway that is connected to TTN, or you have to be within the coverage of a TTN community network.

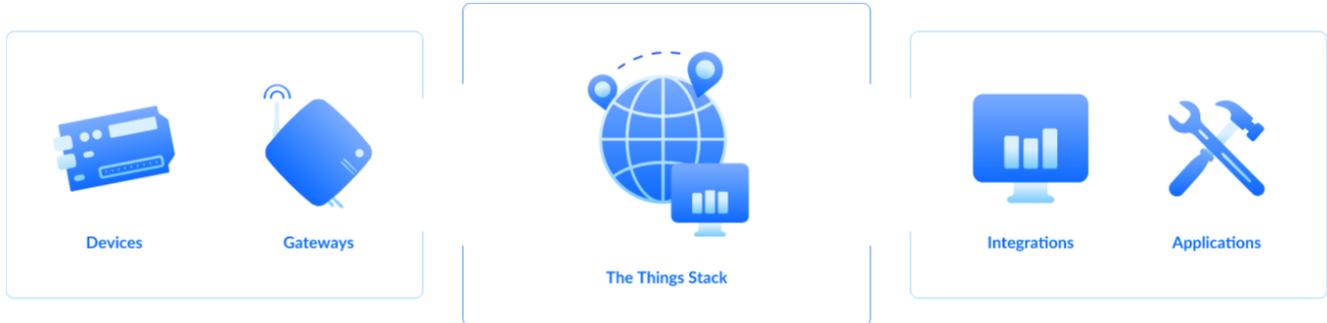


Figure 38: The Things Stack

As shown in **Figure 38**, The Things Stack (TTN V3) is an open-source LoRaWAN Network Server suitable for global, geo-distributed public and private deployments, as well as for small local networks. The architecture follows the LoRaWAN Network Reference Model for standards compliance and interoperability. This project is actively maintained by [The Things Industries](#) .

LoRaWAN is a protocol for low-power wide-area networks. It allows for large-scale Internet of Things deployments where low-powered devices efficiently communicate with Internet-connected applications over long-range wireless connections.

The RAK4630 WisDuo module can be part of this ecosystem as a device, and the objective of this section is to demonstrate how simple it is to send data to The Things Stack using the LoRaWAN protocol. To achieve this, the RAK4630 WisDuo module must be located inside the coverage of a LoRaWAN gateway connected to The Things Stack server.

Registration to TTN and Creating LoRaWAN Applications

1. Go to [The Things Network platform](#) and select a cluster, as shown in **Figure 39**.

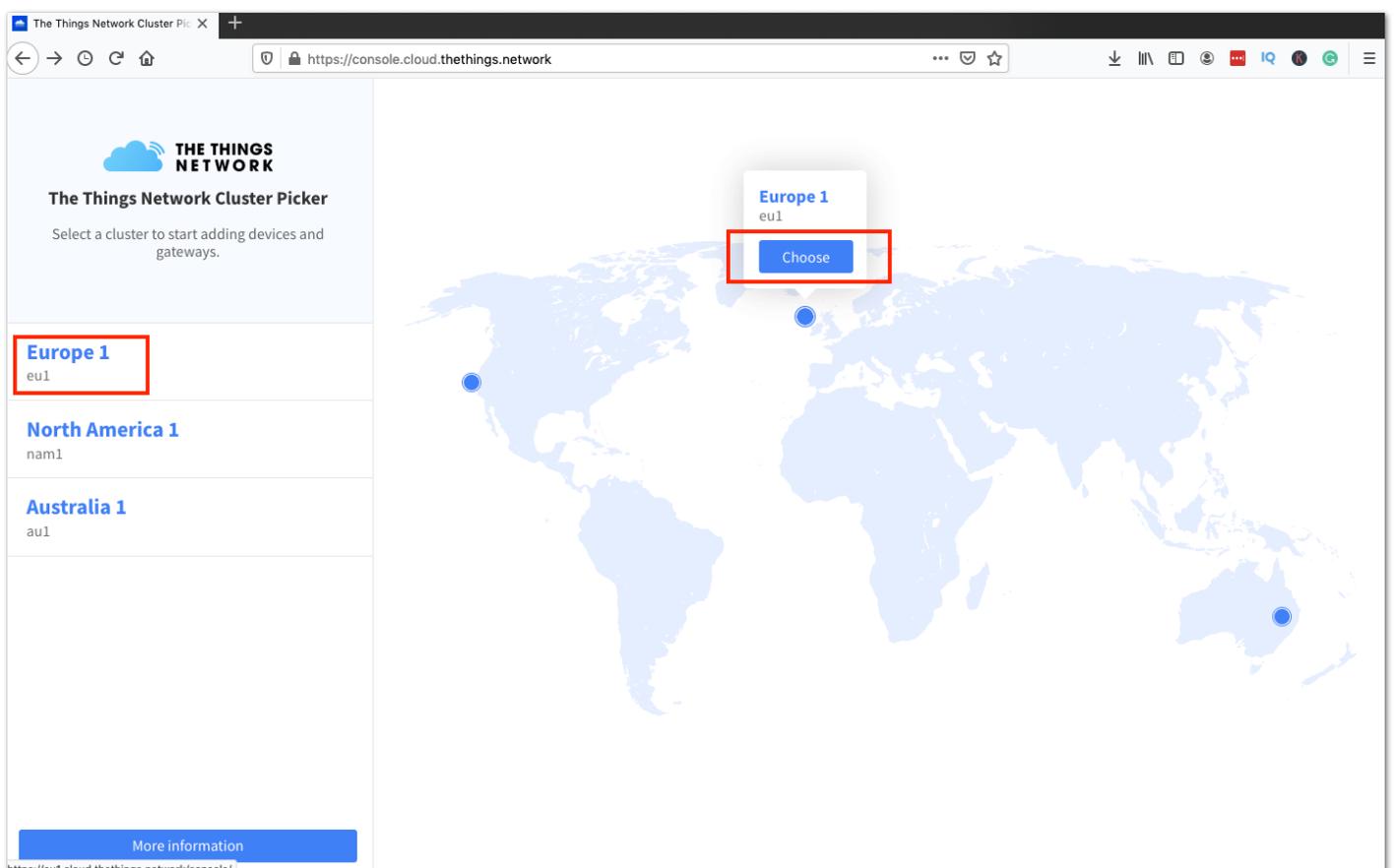


Figure 39: Selecting cluster in TTN V3

You can use the same login credentials on the TTN V2 if you have one. If you have no account yet, you need to create one.

2. To register as a new user to TTN, click on **Login with The Things ID** then select **register** on the next page, as shown in **Figure 40** and **Figure 41**.

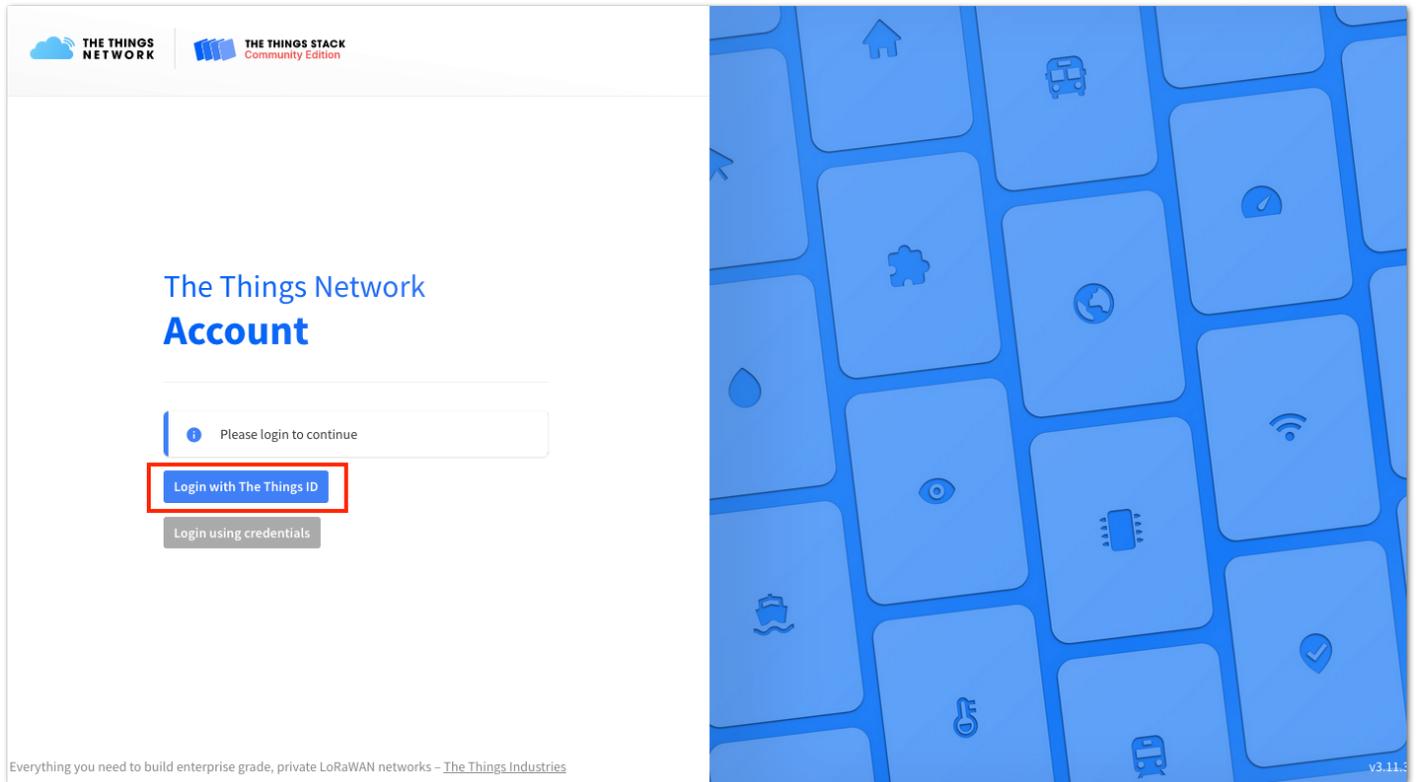


Figure 40: Login using TTN account

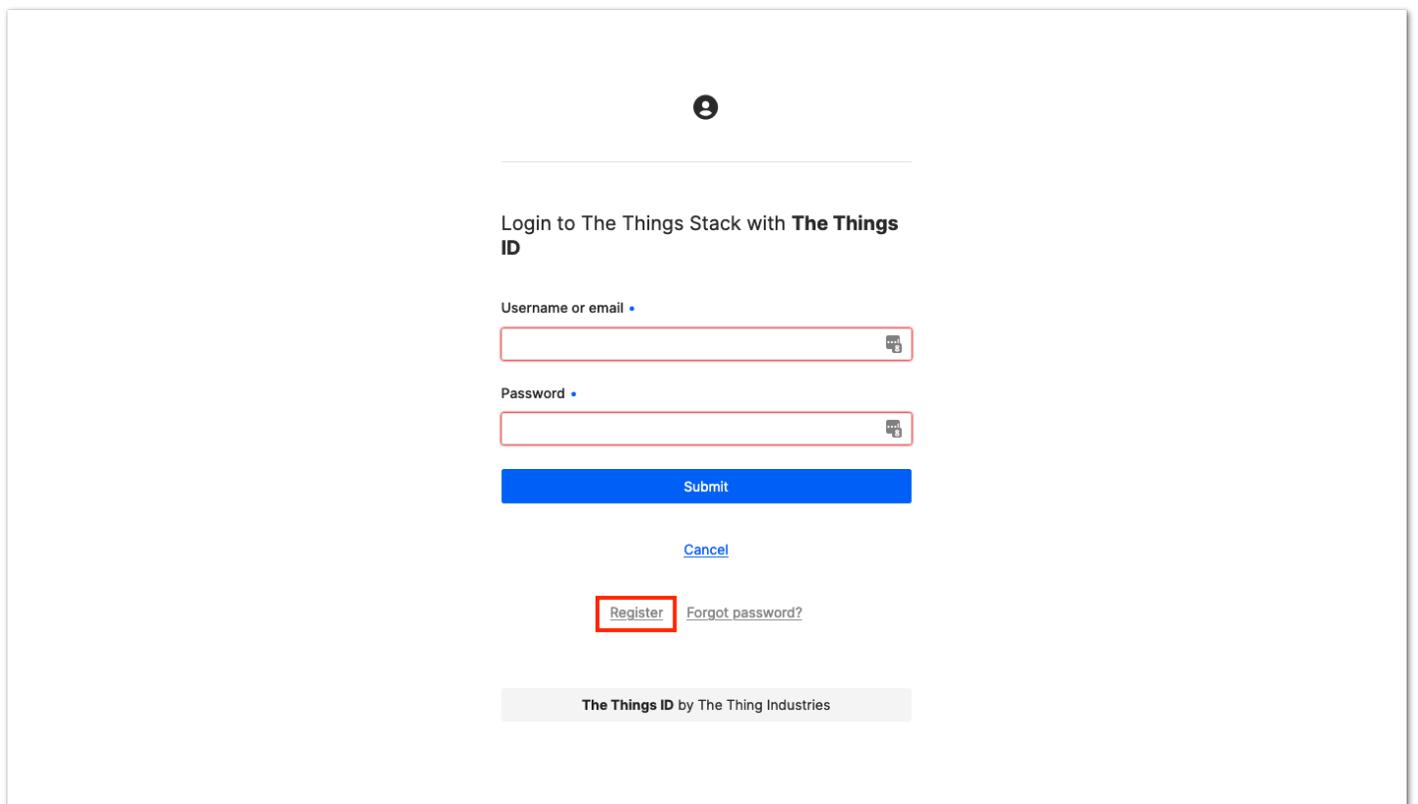


Figure 41: Registration of new account

3. You should now be on the step of creating your TTN account. Fill in all the necessary details and activate your account.

4. After creating an account, log in on the platform using your username/email and password then click **Submit**, as shown in **Figure 42**.

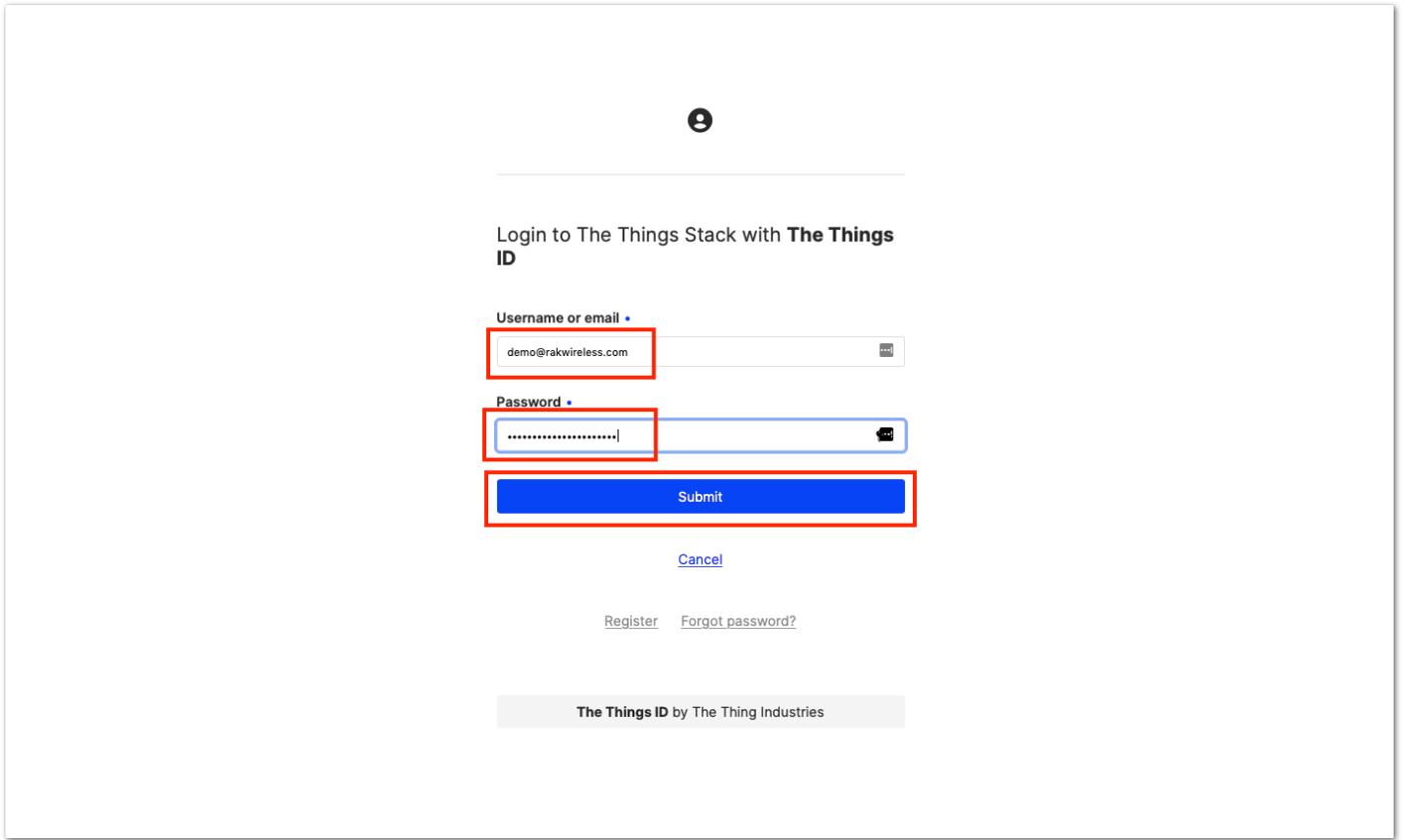


Figure 42: Logging in to TTN platform

5. Click **Authorize** to proceed.

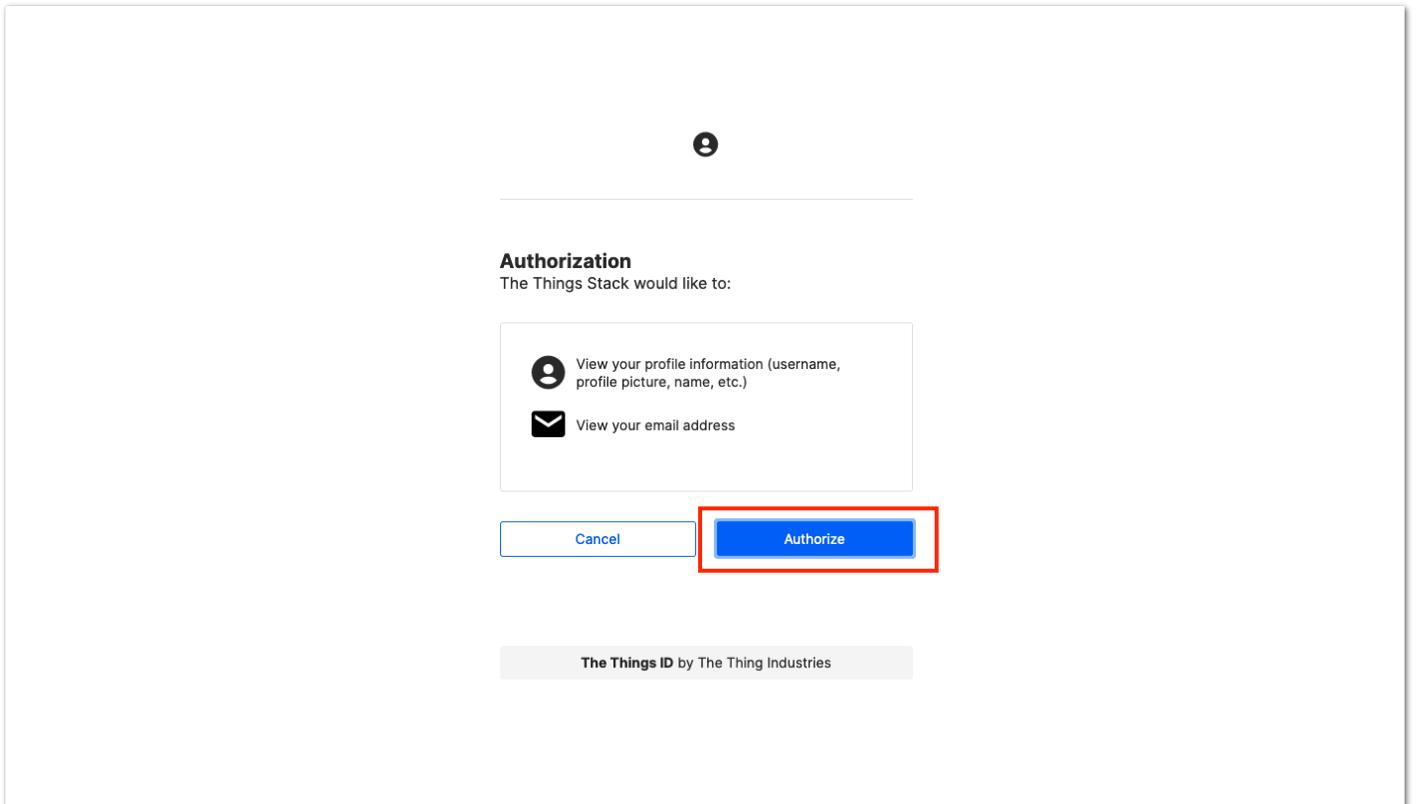


Figure 43: Authorization to TTN

6. Now that you are logged in to the platform, the next step is to create an application. Click **Create an application**.

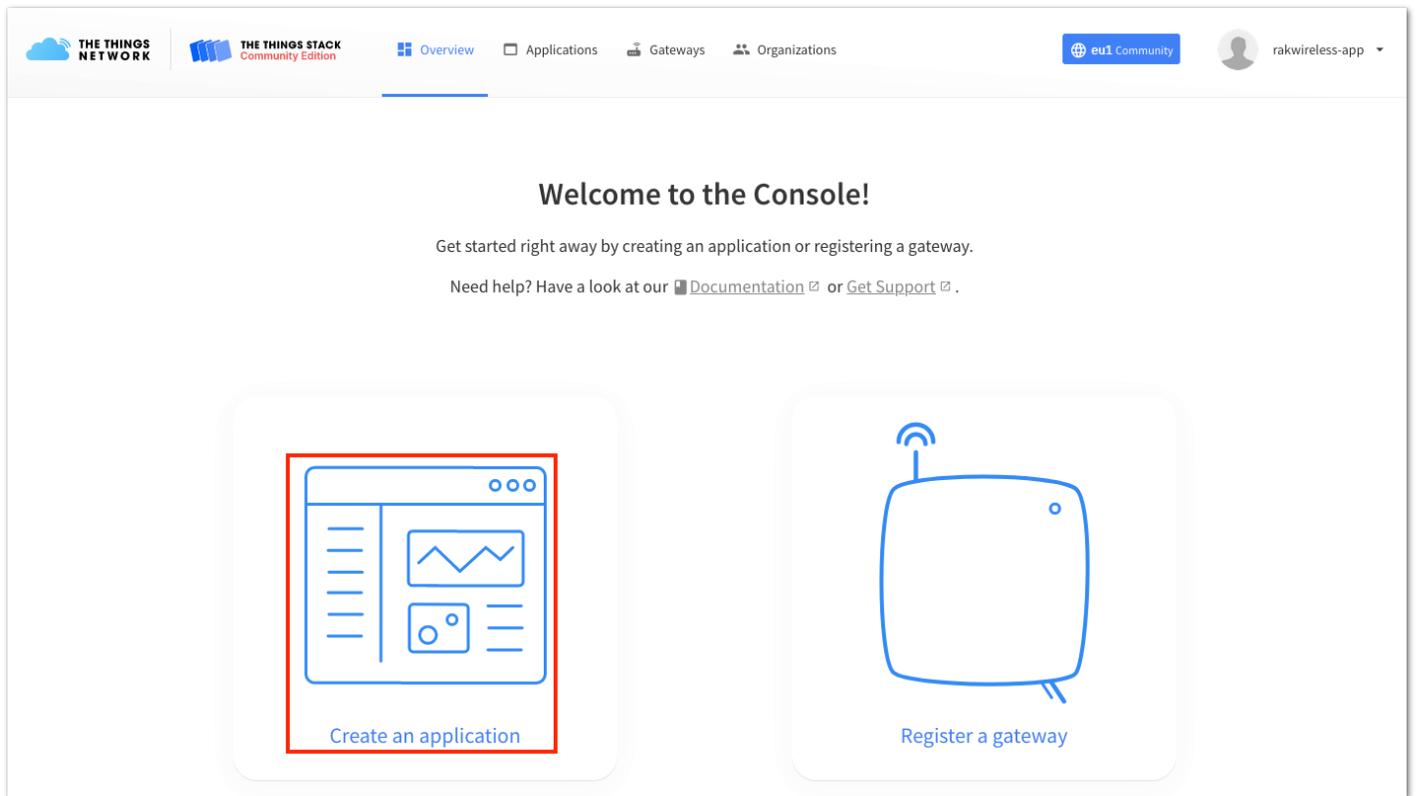


Figure 44: Creating TTN application for your LoRaWAN devices

7. To have an application registered, input first the specific details and necessary information about your application then click **Create application**.

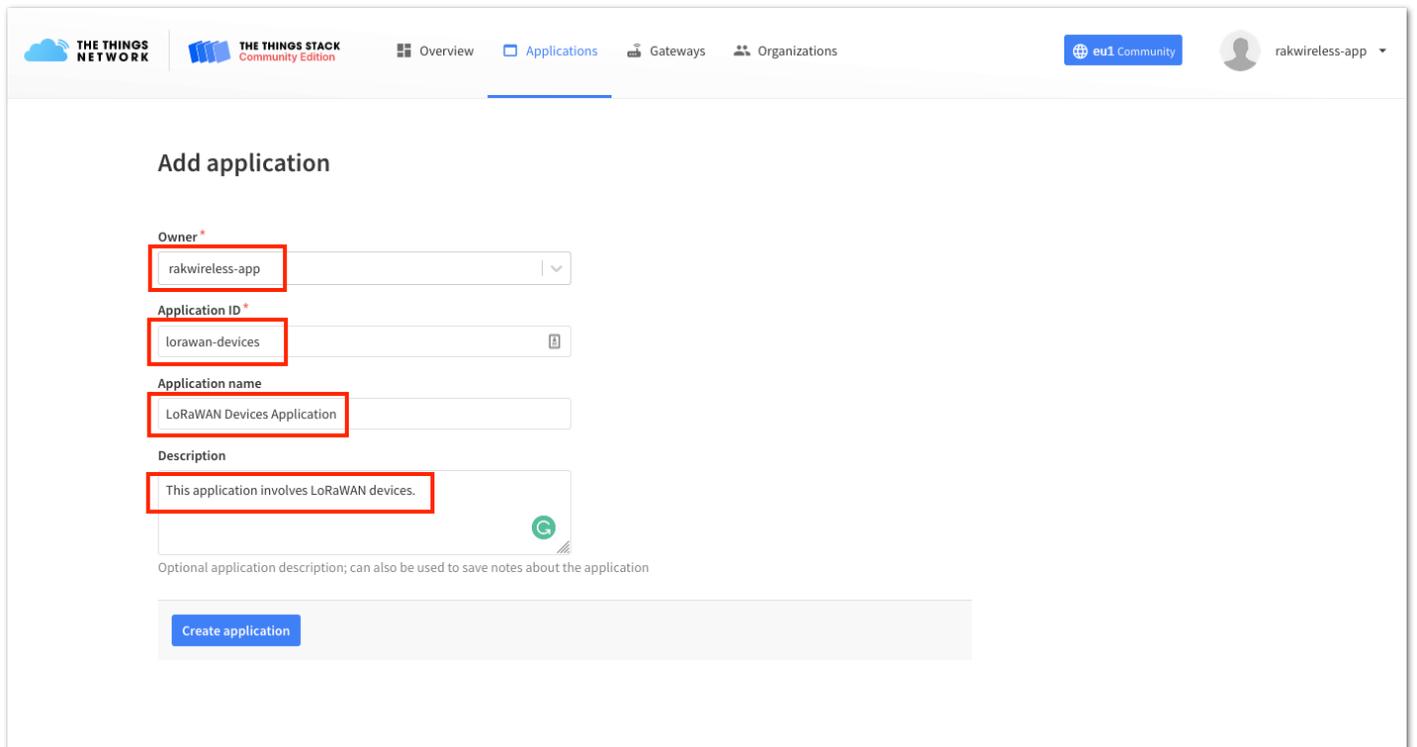


Figure 45: Details of the TTN application

8. If you had no error during the previous step, you should now be on the application console page. The next step is to add end-devices to your TTN application. LoRaWAN specification enforces that each end-device has to be personalized and activated. There are two options for registering devices depending on the activation mode you select. Activation can be done either via Over-The-Air-Activation (OTAA) or Activation-By-Personalization (ABP).

TTN OTAA Device Registration

1. Go to your application console to be able to register a device. To start adding an OTAA end-device, click **+ Add end device**, as shown in Figure 46.

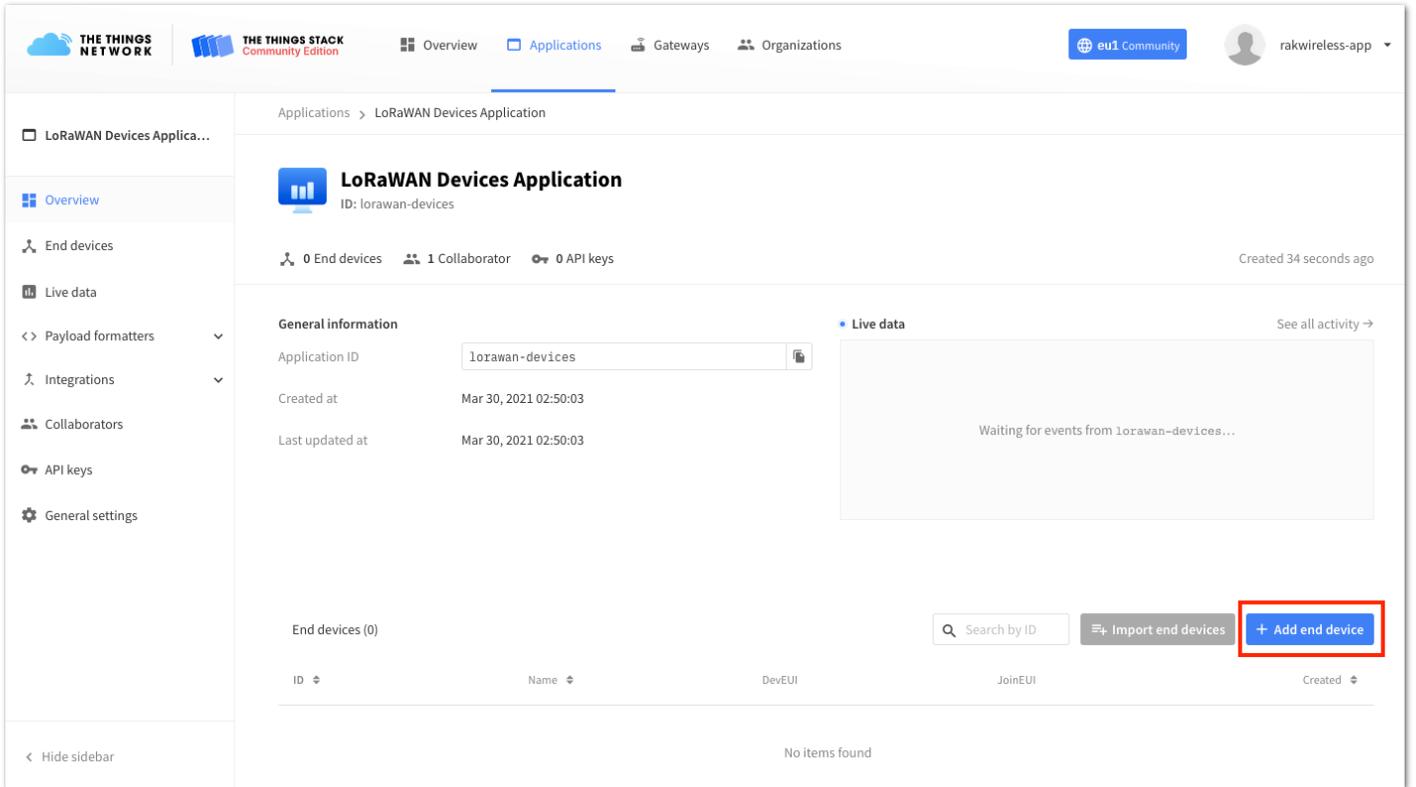


Figure 46: Add end device

2. To register the module, click first **Manually** then configure the activation method by selecting **Over the air activation (OTAA)** and compatible **LoRaWAN version** then click the **Start** button, as shown in **Figure 47** and **Figure 48**.

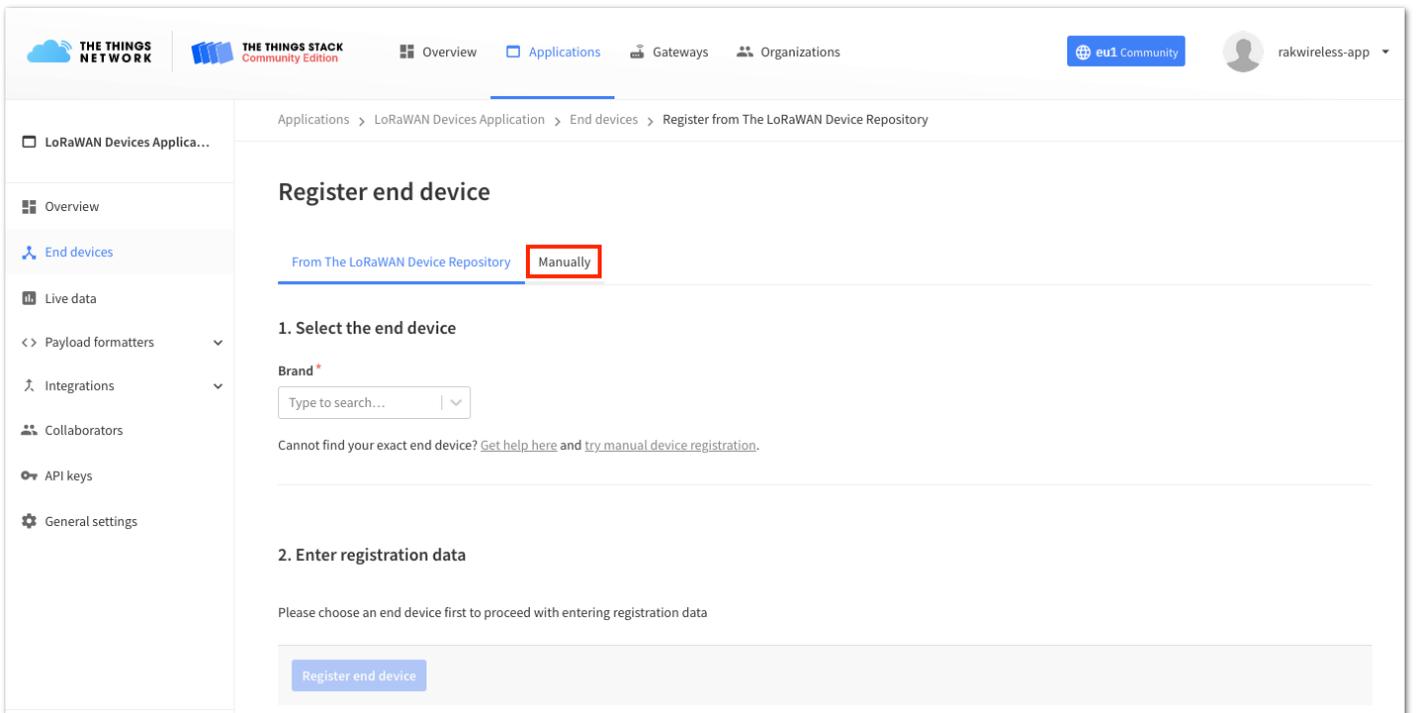


Figure 47: Manually register device to TTN

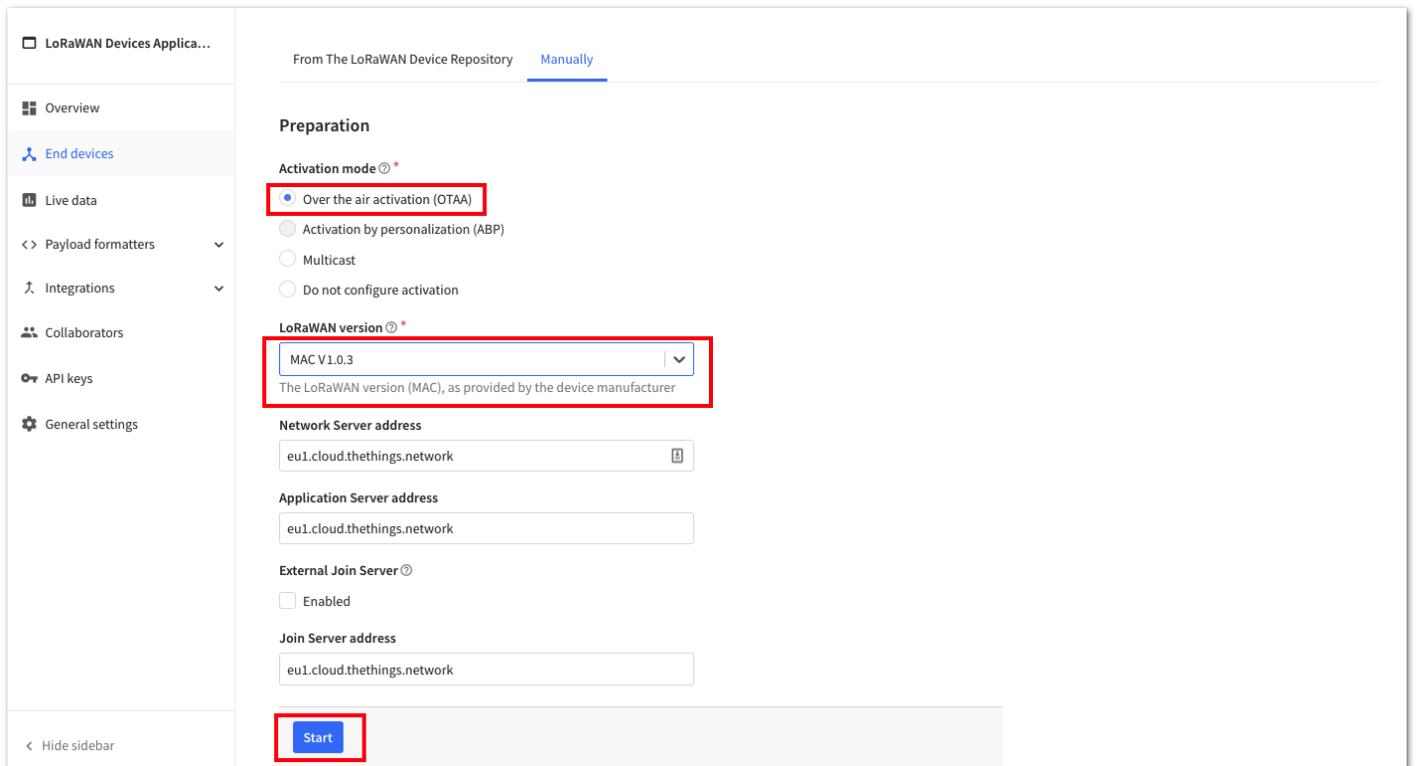


Figure 48: Device activation configuration

3. Then you need to put a unique **End device ID** and EUIs (**DevEUI** and **AppEUI**), as shown in **Figure 49**. Check if your module has a DevEUI on the sticker or QR that you can scan, then use this as the device unique DevEUI.

Optionally, you can add a more descriptive **End device name** and **End device description** about your device.

4. After putting all the details, click **Network layer settings** to proceed to the next step.

NOTE:

It is advisable to use a meaningful end-device ID, end-device name, and end-device description that will match your device purpose. The end-device ID `rak-device` is for illustration purposes only.

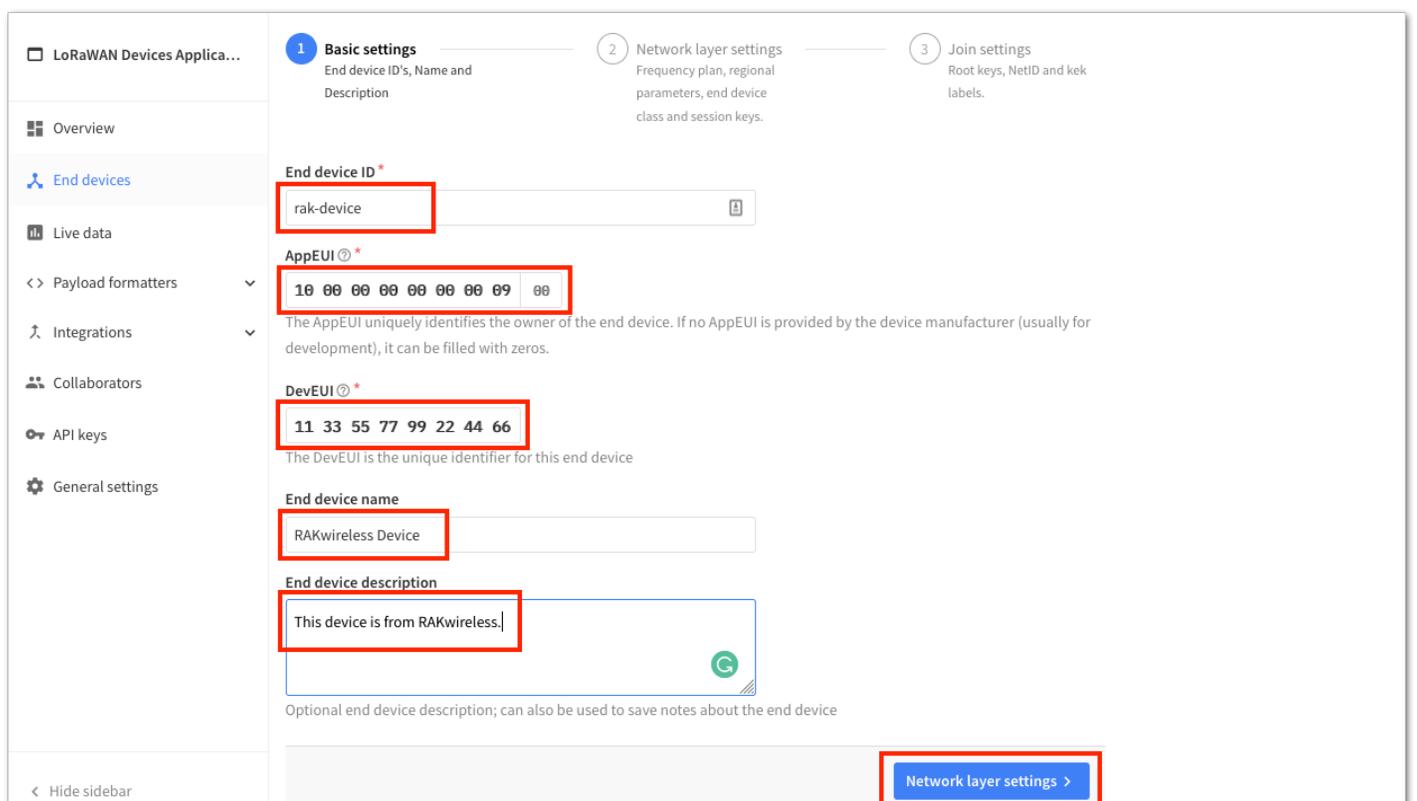


Figure 49: OTAA device information

5. Set up the **Frequency plan**, a compatible **Regional Parameter version**, and the **LoRaWAN class** supported. Then click **Join settings**.

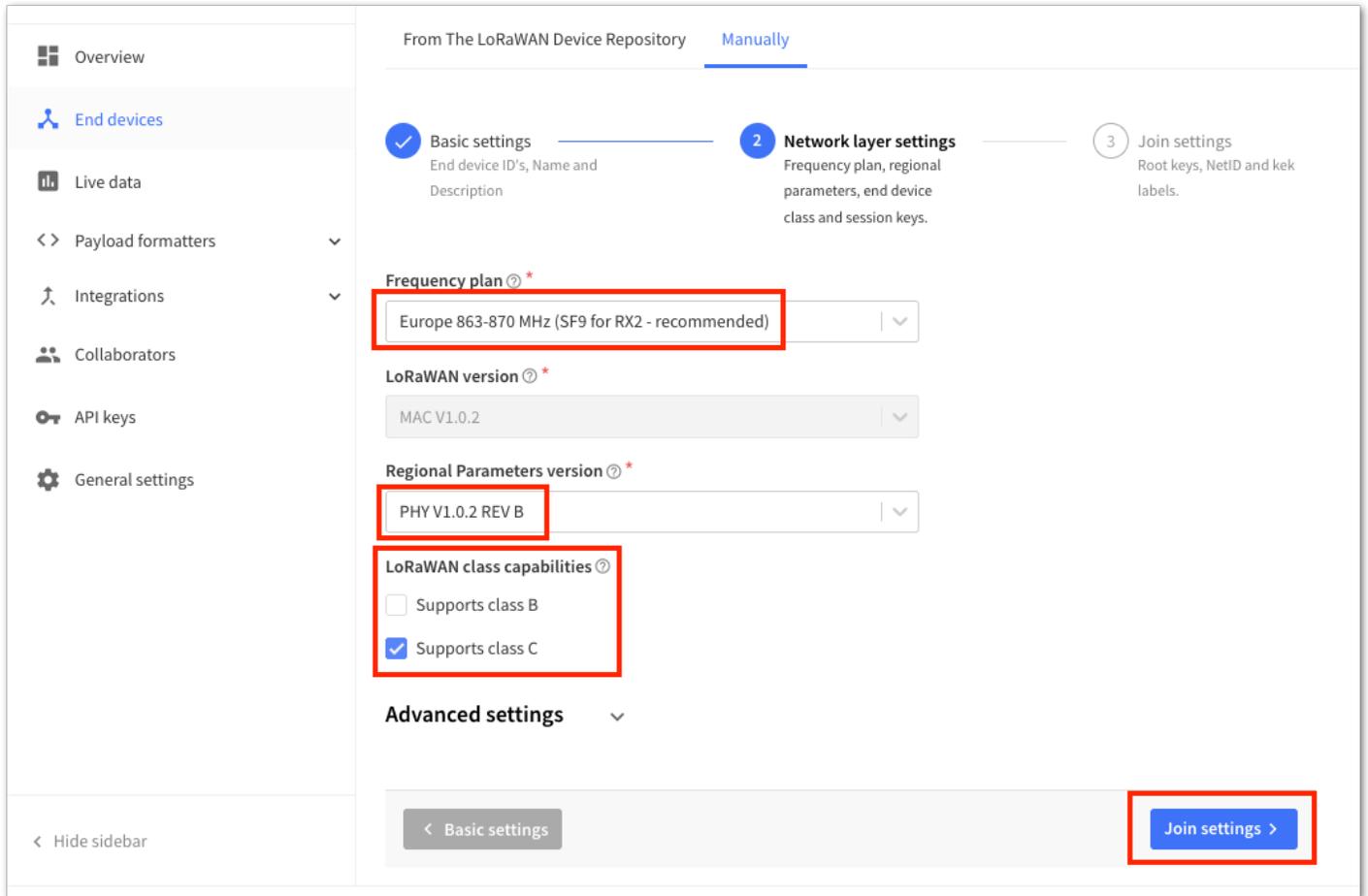


Figure 50: OTAA configuration

6. The last step in the registration of a new OTAA end-device is the configuration of the **AppKey**. To get the AppKey, click the **generate** button, then the **Add end device** to finish your new device registration.

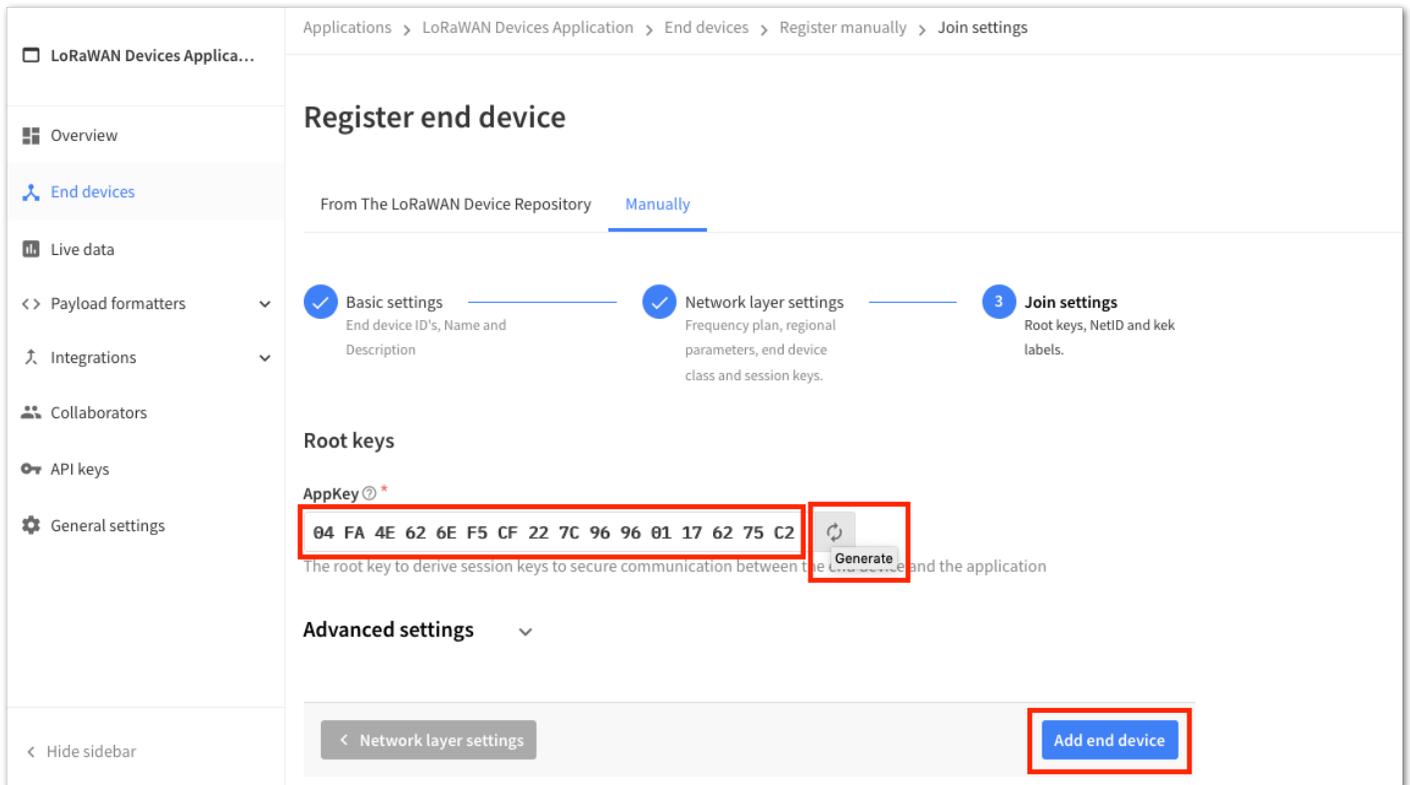


Figure 51: OTAA AppKey generation and device registration

You should now be able to see the device on the TTN console after you fully register your device, as shown in **Figure 52**.

NOTE:

The **AppEUI**, **DevEUI**, and **AppKey** are the parameters that you will need to activate your LoRaWAN end-device via OTAA. The **AppKey** is hidden by default for security reasons, but you can easily show it by clicking the show button. You can also copy the parameters quickly using the copy button.

The three OTAA parameters on the TTN device console are MSB by default.

These parameters are always accessible on the device console page, as shown in **Figure 52**.

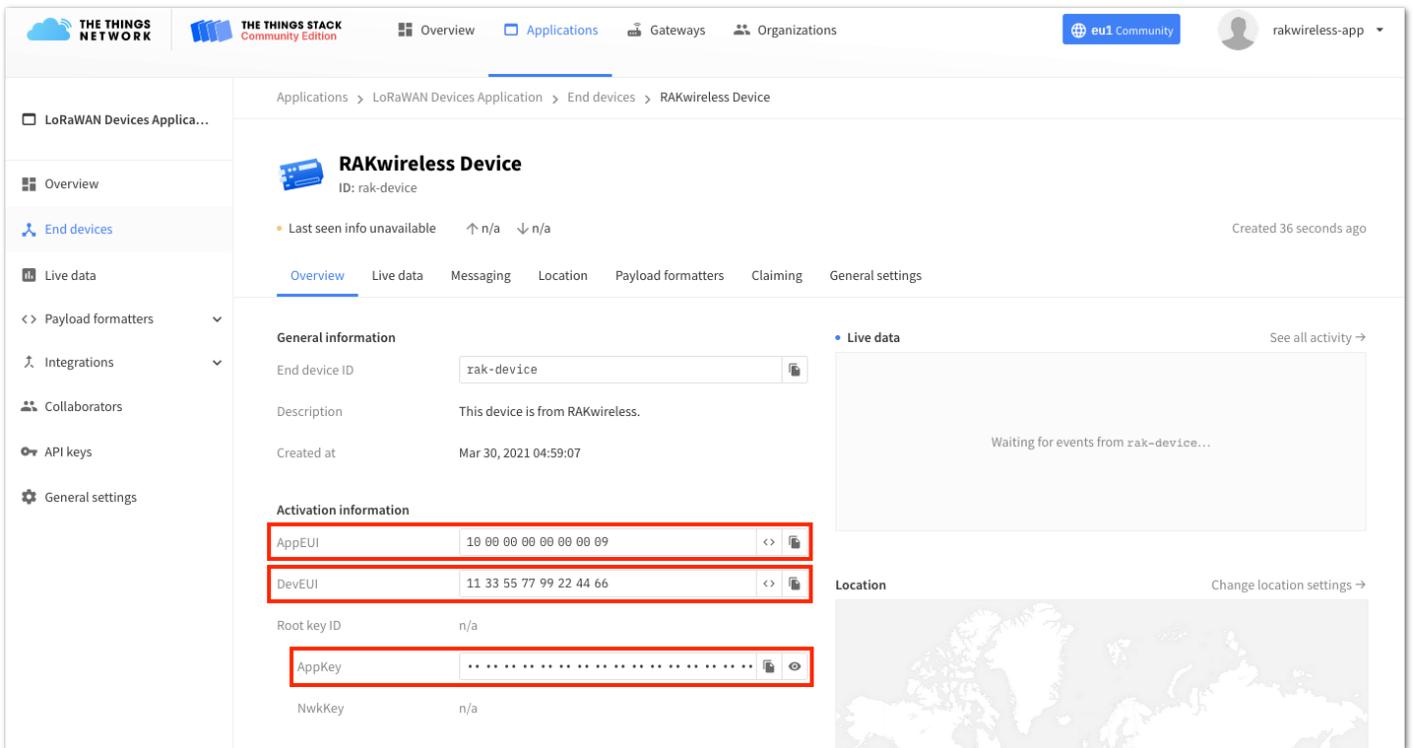


Figure 52: OTAA device successfully registered to TTN

OTAA Configuration for TTN

The RAK4630 module supports a series of AT commands to configure its internal parameters and control the functionalities of the module.

1. To set up the RAK4630 module to join the TTN using OTAA, start by connecting the RAK4630 module to your computer (see [Figure 37](#)) and open the RAK Serial Port Tool. Select the right COM port and set the baud rate to 115200.

It is recommended to start by testing the serial communication and verify that the current configuration is working by sending these two AT commands:

```
AT
```

```
ATE
```

`ATE` will echo the commands you input to the module, which is useful for tracking the commands and troubleshooting.

You will receive `OK` when you input the two commands. After setting `ATE`, you can now see all the commands you input together with the replies. Try again `AT` and you should see it on the terminal followed by `OK`, as shown in **Figure 53**.

NOTE:

If you haven't received an `OK` or any reply, you need to check if you selected the right port. You can also look at the wiring of your UART lines and if the device is powered correctly. If you are getting power from a USB port, ensure that you have a good USB cable.

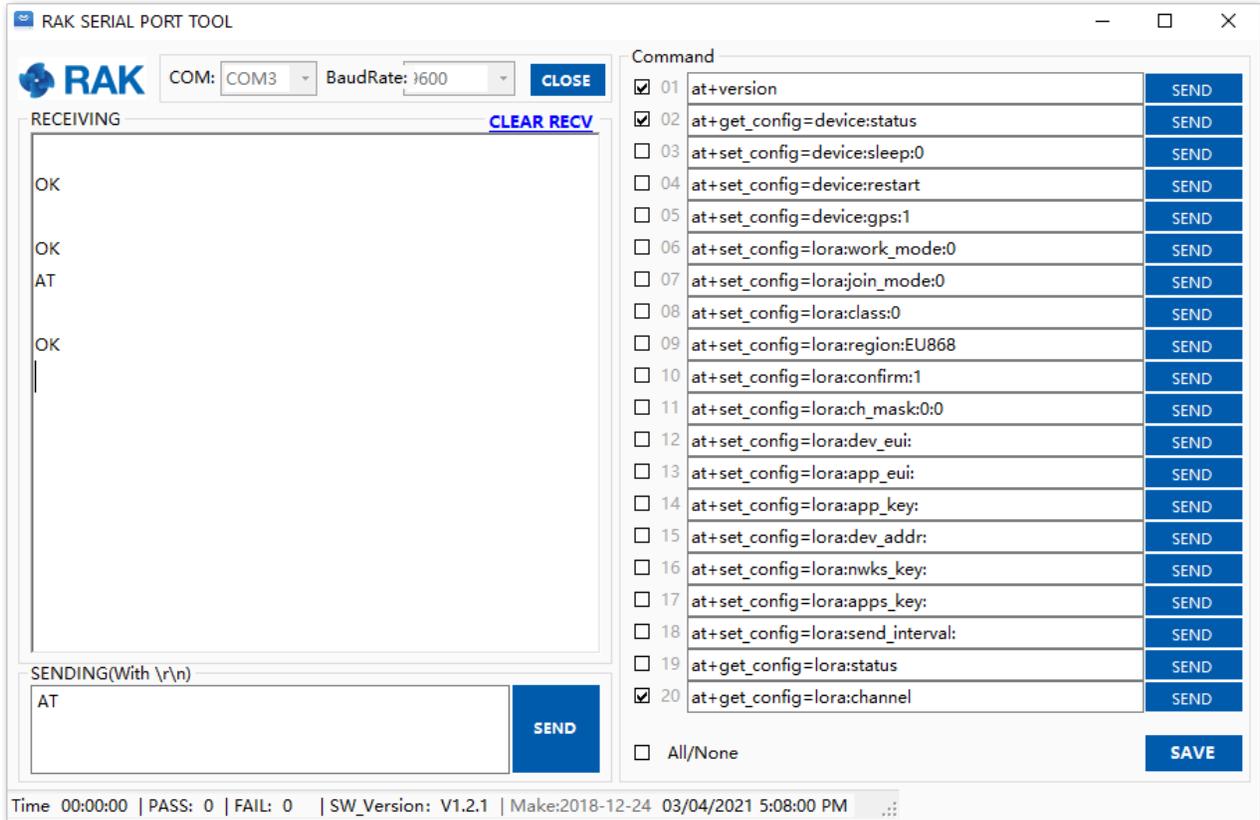


Figure 53: at+version command response

2. The next step is to configure the OTAA LoRaWAN parameters in RAK4630:

- LoRa work mode: **LoRaWAN**
- LoRaWAN join mode: **OTAA**
- LoRaWAN class: **Class A**
- LoRaWAN region: **EU868**

Set the work mode to LoRaWAN. You might need to reconnect on the Serial Terminal Software Tool if you lost connection after using the `AT+NWM=1` command.

```
AT+NWM=1
```

Set the LoRaWAN activation to OTAA.

```
AT+NJM=1
```

Set the LoRaWAN class to Class A.

```
AT+CLASS=A
```

Set the frequency/region to EU868.

AT+BAND=4

 **NOTE:**

Depending on the Regional Band you selected, you might need to configure the sub-band of your RAK4630 to match the gateway and LoRaWAN network server. This is especially important for Regional Bands like US915, AU915, and CN470.

To configure the masking of channels for the sub-bands, you can use the `AT+MASK` command that can be found on the [AT Commands Manual](#) .

To illustrate, you can use sub-band 2 by sending the command `AT+MASK=0002`.

List of band parameter options

Code	Regional Band
0	EU433
1	CN470
2	RU864
3	IN865
4	EU868
5	US915
6	AU915
7	KR920
8	AS923

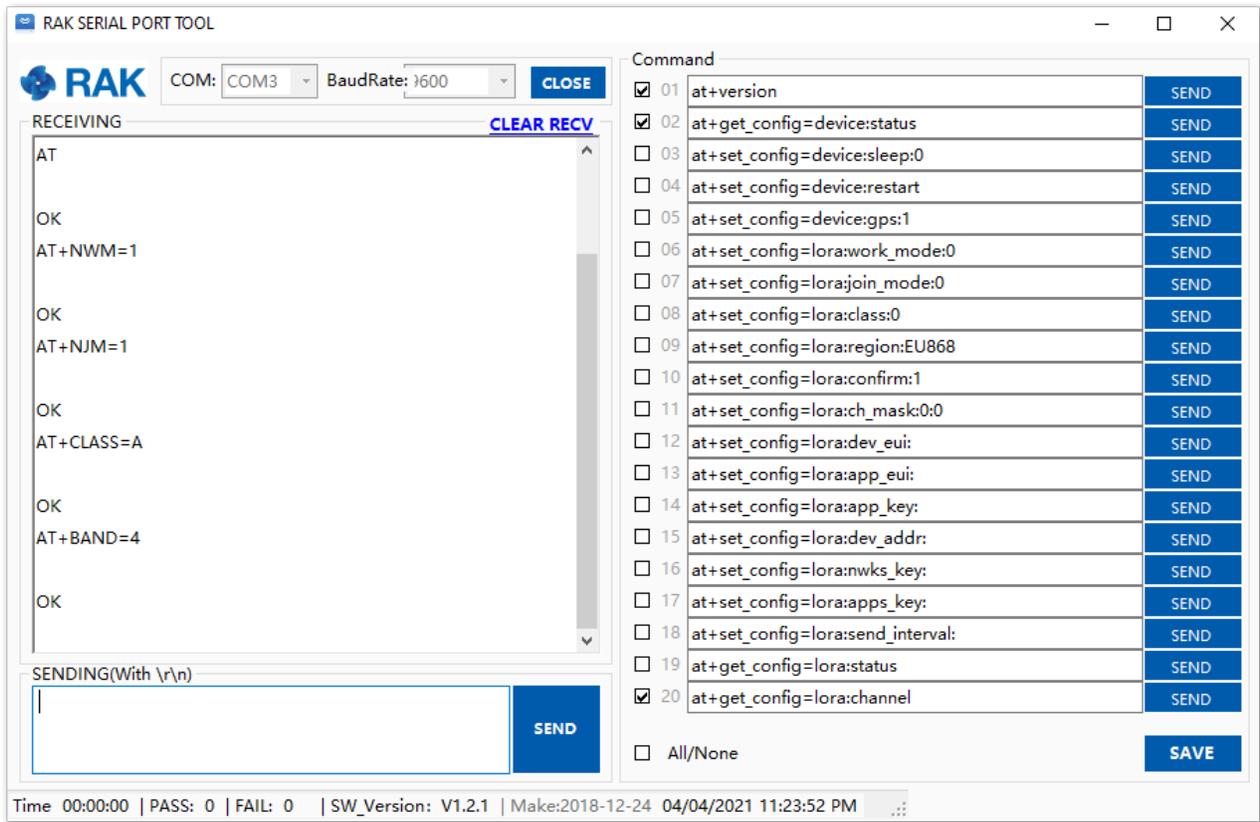


Figure 54: Configuring LoRa Parameters

3. After the configuration of the LoRaWAN parameters, the next step is to set up the EUIs and key. You need the use the values from the TTN console.

- Device EUI: **1133557799224466**
- Application EUI: **1000000000000009**
- Application Key: **04FA4E626EF5CF227C969601176275C2**

Set the Device EUI.

```
AT+DEVEUI=1133557799224466
```

Set the Application EUI.

```
AT+APPEUI=1000000000000009
```

Set the Application Key.

```
AT+APPKEY=04FA4E626EF5CF227C969601176275C2
```

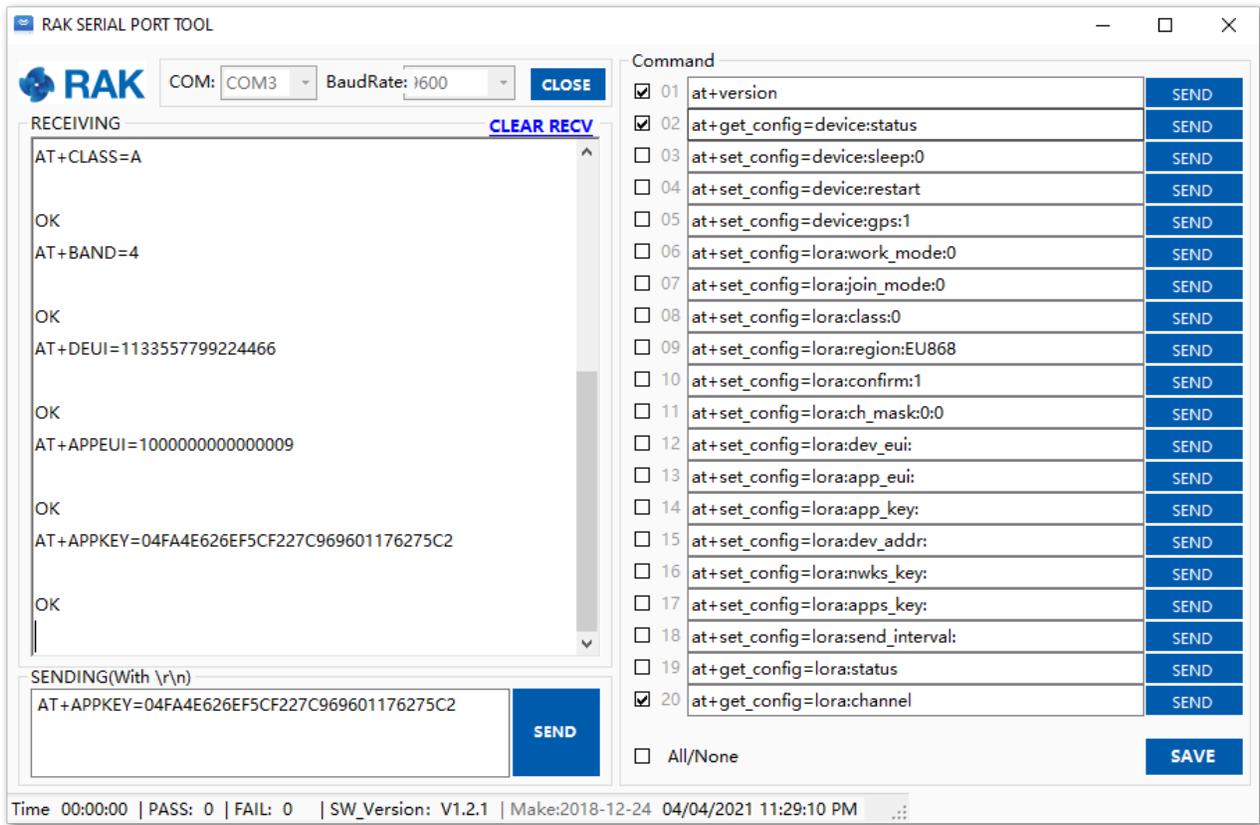


Figure 55: Configuring LoRa parameters

4. After EUI and keys configuration, the device can now join the network and send payloads.

```
AT+JOIN=1:0:10:8
```

Join command format: `AT+JOIN=w:x:y:z`

Parameter	Description
w	Join command - 1: joining, 0: stop joining.
x	Auto-join config - 1: auto-join on power-up, 0: no auto-join
y	Reattempt interval in seconds (7-255) - 8 is the default.
z	Number of join attempts (0-255) - 0 is default.

After 5 or 6 seconds, if the request is successfully received by a LoRa gateway, you should see `+EVT: JOINED` status reply, as shown in **Figure 56**.

NOTE:

If the OTAA device failed to join, you need to check if your device is within reach of a working LoRaWAN gateway that is configured to connect to TTN. It is also important to check that all your OTAA parameters (DEVEUI, APPEUI, and APPKEY) are correct using the `AT+DEVEUI=?`, `AT+APPEUI=?`, and `AT+APPKEY=?` commands. Lastly, ensure that the antenna of your device is properly connected.

After checking all the things above, try to join again.

5. With the end-device properly activated, you can now try to send some payload after a successful join.

AT+SEND=2:12345678

6. Send command format: `AT+SEND=<port>:<payload>`

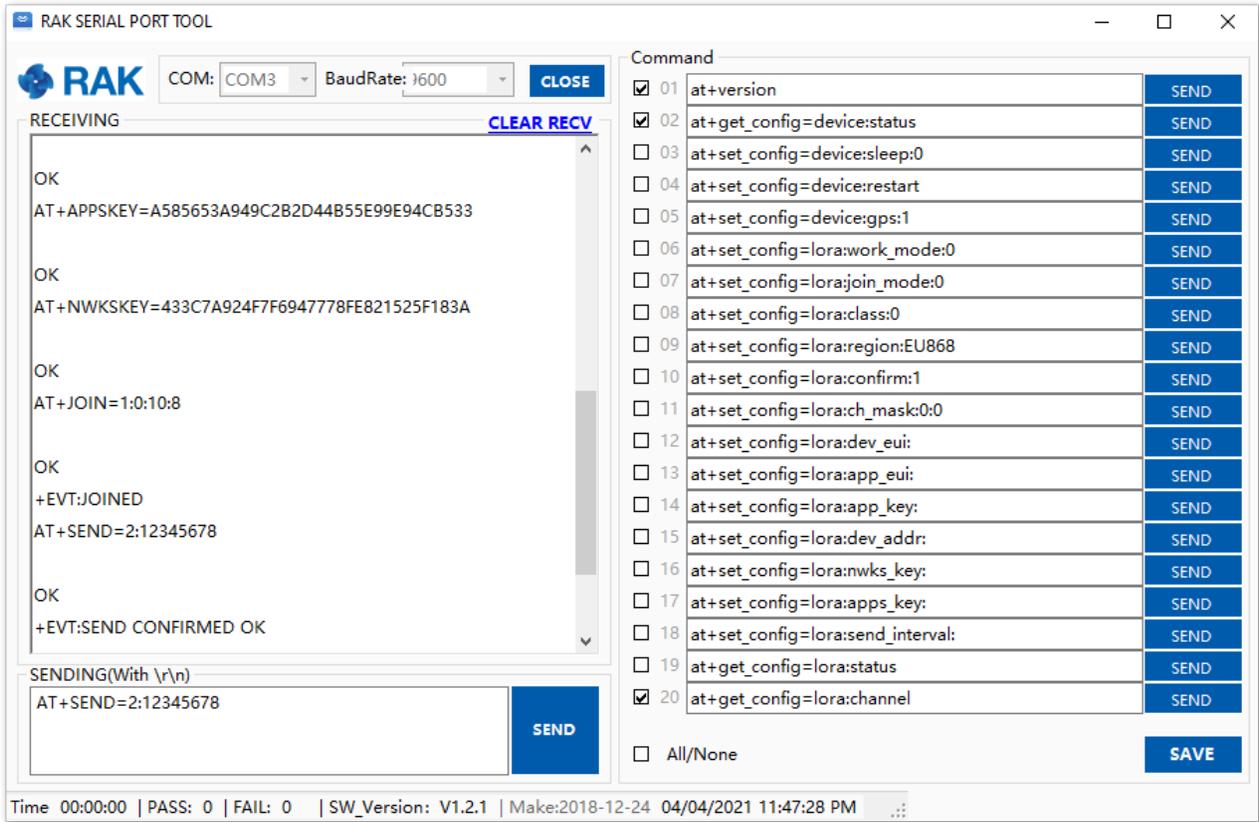


Figure 56: OTAA test sample data sent via RAK Serial Port Tool

7. You can see the data sent by the RAK4630 module on the TTN device console *Live data* section. Also, the *Last seen* info should be a few seconds or minutes ago.

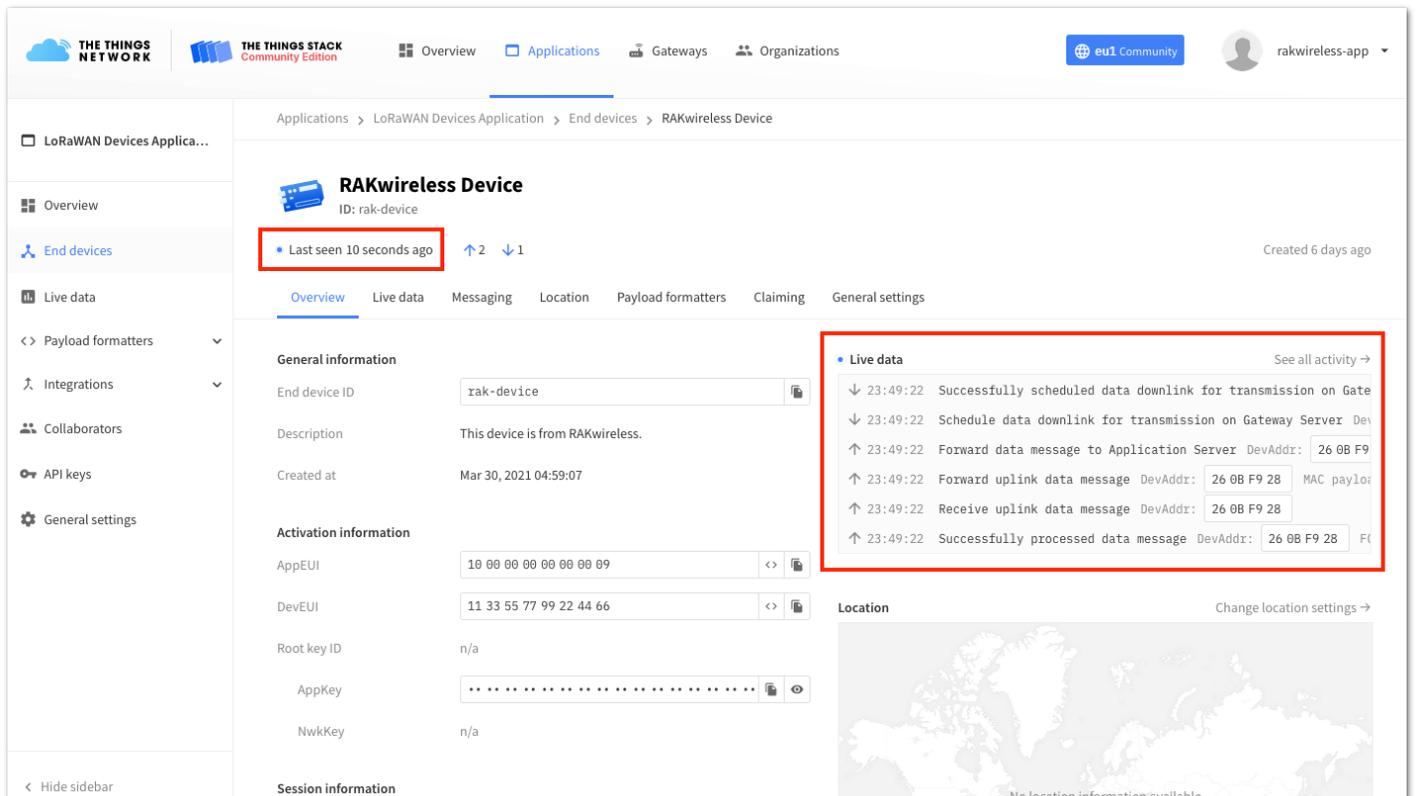


Figure 57: OTAA test sample data sent viewed in TTN

TTN ABP Device Registration

1. To register an ABP device, go to your application console and select the application where you want your device to be added. Then click **+ Add end device**, as shown in **Figure 58**.

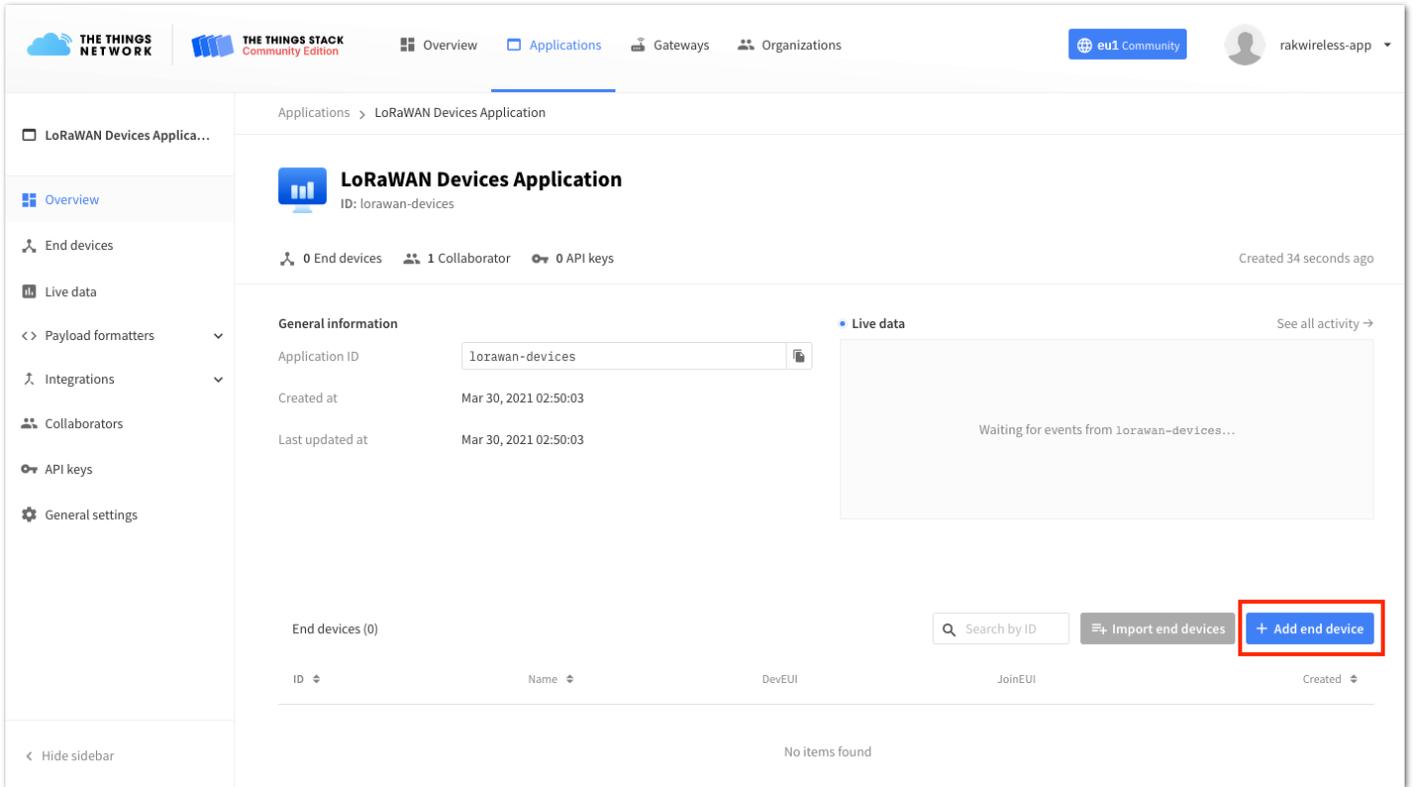


Figure 58: Adding ABP device

2. To register the module, you need to click first **Manually** then configure the activation method by selecting **Activation by personalization (ABP)**, compatible **LoRaWAN version**, and click the **Start** button, as shown in **Figure 59** and **Figure 60**.

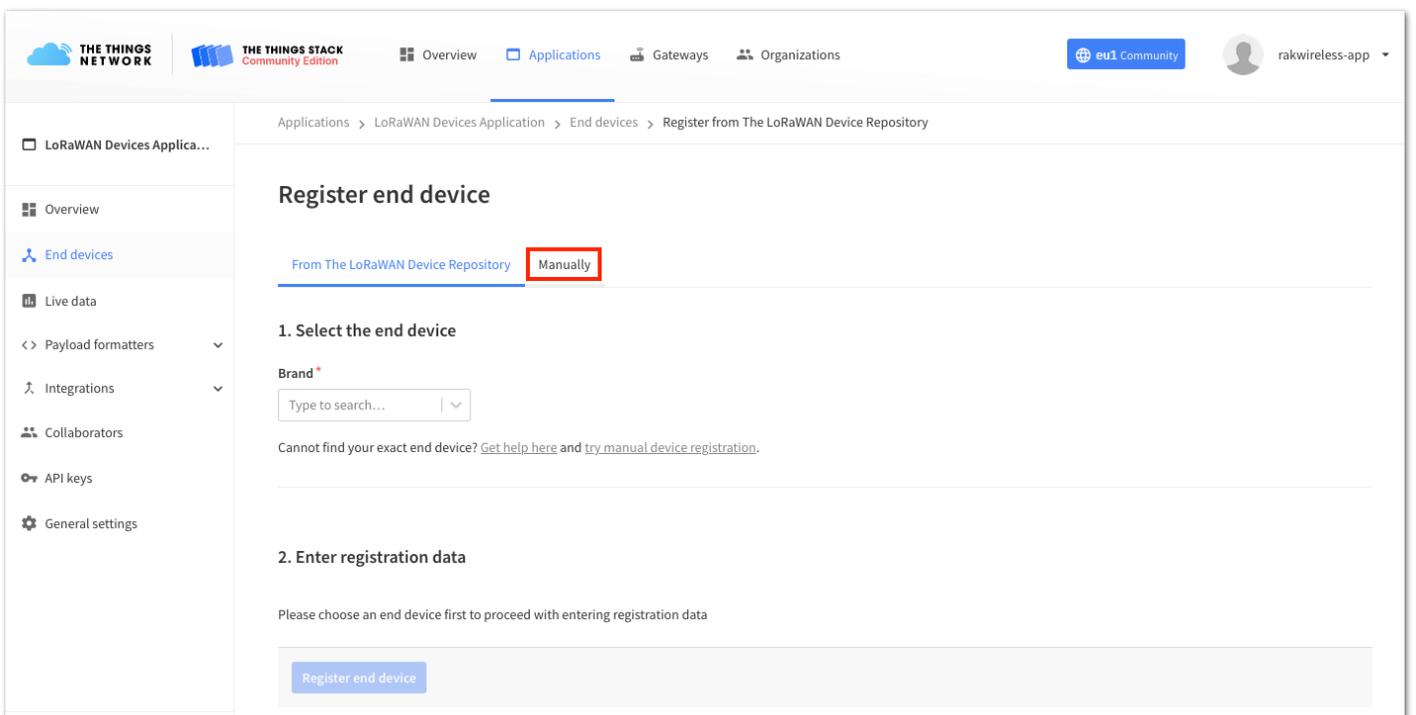


Figure 59: Manually register device to TTN

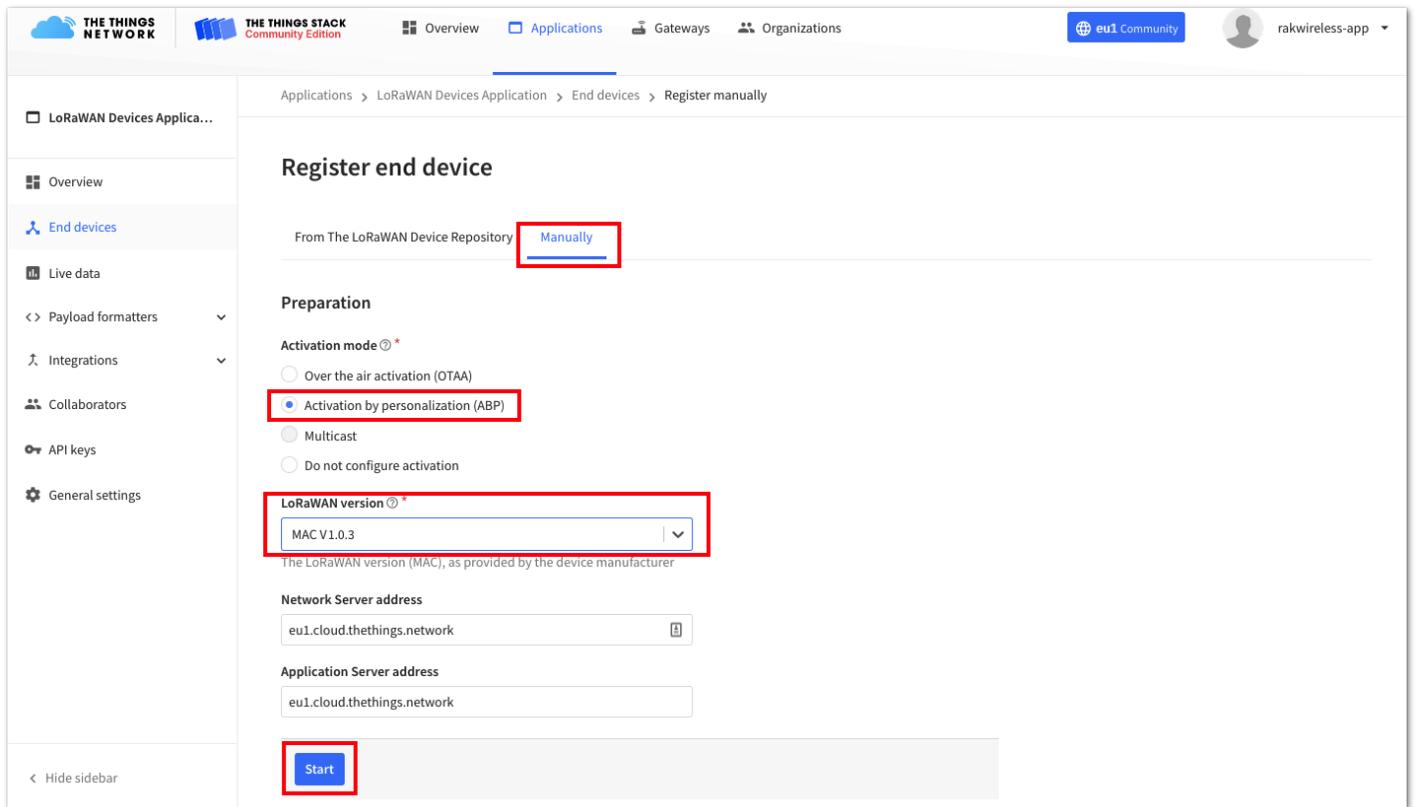


Figure 60: Selecting ABP and LoRaWAN version

3. At this step, you need to put a unique **End device ID** and **DevEUI**, as shown in **Figure 61**. Check if your module has a DevEUI on the sticker or QR that you can scan then use this as the device unique DevEUI.

Optionally, you can add a more descriptive **End device name** and **End device description** about your device.

4. After putting all the details, click **Network layer settings** to proceed to the next step.

NOTE:

It is advisable to use a meaningful end-device ID, end-device name, and end-device description that will match your device purpose. The end-device ID `rak-device-abp` is for illustration purposes only.

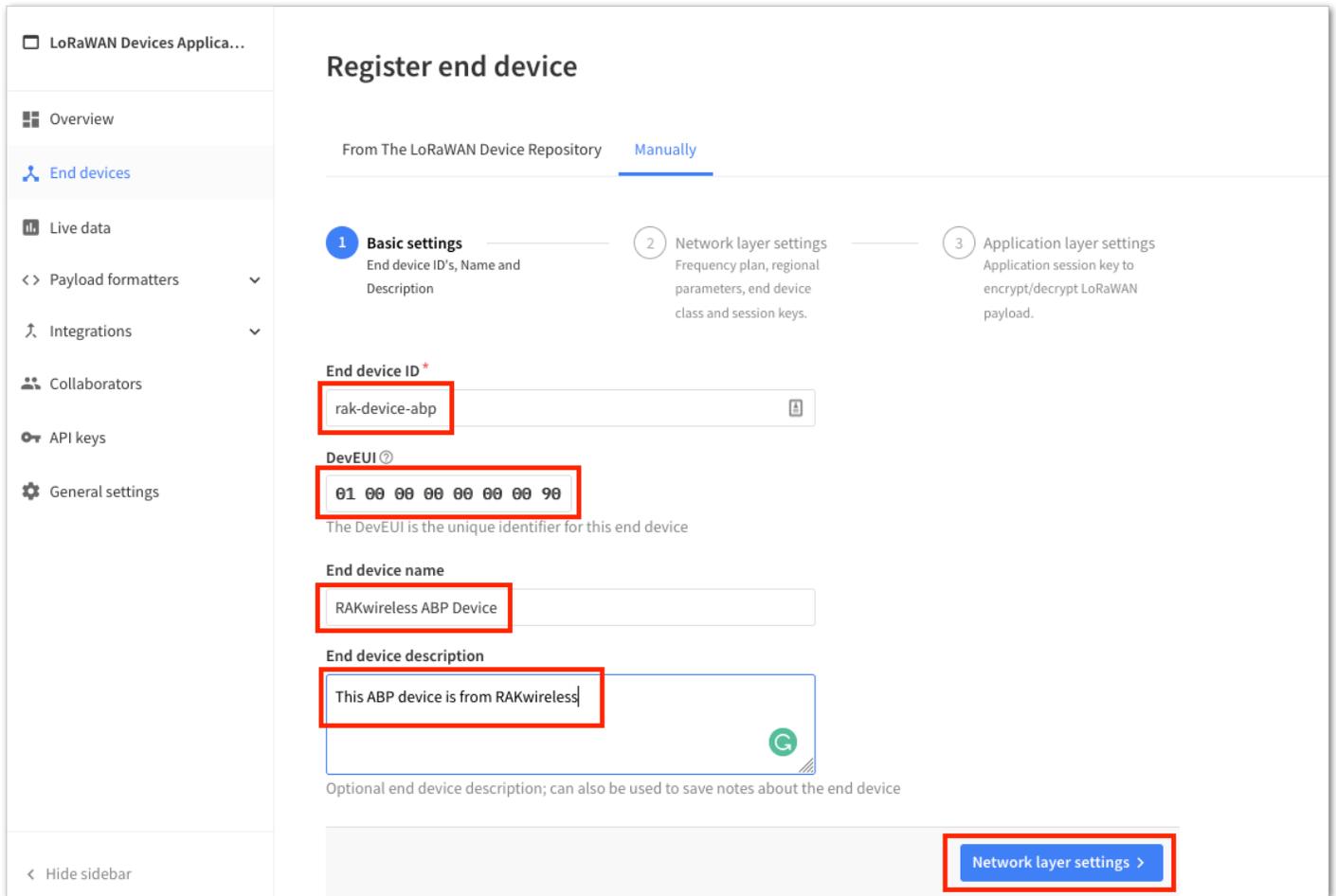


Figure 61: ABP device information

5. Set up the **Frequency plan**, a compatible **Regional Parameter version**, and the **LoRaWAN class** supported. In an ABP device, you also need to generate a **Device Address** and a **NwksKey** (Network Session Key). Then you can click **Application layers settings**.

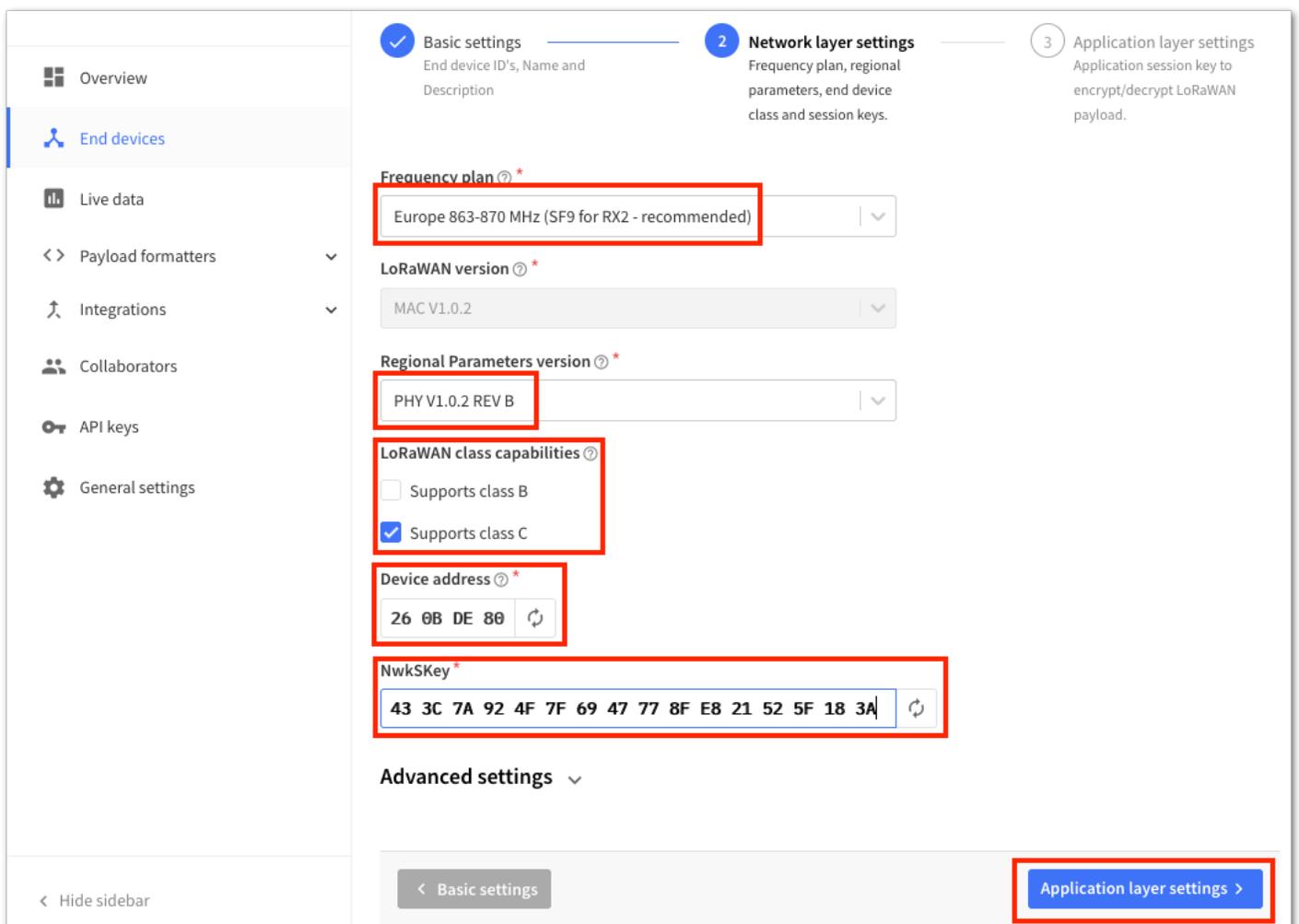


Figure 62: ABP device configuration

6. The last step in the registration of a new ABP end-device is the configuration of the **AppSKey**. To get the AppSKey, you must click the **generate button**, then the **Add end device** to finish your new device registration.

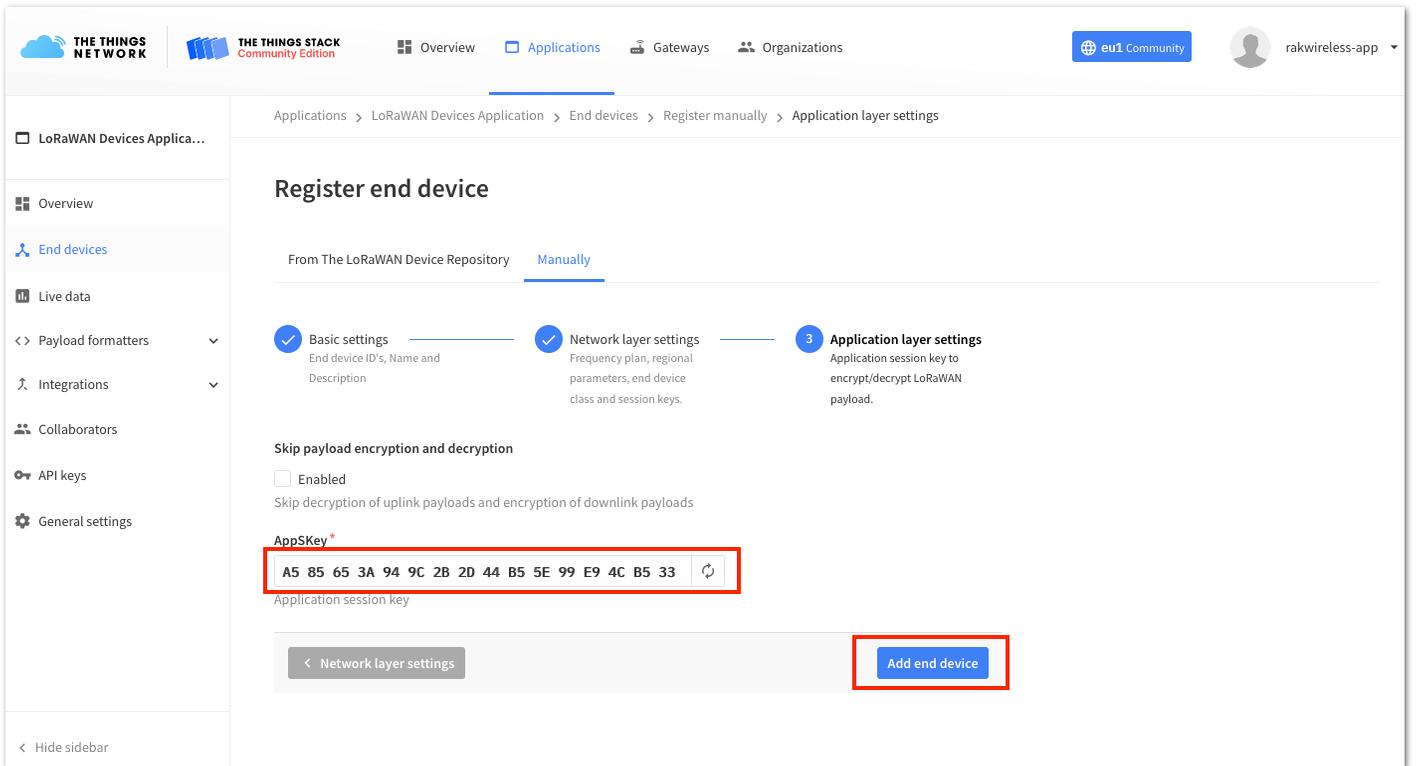


Figure 63: ABP AppSKey generation and device registration

You should now be able to see the device on the TTN console after you fully register your device, as shown in **Figure 64**.

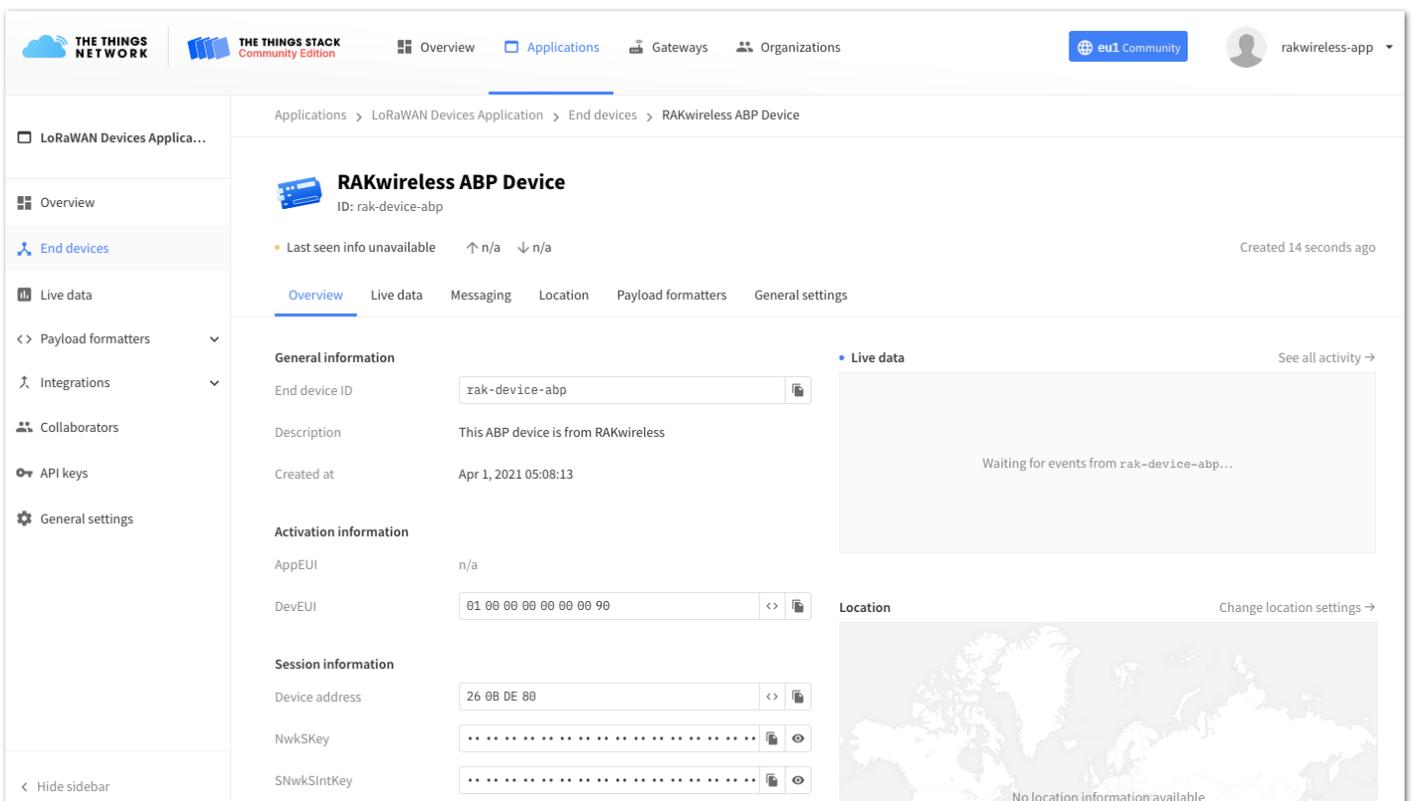


Figure 64: ABP device successfully registered to TTN

ABP Configuration for TTN

1. To set up the RAK4630 module to join the TTN using ABP, start by connecting the RAK4630 module to the computer (see [Figure 37](#)) and open the RAK Serial Port Tool. Select the right COM port and set the baud rate to 115200.

It is recommended to start by testing the serial communication and verify the current configuration is working by sending these two AT commands:

AT

ATE

ATE will echo the commands you input to the module, which is useful for tracking the commands and troubleshooting.

2. You will receive OK when you input the two commands. After setting ATE, you can now see all the commands you input together with the replies. Try again AT and you should see it on the terminal followed by OK, as shown in **Figure 65**.

NOTE:

If you haven't received an OK or any reply, you need to check if the wiring of your UART lines is correct and if the baud is correctly configured to 115200. Also, you can check if the device is powered correctly. If you are getting power from a USB port, ensure that you have a good USB cable.

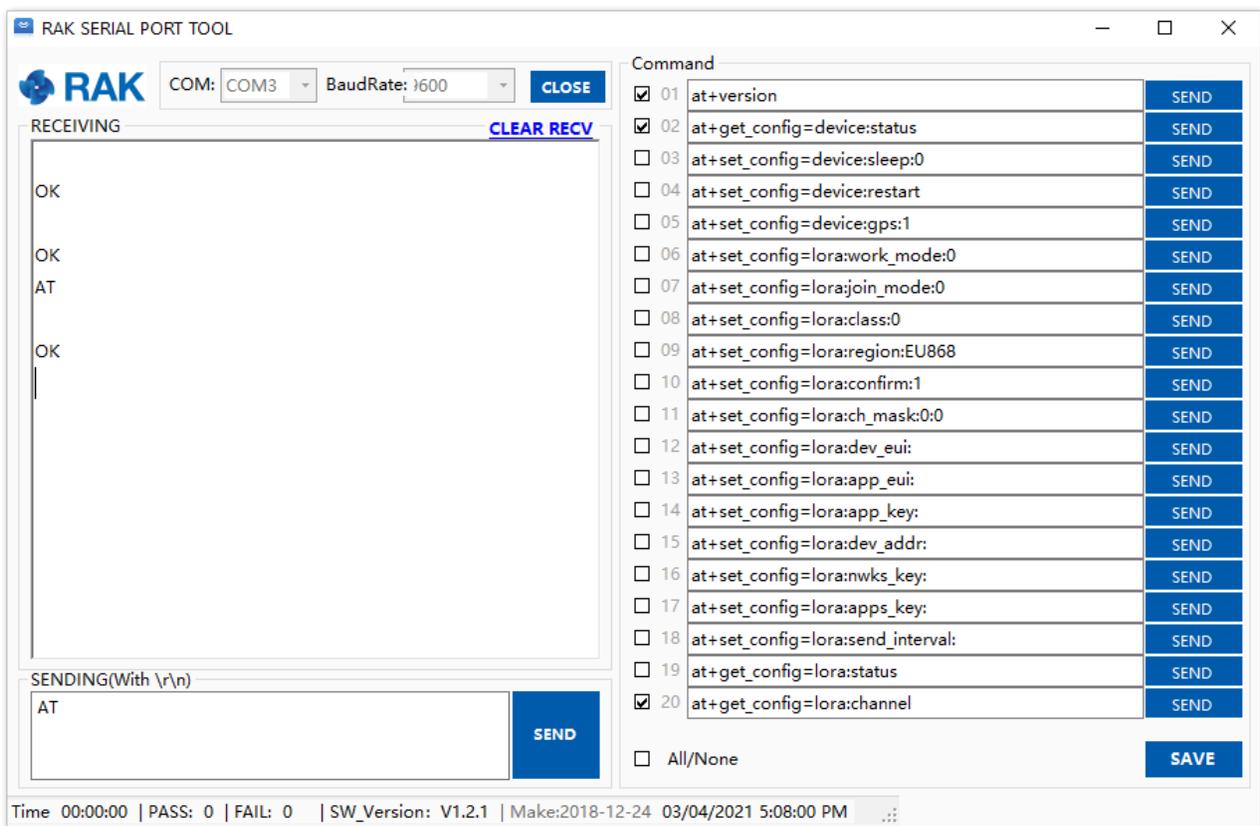


Figure 65: at+version command response

3. The next step is to configure the ABP LoRaWAN parameters in RAK4630:

- LoRa work mode: **LoRaWAN**
- LoRaWAN join mode: **ABP**
- LoRaWAN class: **Class A**
- LoRaWAN region: **EU868**

Set the work mode to LoRaWAN. You might need to reconnect on the Serial Terminal Software Tool if you lost connection after using the **AT+NWM=1** command.

AT+NWM=1

Set the LoRaWAN activation to ABP.

```
AT+NJM=0
```

Set the LoRaWAN class to Class A.

```
AT+CLASS=A
```

Set the frequency/region to EU868.

```
AT+BAND=4
```

 **NOTE:**

Depending on the Regional Band you selected, you might need to configure the sub-band of your RAK4630 to match the gateway and LoRaWAN network server. This is especially important on Regional Bands like US915, AU915, and CN470.

To configure the masking of channels for the sub-bands, you can use the `AT+MASK` command that can be found on the [AT Commands Manual](#) .

To illustrate, you can use sub-band 2 by sending the command `AT+MASK=0002` .

List of band parameter options

Code	Regional Band
0	EU433
1	CN470
2	RU864
3	IN865
4	EU868
5	US915
6	AU915
7	KR920
8	AS923

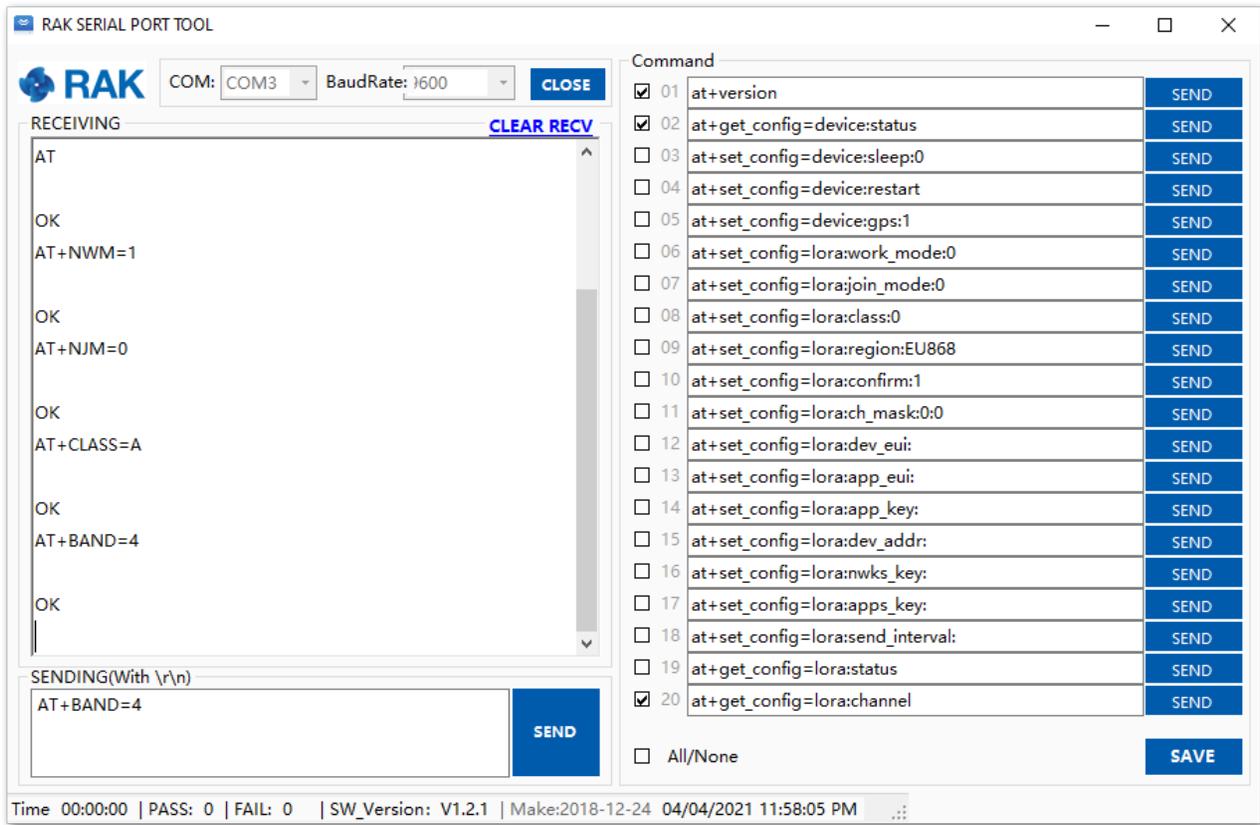


Figure 66: Configuring LoRa parameters

4. After the configuration of the LoRaWAN parameters, the next step is to set up the device address and sessions keys. You need the use the values from the TTN console.

- Device Address: **260BDE80**
- Application Session Key: **A585903A949C2B2D44B55E99E94CB533**
- Network Session Key: **433C7A924F7F6947778FE821525F183A**

Set the Device Address.

```

AT+DEVADDR=260BDE80
  
```

Set the Application Session Key.

```

AT+APPSKEY=A585903A949C2B2D44B55E99E94CB533
  
```

Set the Network Session Key.

```

AT+NWKSKEY=433C7A924F7F6947778FE821525F183A
  
```

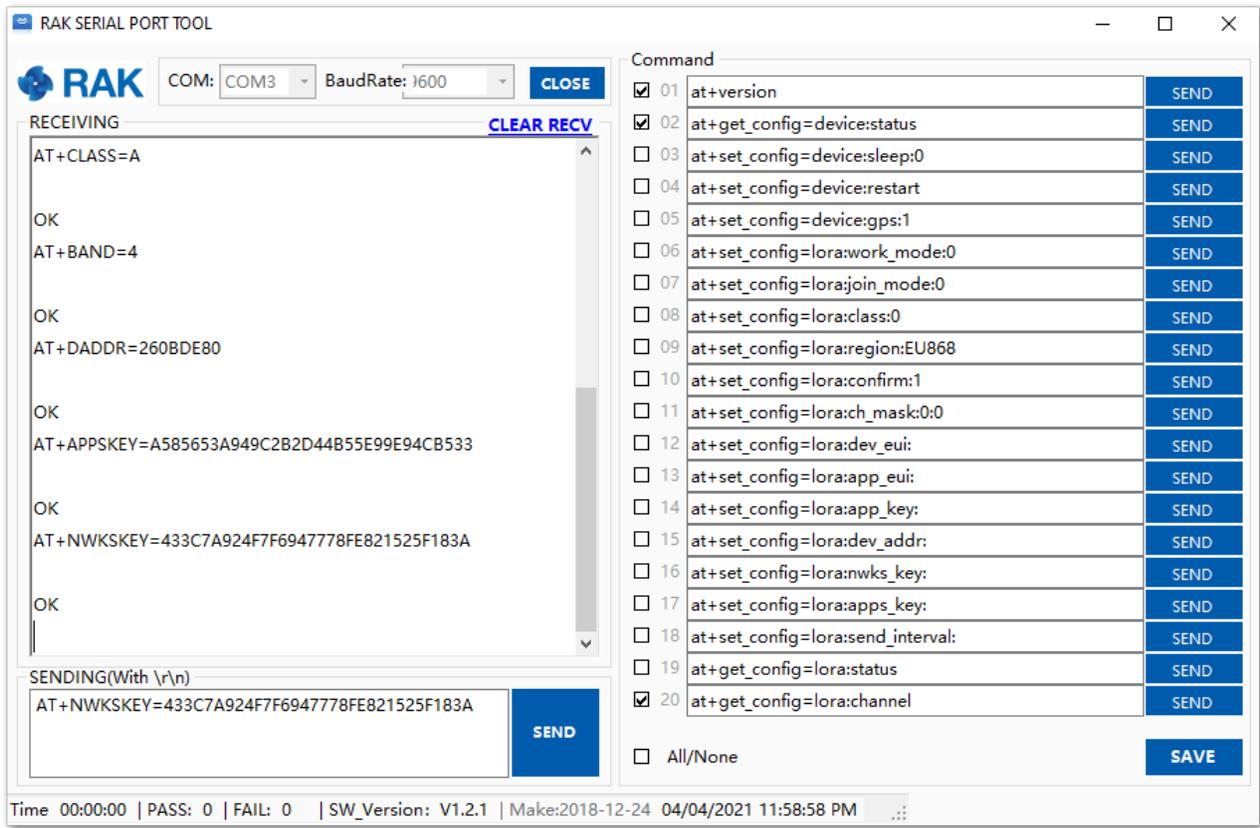


Figure 67: Configuring LoRa parameters

5. After EUI and keys configuration, the device can now join the network and send some payload.

```
AT+JOIN=1:0:8:0
```

Join command format: `AT+JOIN=w:x:y:z`

Parameter	Description
w	Join command - 1: joining, 0: stop joining.
x	Auto-join config - 1: auto-join on power-up, 0: no auto-join
y	Reattempt interval in seconds (7-255) - 8 is the default.
z	Number of join attempts (0-255) - 0 is default.

6. With the end-device properly activated, you can now try to send some payload after a successful join.

```
AT+SEND=3:12341234
```

Send command format: `AT+SEND=<port>:<payload>`

NOTE:

If your LoRaWAN payload didn't reach the TTN, check if your device is within reach of a working LoRaWAN gateway that is configured to connect to TTN. It is also important to check that all your ABP parameters (DEVADDR, APPSKEY, and NWKSKEY) are correct by using `AT+DEVADDR=?`, `AT+APPSKEY=?`, and `AT+NWKSKEY=?` commands. Lastly, ensure that the antenna of your device is properly connected.

After checking all the things above, try to send LoRaWAN payloads again.

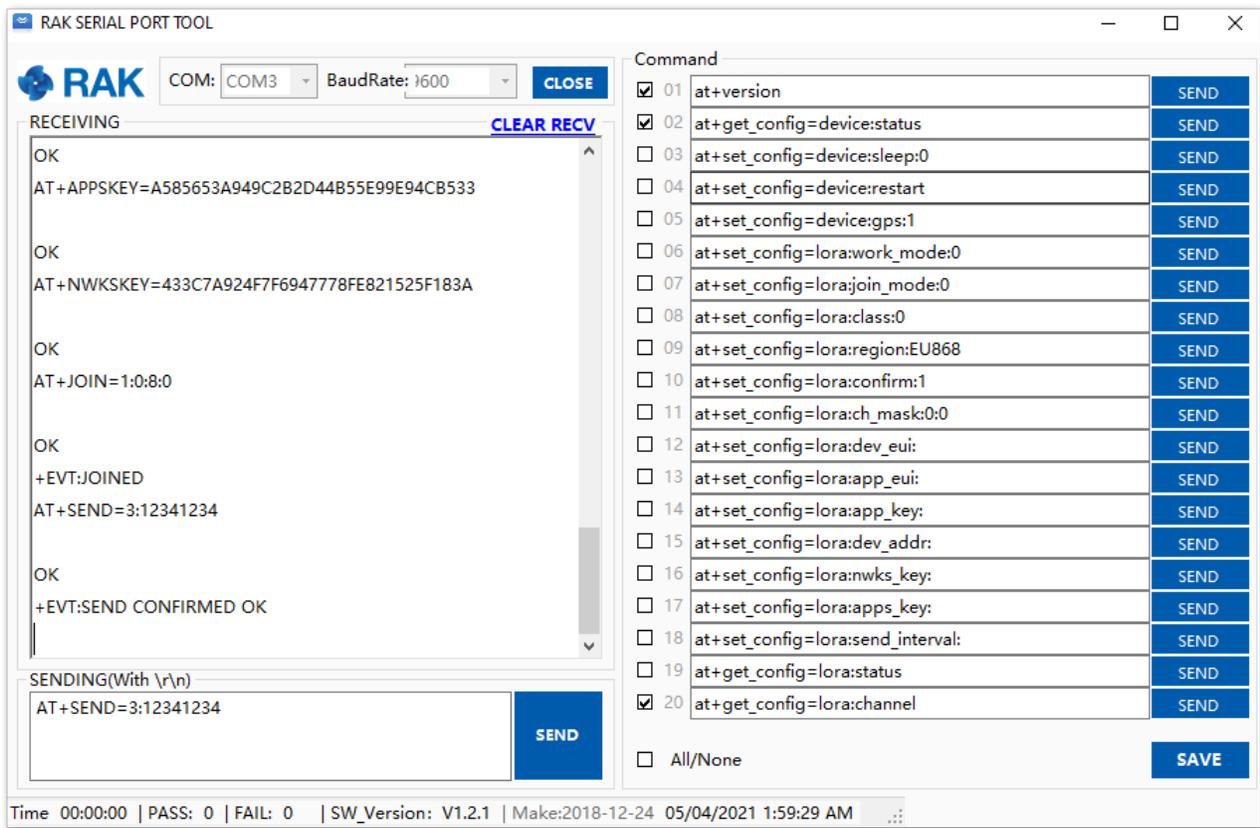


Figure 68: ABP test sample data sent via RAK Serial Port Tool

7. You can see the data sent by the RAK4630 module on the TTN device console *Live data* section and the *Last seen* info should be a few seconds ago.

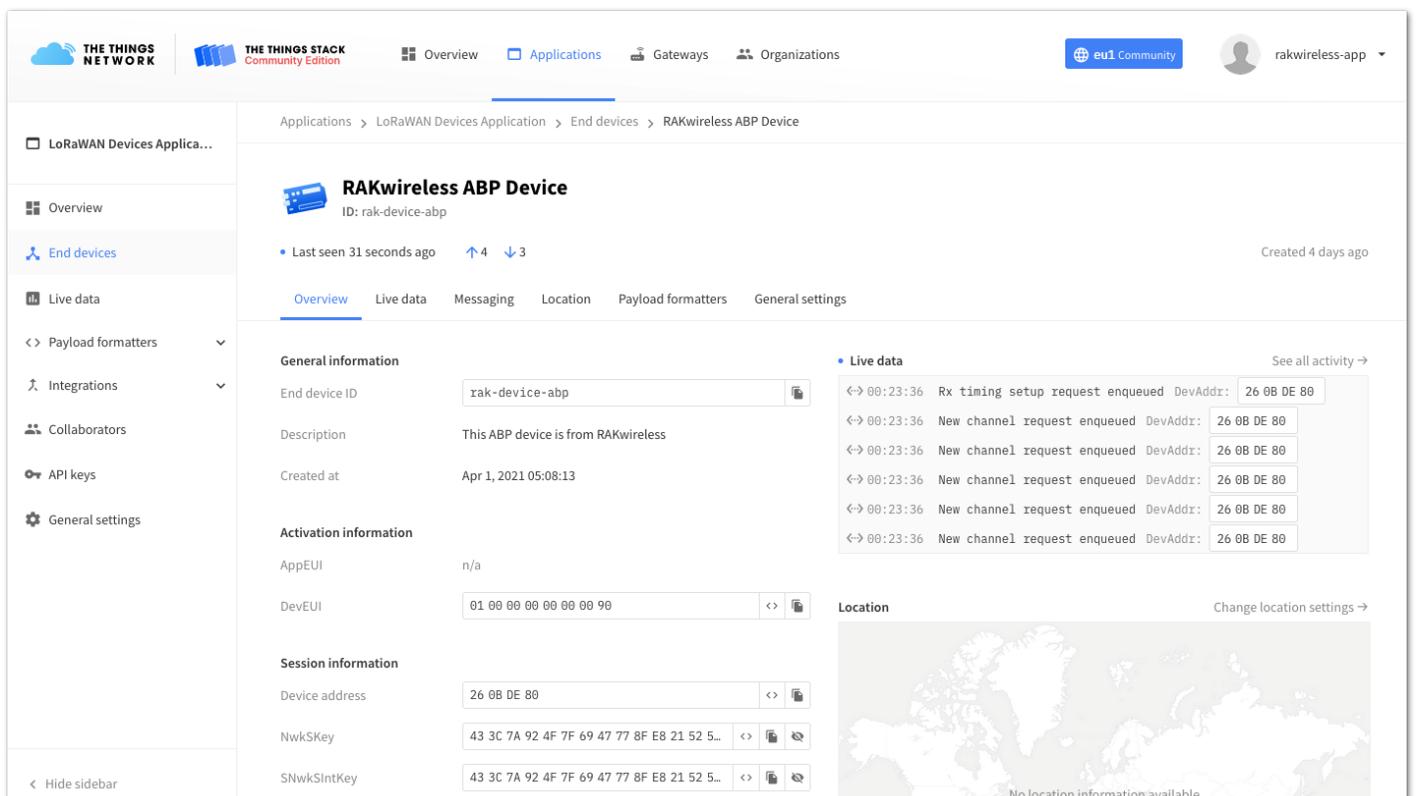


Figure 69: OTAA test sample data sent viewed in TTN

Connecting with ChirpStack

This section shows how to connect the RAK4630 module to the ChirpStack platform.

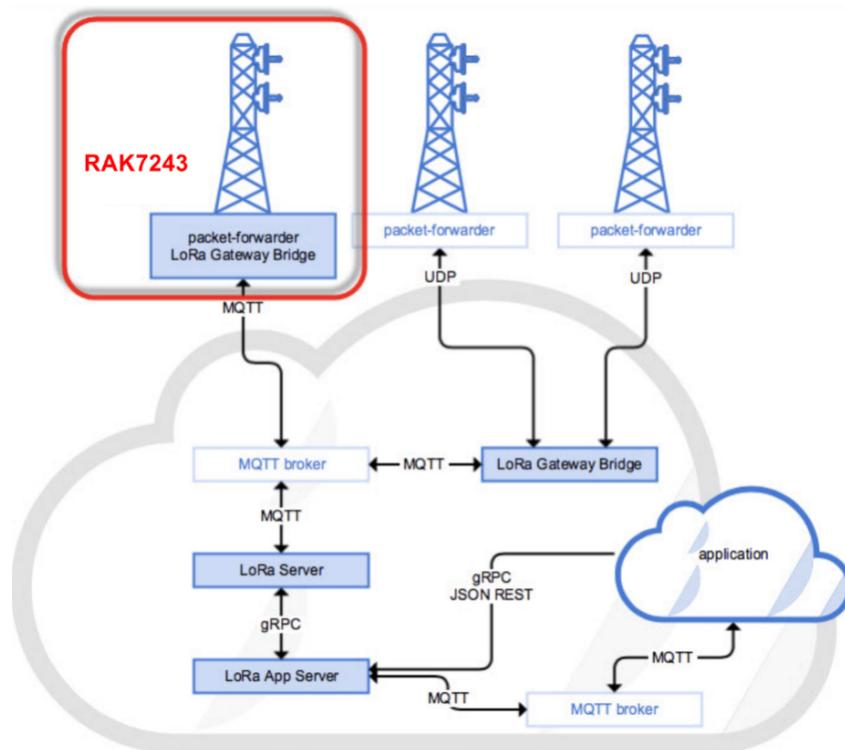


Figure 70: RAK4630 Module in the context of the ChirpStack platform

The ChirpStack, previously known as the LoRaServer project, provides open-source components for building LoRaWAN networks. Like in the case of TTN, the RAK4630 module is located in the periphery and will transmit the data to the backend servers through a LoRaWAN gateway. Learn more about [ChirpStack](#) .

NOTE:

In this guide, it is assumed that you are using a RAK Gateway and its built-in ChirpStack. Also, the gateway with the ChirpStack must be configured successfully. For further information, check the RAK documents for more details.

- In summary, these are the requirements:
 1. In a ChirpStack online gateway, the frequency band of the nodes should be consistent with the frequency band of the gateway in use.
 - [Connect the Gateway with Chirpstack](#)
 2. RAK Serial Port Tool provided by RAK
 3. RAK4630 module

NOTE:

The frequency band used in the demonstration is EU868. Use a high-frequency version of RAK4630. The product number should be “**RAK4630 (H)**”.

Create a New Application

1. Log in to the ChirpStack server using your account and password.
2. Go to the Application section, as shown in **Figure 71**.

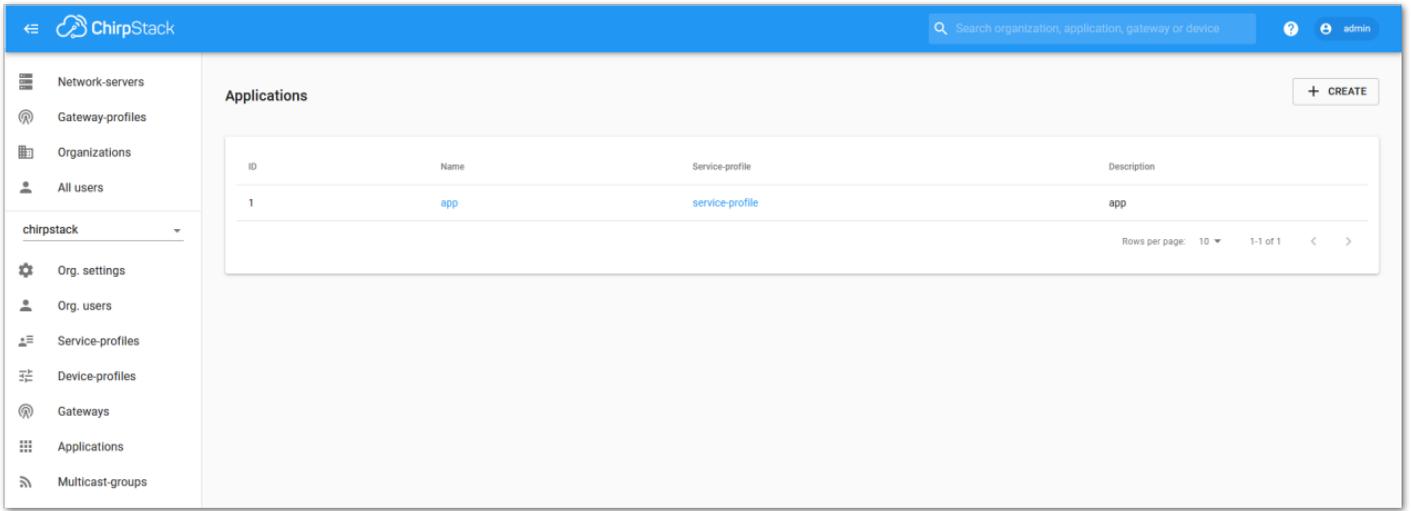


Figure 71: Application section

3. By default, you should create a new application, although you can reuse existing ones. For this setup, create a new Application by clicking on the “**CREATE**” button and filling in the required parameters, as shown in **Figure 72** and **Figure 73**.

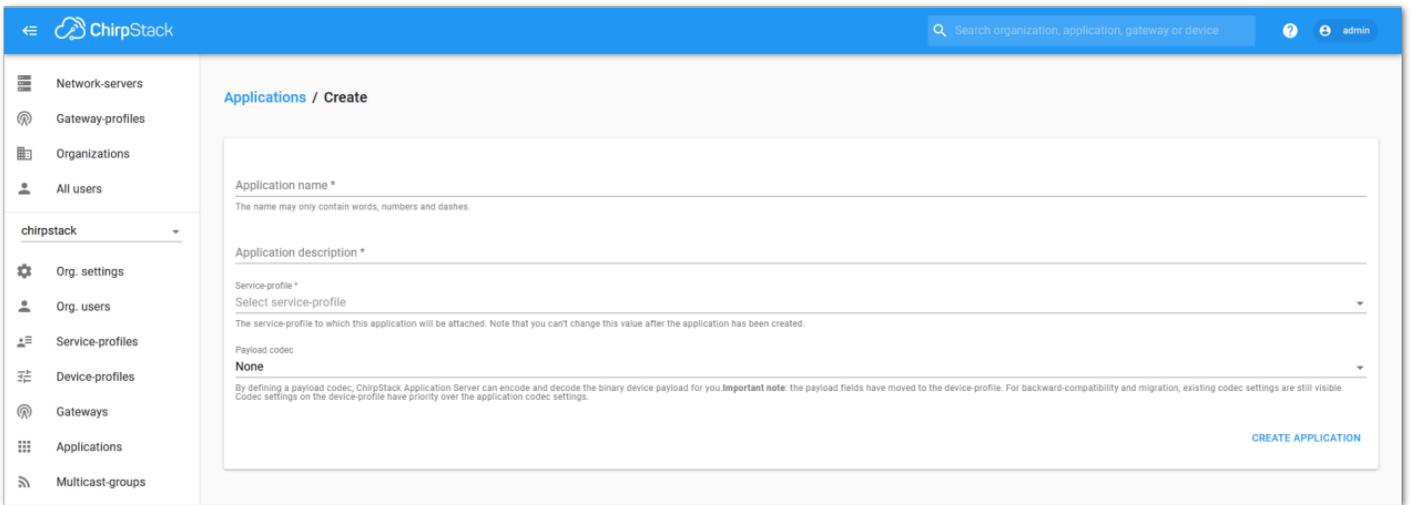


Figure 72: Creating a new application

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

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

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

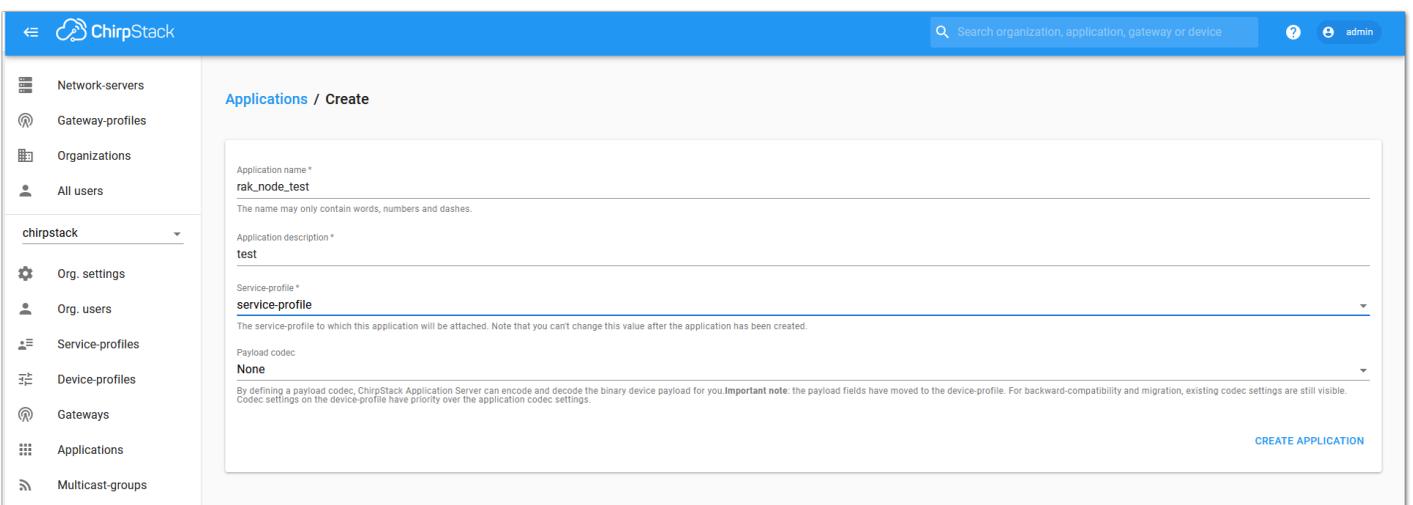


Figure 73: Filling in the parameters of an application

Register a New Device

1. Choose the **Application** created in the previous step, then select the **DEVICES** tab, as shown in **Figure 74** and **Figure 75**.
2. Once done, click “+ **CREATE**”.

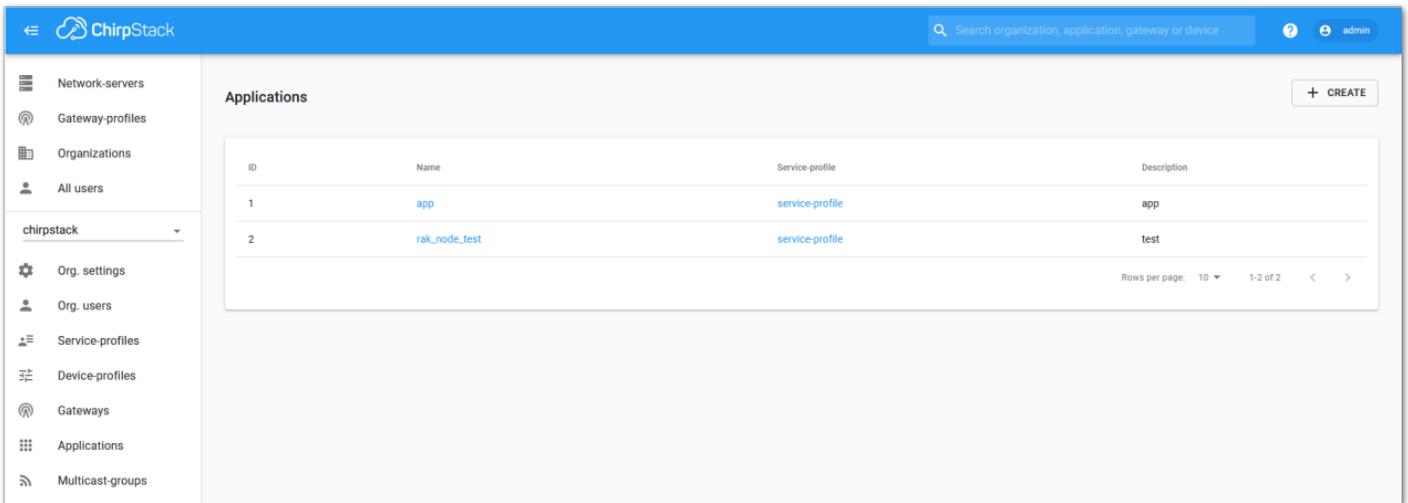


Figure 74: List of applications created

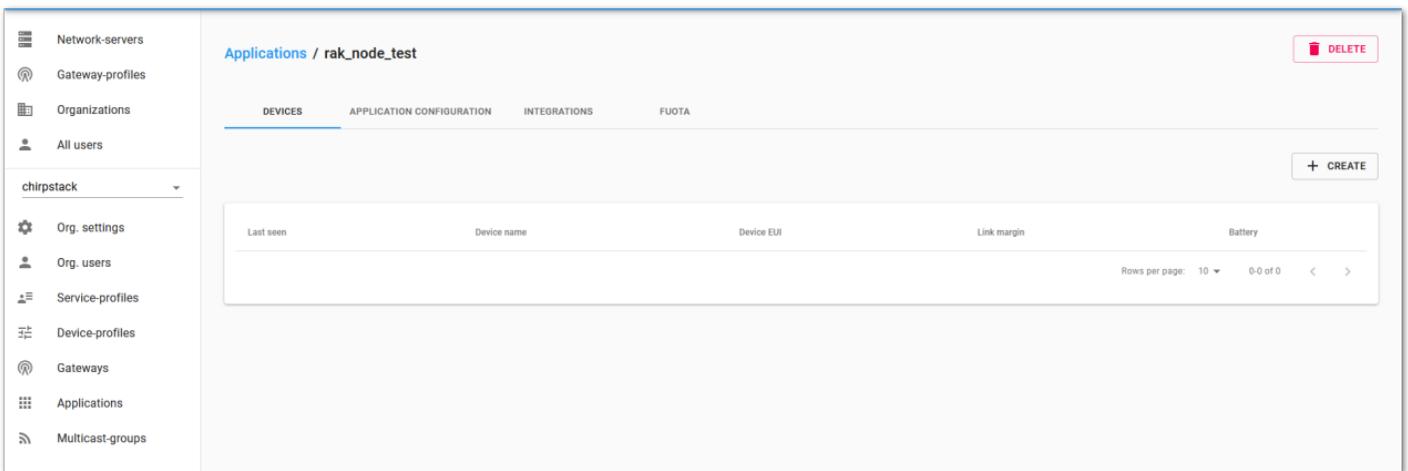


Figure 75: Device tab of an application

3. Once inside of the **DEVICE** tab, create a new device (LoRaWAN node) by clicking on the “+ **CREATE**” button.

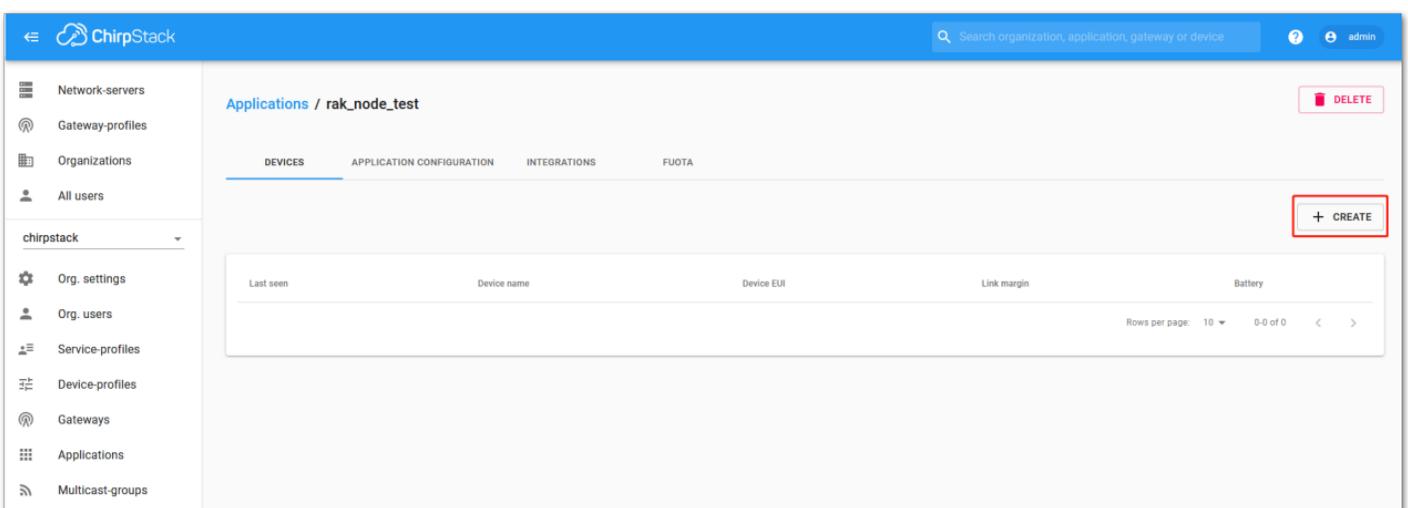


Figure 76: Add a new device

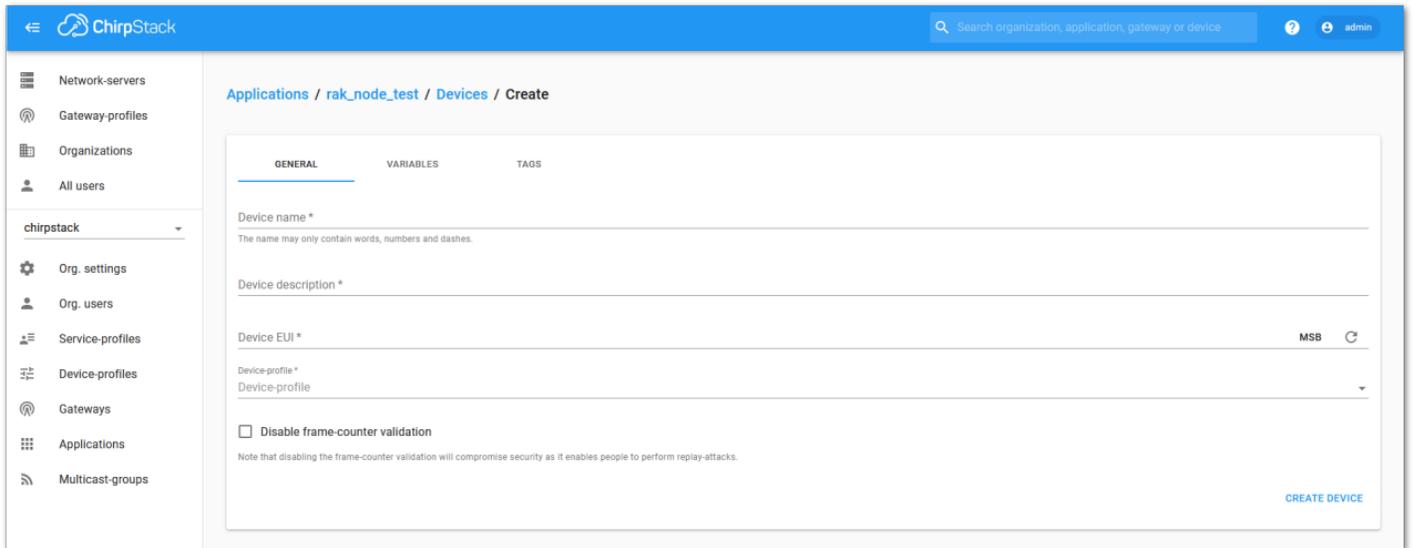


Figure 77: Chirpstack adding node into the RAK4630 Module

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.

Fill in the parameters requested:

- **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 generate icon. 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:

Device profiles **DeviceProfile_OTAA** and **DeviceProfile_ABP** are only available if you are using the built-in Chirpstack LoRaWAN Server of RAK Gateways.

If you have your own Chirpstack installation, you can set up the device profile with **LoRaWAN MAC version 1.0.3** and **LoRaWAN Regional Parameters revision B** to make it compatible with RAK4630.

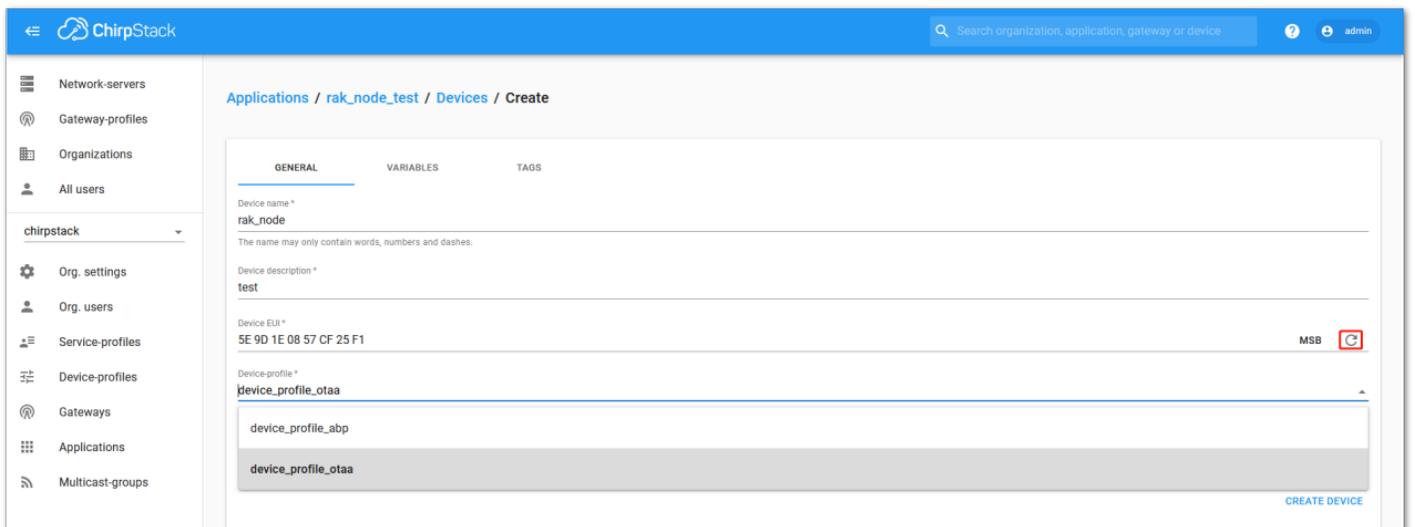


Figure 78: Generate a new Device EUI

Chirpstack OTAA Device Registration

1. If you have selected “**DeviceProfile_OTAA**”, as shown in **Figure 79**, after the device is created, an Application Key must be also created for this device.

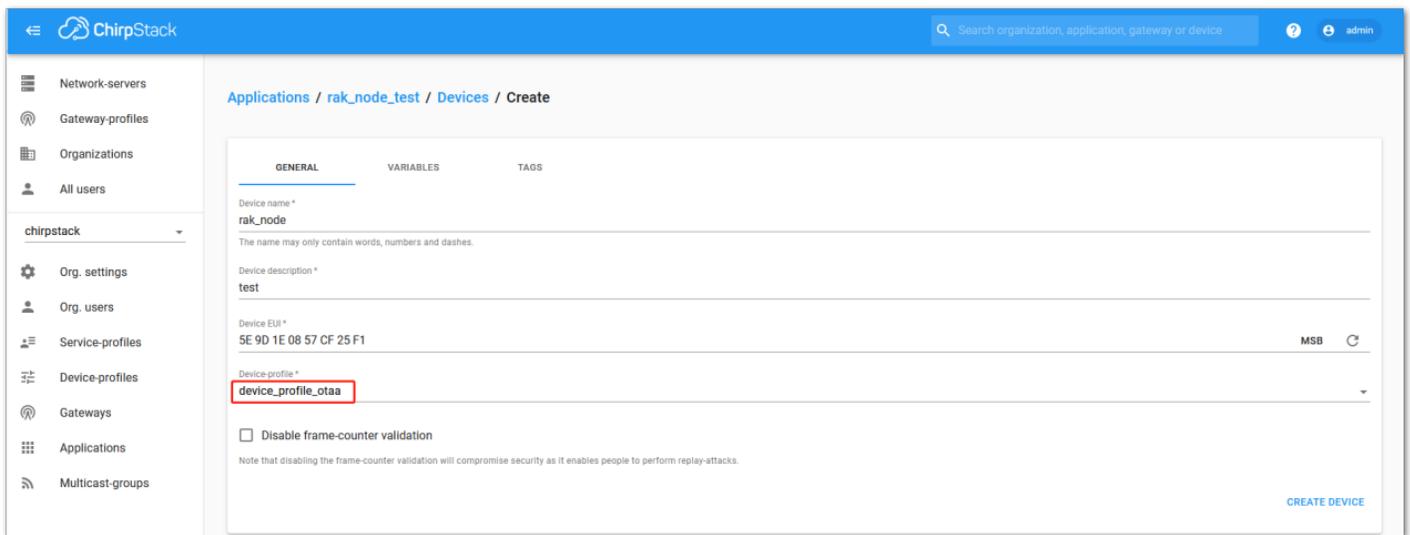


Figure 79: Chirpstack OTAA activation

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 80**.

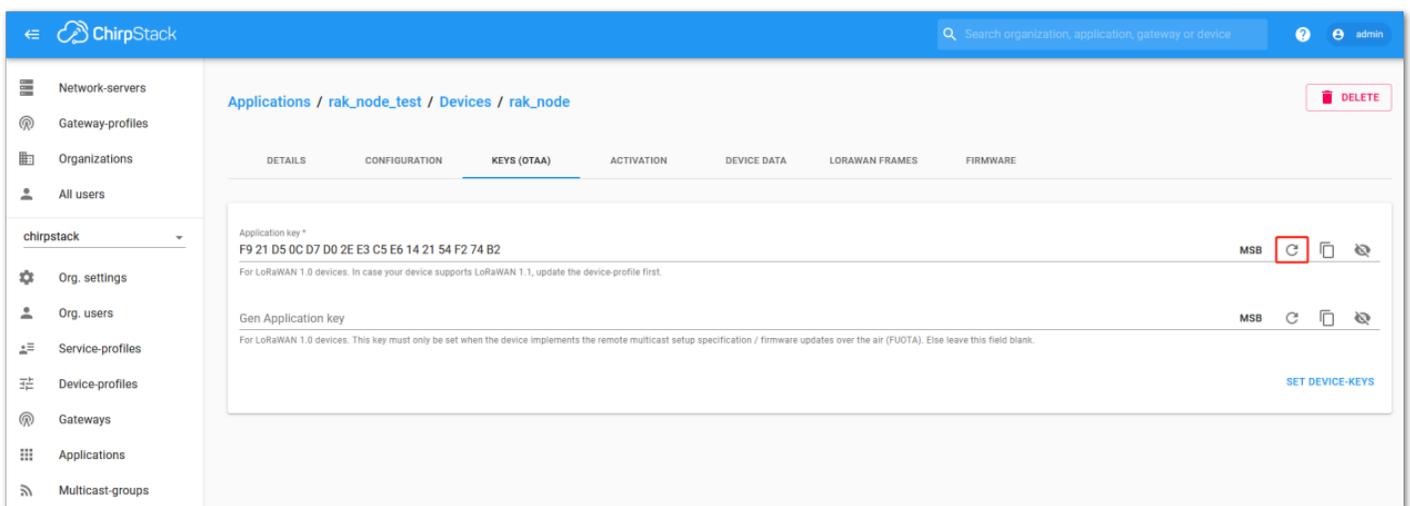


Figure 80: Chirpstack OTAA Set Application Keys

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

- As shown in **Figure 81**, a new device should be listed in the DEVICES tab. The most important parameters, such as the Device EUI, are shown in the summary.

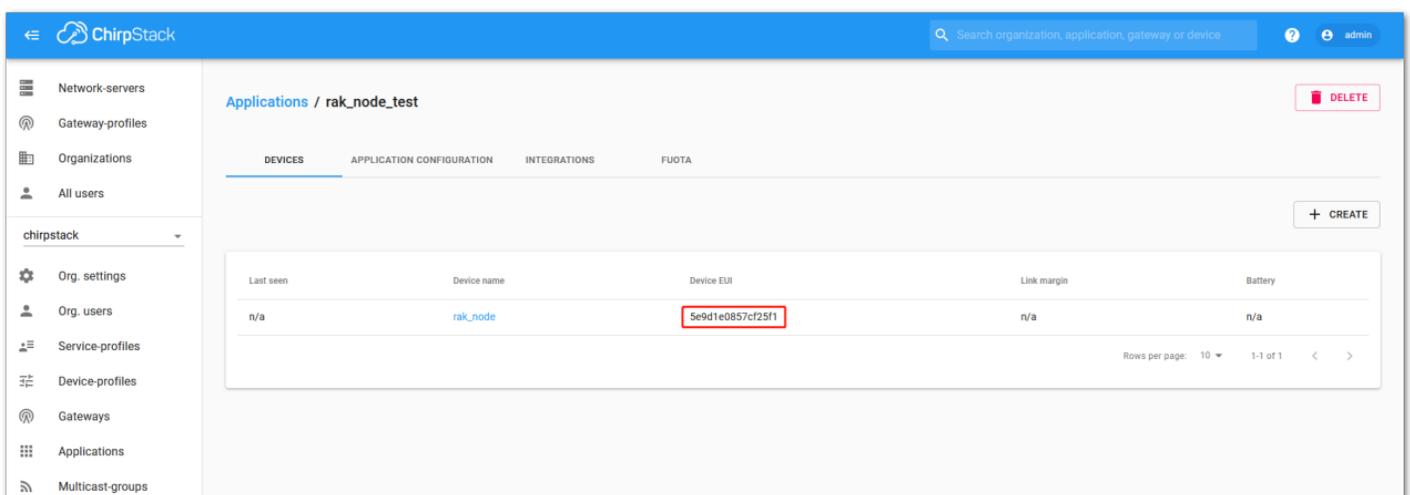


Figure 81: Chirpstack OTAA list of device in the Device Tab

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 82**.

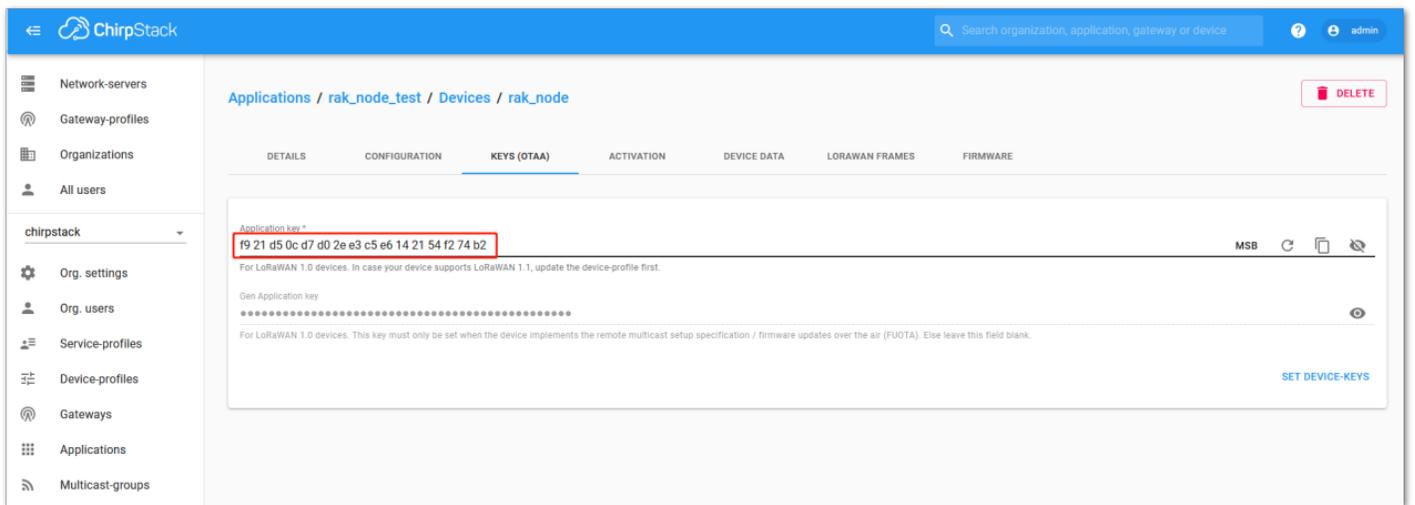


Figure 82: Application key associated with the new device

NOTE:

Standard OTAA mode requires the **Device EUI**, **Application Key**, and **Application EUI**, but in ChirpStack's implementation, only the Device EUI and the Application Key are mandatory. The Application EUI is not required and not recorded in the Application tab. Nevertheless, you can reuse the Device EUI as the Application EUI during the configuration on the side of the node.

OTAA Configuration for Chirpstack

The RAK4630 module supports a series of AT commands to configure its internal parameters and control the functionalities of the module.

1. To set up the RAK4630 module to join the Chirpstack using OTAA, start by connecting the RAK4630 module to the Computer (see [Figure 37](#)) and open the RAK Serial Port Tool. Select the right COM port and set the baud rate to 115200.

It is recommended to start by testing the serial communication and verify that the current configuration is working by sending these two AT commands:

AT

ATE

`ATE` will echo the commands you input to the module, which is useful for tracking the commands and troubleshooting.

You will receive `OK` when you input the two commands. After setting `ATE`, you can now see all the commands you input together with the replies. Try again `AT` and you should see it on the terminal followed by `OK`, as shown in [Figure 68](#).

NOTE:

If you haven't received an `OK` or any reply, you need to check if the wiring of your UART lines is correct and if the baud is correctly configured to 115200. Also, you can check if the device is powered correctly. If you are getting power from a USB port, ensure that you have a good USB cable.

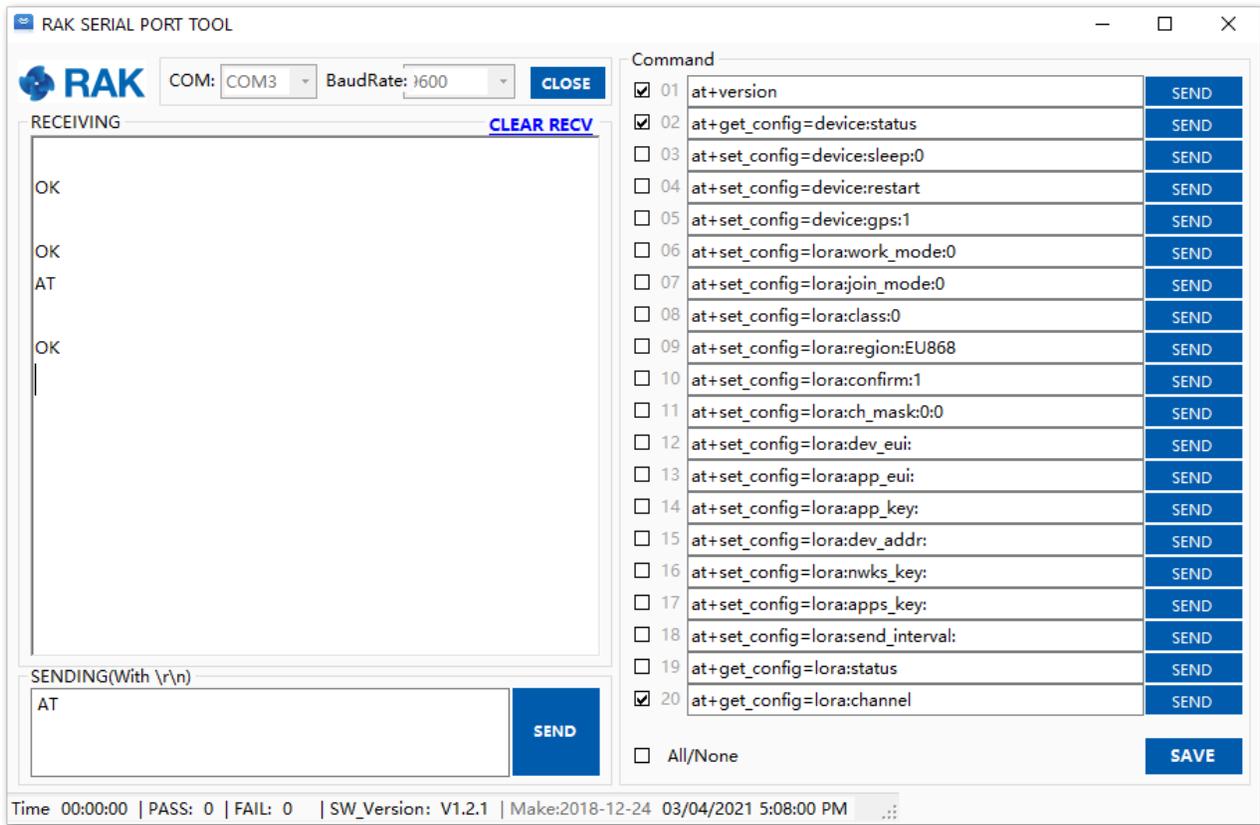


Figure 83: at+version command response

2. The next step is to configure the OTAA LoRaWAN parameters in RAK4630:

- LoRa work mode: **LoRaWAN**
- LoRaWAN join mode: **OTAA**
- LoRaWAN class: **Class A**
- LoRaWAN region: **EU868**

Set the work mode to LoRaWAN. You might need to reconnect on the Serial Terminal Software Tool if you lost connection after using the `AT+NWM=1` command.

```
AT+NWM=1
```

Set the LoRaWAN activation to OTAA.

```
AT+NJM=1
```

Set the LoRaWAN class to Class A.

```
AT+CLASS=A
```

Set the frequency/region to EU868.

```
AT+BAND=4
```

 **NOTE:**

Depending on the Regional Band you selected, you might need to configure the sub-band of your RAK4630 to match the gateway and LoRaWAN network server. This is especially important for Regional Bands like US915, AU915, and CN470.

To configure the masking of channels for the sub-bands, you can use the `AT+MASK` command that can be found on the [AT Command Manual](#) .

To illustrate, you can use sub-band 2 by sending the command `AT+MASK=0002`.

List of band parameter options

Code	Regional Band
0	EU433
1	CN470
2	RU864
3	IN865
4	EU868
5	US915
6	AU915
7	KR920
8	AS923

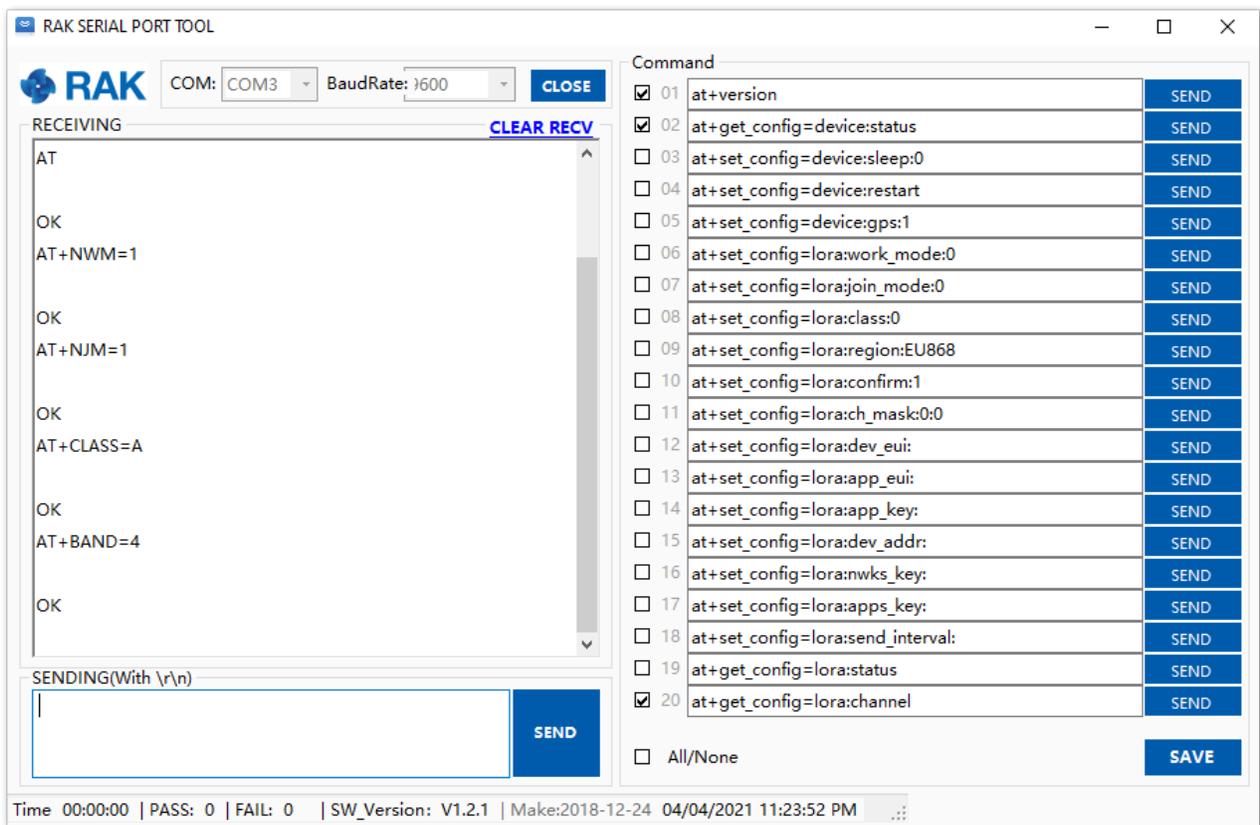


Figure 84: Configuring LoRa parameters

3. After the configuration of the LoRaWAN parameters, the next step is to set up the DevEUI and AppKey. You need the use the values from the Chirpstack device console.

NOTE:

The Application EUI parameter is not required in the ChirpStack platform; therefore, it is possible to use the same id as the Device EUI.

- Device EUI: **5E9D1E0857CF25F1**
- Application EUI: **5E9D1E0857CF25F1**
- Application Key: **F921D50CD7D02EE3C5E6142154F274B2**

Set the Device EUI.

```
AT+DEVEUI=5E9D1E0857CF25F1
```

Set the Application EUI.

```
AT+APPEUI=5E9D1E0857CF25F1
```

Set the Application Key.

```
AT+APPKEY=F921D50CD7D02EE3C5E6142154F274B2
```

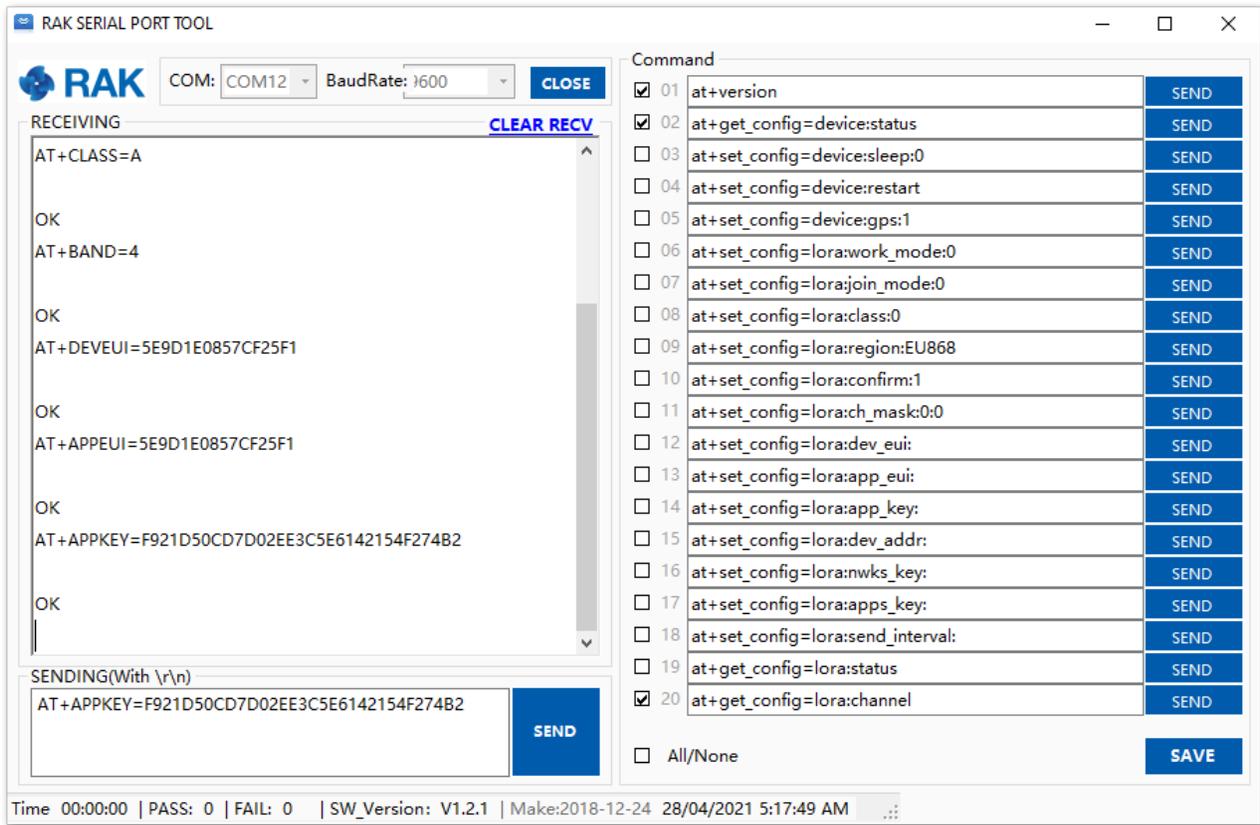


Figure 85: Configuring LoRa Parameters

4. After the EUI and key configuration, the device can now join the network and send some payload.

```
AT+JOIN=1:0:10:8
```

Join command format: `AT+JOIN=w:x:y:z`

Parameter	Description
w	Join command - 1: joining, 0: stop joining.
x	Auto-join config - 1: auto-join on power-up, 0: no auto-join
y	Reattempt interval in seconds (7-255) - 8 is the default.
z	Number of join attempts (0-255) - 0 is default.

After 5 or 6 seconds, if the request is successfully received by a LoRa gateway, you should see the JOINED status reply.

NOTE:

If the OTAA device failed to join, you need to check if your device is within reach of a working LoRaWAN gateway that is configured to connect to Chirpstack. It is also important to check that all your OTAA parameters (DEVEUI and APPKEY) are correct, using the `AT+DEVEUI=?` and `AT+APPKEY=?` commands. Lastly, ensure that the antenna of your device is properly connected.

After checking all the things above, try to join again.

5. With the end-device properly activated, you can now try to send some payload after a successful join.

AT+SEND=2:12345678

Send command format: `AT+SEND=<port>:<payload>`

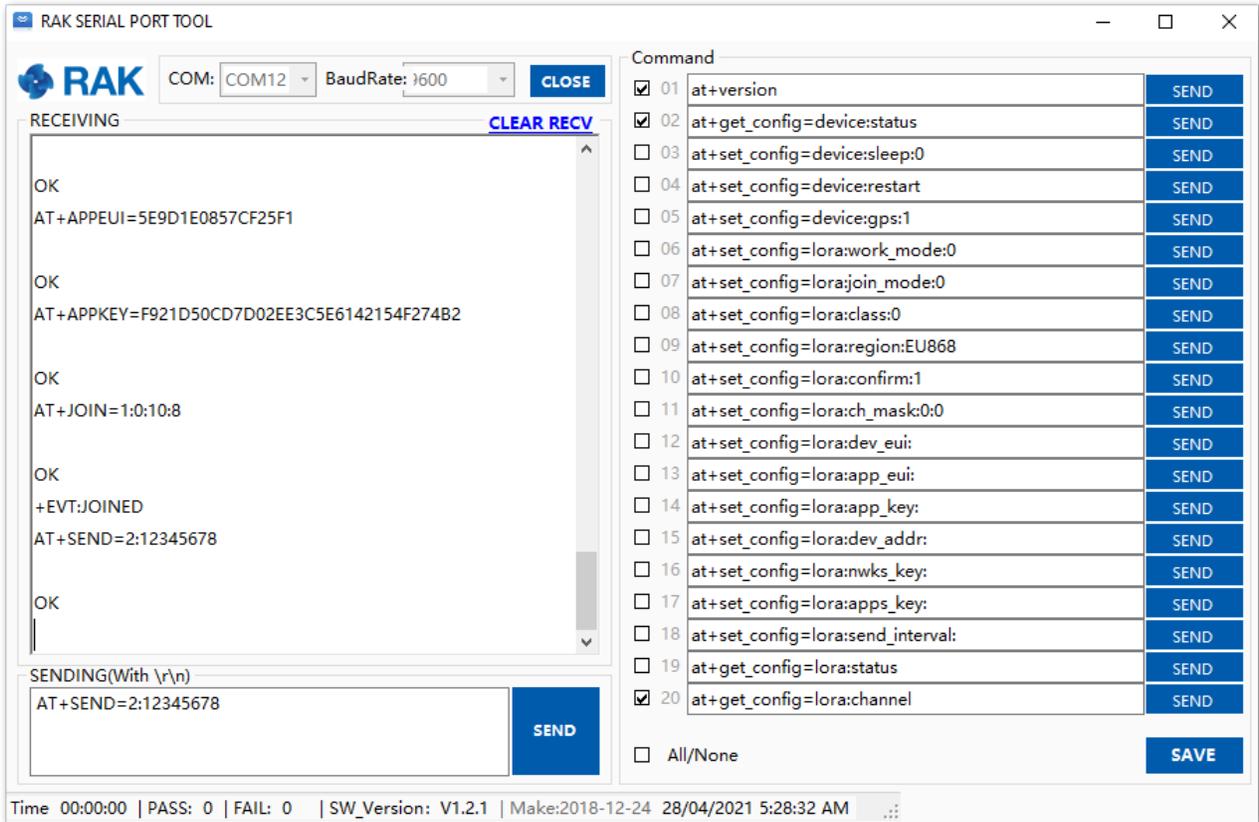


Figure 86: OTAA test sample data sent via RAK Serial Port Tool

On the ChirpStack platform, you should see the join and uplink messages in the LORAWAN FRAMES tab, as shown in Figure 87. By convention, messages sent from nodes to gateways are considered as **Uplinks** while messages sent by gateways to nodes are considered as **Downlinks**.

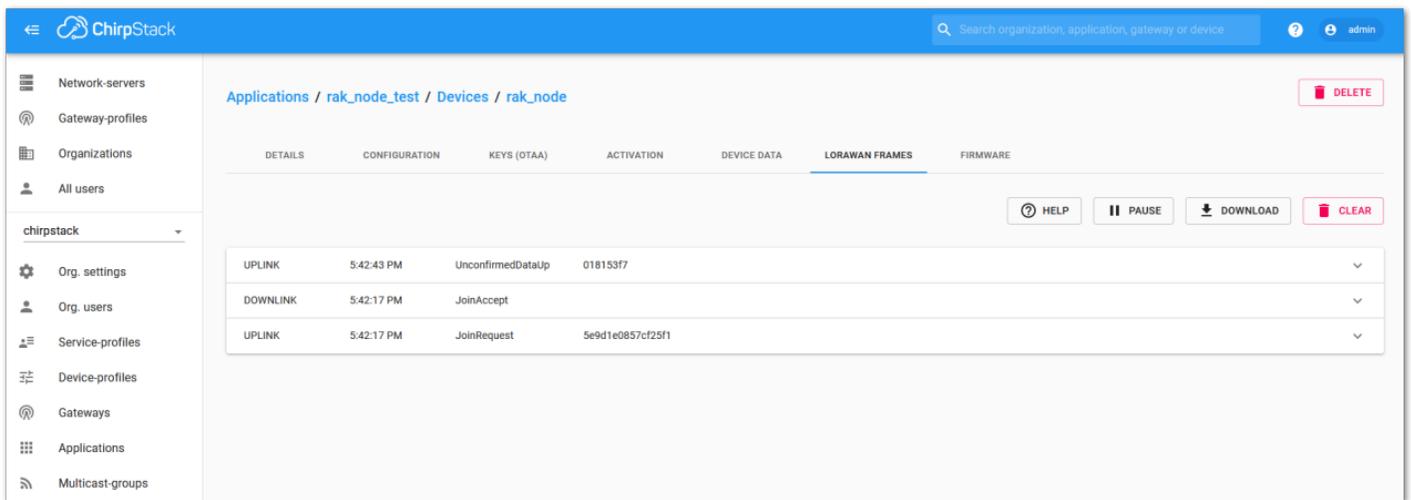


Figure 87: Chirpstack data received preview

Chirpstack ABP Device Registration

1. During the registration of a new device, if you select “**DeviceProfile_ABP**”, as shown in Figure 88, then the ChirpStack platform will assume that this device will join 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.

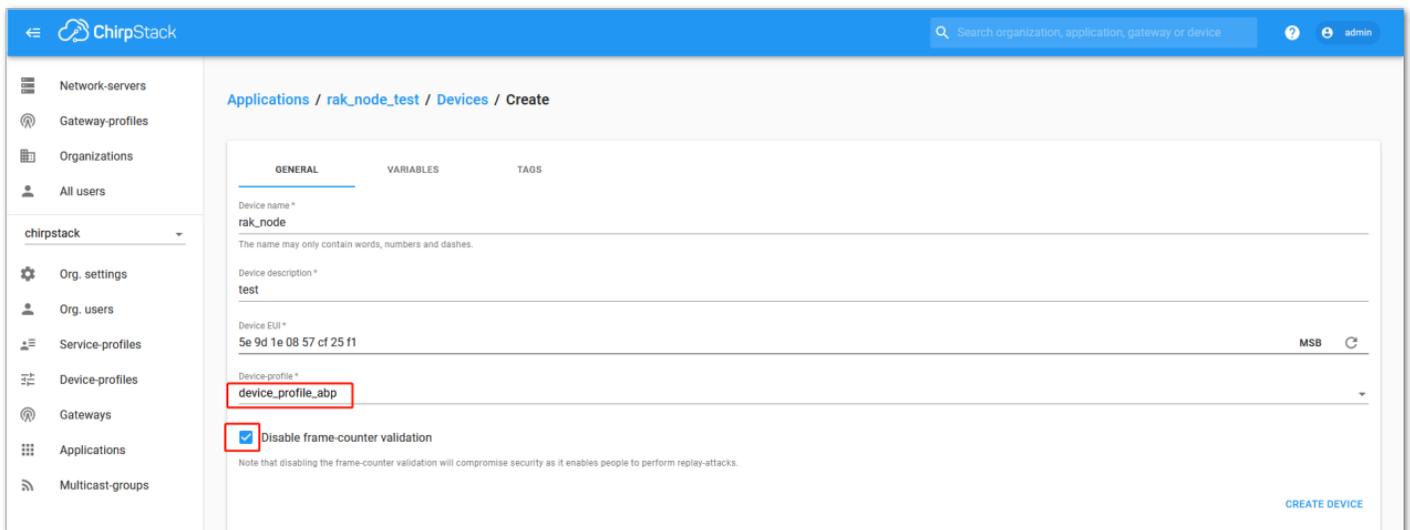


Figure 88: ChirpStack console, configuring a device

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

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

- **Device Address**
- **Network Session Key**
- **Application Session Key**

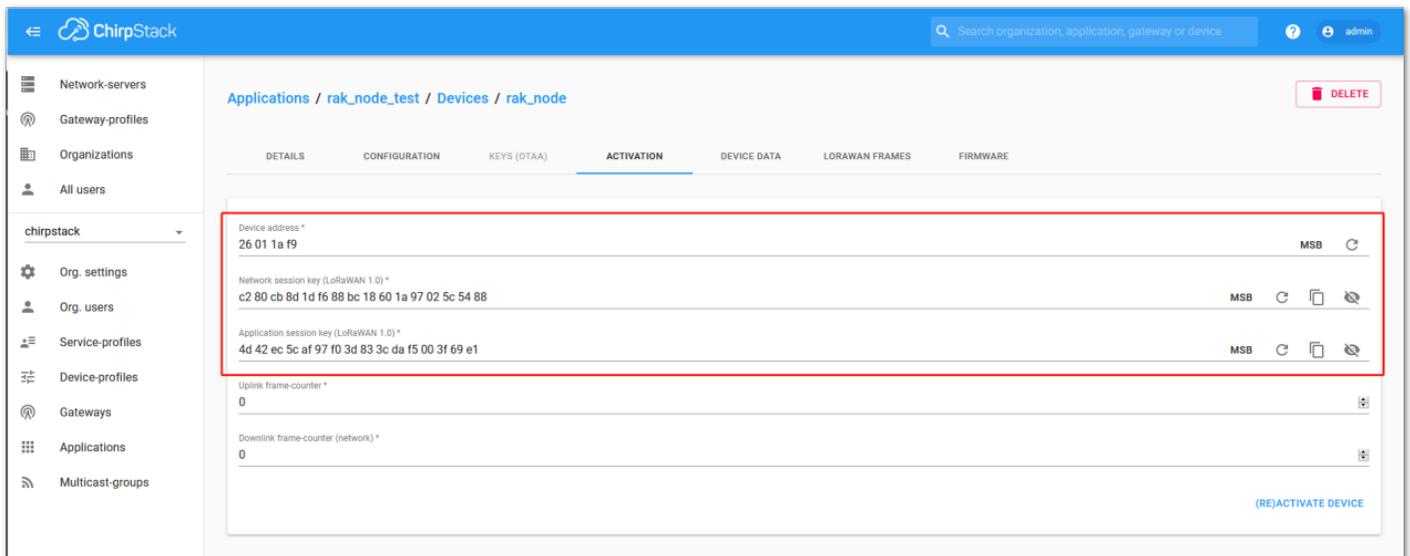


Figure 89: Chirpstack ABP activation parameters needed

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

ABP Configuration for Chirpstack

1. To set up the RAK4630 module to join the Chirpstack using ABP, start by connecting the RAK4630 module to the Computer (see [Figure 37](#)) and open the RAK Serial Port Tool. Select the right COM port and set the baud rate to 115200.

It is recommended to start by testing the serial communication and verify that the current configuration is working by sending these two AT commands:

```
AT
```

```
ATE
```

`ATE` will echo the commands you input to the module, which is useful for tracking the commands and troubleshooting.

You will receive `OK` when you input the two commands. After setting `ATE`, you can now see all the commands you input together with the replies. Try again `AT` and you should see it on the terminal followed by `OK`, as shown in **Figure 90**.

NOTE:

If haven't received an `OK` or any reply, you need to check if the wiring of your UART lines is correct and if the baud is correctly configured to 115200. Also, you can check if the device is powered correctly. If you are getting power from a USB port, ensure that you have a good USB cable.

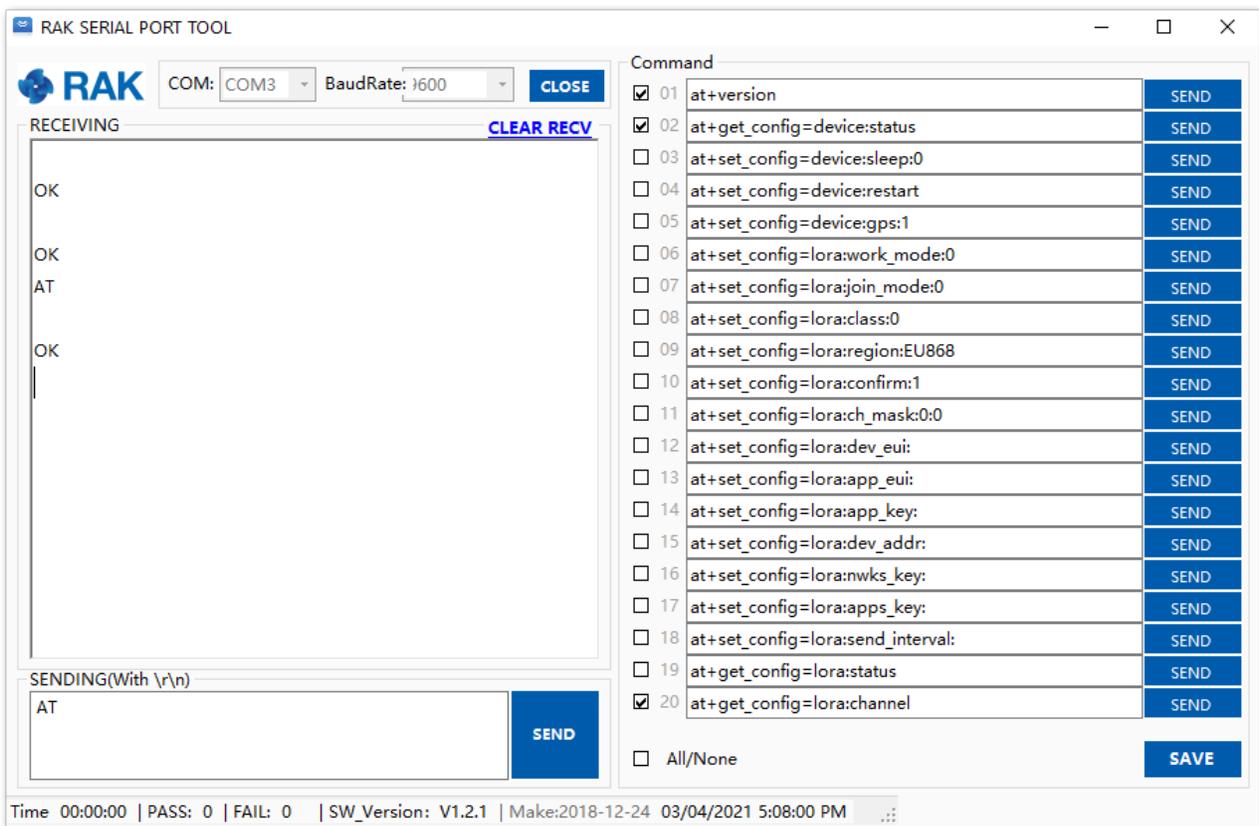


Figure 90: at+version command response

2. The next step is to configure the ABP LoRaWAN parameters in RAK4630:

- LoRa work mode: **LoRaWAN**
- LoRaWAN join mode: **ABP**
- LoRaWAN class: **Class A**
- LoRaWAN region: **EU868**

Set the work mode to LoRaWAN. You might need to reconnect on the Serial Terminal Software Tool if you lost connection after using the `AT+NWM=1` command.

```
AT+NWM=1
```

Set the LoRaWAN activation to ABP.

```
AT+NJM=0
```

Set the LoRaWAN class to Class A.

```
AT+CLASS=A
```

Set the frequency/region to EU868.

```
AT+BAND=4
```

 **NOTE:**

Depending on the Regional Band you selected, you might need to configure the sub-band of your RAK4630 to match the gateway and LoRaWAN network server. This is especially important on Regional Bands like US915, AU915, and CN470.

To configure the masking of channels for the sub-bands, you can use the `AT+MASK` command that can be found on the [AT Commands Manual](#) .

To illustrate, you can use sub-band 2 by sending the command `AT+MASK=0002` .

List of band parameter options

Code	Regional Band
0	EU433
1	CN470
2	RU864
3	IN865
4	EU868
5	US915
6	AU915
7	KR920

Code

Regional Band

8

AS923

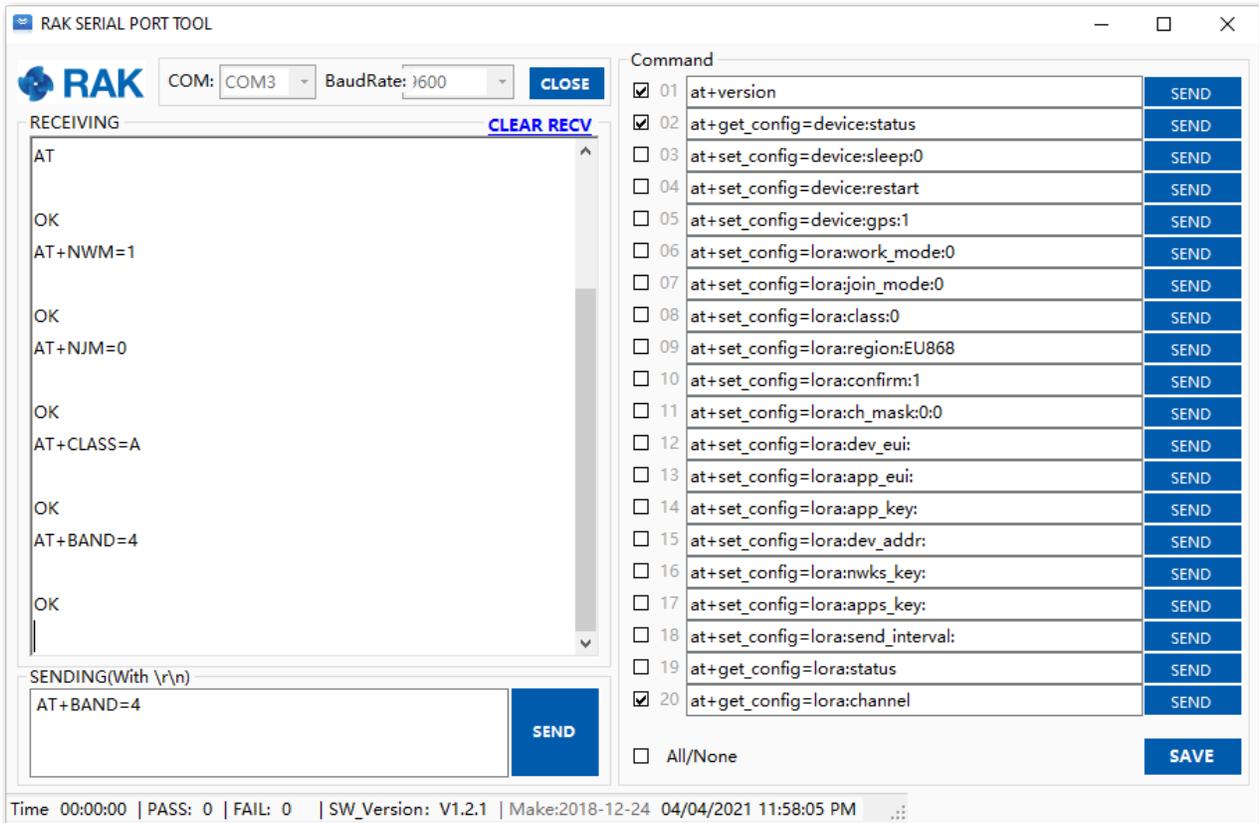


Figure 91: Configuring LoRa parameters

3. After the configuration of the LoRaWAN parameters, the next step is to set up the device address and session keys. You need the use the values from the Chirpstack device console.

- Device Address: **26011AF9**
- Application Session Key: **4D42EC5CAF97F03D833CDAf5003F69E1**
- Network Session Key: **C280CB8D1DF688BC18601A97025C5488**

Set the Device Address.

```
AT+DEVADDR=26011AF9
```

Set the Application Session Key.

```
AT+APPSKEY=4D42EC5CAF97F03D833CDAf5003F69E1
```

Set the Network Session Key.

```
AT+NWKSKEY=C280CB8D1DF688BC18601A97025C5488
```

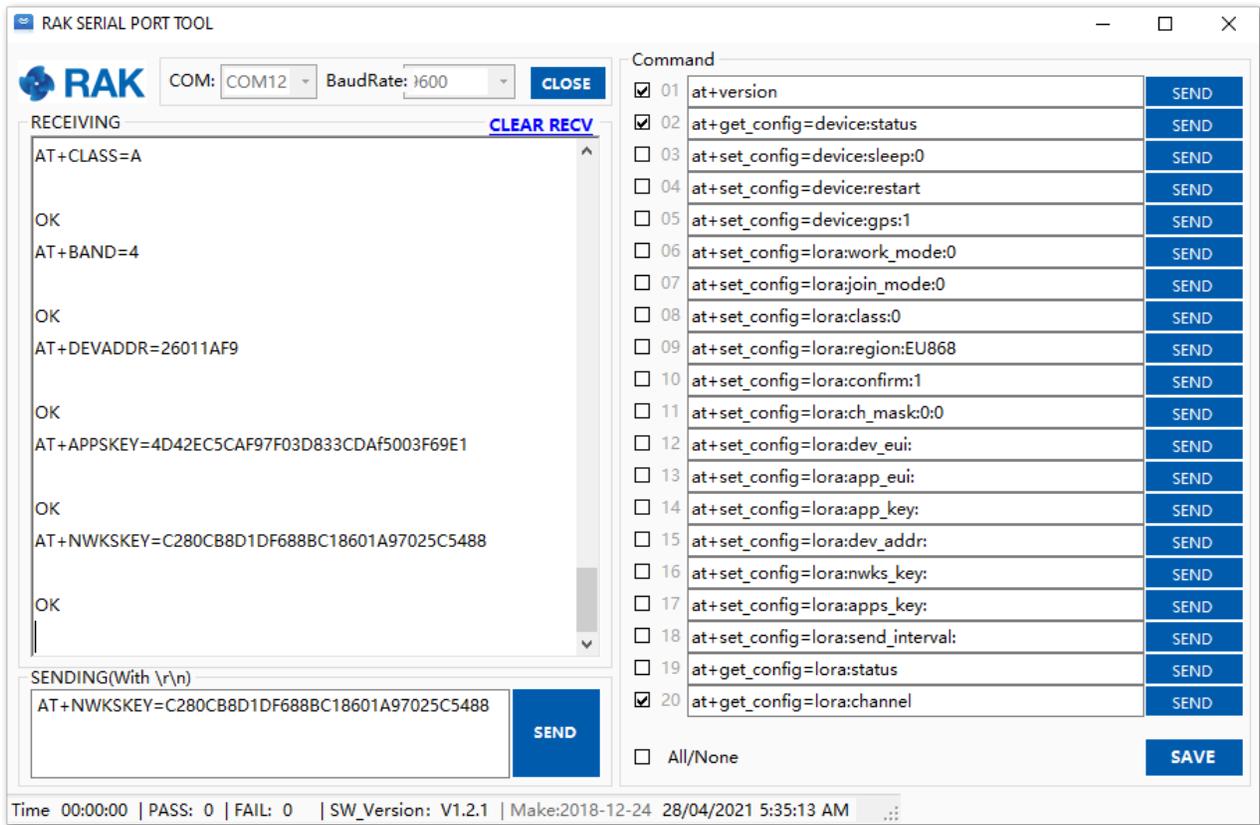


Figure 92: Configuring LoRa parameters

After the EUI and keys configuration, the device can now join the network and send some payload.

```
AT+JOIN=1:0:10:8
```

Join command format: `AT+JOIN=w:x:y:z`

Parameter	Description
w	Join command - 1: joining, 0: stop joining.
x	Auto-join config - 1: auto-join on power-up, 0: no auto-join
y	Reattempt interval in seconds (7-255) - 8 is the default.
z	Number of join attempts (0-255) - 0 is default.

4. After 5 or 6 seconds, if the request is successfully received by a LoRa gateway, then you should see the JOINED status reply.

5. You can now try to send some payload after a successful join.

```
AT+SEND=2:12341234
```

Send command format: `AT+SEND=<port>:<payload>`

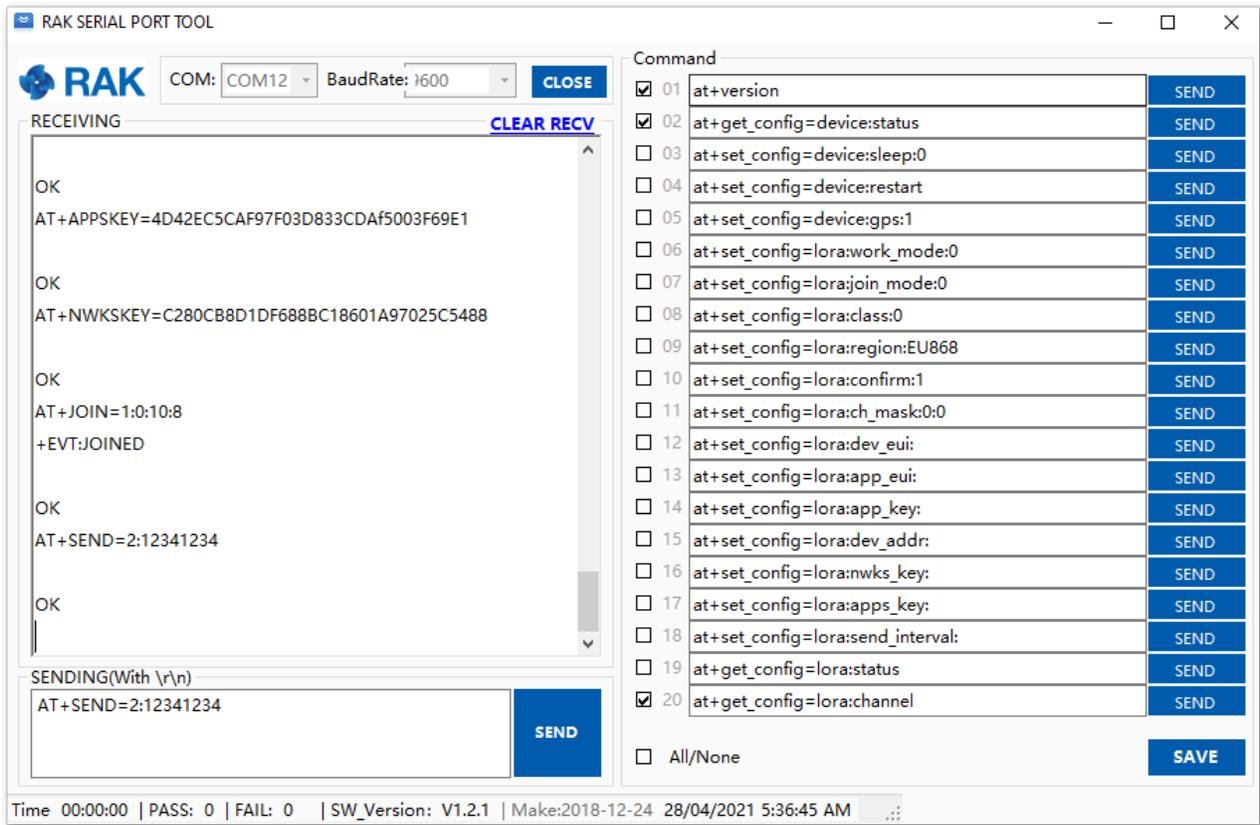


Figure 93: ABP test sample data sent via RAK Serial Port Tool

LoRa P2P Mode

This section shows how to set up and connect two RAK4630 units to work in the LoRa P2P mode. The configuration of the RAK4630 units is done by connecting the two modules to a general-purpose computer using a USB-UART converter. The setup of each RAK4630 can be done separately, but testing the LoRa P2P mode will require having both units connected simultaneously. This could be done by having one computer with two USB ports or two computers with one USB port each.

It is recommended to start by testing the serial communication and verify the current configuration is working by sending these two AT commands:

```
AT
```

```
ATE
```

`ATE` will echo the commands you input to the module, which is useful for tracking the commands and troubleshooting.

You will receive `OK` when you input the two commands. After setting `ATE`, you can now see all the commands you input together with the replies.

Try again `AT` and you should see it on the terminal followed by `OK`.

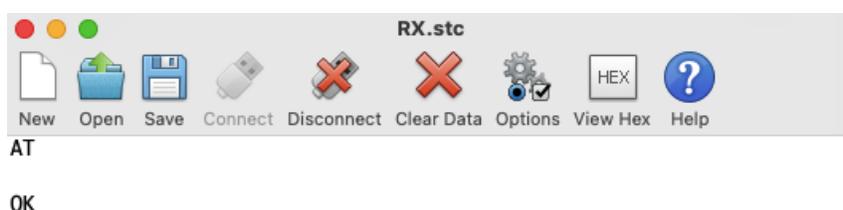


Figure 94: AT command response

1. To set up the RAK4630 to work in LoRa P2P mode, you need to input the work mode command on both RAK4630 modules. You might need to disconnect and reconnect the module as it changes its mode of operation if you are sending commands via USB.

```
AT+NWM=0
```

2. For this P2P setup, the LoRa parameters are the following:

- Link frequency: **868000000 Hz**
- Spreading factor: **7**
- Bandwidth: **125 kHz**
- Coding Rate: 0 = **4/5**
- Preamble Length: **10**
- Power: **14 dBm**

```
AT+P2P=868000000:7:125:0:10:14
```

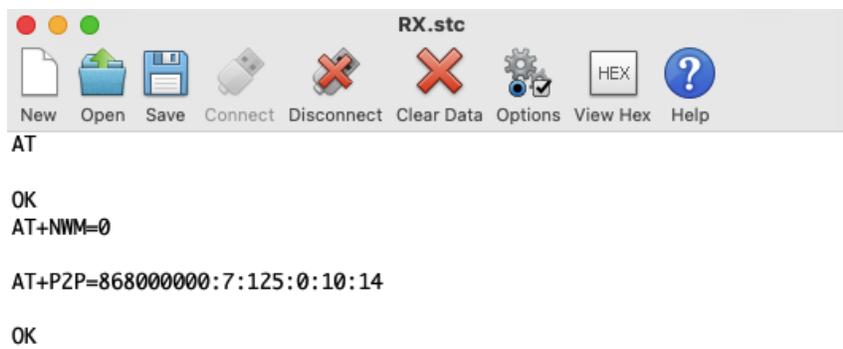


Figure 95: P2P Setup

NOTE:

Refer to the [P2P Mode](#) section of the AT command documentation to learn more about the definition of the parameters used.

3. To set one module as a receiver (RX), you need to set the value of the P2P receive command to 65535.

```
AT+PRECV=65535
```

With one module configured as RX, the other device will be the TX. You can now try to send a P2P payload.

```
AT+PSEND=11223344
```

NOTE:

- If the `AT+PRECV` value is set to 65534, the device will continuously listen to P2P LoRa TX packets without any timeout. This is the same as setting the device in RX mode.
- If the `AT+PRECV` value is set to 65535, the device will listen to P2P TX without a timeout. But it will stop listening once a P2P LoRa packet is received to save power.
- If the `AT+PRECV` value is 0, the device will stop listening to P2P TX data. The device is in TX mode.

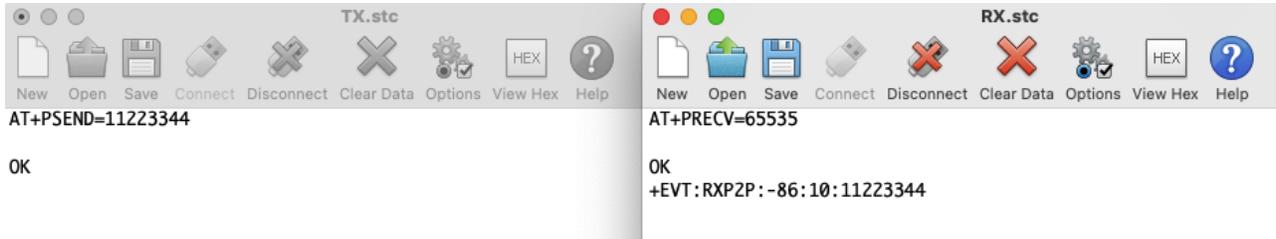


Figure 96: P2P Sending and Receiving LoRa packets

AT Command over BLE

This section shows how to use AT Commands over BLE using a Serial Bluetooth Terminal.

All available commands can be found in the [AT Command Manual](#) section of this document.

1. Download and install the Serial Bluetooth Terminal to connect the device.
2. Make sure the Bluetooth on your mobile is turned on.
3. Select **Category** then **Devices**.

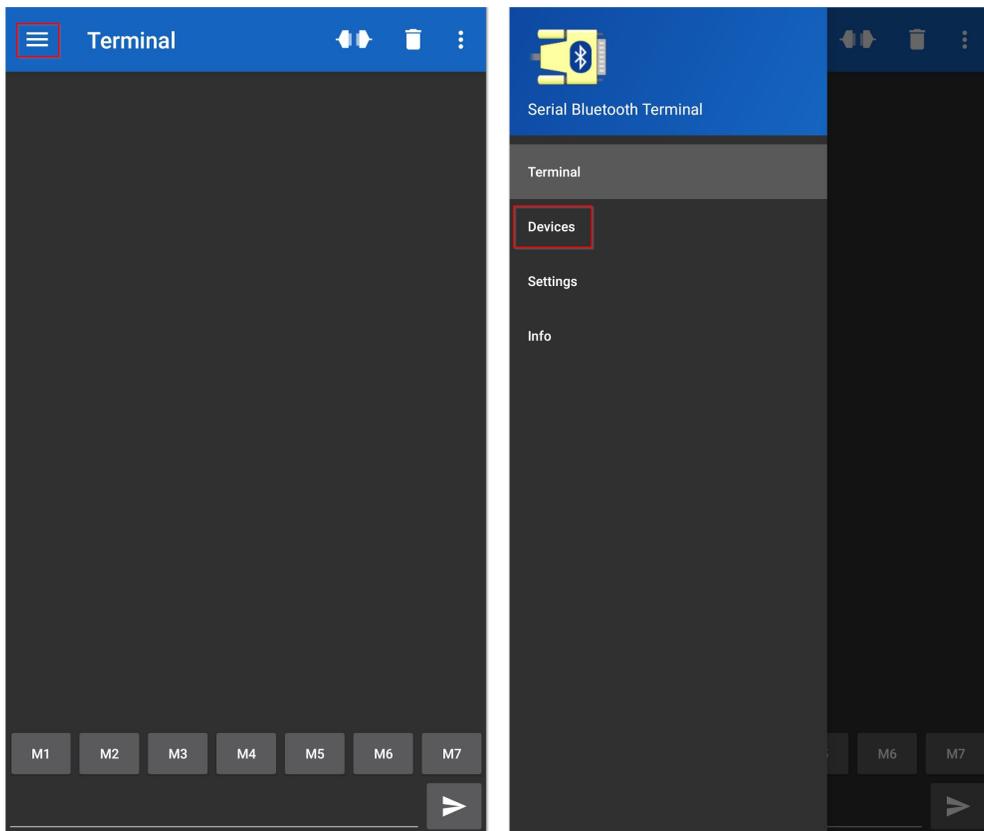


Figure 97: Available Serial Bluetooth Terminal

4. Select the **Bluetooth LE** icon and click the **SCAN** icon to scan the device.
5. Look for a BLE Device named "**RAK.XXXXXX**" in the scanner list of the app and connect to this device.

NOTE

By default, the BLE signal of the RAK4630 is turned off automatically if no connection is established after 30 seconds. Connect to the BLE signal of the RAK4630 immediately after pressing the reset button.

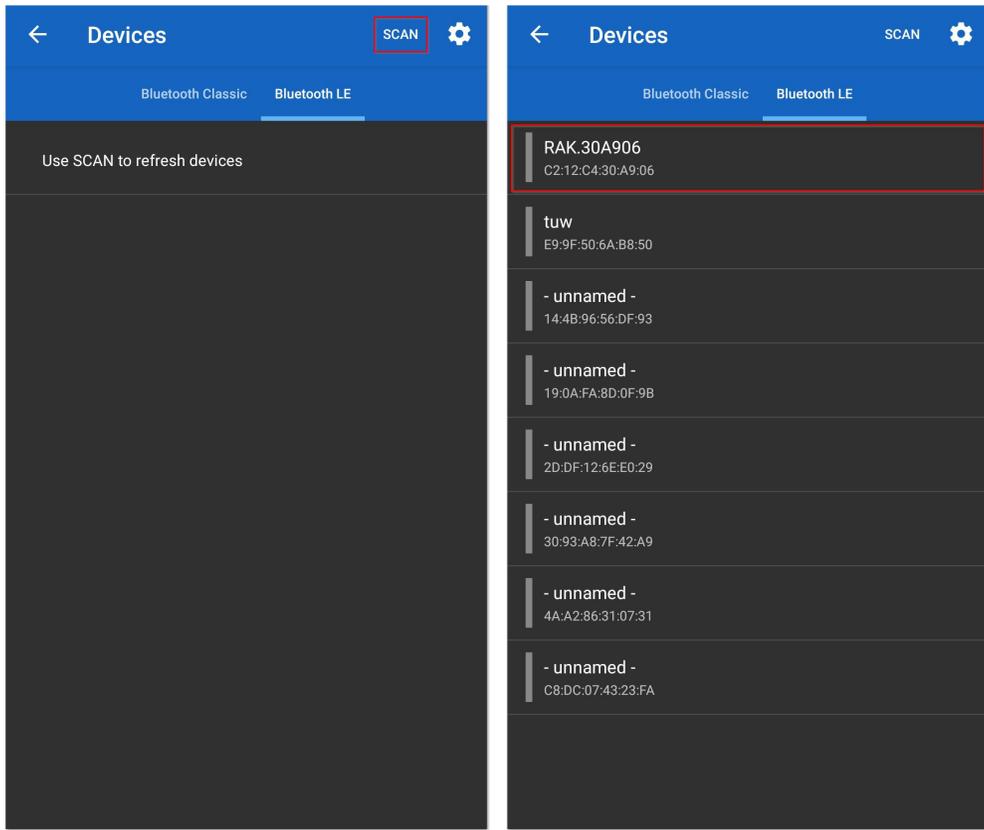


Figure 98: Scanning devices

6. Make sure the connection is successful with "RAK.XXXXXX".

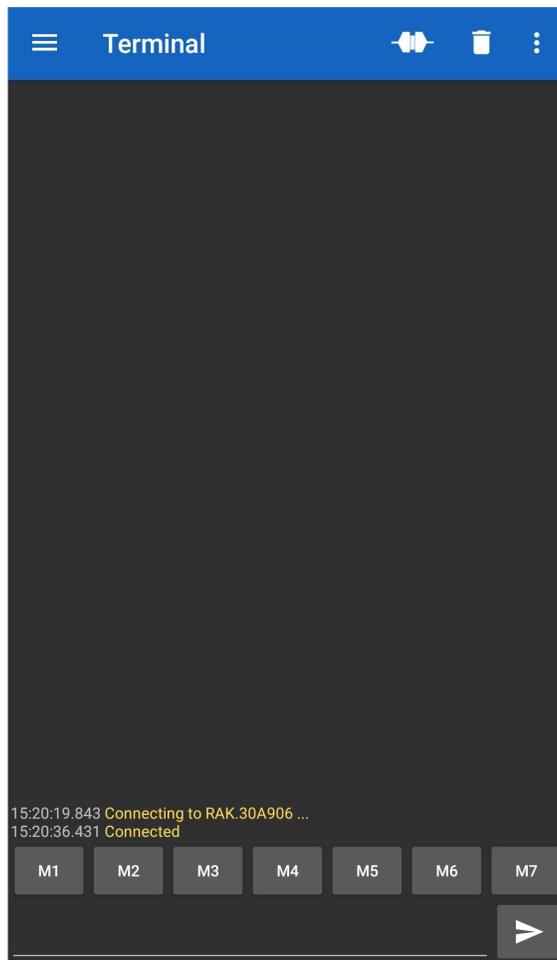


Figure 99: Connect with the device

7. Send an AT Command and check remote console is received or not.

at+ver=?

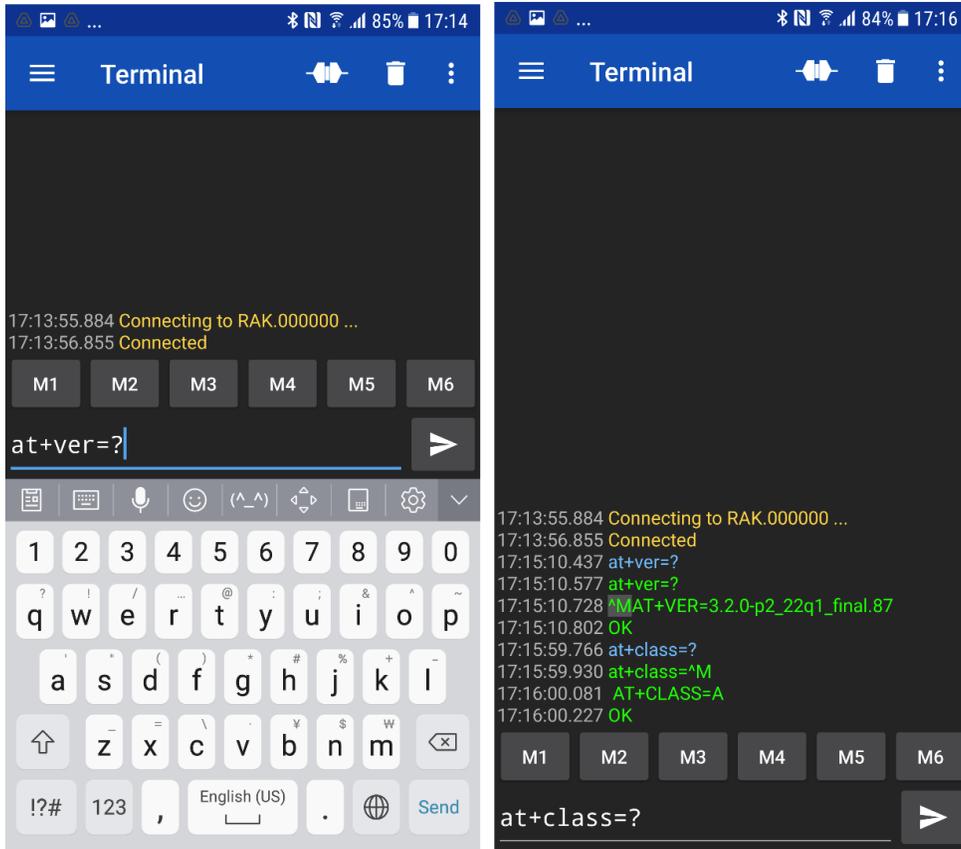


Figure 100: RAK4630 default Bluetooth ID after resetting

8. The remote device will receive the same AT Commands.

```
at+ver=?
AT+VER=3.2.0-p2_22q1_final.87

OK

at+class=?
AT+CLASS=A

OK
```

Miscellaneous

Arduino Installation

Go to [Arduino official website](#) and download the Arduino IDE. You can see the multiple versions available for Windows, Linux, and Mac OS X. Choose the correct version of Arduino IDE and download it.



Arduino IDE 1.8.16

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 [Get](#)

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes Checksums (sha512)

Figure 101: Arduino IDE latest version

For Windows

NOTE

For Windows 10 users:

Do **NOT** install the Arduino IDE from the Microsoft App store. Install the original Arduino IDE from the Arduino official website instead, since the Arduino app from the Microsoft App Store has problems using third-party Board Support Packages.

1. Install the Arduino IDE, which you just downloaded, on your Windows PC.
2. Click **I Agree** then **Next** to proceed.

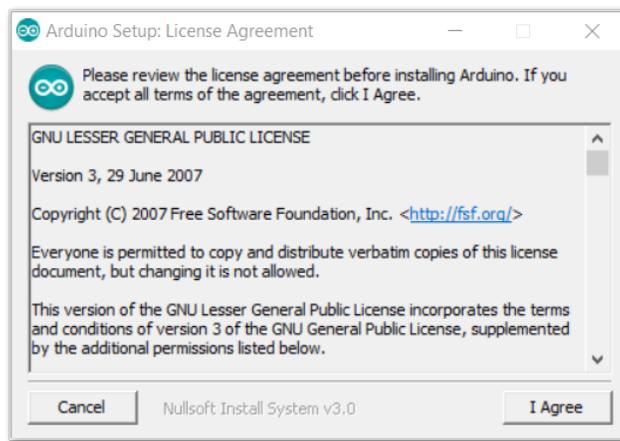


Figure 102: Arduino setup license agreement

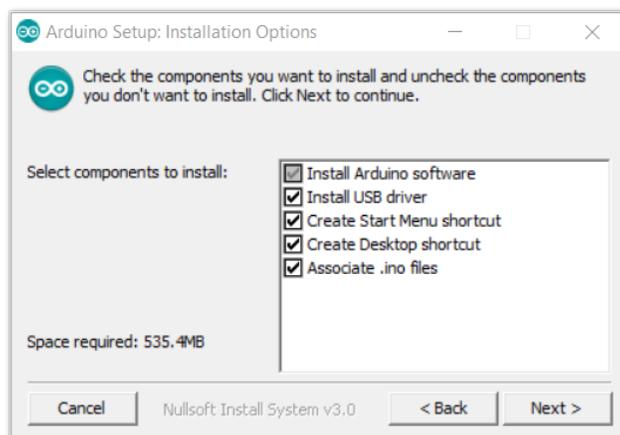


Figure 103: Arduino setup installation options

3. Click **Install**.

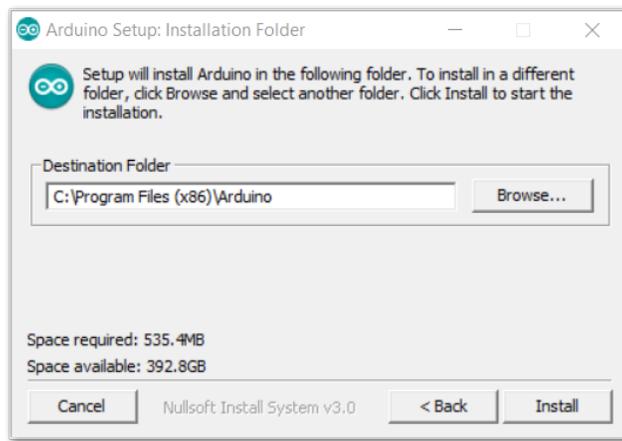


Figure 104: Installing Arduino IDE

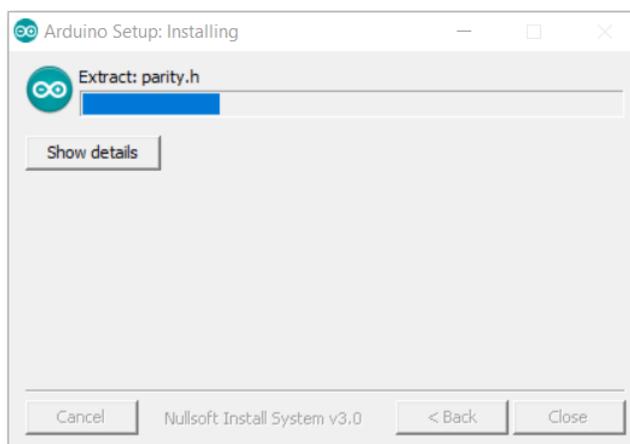


Figure 105: Ongoing installation

After 100% progress, the Arduino IDE has been installed successfully.

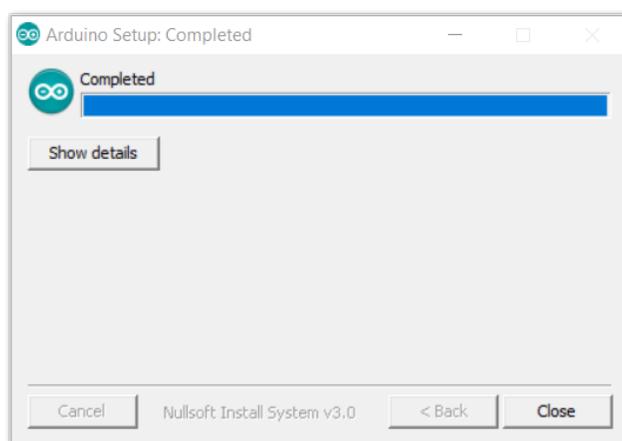


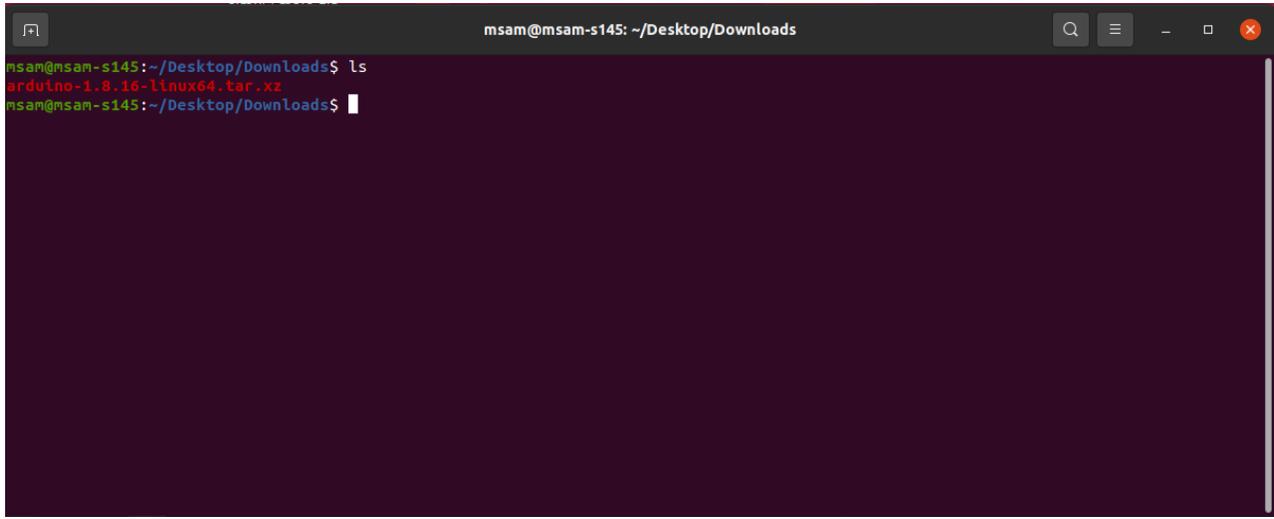
Figure 106: Successful installation

For Linux

First, you need to check the compatibility with your system and choose between the 32-bit, 64-bit, and ARM versions of the Arduino IDE for Linux.

Installing via a tarball

After downloading the correct Arduino version, open a terminal, then run `ls` to check the installation file on the download folder.

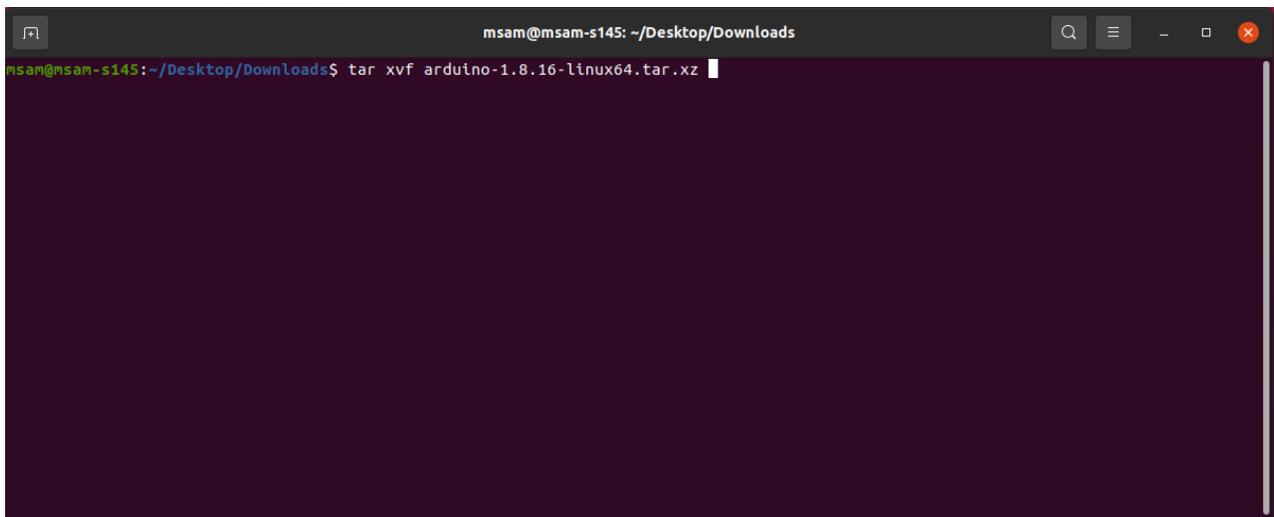


```
msam@msam-s145: ~/Desktop/Downloads
msam@msam-s145:~/Desktop/Downloads$ ls
arduino-1.8.16-linux64.tar.xz
msam@msam-s145:~/Desktop/Downloads$
```

Figure 107: Check the download folder

A tarball is a type of compressed folder, like a `.zip` file, commonly used to distribute software in Linux. To extract the files from the tarball, change the directory to where the downloaded tarball is, then run:

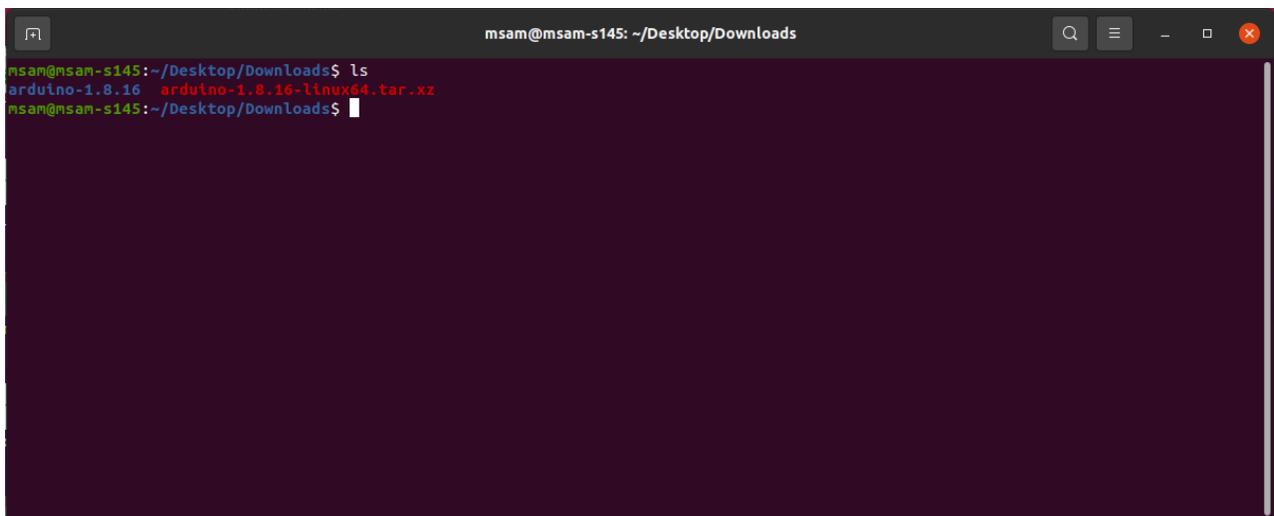
```
tar xvf arduino-version.xz
```



```
msam@msam-s145: ~/Desktop/Downloads
msam@msam-s145:~/Desktop/Downloads$ tar xvf arduino-1.8.16-linux64.tar.xz
```

Figure 108: Tarball extract command

When the tar command finishes, run `ls` again. A folder named **arduino-version** will be created.



```
msam@msam-s145: ~/Desktop/Downloads
msam@msam-s145:~/Desktop/Downloads$ ls
arduino-1.8.16  arduino-1.8.16-linux64.tar.xz
msam@msam-s145:~/Desktop/Downloads$
```

Figure 109: Arduino install folder created

Change the current directory and go to the newly created folder directory. There will be a file named `install.sh` in the folder. Execute `sudo ./install.sh` to install the Arduino IDE.

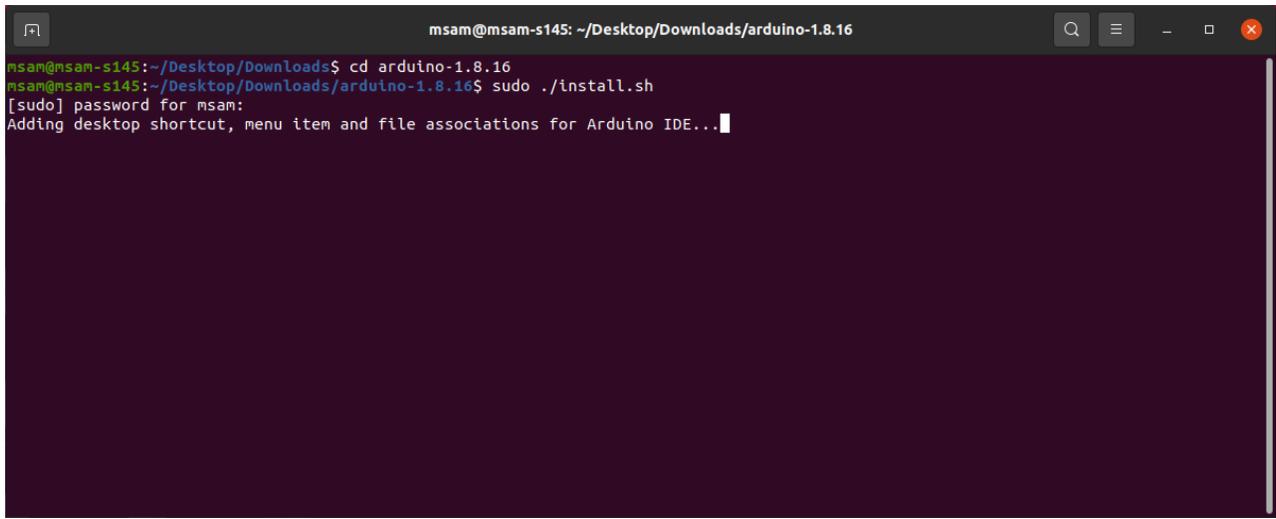


Figure 110: Arduino install script running

The `sudo` command temporarily elevates privileges allowing the installer to complete sensitive tasks without logging in as the root user.

For Mac OS X

In Mac OS X, the same as Linux, there is no installation process. It is just a process of decompression, then you can open Arduino IDE successfully.

Arduino IDE Parts Guide

Figure 111 shows the five (5) parts of Arduino IDE.

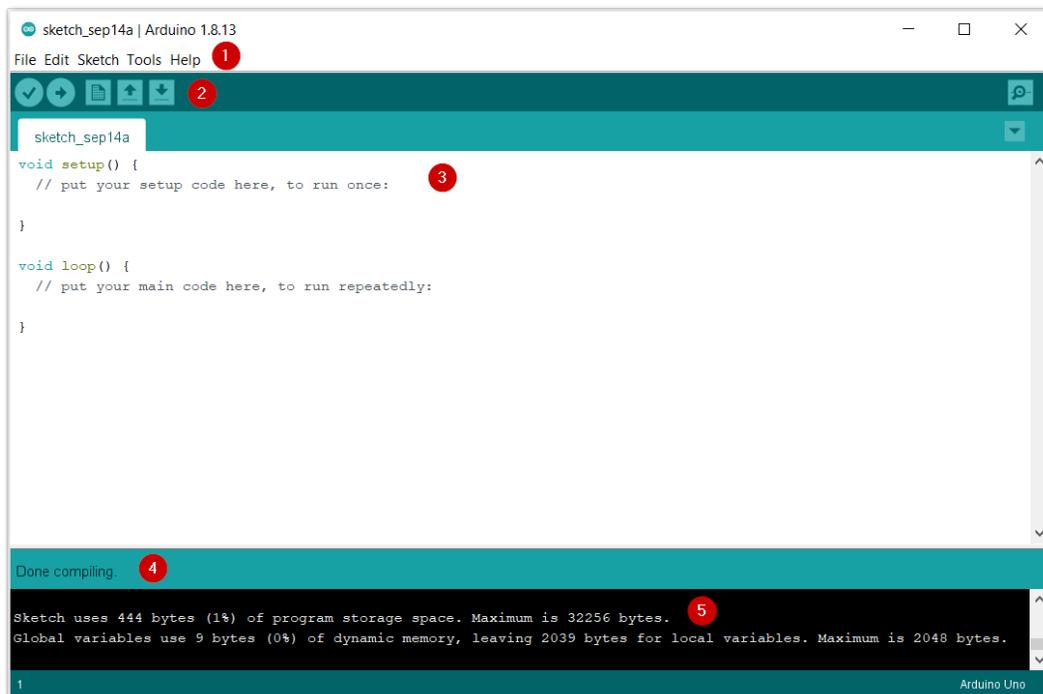


Figure 111: Arduino IDE

1. IDE Option Menu

You can configure some general parameters such as the serial port, the board information, the libraries, the edit parameters, and so on.

2. Operating Buttons

The operating buttons have five operations:

- **Verify/Compile** the source code.
- **Upload** the compiled code into WisBlock.

- **Open** a **New** Arduino IDE window or existing application.
- **Save** the current application.

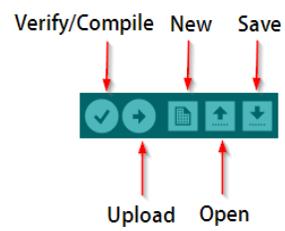


Figure 112: Operating buttons

3. Code Area

You can edit the source code, which will be compiled and uploaded into WisBlock later in this area.

4. State Area

5. **Output Message Area** You can see the output message in this area, whether it's failure or success information.