RAK3172 Module Quick Start Guide

This guide covers the following topics:

Prerequisites

What Do You Need?

Before going through the steps in the installation guide of the RAK3172 WisDuo LPWAN Module, make sure to prepare the necessary items listed below:

Hardware

- RAK3172 WisDuo LPWAN Module
- Computer
- USB to UART TTL adapter

Software

- Download and install the Arduino IDE .

> ⚠ **WARNING**
>
> ***If you are using Windows 10***.
> Do ***NOT*** install the Arduino IDE from the Microsoft App Store. Instead, install the original Arduino IDE from the Arduino official website. The Arduino app from the Microsoft App Store has problems using third-party Board Support Packages.

- Add RAK3172 as a supported board in Arduino IDE by updating Board Manager URLs in **Preferences** settings of Arduino IDE with the JSON URL below.

```json
https://raw.githubusercontent.com/RAKWireless/RAKwireless-Arduino-BSP-Index/main/package_rakwirel
```

After that, you can then add **RAKwireless RUI STM32 Boards** via Arduino board manager.

- RAK Serial Port Tool

List of Acronyms

Acronym	Definition
DFU	Device Firmware Upgrade
JTAG	Joint Test Action Group
LoRa	Long Range
OTAA	Over-The-Air-Activation (OTAA)
ABP	Activation-By-Personalization (ABP)
TTN	The Things Network
DEVEUI	Device EUI (Extended Unique Identification)
APPEUI	Application EUI (Extended Unique Identification)
APPKEY	Application Key
DEVADDR	Device Address
NWKSKEY	Network Session Key
APPSKEY	Application Session Key
P2P	Point-to-Point
MSB	Most Significant Bit
LNS	LoRaWAN Network Service

Product Configuration

RAK3172 as a Stand-Alone Device Using RUI3

Hardware Setup

The RAK3172 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.

> ⚠️ **WARNING**
>
> Firmware update is done via UART2 pins. If you will connect the module to an external device that will be interfacing with UART2, take extra precautions in your board design to ensure you can still perform FW update to it. There should be a way in your board design that can disconnect the external device to RAK3172 UART2 before connecting the module to the PC (via USB-UART converter) for the FW update process.
>
> An alternative option to update firmware aside from UART2 is to use SWD pins (SWCLK & SWDIO). This method will require you to use external tools like ST-LINK and RAKDAP1.

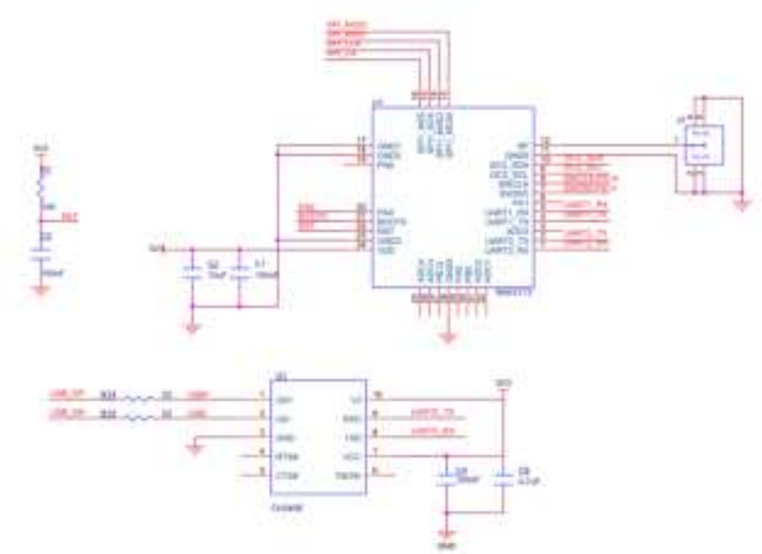

Figure 1: RAK3172 Minimum Schematic

Ensure that the antenna is properly connected to have a good LoRa 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 connector.

Figure 2: LoRa Antenna

RAK3172 has a module variant with an IPEX connector where you can connect the LoRa antenna, as shown in **Figure 3**. If the RAK3172 module you ordered is the variant with no IPEX connector, you need to ensure that there is an external antenna connected to **RF pin** (PIN 12) of the module.

Figure 3: IPEX Connector of RAK3172 for LoRa Antenna

> 📝 **NOTE**
>
> Detailed information about the RAK3172 LoRa antenna can be found on the antenna datasheet .

> ⚠️ **WARNING**
>
> When using the LoRa transceiver, make sure that an antenna is always connected. Using this transceiver without an antenna can damage the module.

Software Setup

The default firmware of RAK3172 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 STM32 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 UART. The AT commands of RAK3172 is still available even if you compile custom firmware via RUI3. You can send AT commands via UART2 connection.

RAK3172 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 RAK3172 in its board selection by following these steps.

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

Figure 4: Arduino preferences

2. To add the RAK3172 to your Arduino Boards list, edit the **Additional Board Manager URLs**. Click the icon, as shown in **Figure 5**.

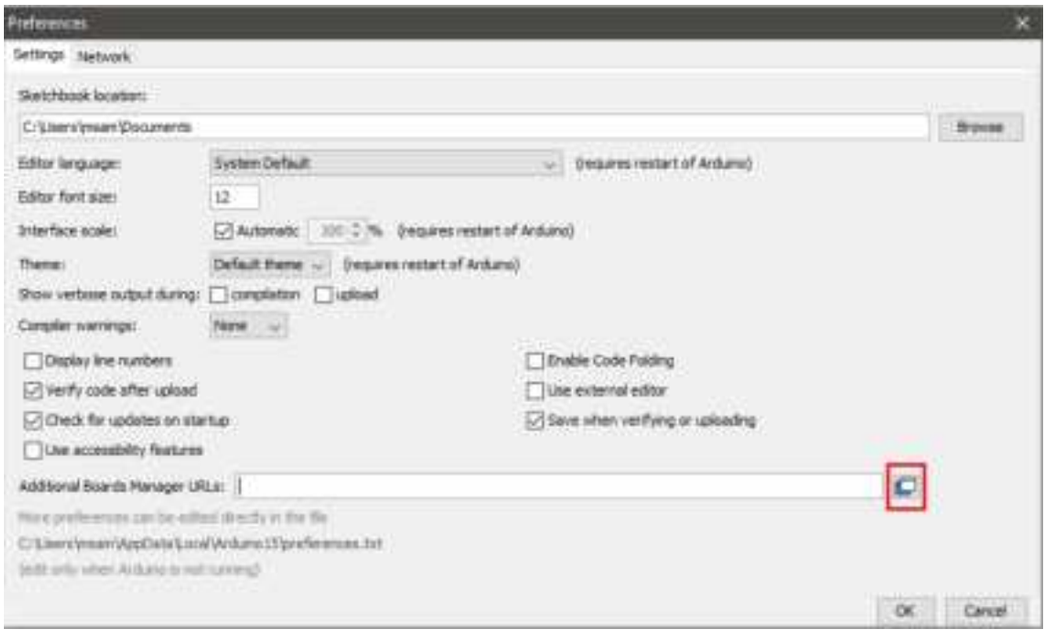

Figure 5: Modifying Additional Board Manager URLs

3. Copy the URL below and paste it on the field, as shown in **Figure 6**. If there are other URLs already there, just add them on the next line. After adding the URL, click **OK**.

```json
https://raw.githubusercontent.com/RAKWireless/RAKwireless-Arduino-BSP-Index/main/package_rakwirel
```

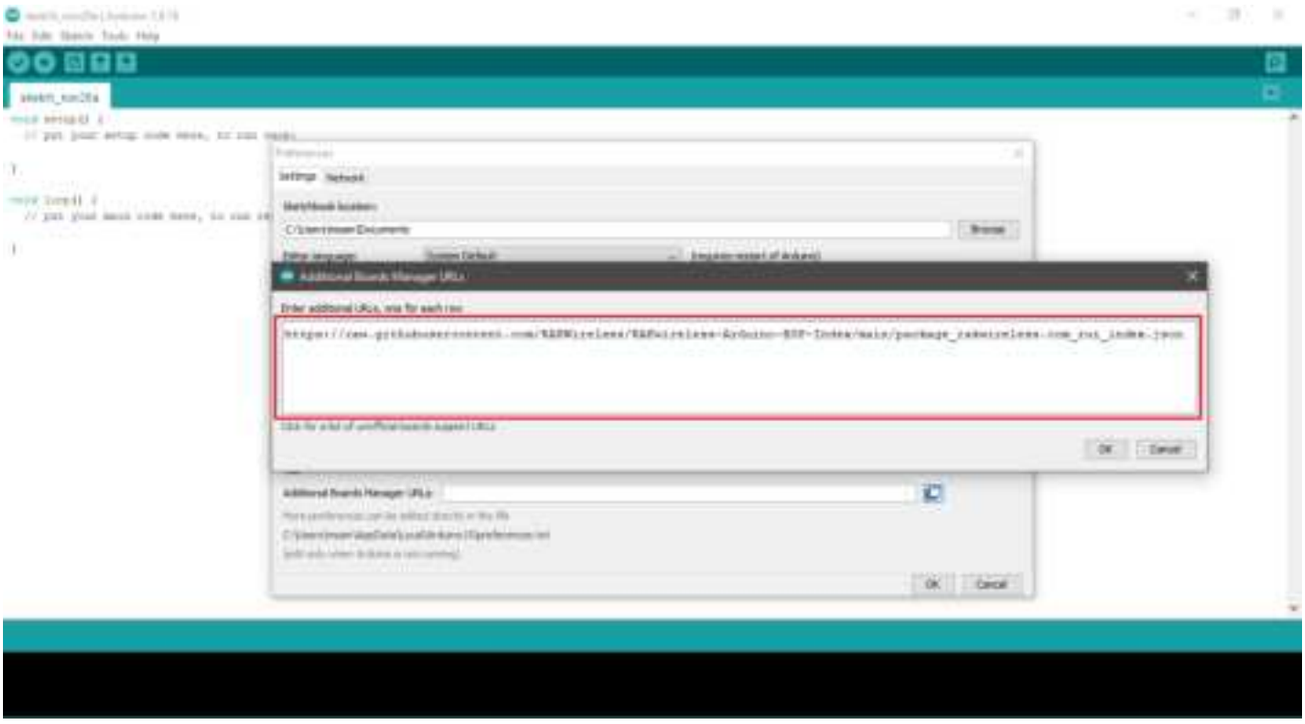

Figure 6: Add additional board manager URLs

4. Restart the Arduino IDE.

5. Open the Boards Manager from Tools Menu.

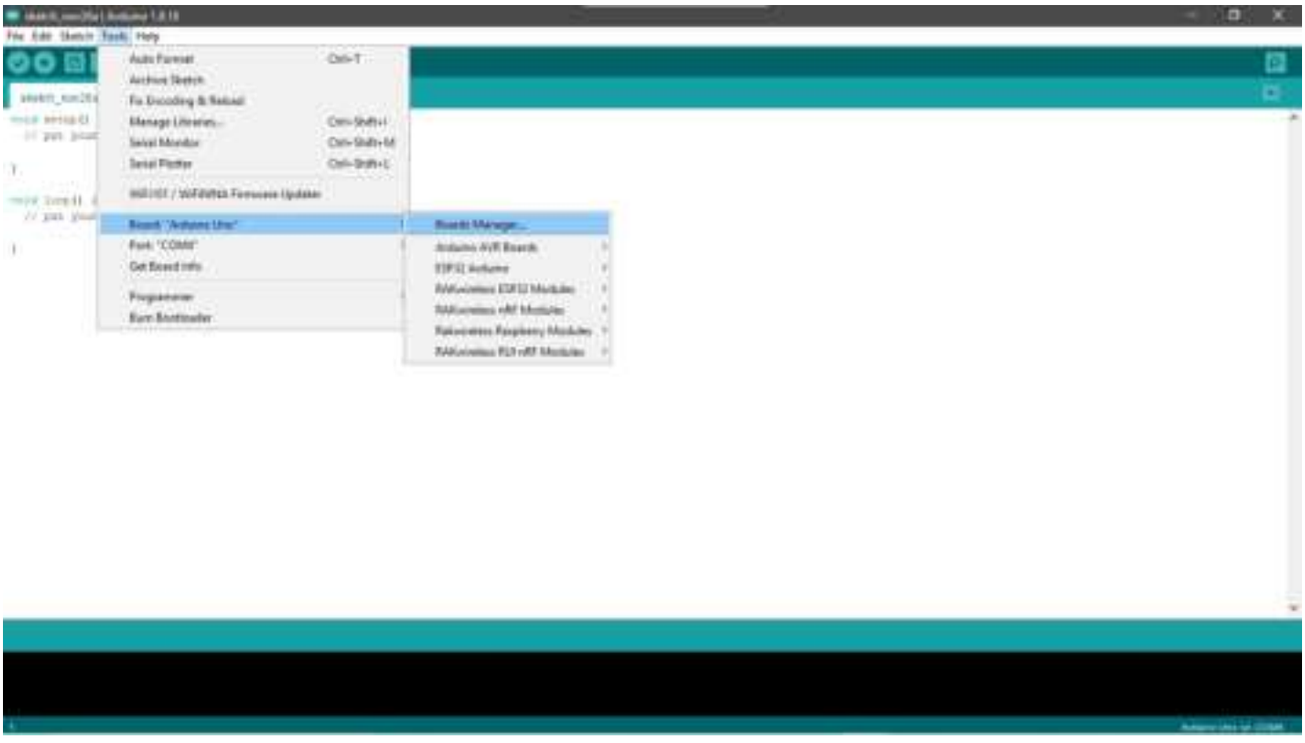

Figure 7: Opening Arduino boards manager

6. Write `RAK` in the search bar, as shown in **Figure 8**. This will show the available RAKwireless module boards that you can add to your Arduino Board list.

7. Click on the area highlighted in blue to select **RAKwireless RUI STM32 Boards**. Install the latest version of the **RAKwireless RUI STM32 Boards** by clicking on **Install** button.

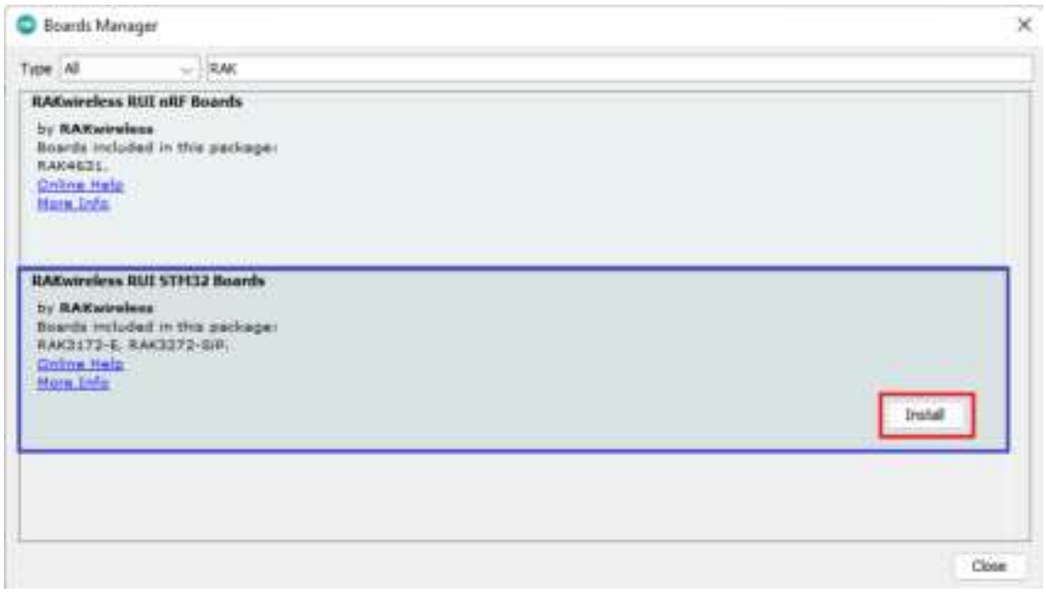

Figure 8: Installing RAKwireless RUI STM32 Boards

Configure RAK3172 on Boards Manager

8. Once the BSP is installed, select **Tools** > **Boards Manager** > **RAKWireless RUI STM32 Modules** > **WisDuo RAK3172 Evaluation Board**. The RAK3172 Evaluation board uses RAK3172 WisDuo module.

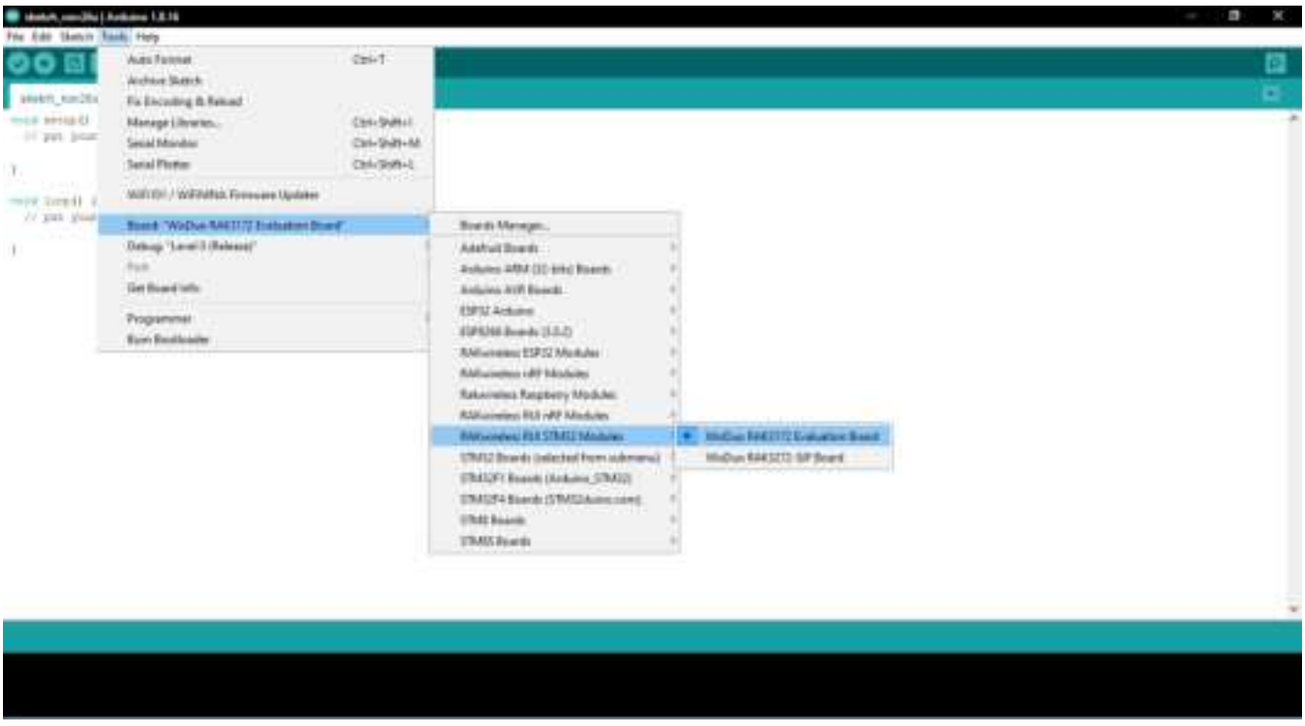

Figure 9: Selecting RAK3172 Module

RAK3172 COM Port on Device Manager

Connect the RAK3172 via UART and check RAK3172 COM Port using Windows **Device Manager**. Double-click the reset button if the module is not detected.

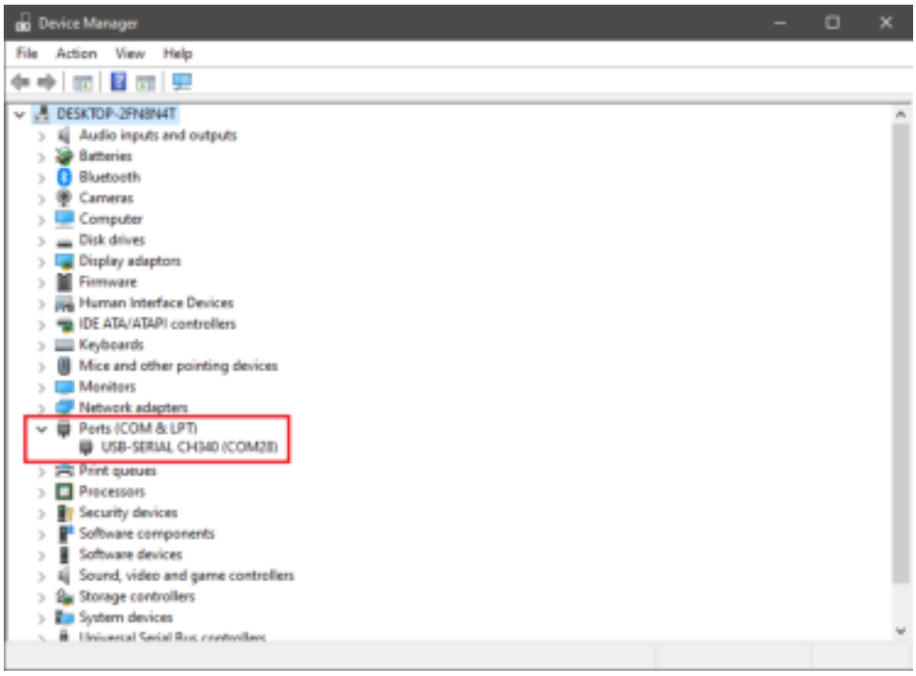

Figure 10: Device manager ports (COM & LPT)

Compile an Example with Arduino LED Breathing

1. After completing the steps on adding your RAK3172 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 RAK3172 module, as shown in **Figure 11**.

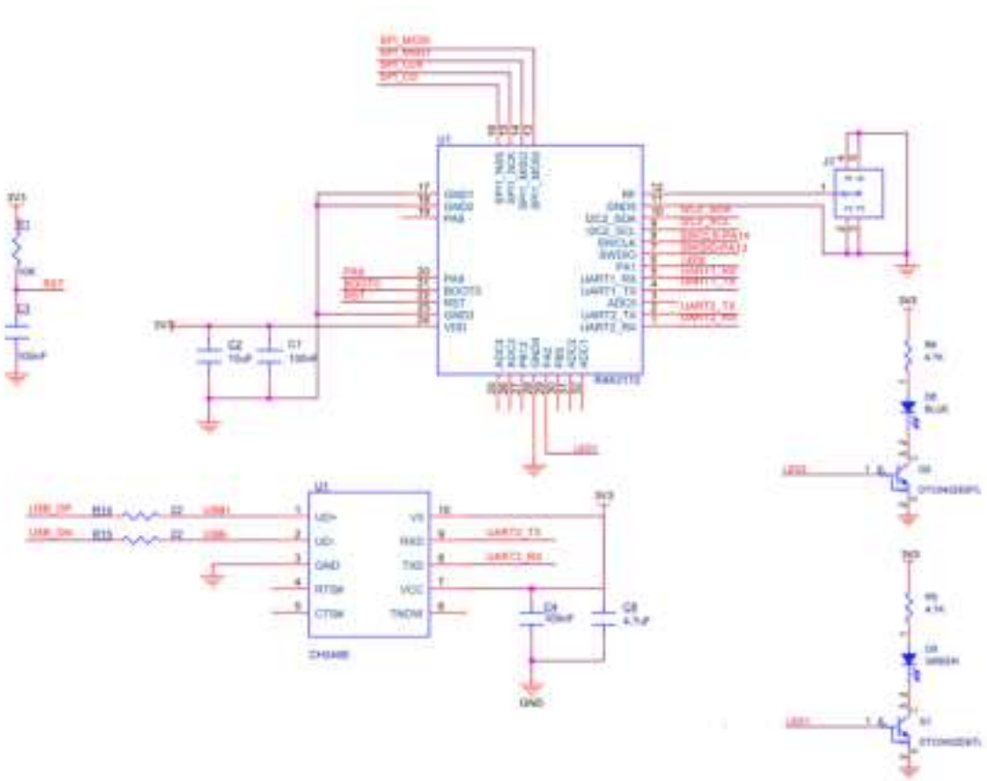

Figure 11: RAK3172 with two LEDs

2. Launch Arduino IDE and configure WisDuo RAK3172 Evaluation Board on board selection. See **Figure 9**.

3. Connect the RAK3172 via UART and check RAK3172 COM Port. See **Figure 10**.

4. Open **Tools** Menu and select a COM port. **COM28** is currently used.

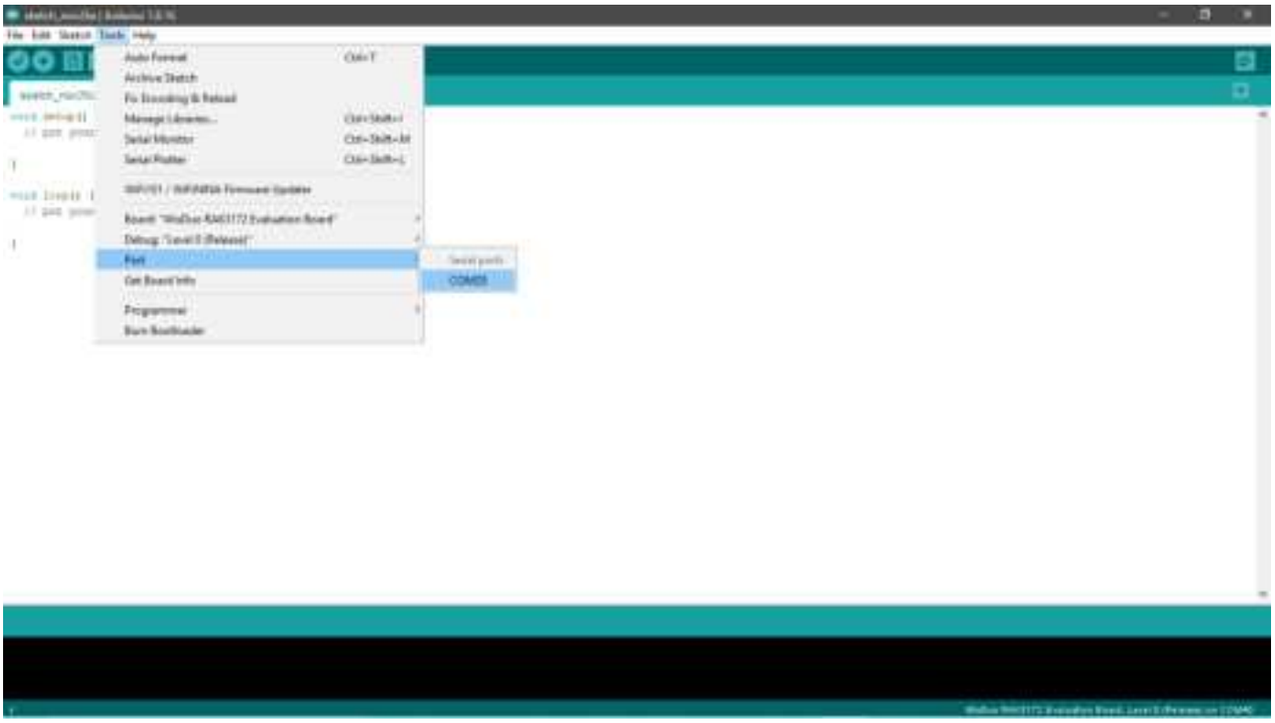

Figure 12: Select COM port

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

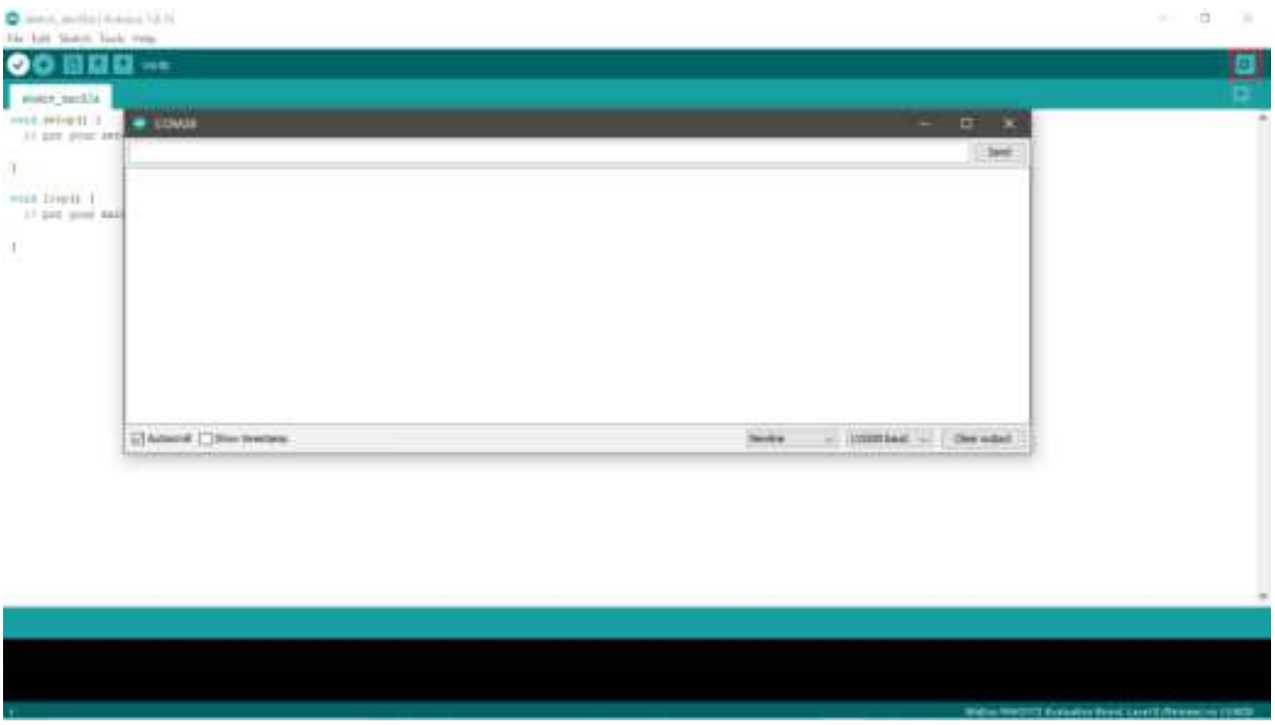

Figure 13: Open Arduino serial monitor

6. If the connection is successful, you can send AT Commands to RAK3172. For example: To check the RUI version, type `AT+VER=?` on the text area, then click on the **Send** button, as shown in **Figure 14**.

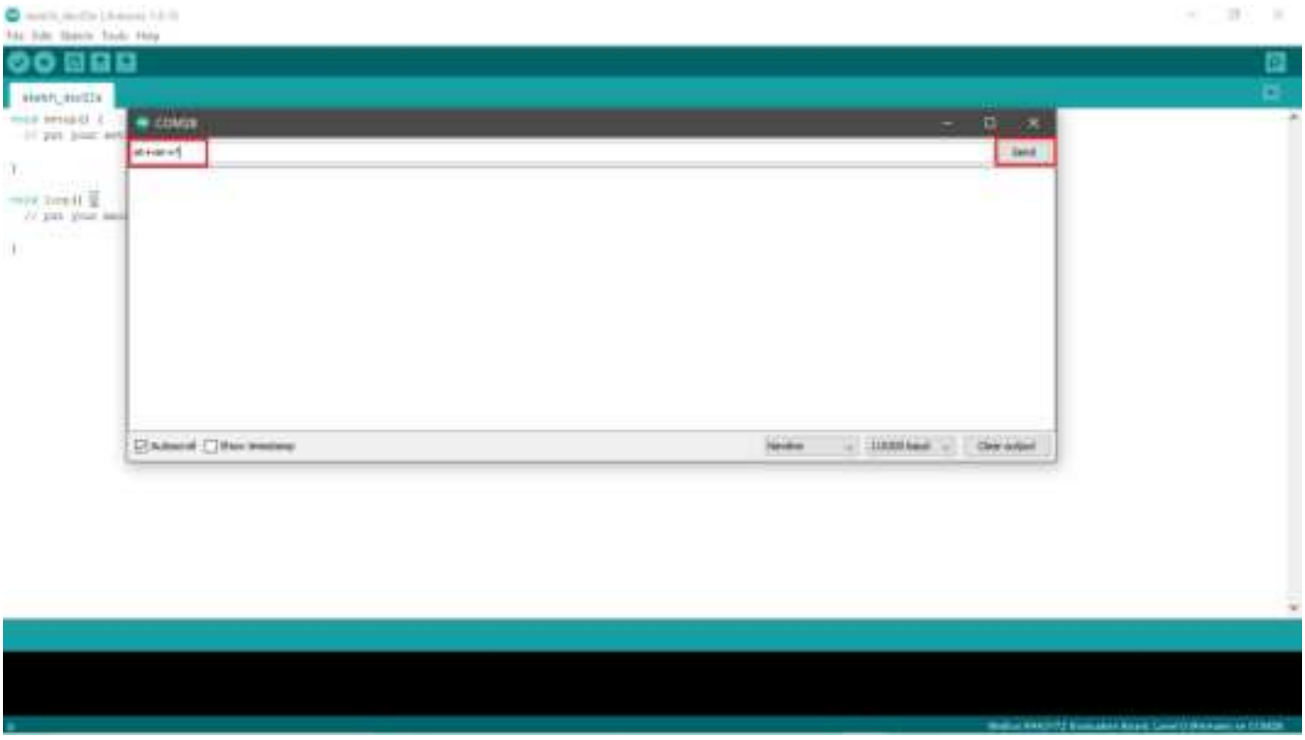

Figure 14: Send AT command

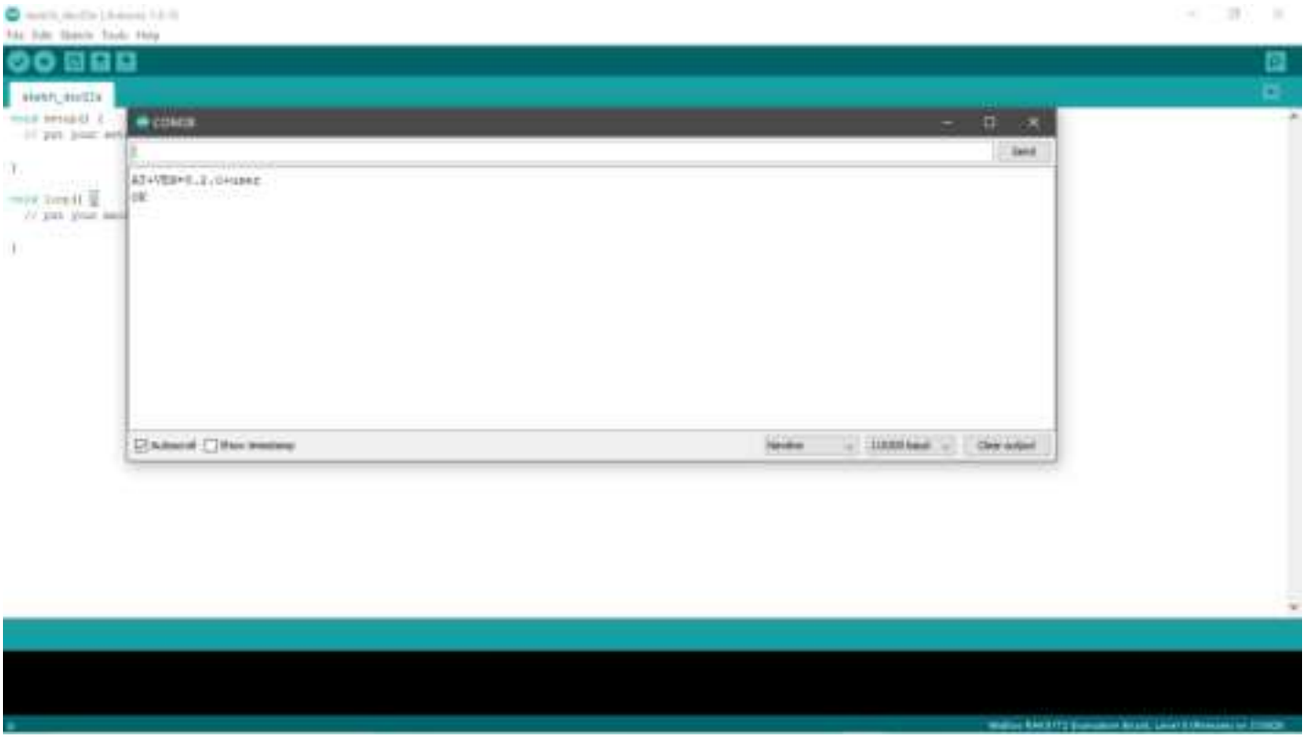

Figure 15: Arduino serial monitor COM28

7. Open **Arduino_Led_Breathing** example code.

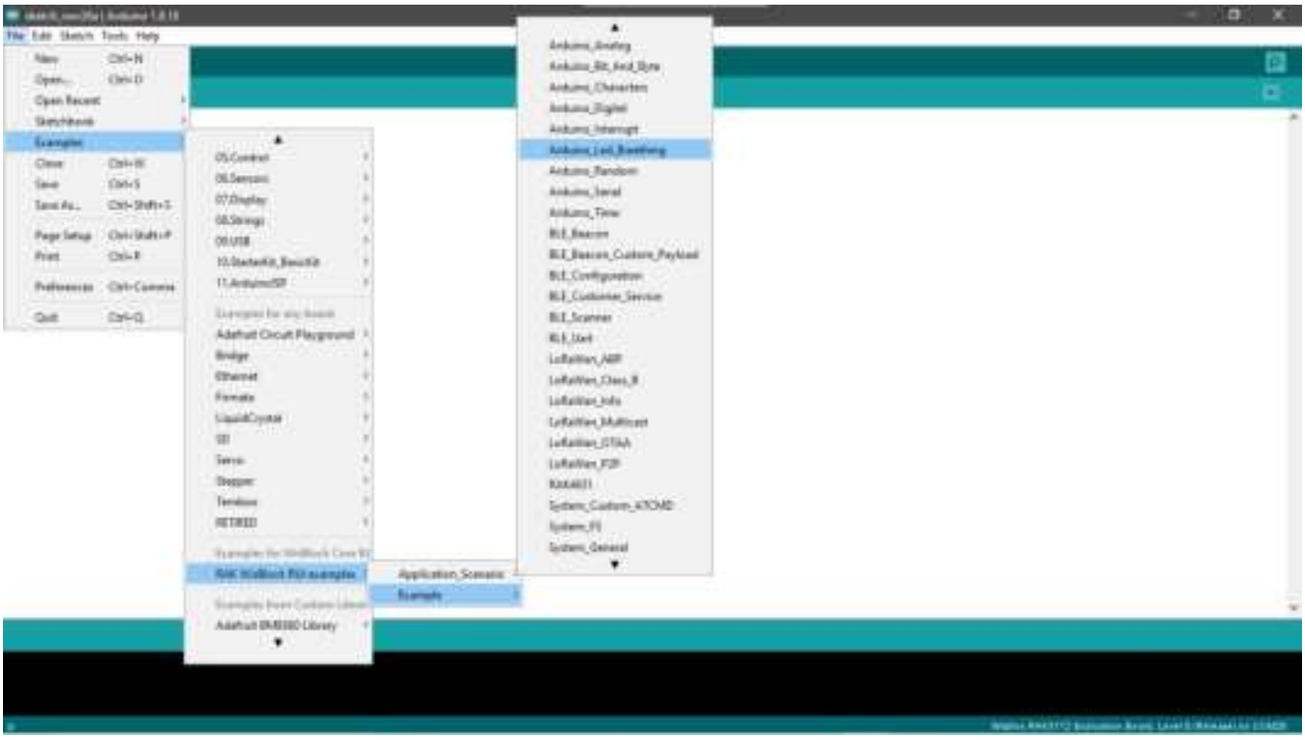

Figure 16: Arduino Led Breathing example

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

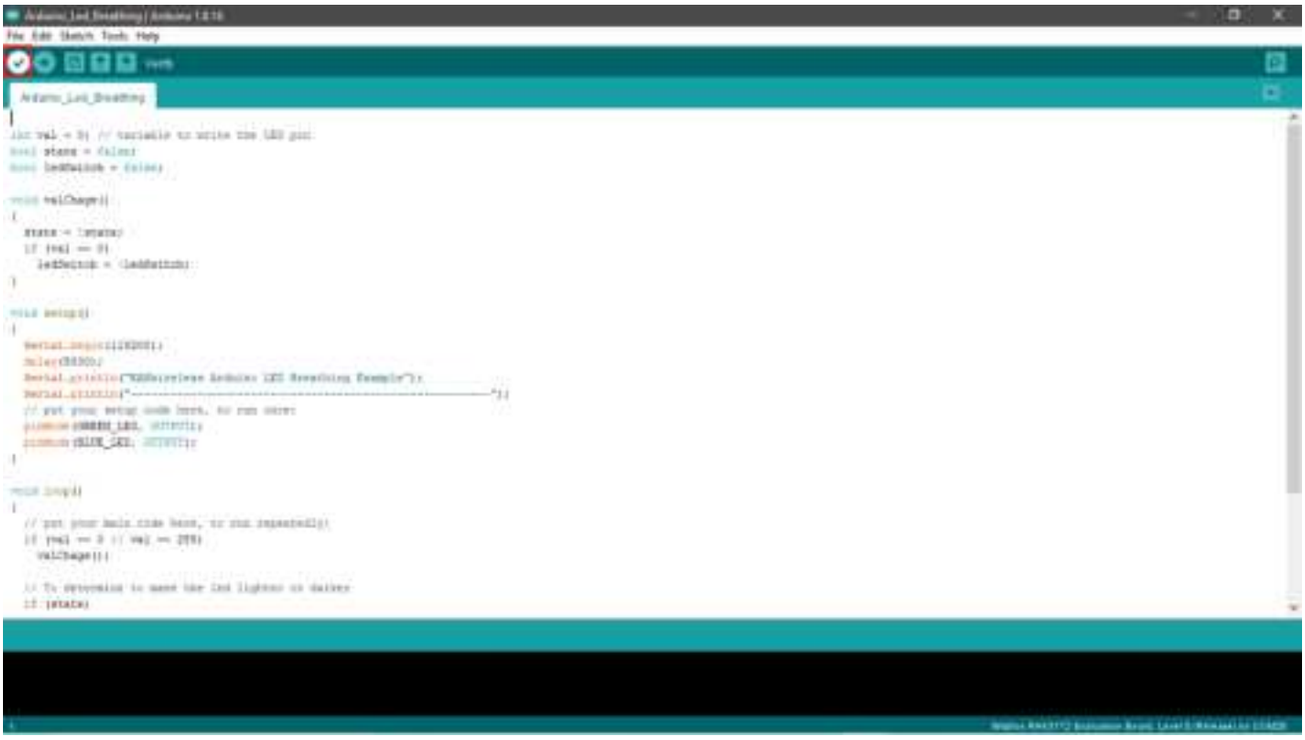

Figure 17: Verify the example code

9. Click the **Upload** icon to send the compiled firmware to your RAK3172 module.

> 📝 **NOTE:**
>
> RAK3172 should automatically go to BOOT mode when the firmware is uploaded via Arduino IDE.
>
> If BOOT mode is not initiated, pull to ground the RESET pin twice (or double click the reset button if available) to force BOOT mode.

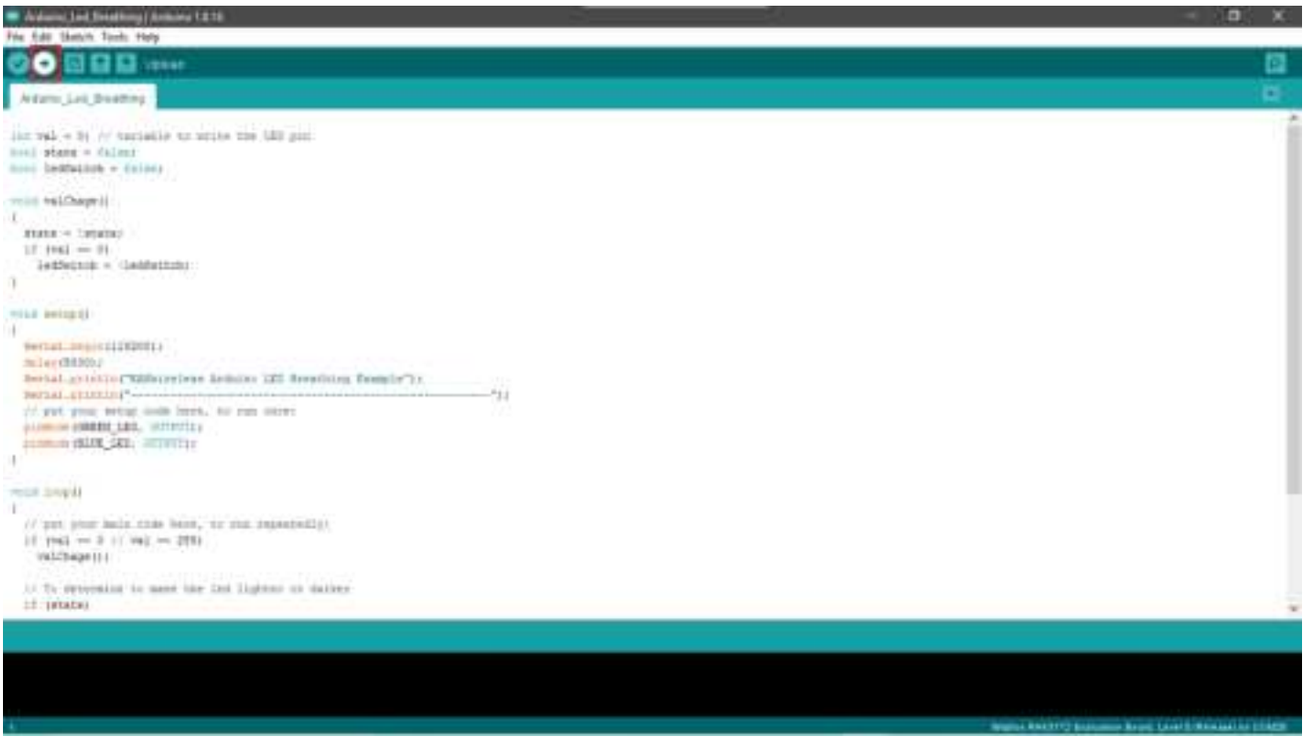

Figure 18: Upload the example code

10. If the upload is successful, you will see the **Upgrade Complete** message.

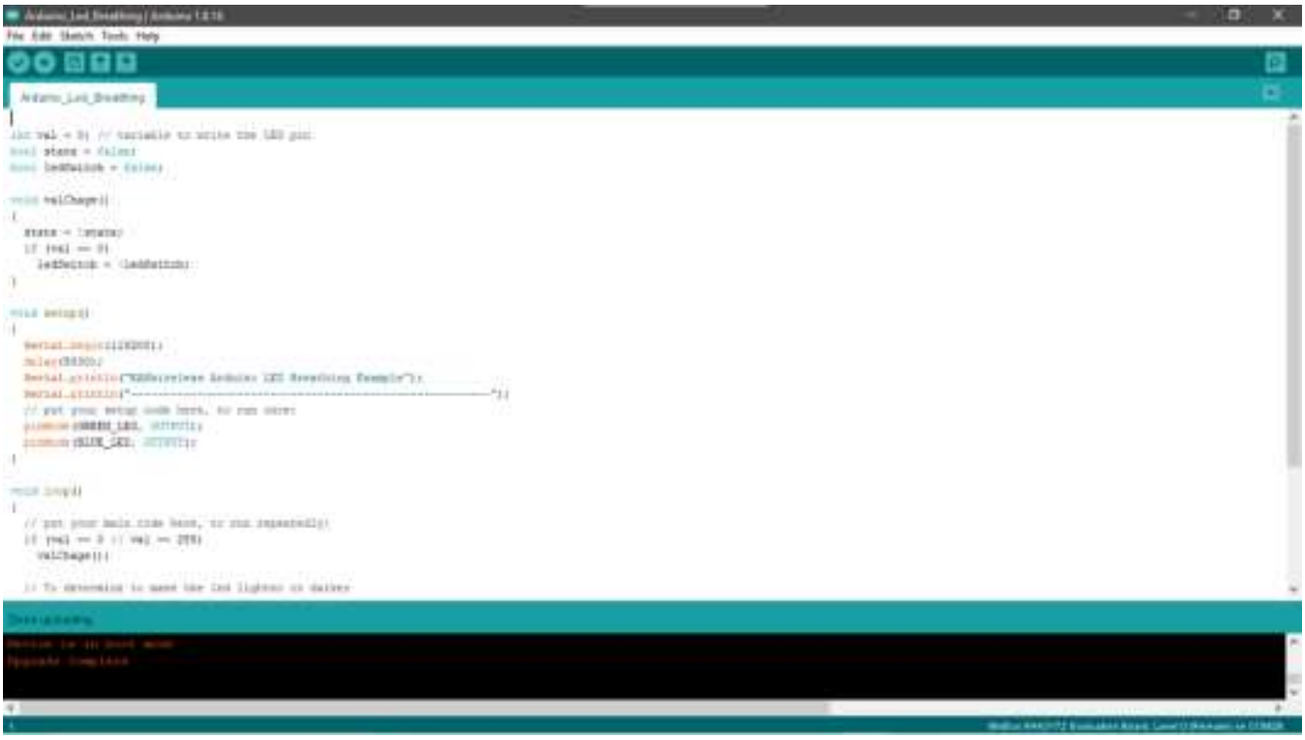

Figure 19: Device programmed successfully

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

RAK3172 I/O Pins and Peripherals

This section discusses how to use and access RAK3172 pins using RUI3 API. It shows basic code on using digital I/O, analog input, UART, and I2C.

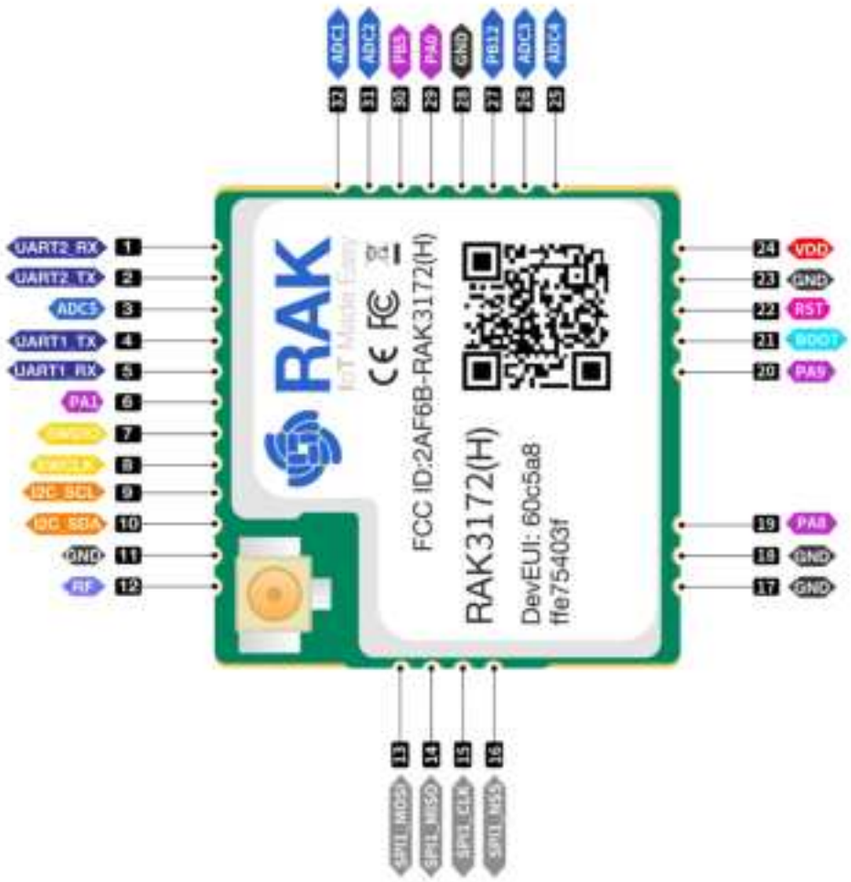

Figure 20: Available Peripherals and Digital I/O pins in RAK3172 module

How to Use Digital I/O

You can use any of the pins below as Digital Pin.

Pin Name	Alternative Pin Usage
PA0	
PA1	
PA4	SPI
PA5	SPI
PA6	SPI
PA7	SPI
PA8	
PA9	I2C_SCL
PA15	

Pin Name	Alternative Pin Usage
PB2	
PB3	ADC1
PB4	ADC2
PB5	
PB6	UART1_TX
PB7	UART1_RX
PB12	

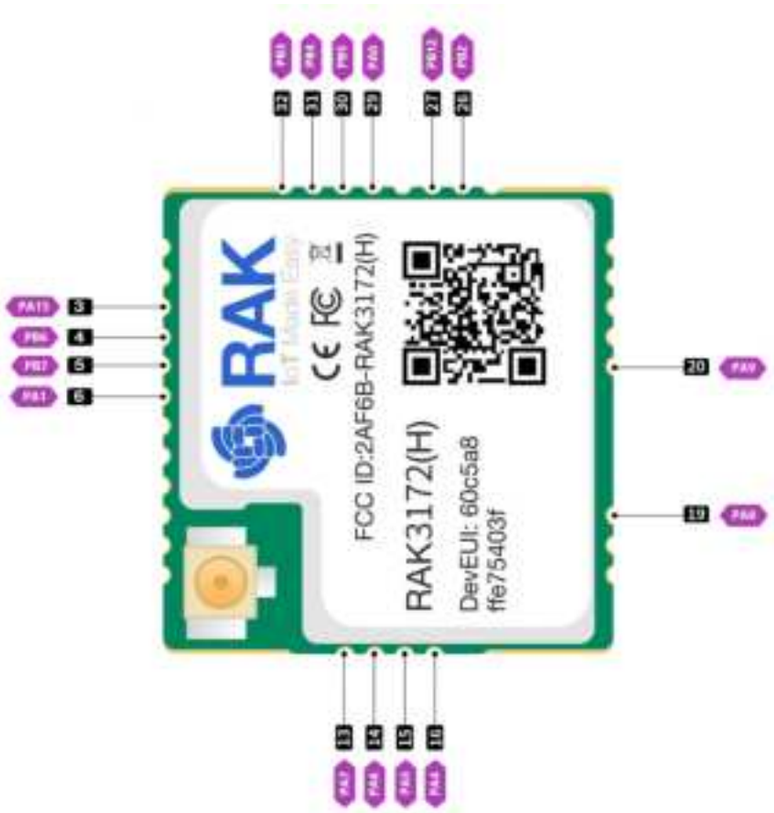

Figure 21: Available Digital I/O pins in RAK3172 module

The pins listed below must not be used.

Pin name	Pin Usage
PA2	UART2_TX
PA3	UART2_RX
PA13	SWDIO
PA14	SWCLK

Pin name	Pin Usage
PB8	RAK3172 Internal

- Use Arduino digitalRead to read the value from a specified Digital I/O pin, either HIGH or LOW.
- Use Arduino digitalWrite to write a HIGH or a LOW value to a Digital I/O pin.

> 📝 **NOTE:**
>
> The GPIO Pin Name is the one to be used on the digitalRead and digitalWrite and NOT the pin numbers.

Example code

```c
void setup()
{
  pinMode(PA0, OUTPUT); //Change the P0_04 to any digital pin you want. Also, you can set this to
}

void loop()
{
  digitalWrite(PA0,HIGH); //Change the PA0 to any digital pin you want. Also, you can set this to
  delay(1000); // delay for 1 second
  digitalWrite(PA0,LOW); //Change the PA0 to any digital pin you want. Also, you can set this to
  delay(1000); // delay for 1 second
}
```

How to Use Analog Input

You can use any of the pins below as Analog Input.

Analog Port	Pin Name
ADC1	PB3
ADC2	PB4

Use Arduino analogRead to read the value from the specified Analog Input pin.

Figure 22: Available Analog pins in RAK3172

Example code

```c
#define analogPin PB3

int val = 0;  // variable to store the value read

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  val = analogRead(analogPin);  // read the input pin
  Serial.println(val);          // debug value
  delay(100);
}
```

How to Use Serial Interfaces

UART

There are two UART peripherals available on the RAK3172 module. There are also different Serial Operating Modes possible in RUI3, namely Binary Mode , AT Mode , and Custom Mode .

Serial Port	Serial Instance Assignment	Default Mode

Serial Port	Serial Instance Assignment	Default Mode
UART1 (pins 4, 5)	Serial1	Custom Mode
UART2 (pins 1, 2)	Serial	AT Command

Figure 23: Available UART pins in RAK3172

Example Code

```c
void setup()
{
  Serial1.begin(115200); // use Serial1 for UART1 and Serial for UART2
                         // you can designate separate baudrate for each.
  Serial.begin(115200);
}

void loop()
{
  Serial1.println("RAK3172 UART1 TEST!");
  Serial.println("RAK3172 UART2 TEST!");
  delay(1000); // delay for 1 second
}
```

I2C

There is one I2C peripheral available on RAK3172.

I2C Pin Number	I2C Pin Name
PA9	I2C_SCL

I2C Pin Number	I2C Pin Name
PA10	I2C_SDA

- Use Arduino Wire library to communicate with I2C devices.

Figure 24: Available I2C pins in RAK3172

Example Code

Make sure you have an I2C device connected to specified I2C pins to run the I2C scanner code below:

```c
#include <Wire.h>

void setup()
{
  Wire.begin();

  Serial.begin(115200);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}


void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);           // wait 5 seconds for next scan
}
```

The Arduino Serial Monitor shows the I2C device found.

```
17:29:15.690 -> Scanning...
17:29:15.738 -> I2C device found at address 0x28  !
17:29:15.831 -> done
17:29:15.831 ->
17:29:20.686 -> Scanning...
17:29:20.733 -> I2C device found at address 0x28  !
17:29:20.814 -> done
17:29:20.814 ->
```

SPI

If your RUI3 project uses SPI, then PA4 to PA7 pins are reserved for RUI3 SPI interface.

> 📝 **NOTE:**
>
> PA13 and PA14 pins are reserved for SWD debug interface. Check the Connect to the RAK3172 section.

LoRaWAN Example

This example illustrates how to program RAK3172 module as a stand-alone LoRaWAN end-device via RUI3 Arduino APIs . To use RAK3172 module 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.

> 📝 **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
>
> RAKwireless 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 .
>
> To correctly run this example, it is necessary to configure the LoRaWAN parameters and create an OTAA application on your LoRaWAN gateway.

Register the LoRaWAN Gateway on TTNv3 Community Edition

After configuring your gateway, you need to register it in TTNv3:

1. Login to TTNv3 Network Server with a web browser.

- Europe
- North America
- Australia

2. Navigate to the **Console** page and click on **gateway** icon, as shown in **Figure 20**.

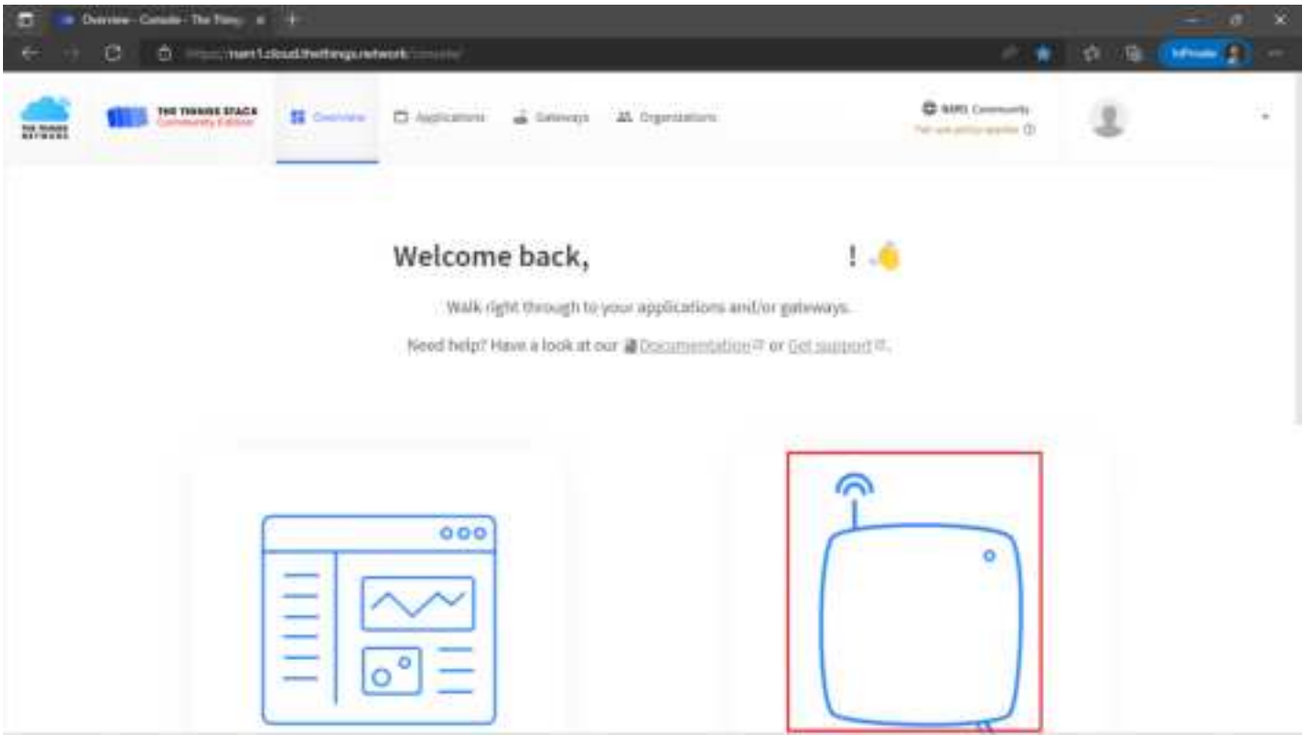

Figure 25: TTNv3 gateway registration and configuration

3. On **General Settings**, enter the **Gateway ID**, **Gateway EUI**, and **Gateway Name**. This information is available in your LoRaWAN gateway configuration. Select the **Gateway Server address** according to the region where the LoRaWAN gateway will be installed.

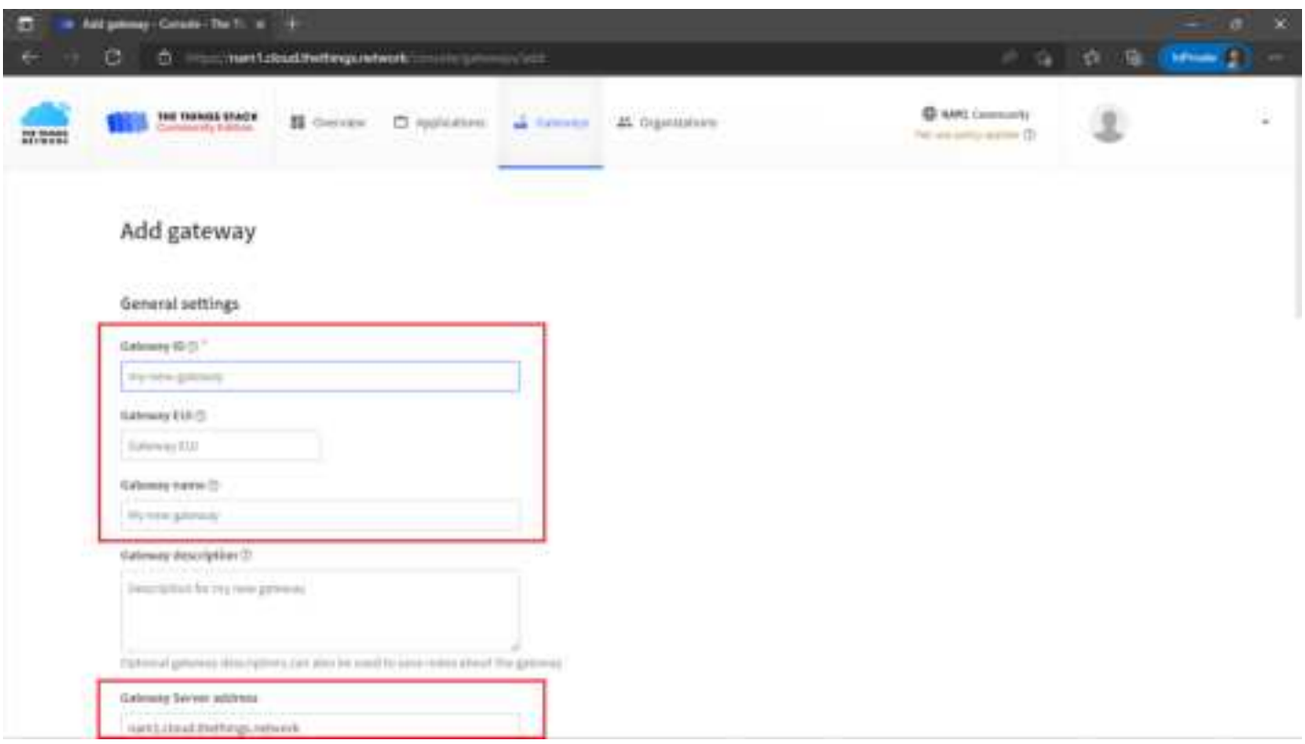

Figure 26: TTNv3 gateway registration and configuration

4. Select the **Frequency plan** for your region (with used by TTN), then click on the **Create gateway** button. This will add a new gateway to TTNv3.

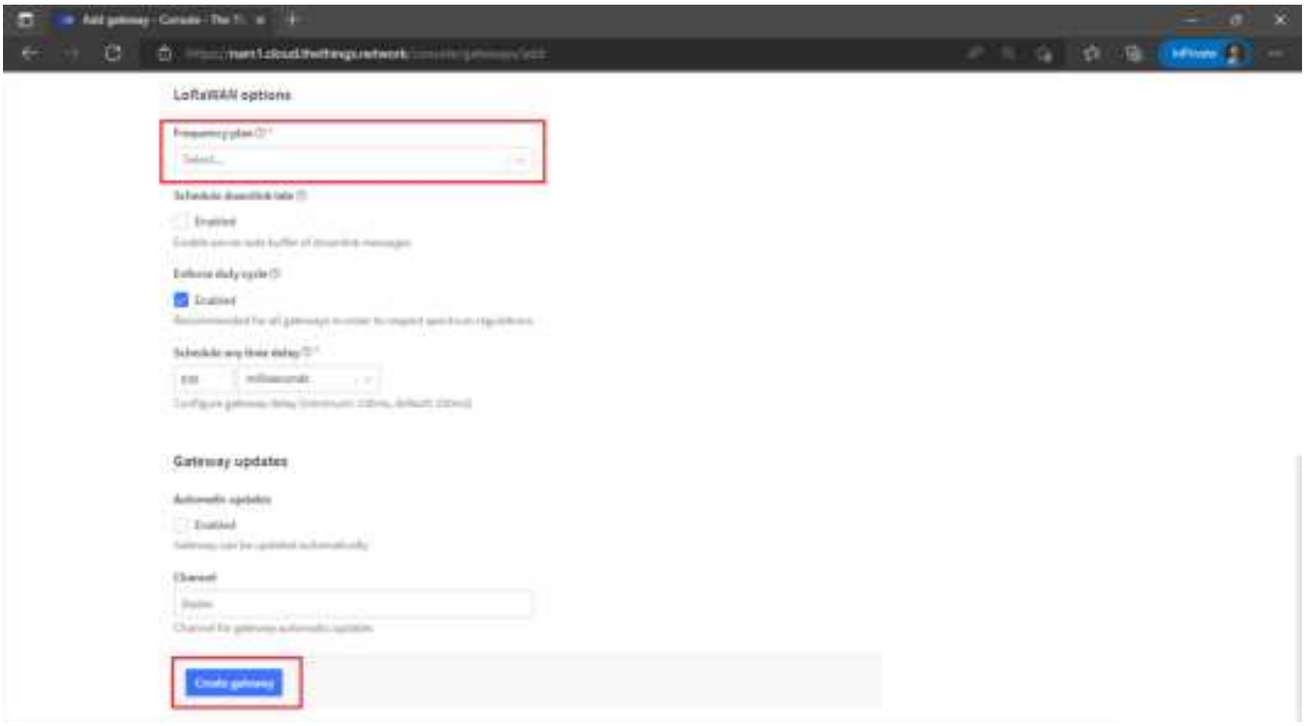

Figure 27: TTNv3 add new Gateway

Register the Device on TTNv3

The next step is to follow the procedure described in the section TTN OTAA Device Registration.

Uploading LoRaWAN Example to RAK3172

After a successful registration of the RAK3172 device on the LNS, you can now proceed with running the LoRaWAN OTAA demo application example.

1. Launch Arduino IDE and configure WisDuo RAK3172 Evaluation Board on board selection. See **Figure 9**.
2. Connect the RAK3172 via UART and check RAK3172 COM Port. See **Figure 10**.
3. Open the example code under **RAK WisBlock RUI examples**: **File** -> **Examples** -> **RAK WisBlock RUI examples** -> **Example** -> **LoRaWan_OTAA**.

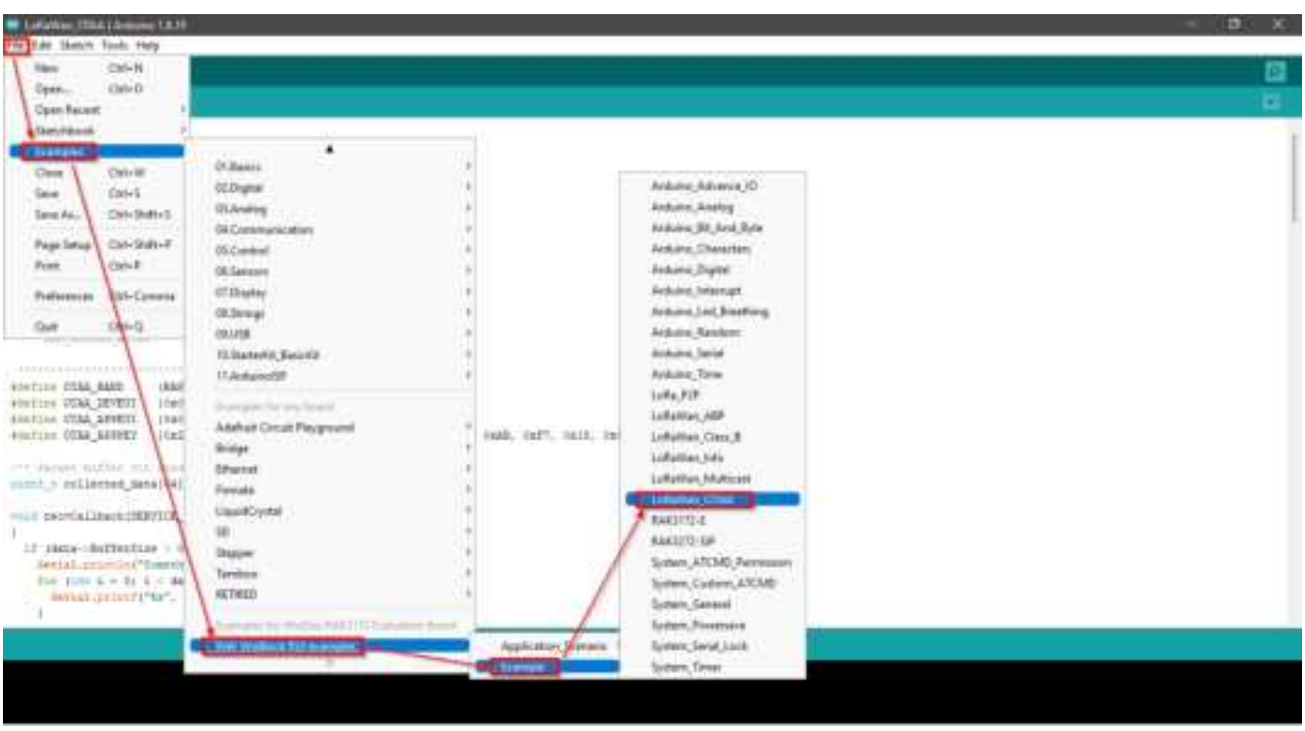

Figure 28: OTAA LoRaWAN application example

4. In the example code, you need to modify the device EUI **OTAA_DEVEUI**, the application EUI **OTAA_APPEUI**, and application key **OTAA_APPKEY**.

- The **OTAA_DEVEUI** parameter should match the device EUI registered in your network server. Note that your RAK3172 module has a sticker with QR code printed on it. You can use the QR code information to configure the **OTAA_DEVEUI** parameter. The **OTAA_DEVEUI** format is MSB first.

```c
  // OTAA Device EUI MSB
#define OTAA_DEVEUI    {0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x04, 0xF1, 0xC0}
```

- The **OTAA_APPEUI** parameter. This parameter should also be the same as the **APPEUI** in the network server you configured.

```c
  // OTAA Application EUI MSB
#define OTAA_APPEUI    {0x70, 0xB3, 0xD5, 0x7E, 0xD0,0x03, 0xAB, 0xA2}
```

- Another important parameter to change is the application key **OTAA_APPKEY**. This parameter should also be the same as the **APPKEY** in the network server you configured. The **OTAA_APPKEY** format is MSB first.

```c
// OTAA Application Key MSB
#define OTAA_APPKEY    {0xB4, 0x85, 0x7E, 0xFE, 0x1C, 0xB5, 0x15, 0xEB, 0x44, 0x61, 0x0D, 0x9B, 0x
```

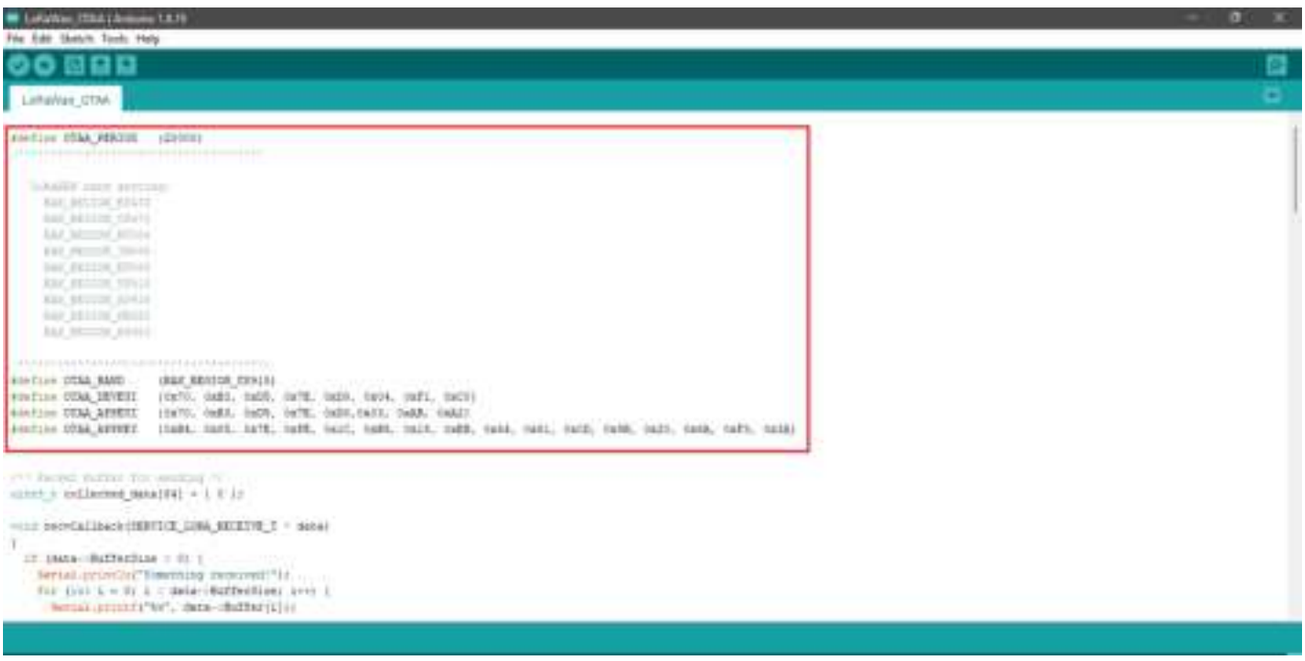

Figure 29: Configuring DEVEUI, APPEUI and APPKEY

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

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).

```c
if (!api.lorawan.band.set(OTAA_BAND)) {
  Serial.printf("LoRaWan OTAA - set band is incorrect! \r\n");
  return;
}
uint16_t maskBuff = 0x0002;
if (!api.lorawan.mask.set(&maskBuff)) {
  Serial.printf("LoRaWan OTAA - set mask is incorrect! \r\n");
  return;
}
```

> 📝 **NOTE:**
>
> RAK3172 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
> - RAK_REGION_AS923-2 = 9
> - RAK_REGION_AS923-3 = 10
> - RAK_REGION_AS923-4 = 11

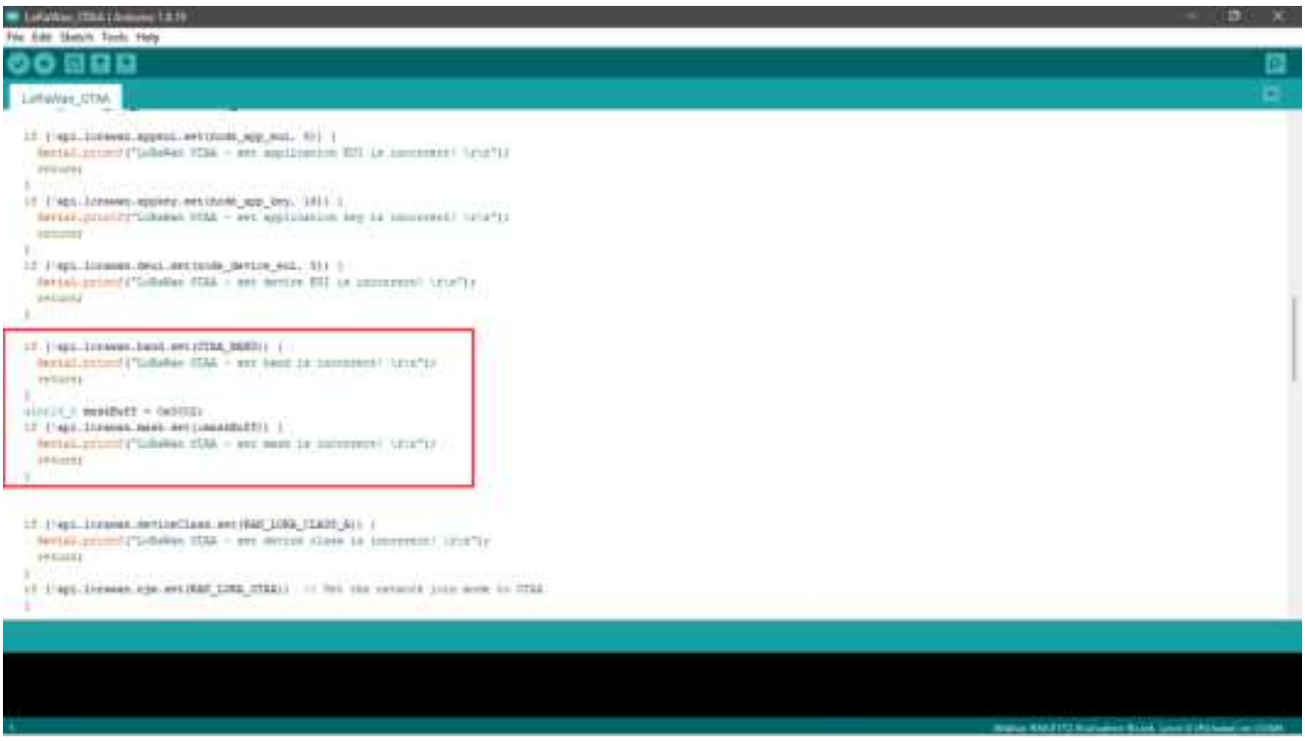

Figure 30: Updating to US915 and setting up channel mask

> 📝 **NOTE:**
>
> - Make sure you have configured the correct RAK board before uploading the code. See Configure RAK3172 on Boards Manager section.
> - Also, check RAK3172 COM Port on Device Manager section.

4. Open **Tools** Menu and select a COM port. **COM28** is currently used.

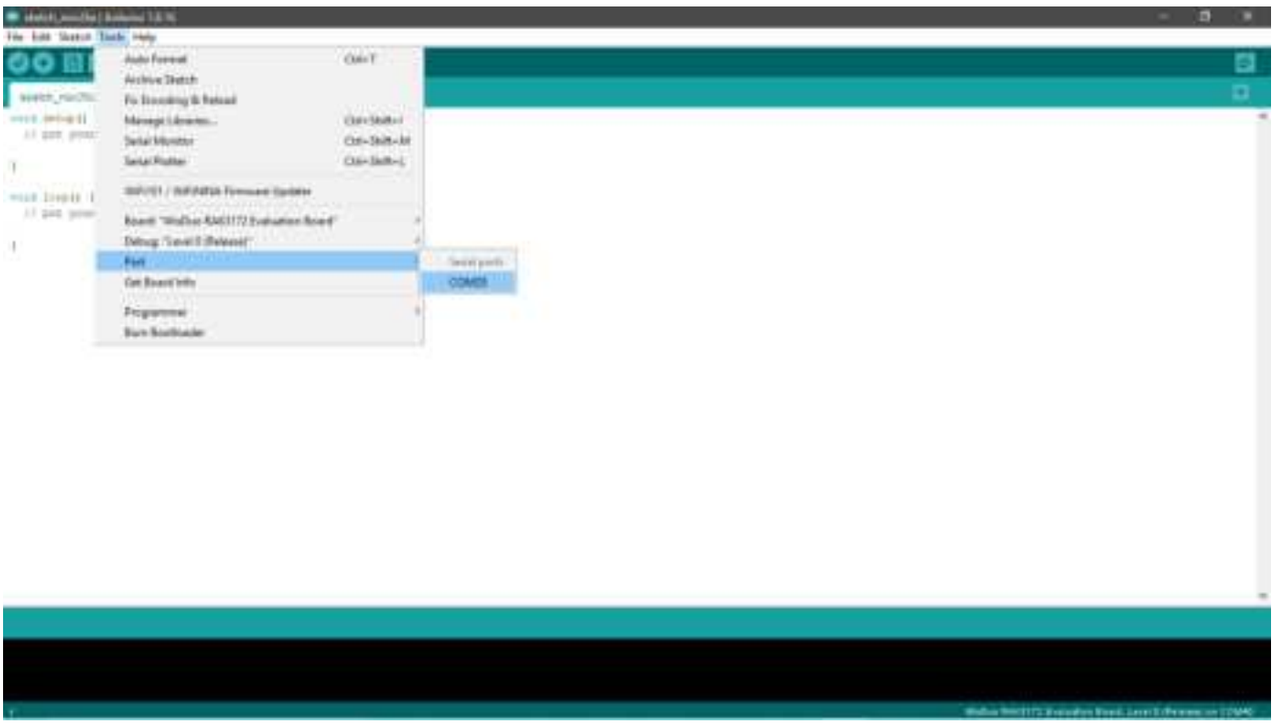

Figure 31: Select COM port

5. The last step is to upload the code by clicking the **Upload** icon on Arduino IDE. Take note that you should select the right board and COM port.

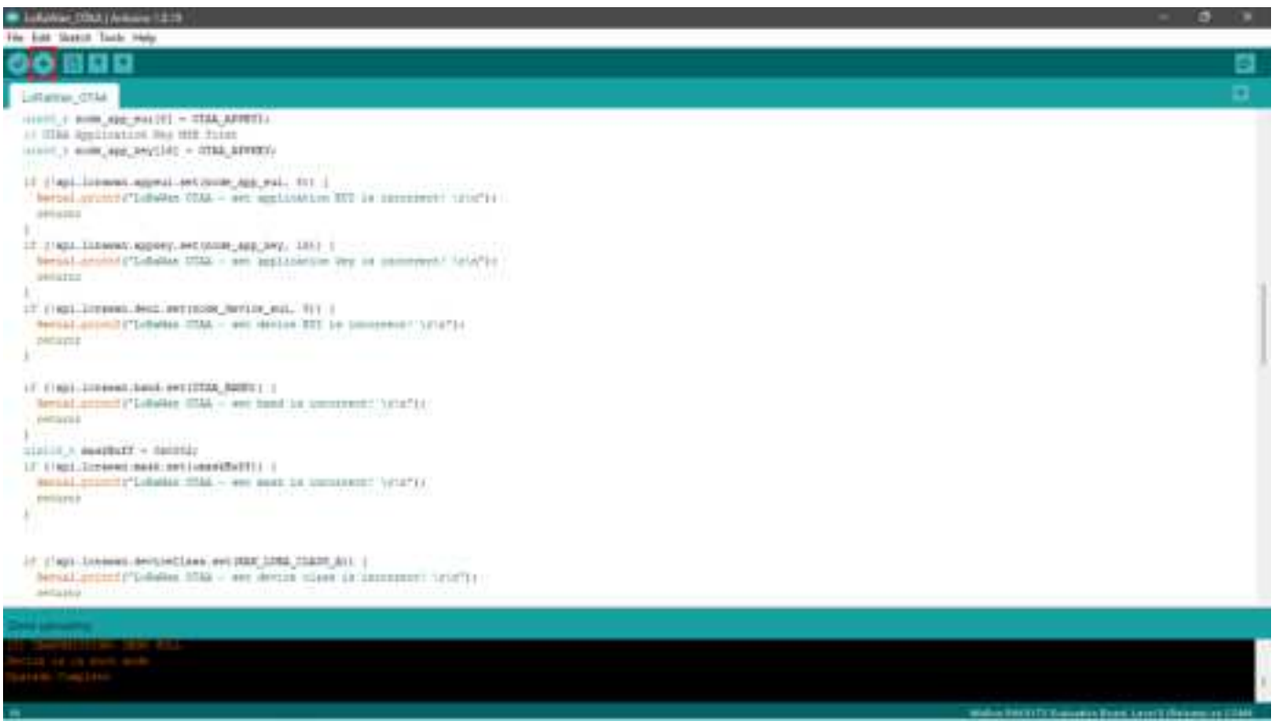

Figure 32: Uploading the code

6. You should now be able to see the console logs using 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.

Figure 33: Arduino serial monitor logs

7. Check on the LoRaWAN network TTN console logs if your device has been successfully joined with the `join-request` and `join-accept` messages.

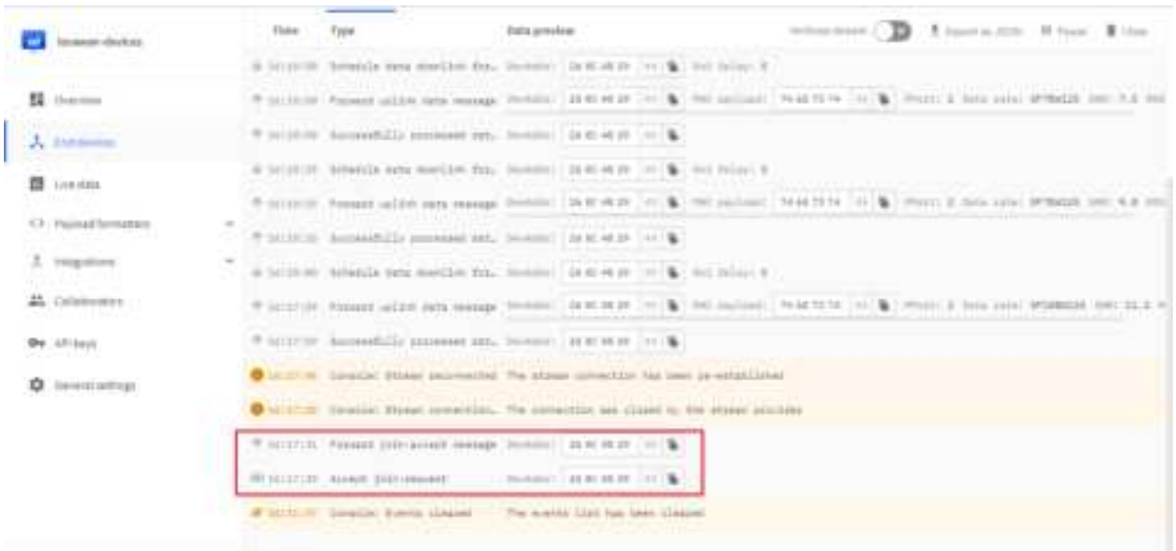

Figure 34: TTN console log

The modified `LoRaWAN_OTAA` project is available below.

```c
#define OTAA_PERIOD   (20000)
/************************************

   LoRaWAN band setting:
     RAK_REGION_EU433
     RAK_REGION_CN470
     RAK_REGION_RU864
     RAK_REGION_IN865
     RAK_REGION_EU868
     RAK_REGION_US915
     RAK_REGION_AU915
     RAK_REGION_KR920
     RAK_REGION_AS923

 ************************************/
#define OTAA_BAND     (RAK_REGION_US915)
#define OTAA_DEVEUI   {0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x04, 0xF1, 0xC0}
#define OTAA_APPEUI   {0x70, 0xB3, 0xD5, 0x7E, 0xD0,0x03, 0xAB, 0xA2}
#define OTAA_APPKEY   {0xB4, 0x85, 0x7E, 0xFE, 0x1C, 0xB5, 0x15, 0xEB, 0x44, 0x61, 0x0D, 0x9B, 0x


/** Packet buffer for sending */
uint8_t collected_data[64] = { 0 };

void recvCallback(SERVICE_LORA_RECEIVE_T * data)
{
  if (data->BufferSize > 0) {
    Serial.println("Something received!");
    for (int i = 0; i < data->BufferSize; i++) {
      Serial.printf("%x", data->Buffer[i]);
    }
    Serial.print("\r\n");
  }
}

void joinCallback(int32_t status)
{
  Serial.printf("Join status: %d\r\n", status);
}

void sendCallback(int32_t status)
{
  if (status == 0) {
    Serial.println("Successfully sent");
  } else {
    Serial.println("Sending failed");
  }
}

void setup()
{
  Serial.begin(115200, RAK_AT_MODE);

  Serial.println("RAKwireless LoRaWan OTAA Example");
  Serial.println("------------------------------------------------------");

  // OTAA Device EUI MSB first
  uint8_t node_device_eui[8] = OTAA_DEVEUI;
  // OTAA Application EUI MSB first
  uint8_t node_app_eui[8] = OTAA_APPEUI;
  // OTAA Application Key MSB first
```

```
uint8_t node_app_key[16] = OTAA_APPKEY;

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

if (!api.lorawan.band.set(OTAA_BAND)) {
  Serial.printf("LoRaWan OTAA - set band is incorrect! \r\n");
  return;
}
uint16_t maskBuff = 0x0002;
if (!api.lorawan.mask.set(&maskBuff)) {
  Serial.printf("LoRaWan OTAA - set mask is incorrect! \r\n");
  return;
}


if (!api.lorawan.deviceClass.set(RAK_LORA_CLASS_A)) {
  Serial.printf("LoRaWan OTAA - set device class is incorrect! \r\n");
  return;
}
if (!api.lorawan.njm.set(RAK_LORA_OTAA))  // Set the network join mode to OTAA
{
  Serial.printf
  ("LoRaWan OTAA - set network join mode is incorrect! \r\n");
  return;
}
if (!api.lorawan.join())  // Join to Gateway
{
  Serial.printf("LoRaWan OTAA - join fail! \r\n");
  return;
}

/** Wait for Join success */
while (api.lorawan.njs.get() == 0) {
  Serial.print("Wait for LoRaWAN join...");
  api.lorawan.join();
  delay(10000);
}

if (!api.lorawan.adr.set(true)) {
  Serial.printf
  ("LoRaWan OTAA - set adaptive data rate is incorrect! \r\n");
  return;
}
if (!api.lorawan.rety.set(1)) {
  Serial.printf("LoRaWan OTAA - set retry times is incorrect! \r\n");
  return;
}
if (!api.lorawan.cfm.set(1)) {
  Serial.printf("LoRaWan OTAA - set confirm mode is incorrect! \r\n");
  return;
}
```

```
  /** Check LoRaWan Status*/
  Serial.printf("Duty cycle is %s\r\n", api.lorawan.dcs.get()? "ON" : "OFF");   // Check Duty Cyc
  Serial.printf("Packet is %s\r\n", api.lorawan.cfm.get()? "CONFIRMED" : "UNCONFIRMED");    // Ch
  uint8_t assigned_dev_addr[4] = { 0 };
  api.lorawan.daddr.get(assigned_dev_addr, 4);
  Serial.printf("Device Address is %02X%02X%02X%02X\r\n", assigned_dev_addr[0], assigned_dev_addr
  Serial.printf("Uplink period is %ums\r\n", OTAA_PERIOD);
  Serial.println("");
  api.lorawan.registerRecvCallback(recvCallback);
  api.lorawan.registerJoinCallback(joinCallback);
  api.lorawan.registerSendCallback(sendCallback);
}

void uplink_routine()
{
  /** Payload of Uplink */
  uint8_t data_len = 0;
  collected_data[data_len++] = (uint8_t) 't';
  collected_data[data_len++] = (uint8_t) 'e';
  collected_data[data_len++] = (uint8_t) 's';
  collected_data[data_len++] = (uint8_t) 't';

  Serial.println("Data Packet:");
  for (int i = 0; i < data_len; i++) {
    Serial.printf("0x%02X ", collected_data[i]);
  }
  Serial.println("");

  /** Send the data package */
  if (api.lorawan.send(data_len, (uint8_t *) & collected_data, 2, true, 1)) {
    Serial.println("Sending is requested");
  } else {
    Serial.println("Sending failed");
  }
}

void loop()
{
  static uint64_t last = 0;
  static uint64_t elapsed;

  if ((elapsed = millis() - last) > OTAA_PERIOD) {
    uplink_routine();

    last = millis();
  }
  //Serial.printf("Try sleep %ums..", OTAA_PERIOD);
  api.system.sleep.all(OTAA_PERIOD);
  //Serial.println("Wakeup..");
}
```

RAK3172 as a LoRa/LoRaWAN Modem via AT Command

AT Command via UART2

RAK3172 module can be configured using AT commands via the UART2 interface. You need a USB to UART TTL adapter to connect the RAK3172 to your computer's USB port and a serial terminal tool. You can use the RAK

Serial Port Tool so you can easily send AT commands and view the replies from the console output. The RAK Serial Port Tool commands still uses the RUI V2 AT commands by default. You can modify it to have RUI3 AT commands and then save it.

> ⚠ **WARNING**
>
> Firmware update and AT command functionality are done via UART2 pins. If you will connect the module to an external host MCU that will send AT commands via UART2, take extra precautions in your board design to ensure you can still perform FW update to it. There should be a way in your board design that can disconnect the host MCU UART to connect to RAK3172 UART2 before connecting the module to the PC (via USB-UART converter) for the FW update process.
>
> An alternative option to update firmware aside from UART2 is to use SWD pins (SWCLK & SWDIO). This method will require you to use external tools like ST-LINK and RAKDAP1.

Connect to the RAK3172

1. Connect the RAK3172 to the serial port of a general-purpose computer (USB port) using a USB to UART TTL adapter like RAKDAP1 , as shown in **Figure 30**.

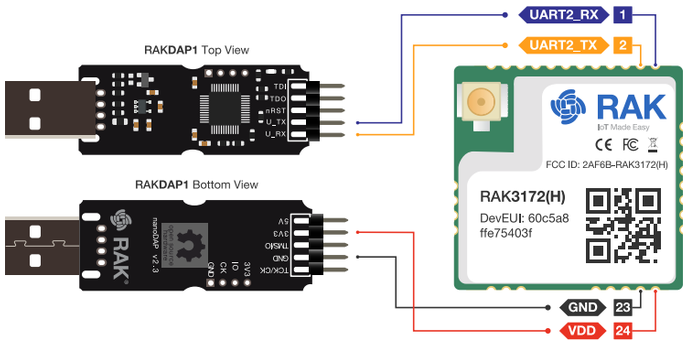

Figure 35: RAK3172 Module Connection

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**

RAK3172 Configuration for LoRaWAN or LoRa P2P

To enable the RAK3172 module as a LoRa P2P module or a LoRaWAN end-device, the module must be configured and parameters must be set by sending AT commands. You can configure the RAK3172 in two ways:

- LoRaWAN End-Device - RAK3172 as LoRaWAN IoT device.
- LoRa P2P - Point-to-point communication between two RAK3172 modules.

Configuring RAK3172 as LoRaWAN End-Device

To enable the RAK3172 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 RAK3172.

This guide covers the following topics:

- TheThingsNetwork Guide - How to login, register new accounts and create new applications on TTN.
- RAK3172 TTN OTAA Guide - How to add OTAA device on TTN and what AT commands to use on RAK3172 OTAA activation.
- RAK3172 TTN ABP Guide - How to add ABP device on TTN and what AT commands to use on RAK3172 ABP activation.
- Chirpstack Guide - How to create new applications on Chirpstack.
- RAK3172 Chirpstack OTAA Guide - How to add OTAA device to Chirpstack and what AT commands to use on RAK3172 OTAA activation.
- RAK3172 Chirpstack ABP Guide - How to add ABP device on Chirpstack and what AT commands to use on RAK3172 ABP activation.

Connecting to The Things Network (TTN)

In this section, a quick tutorial guide will show how to connect the RAK3172 module to the TTN platform.

> 📝 **NOTE:**
>
> In this guide, 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 36: RAK3172 EVB in the context of the TTN

As shown in **Figure 31**, 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 RAK3172 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 RAK3172 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. The first step is to go to The Things Network platform and select a cluster, as shown in **Figure 32**.

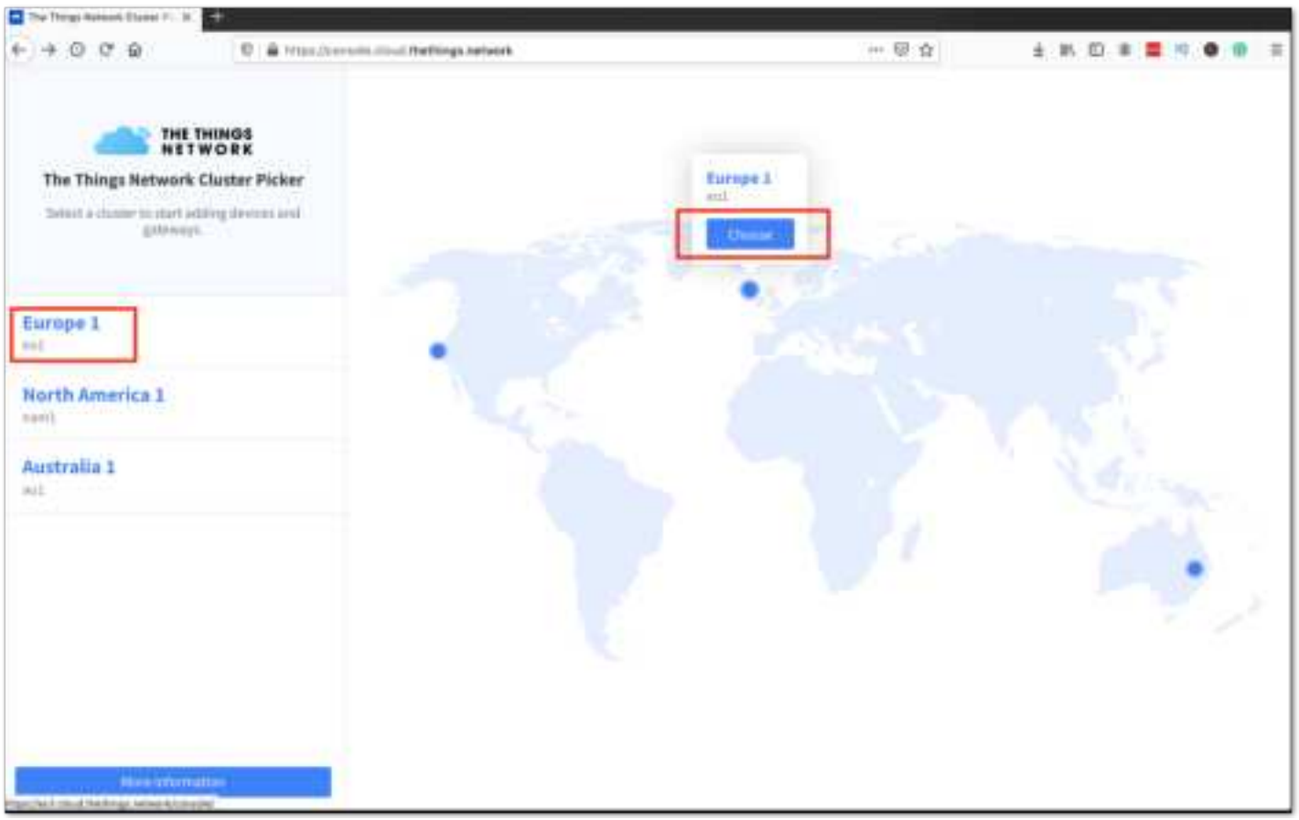

Figure 37: Selecting Cluster in TTN V3

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 33** and **Figure 34**.

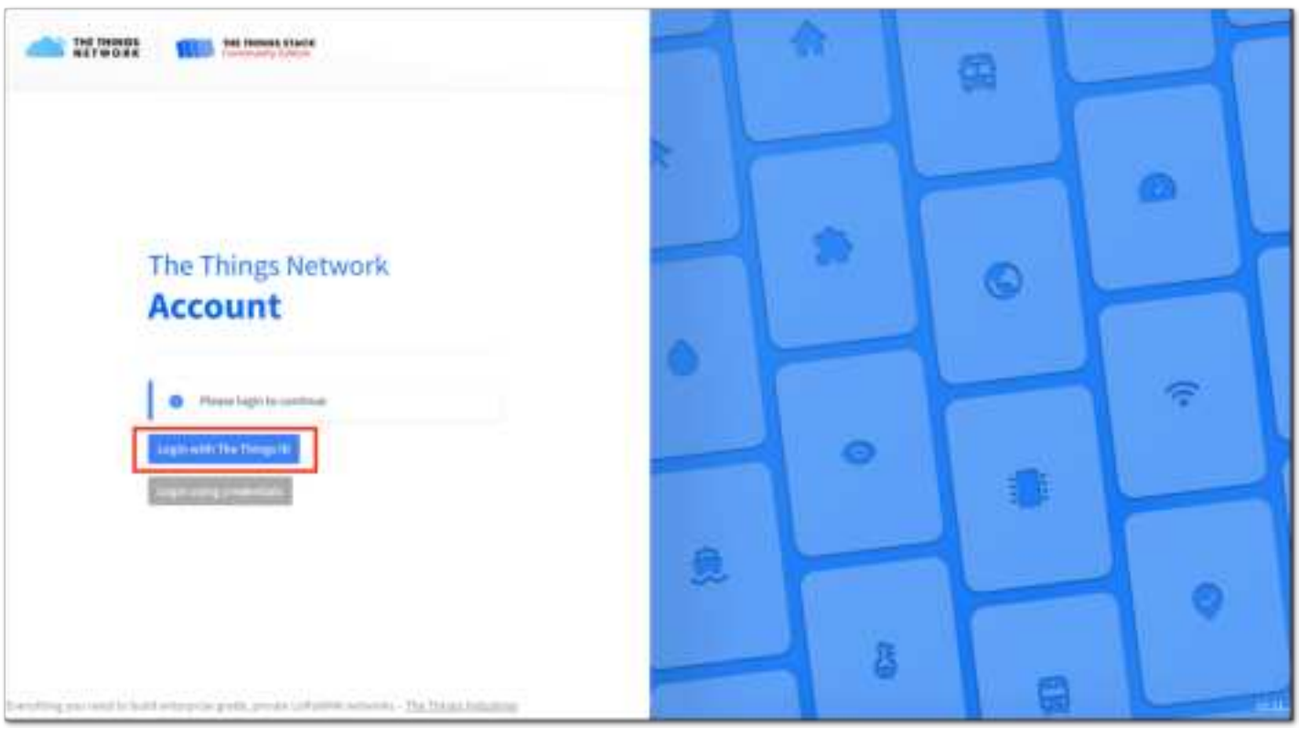

Figure 38: Login using TTN account

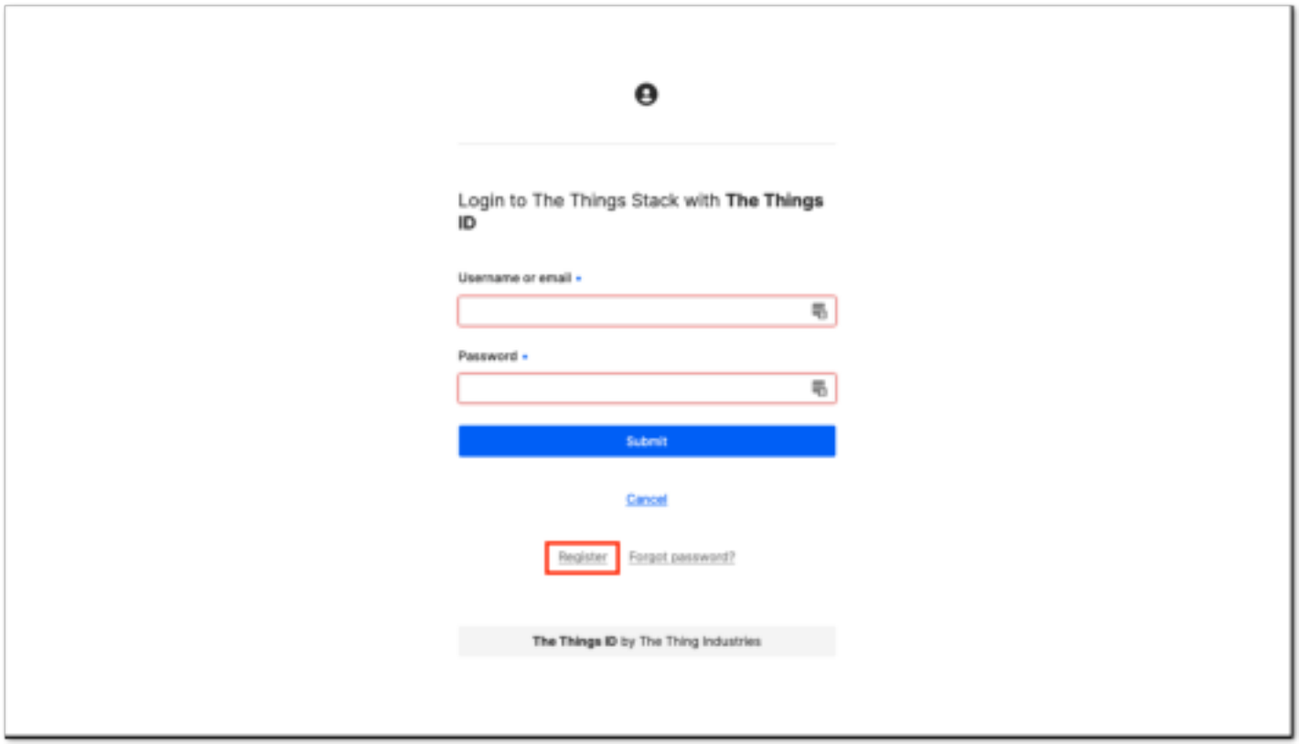

Figure 39: Registration of new account

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, you should log in on the platform using your username/email and password then click **Submit**, as shown in **Figure 35**.

Figure 40: Logging in to TTN platform

5. Click **Authorize** to proceed.

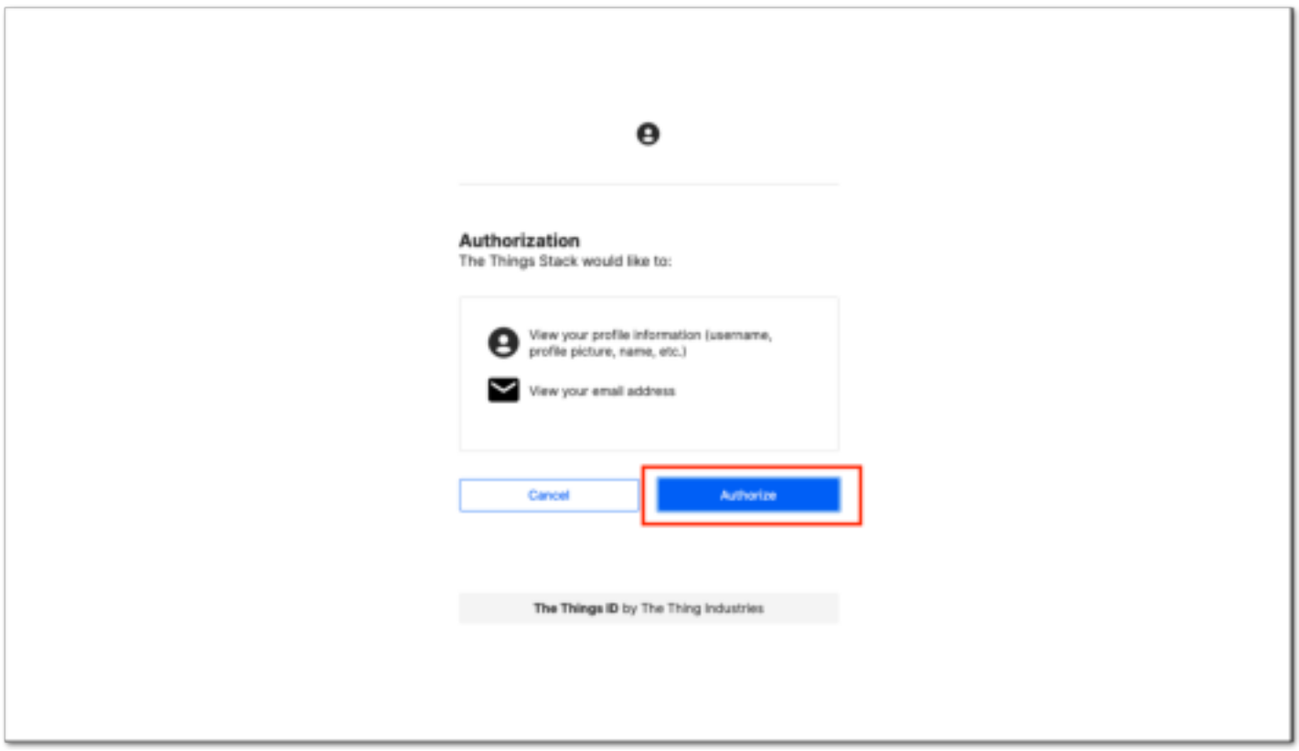

Figure 41: Authorization to TTN

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

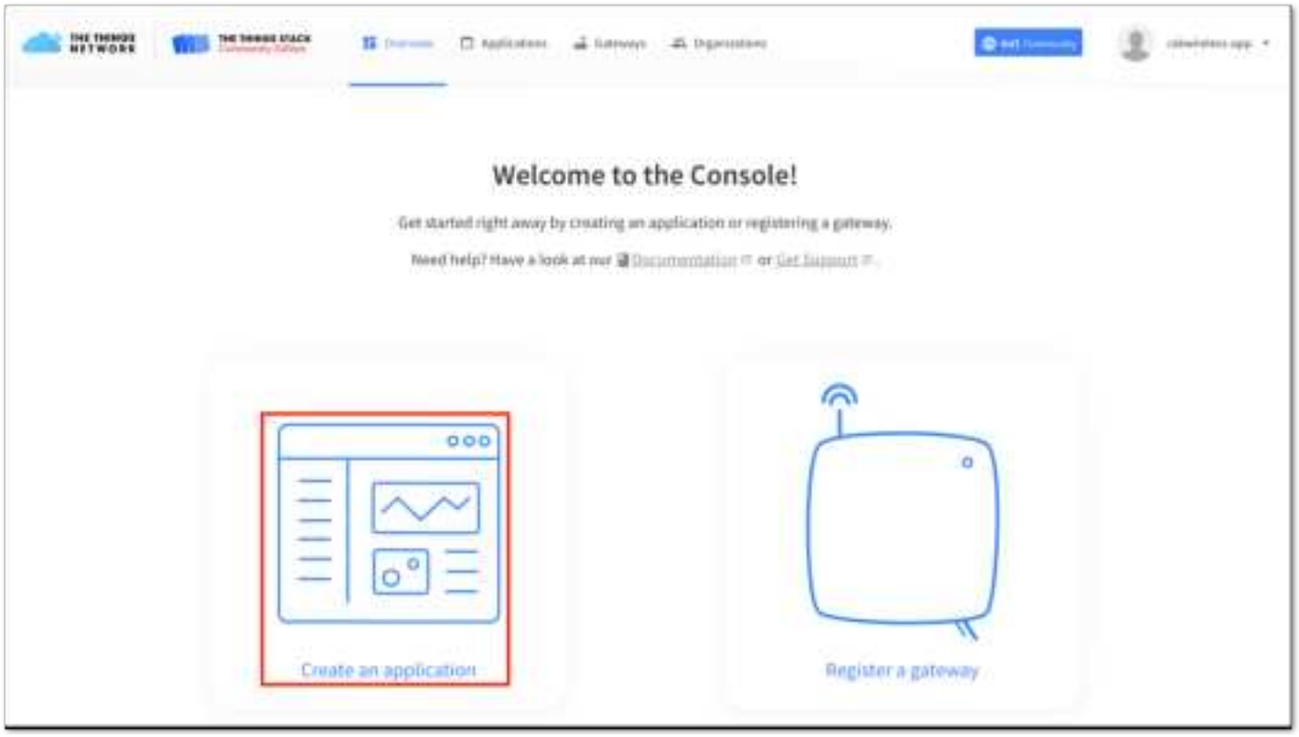

Figure 42: Creating TTN application for your LoRaWAN devices

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

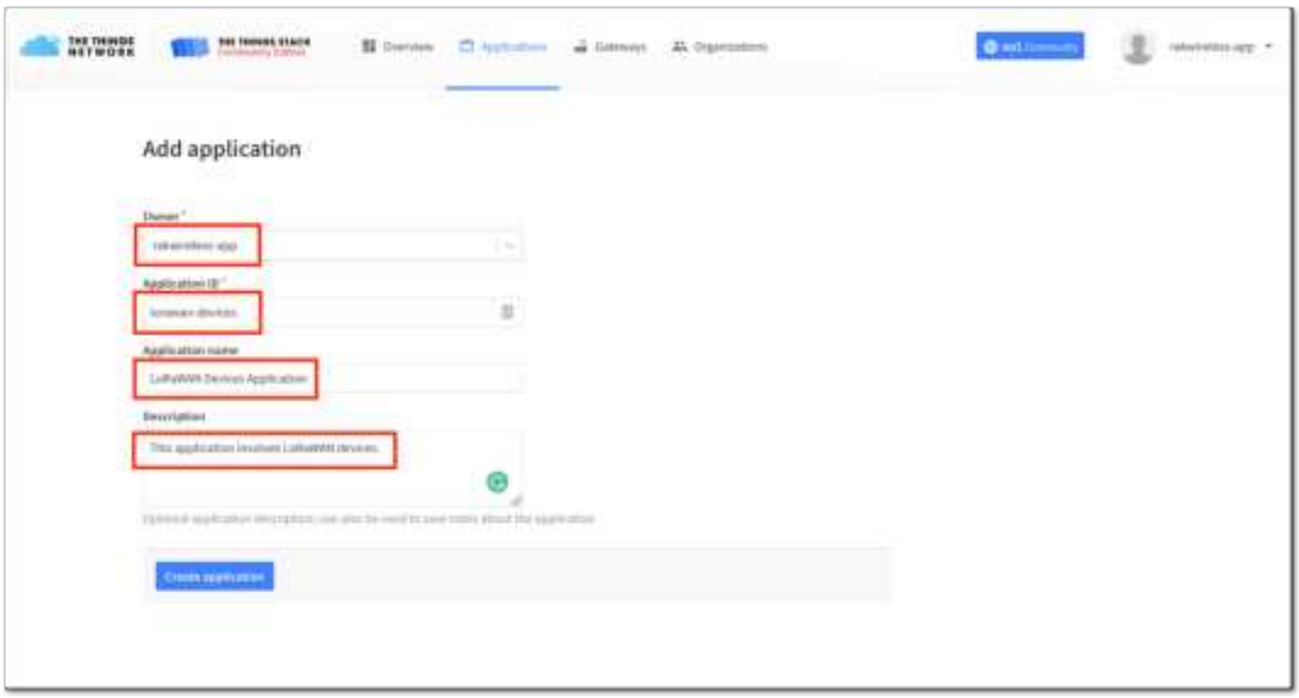

Figure 43: Details of the TTN application

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 39**.

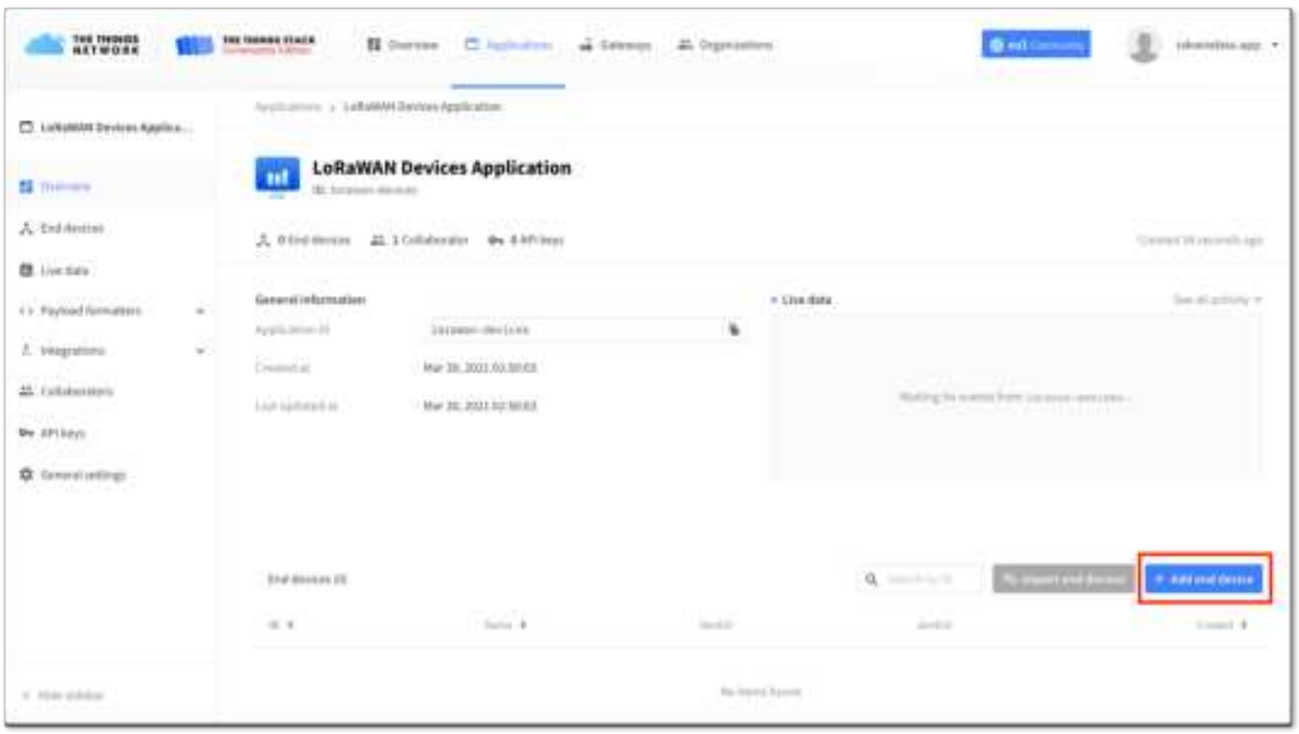

Figure 44: Add end device

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 40** and **Figure 41**.

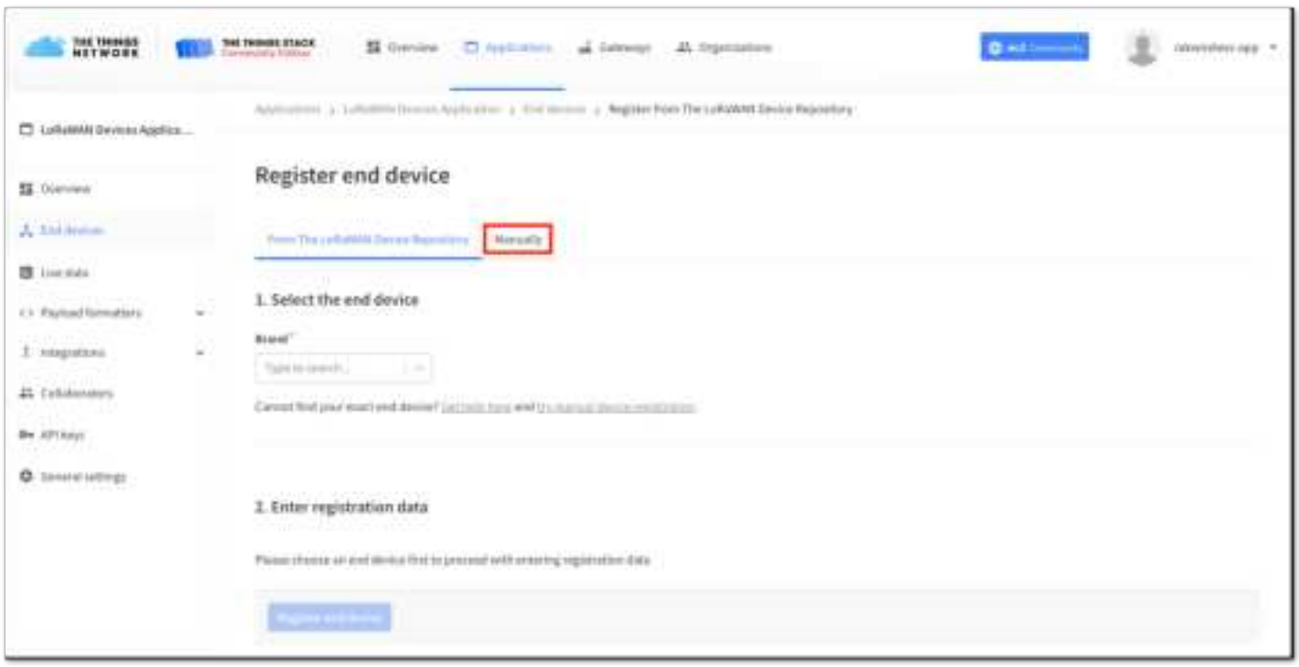

Figure 45: Manually register device to TTN

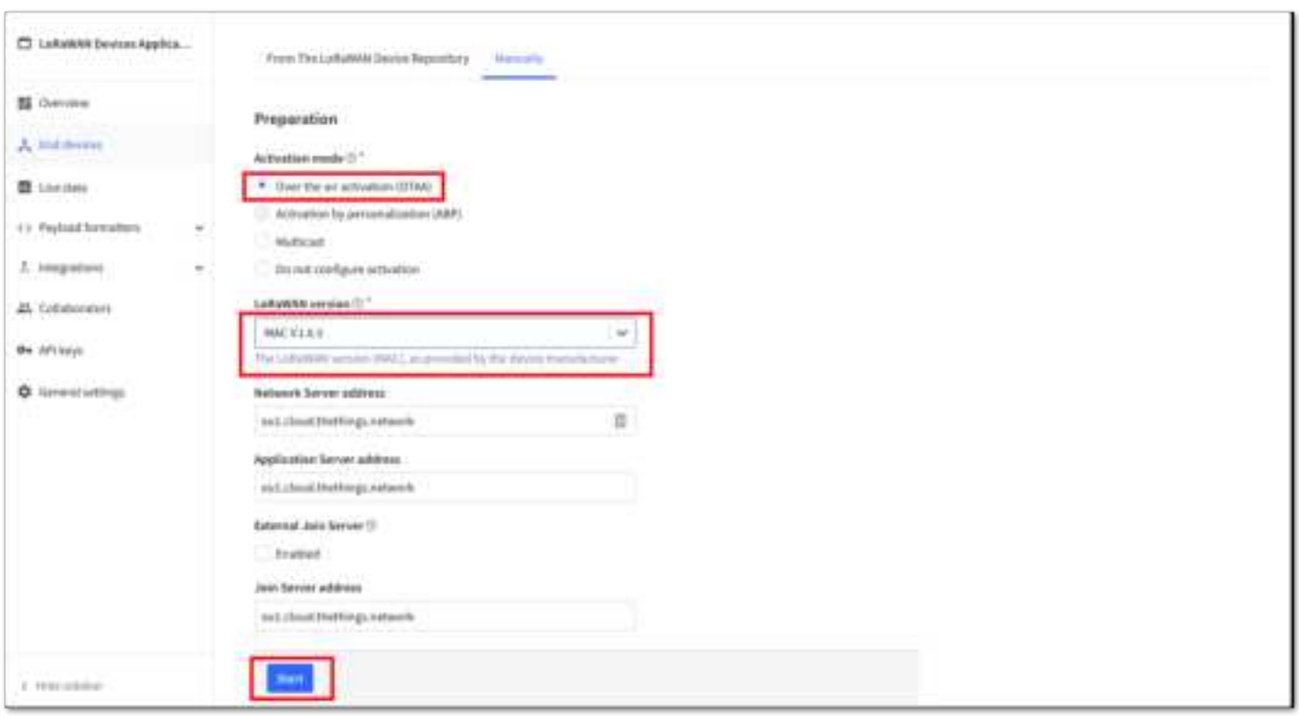

Figure 46: Device activation configuration

3. Then you need to put a unique **End device ID** and EUIs (**DevEUI** and **AppEUI**), as shown in **Figure 42**. 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, you need to 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's purpose. The end-device ID `rak-device` is for illustration purposes only.

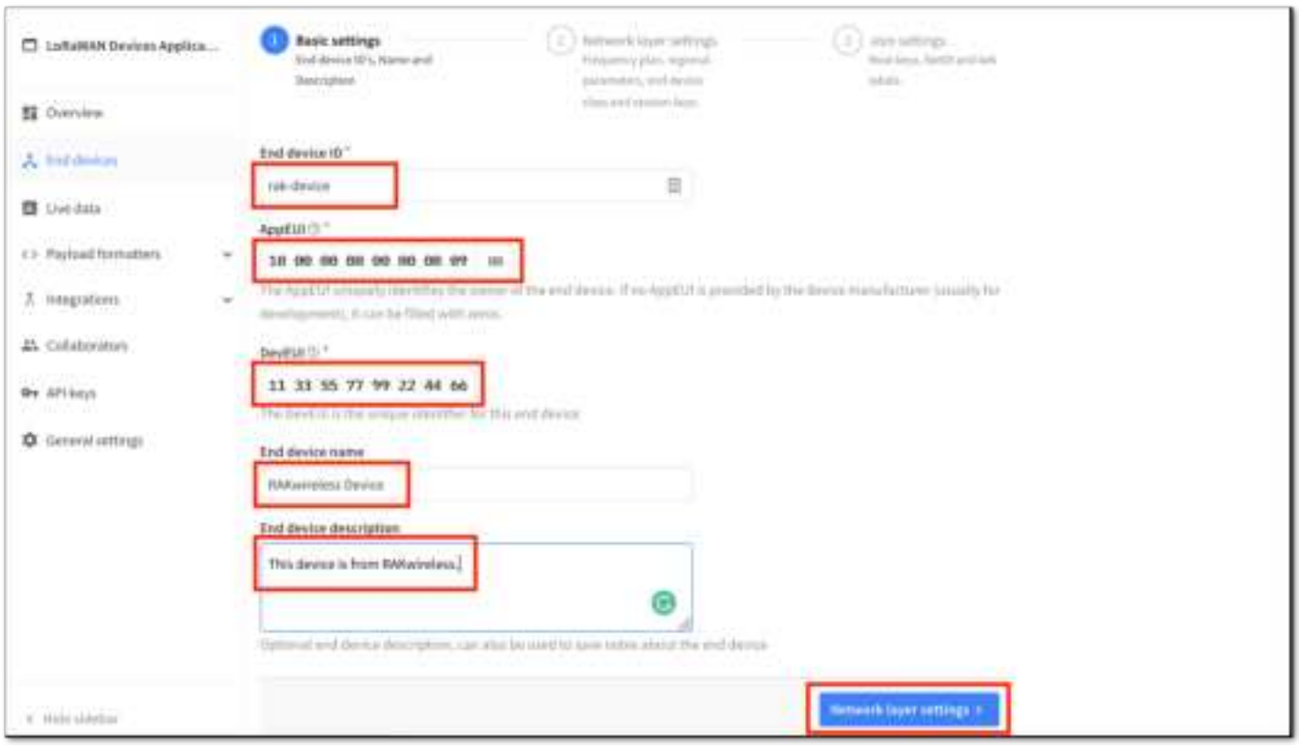

Figure 47: OTAA Device Information

5. The next step is to set up the **Frequency plan**, a compatible **Regional Parameter version**, and the **LoRaWAN class** supported. Then you can click **Join settings**.

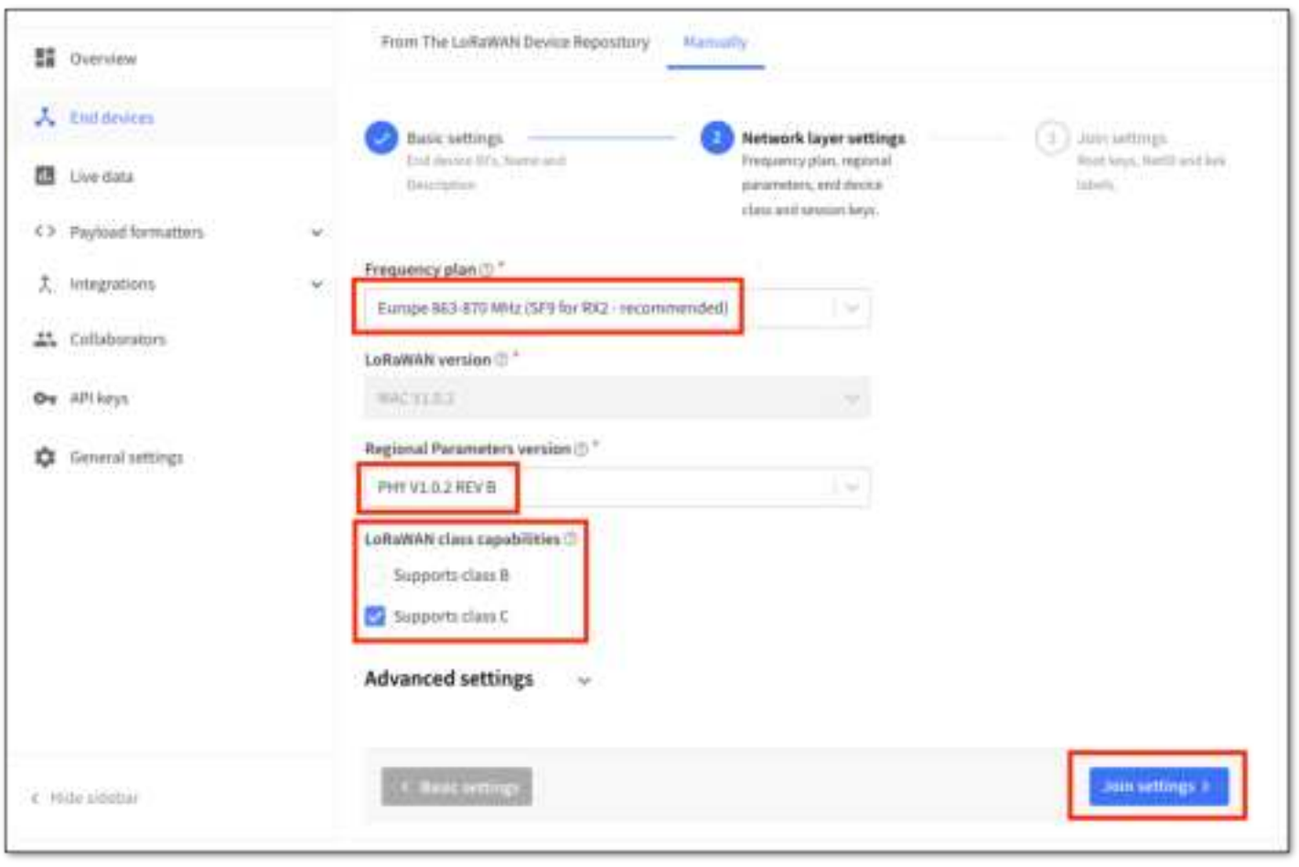

Figure 48: OTAA Configuration

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

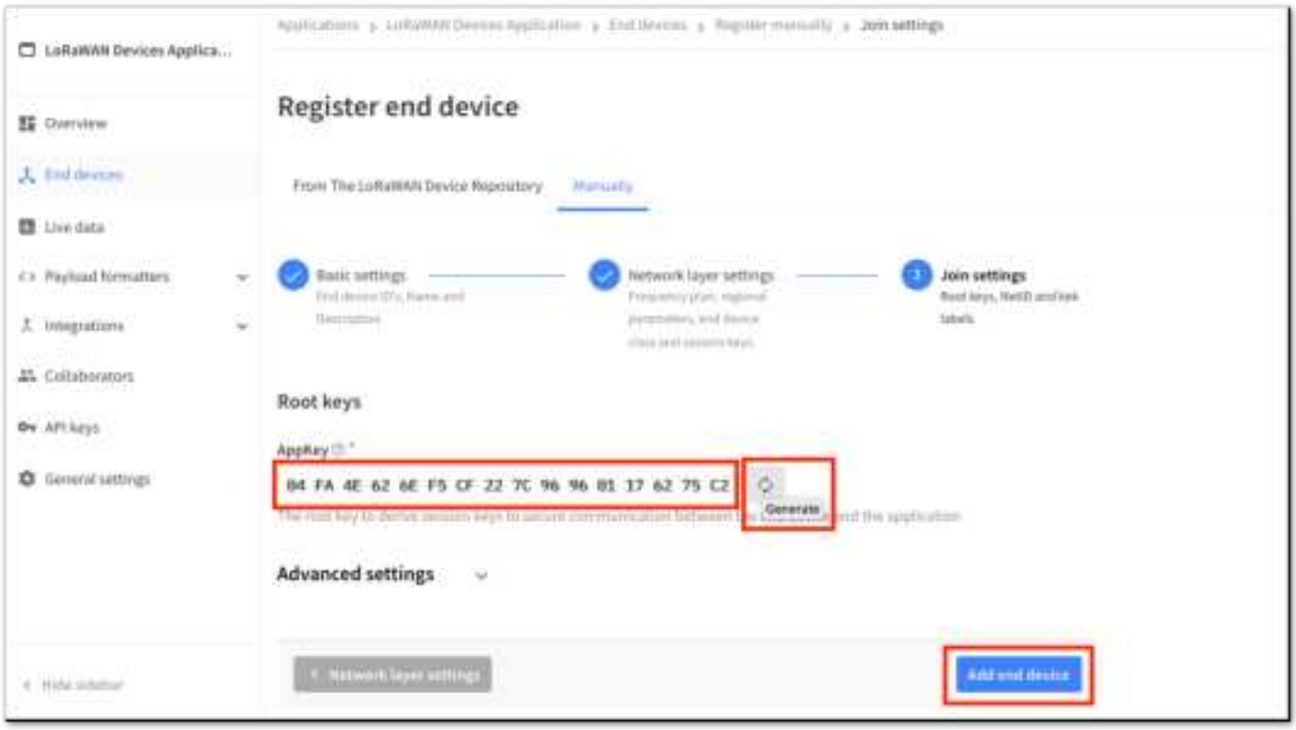

Figure 49: OTAA AppKey generation and device registration

You should now be able to see the device on the TTN console after you fully register your device, as shown in **Figure 44**.

> 📝 **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 45**.

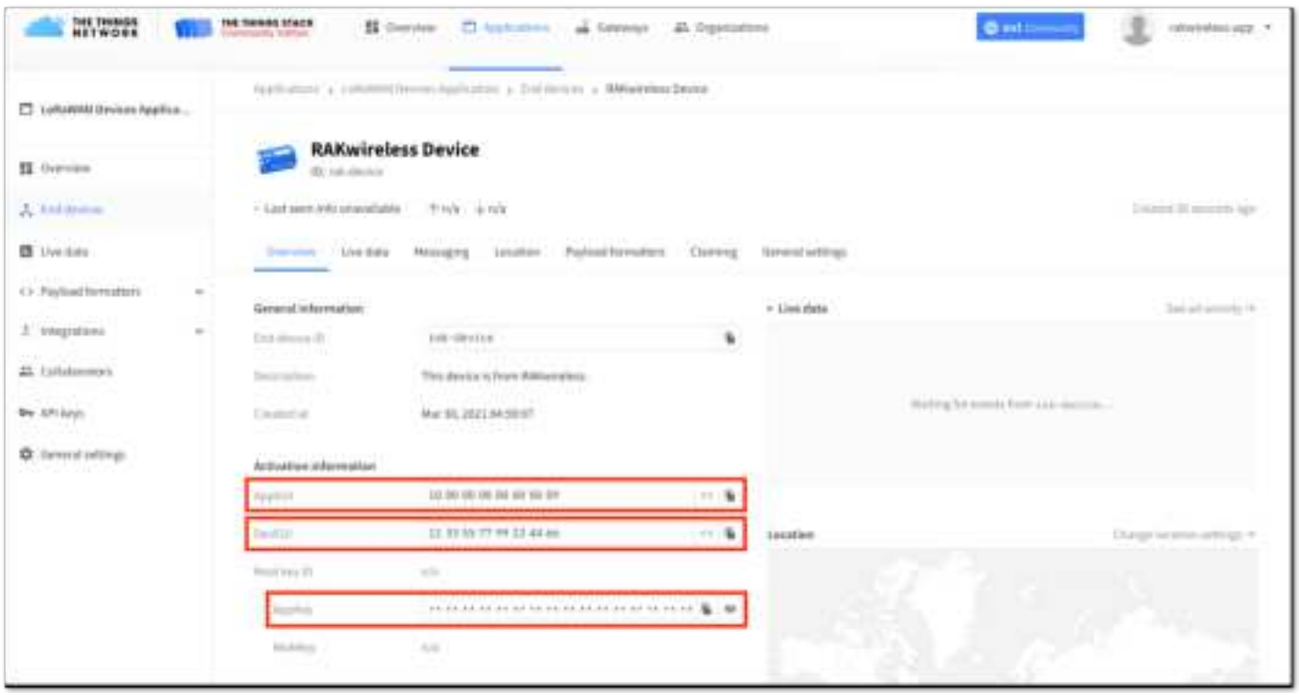

Figure 50: OTAA device successfully registered to TTN

OTAA Configuration for TTN

The RAK3172 module supports a series of AT commands to configure its internal parameters and control the functionalities of the module.

1. To set up the RAK3172 module to join the TTN using OTAA, start by connecting the RAK3172 module to your computer (see **Figure 30**) 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 46**.

> 📝 **NOTE:**
>
> If there is no `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.

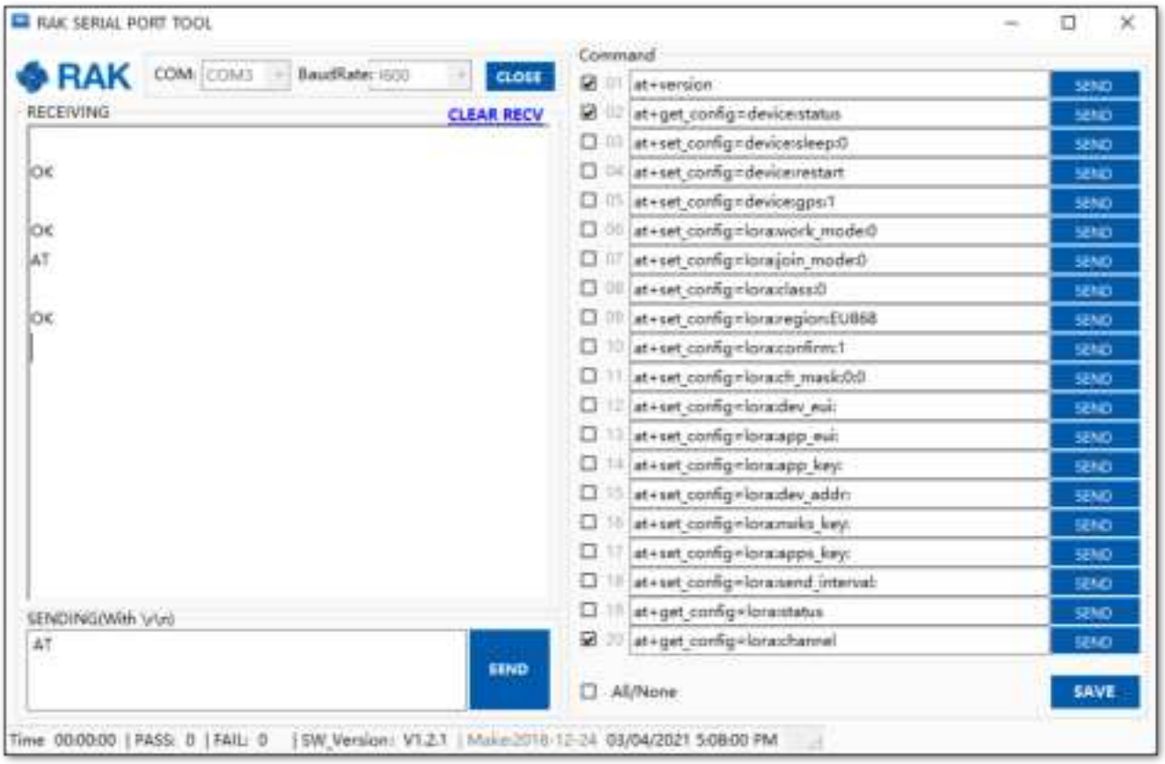

Figure 51: at+version command response

2. The next step is to configure the OTAA LoRaWAN parameters in RAK3172:

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

Set the work mode to LoRaWAN.

```
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 RAK3172 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	AS923-1
9	AS923-2
10	AS923-3
11	AS923-4

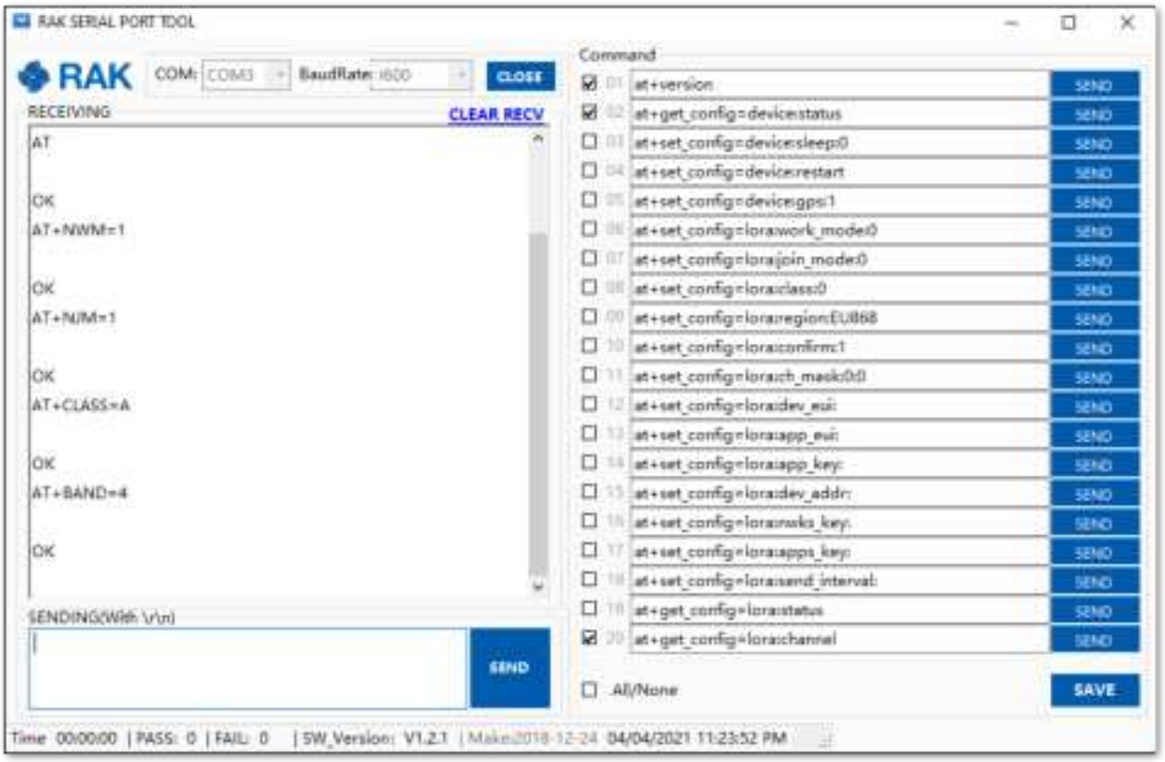

Figure 52: Configuring LoRa Parameters

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
```

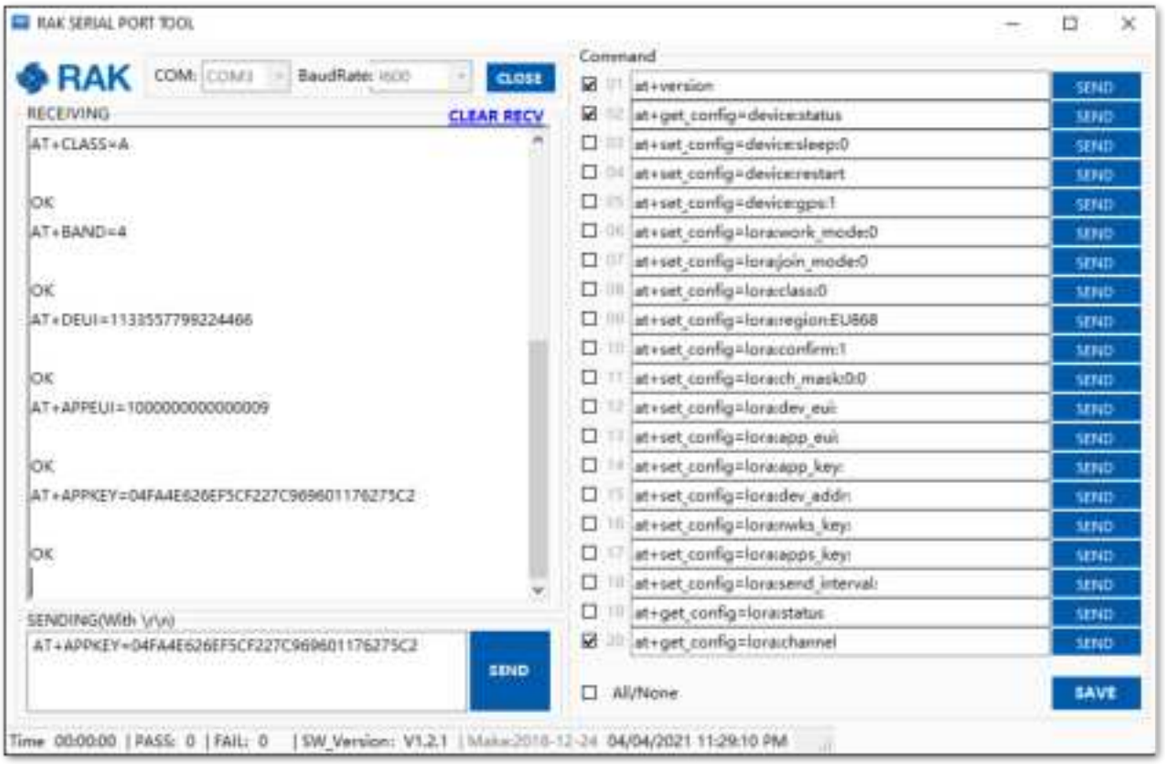

Figure 53: Configuring LoRa Parameters

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

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

> 📝 **NOTE:**
>
> `AT+JOIN` command parameters are optional. You can configure the settings for auto-join, reattempt interval, and the number of join attempts if your application needs it. If not configured, it will use the default parameter values.
>
> `AT+JOIN` and `AT+JOIN=1` also share the common functionality of trying to join the network.

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 49**.

> 📝 **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.

1. 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>`

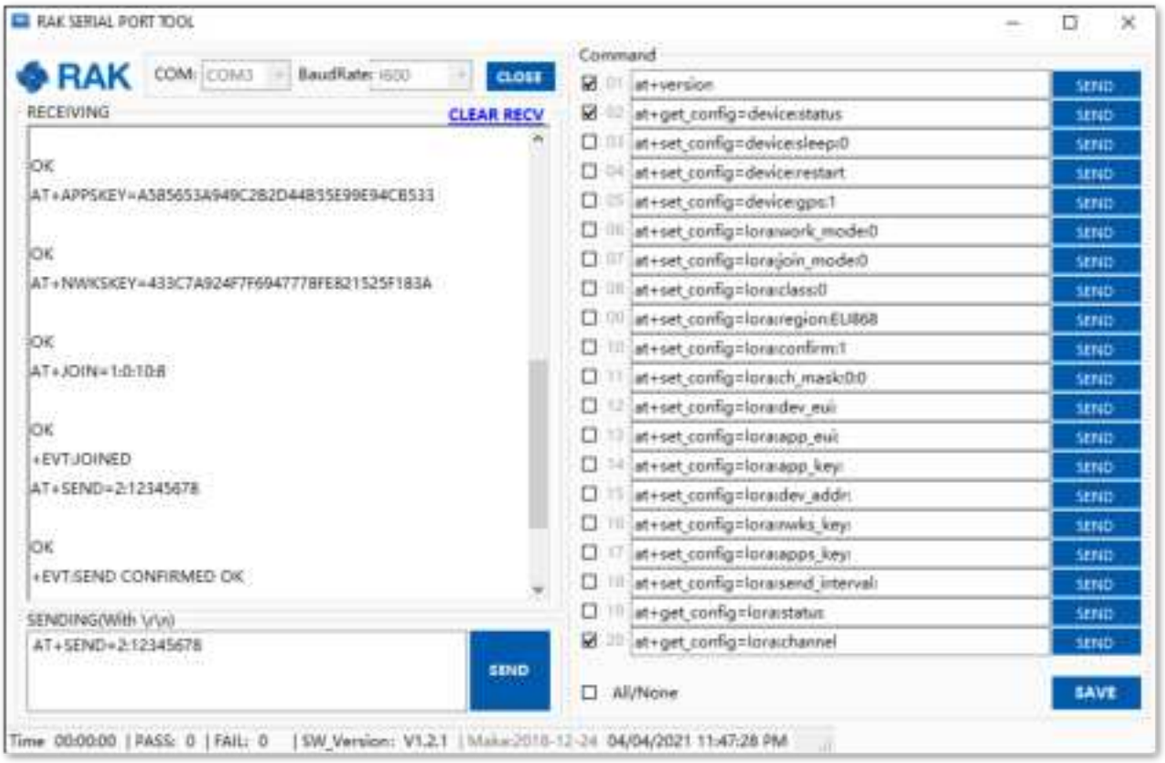

Figure 54: OTAA Test Sample Data Sent via RAK Serial Port Tool

7. You can see the data sent by the RAK3172 module on the TTN device console *Live data* section. Also, the *Last seen* info should be a few seconds or minutes ago.

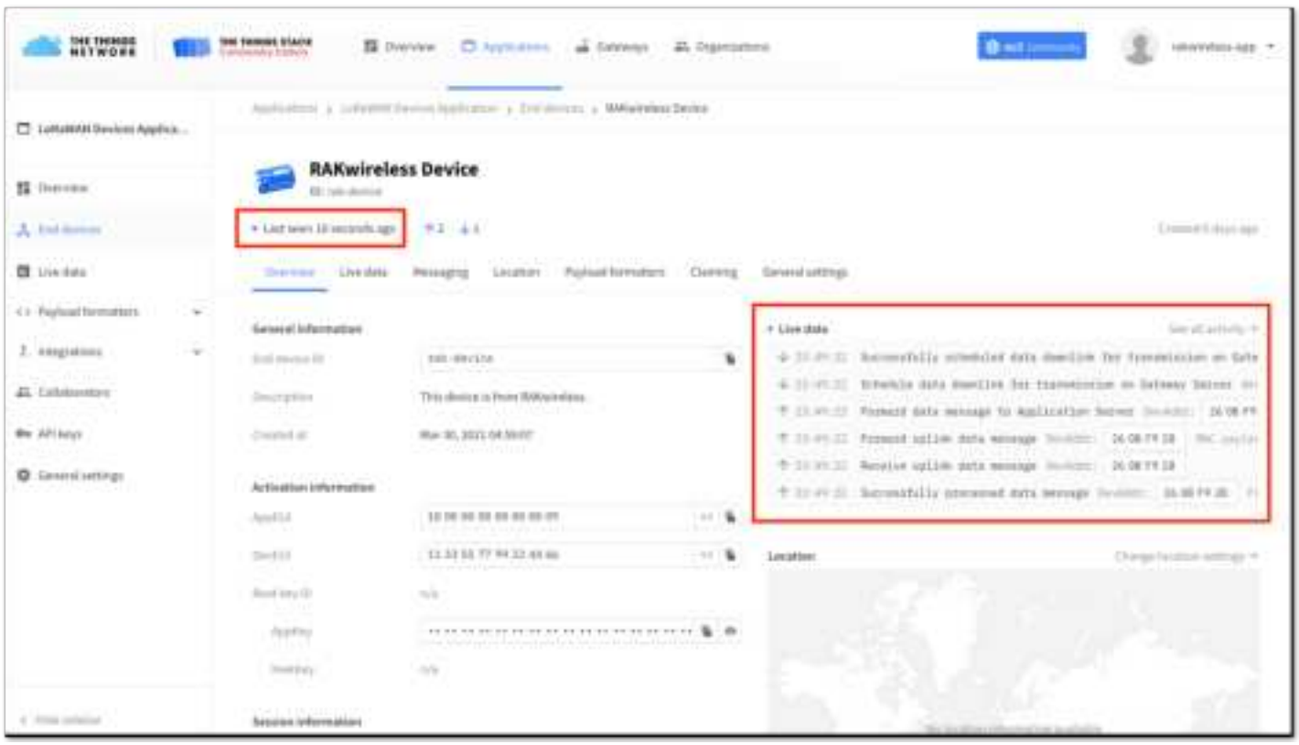

Figure 55: OTAA Test Sample Data Sent Viewed in TTN

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 51**.

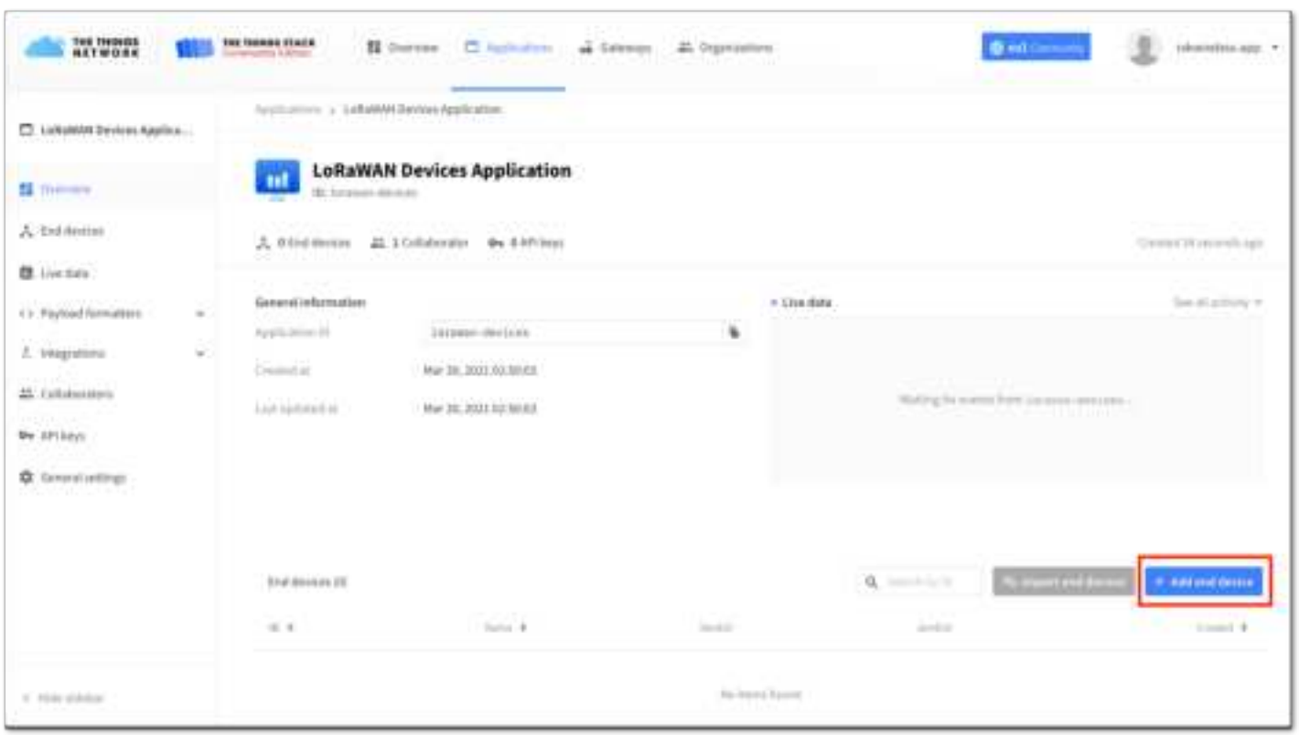

Figure 56: Adding ABP Device

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 52** and **Figure 53**.

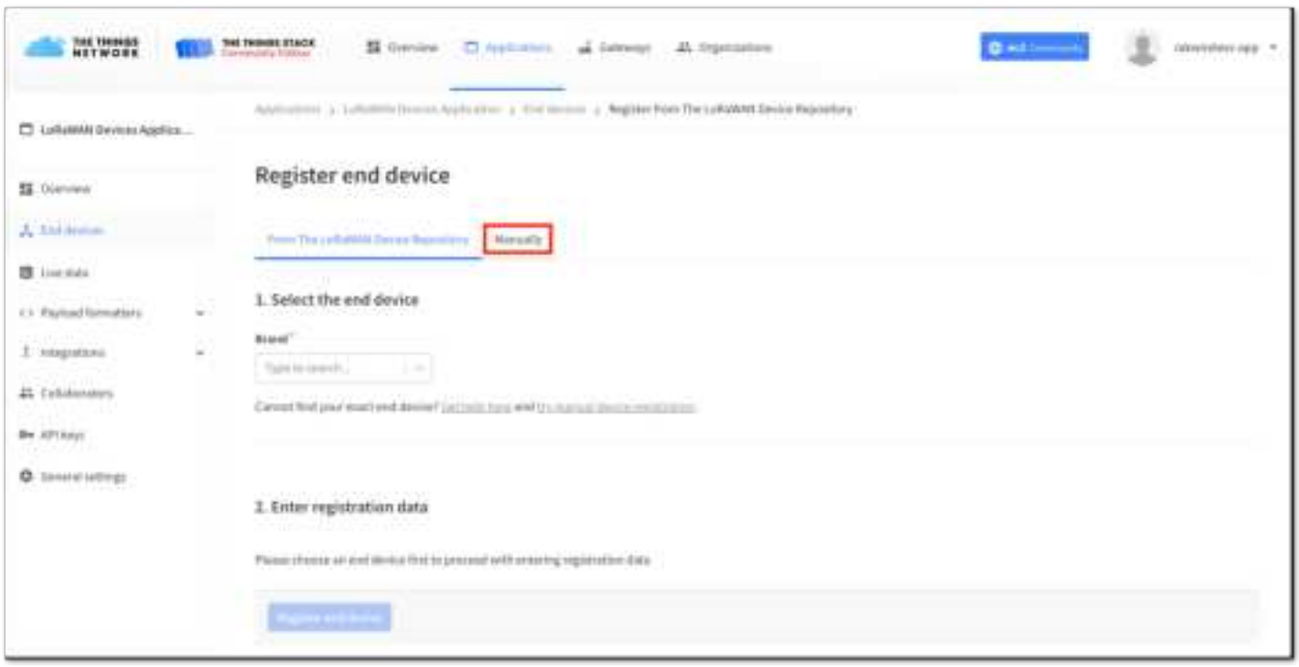

Figure 57: Manually register device to TTN

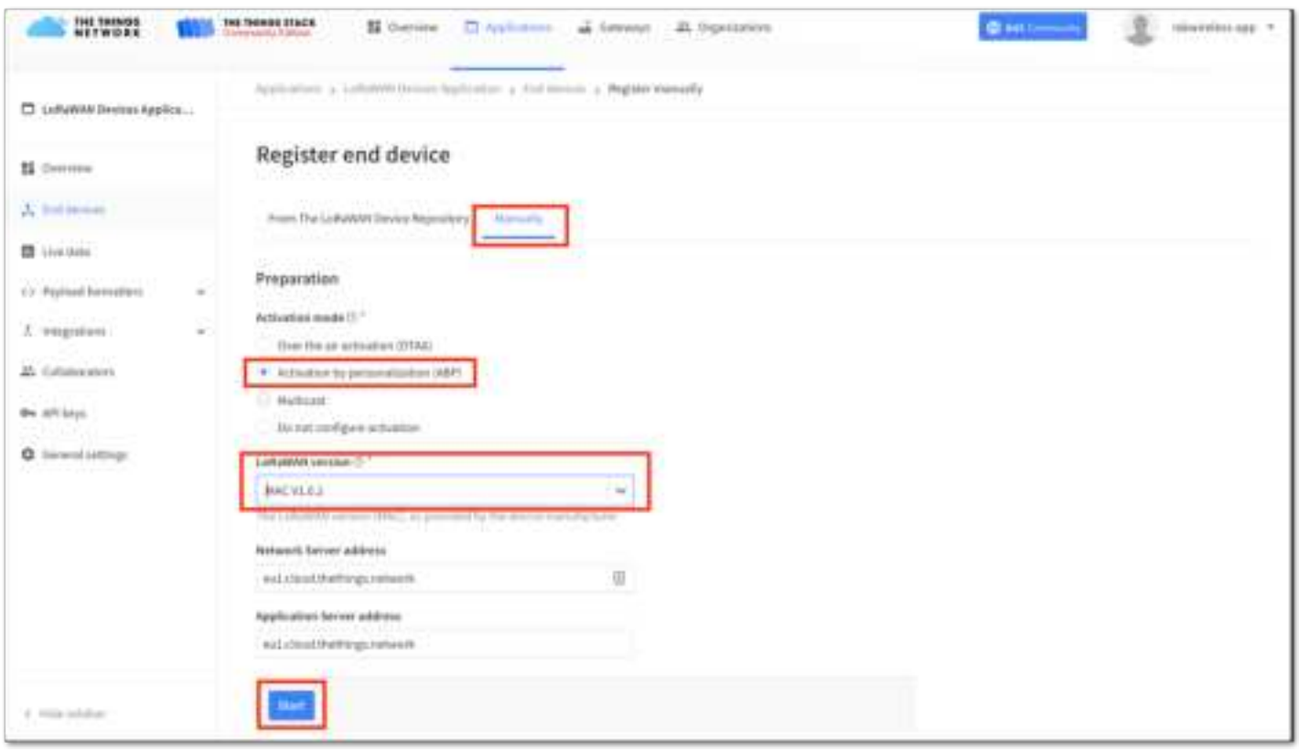

Figure 58: Selecting ABP and LoRaWAN version

3. At this step, you need to put a unique **End device ID** and **DevEUI**, as shown in **Figure 54**. 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.

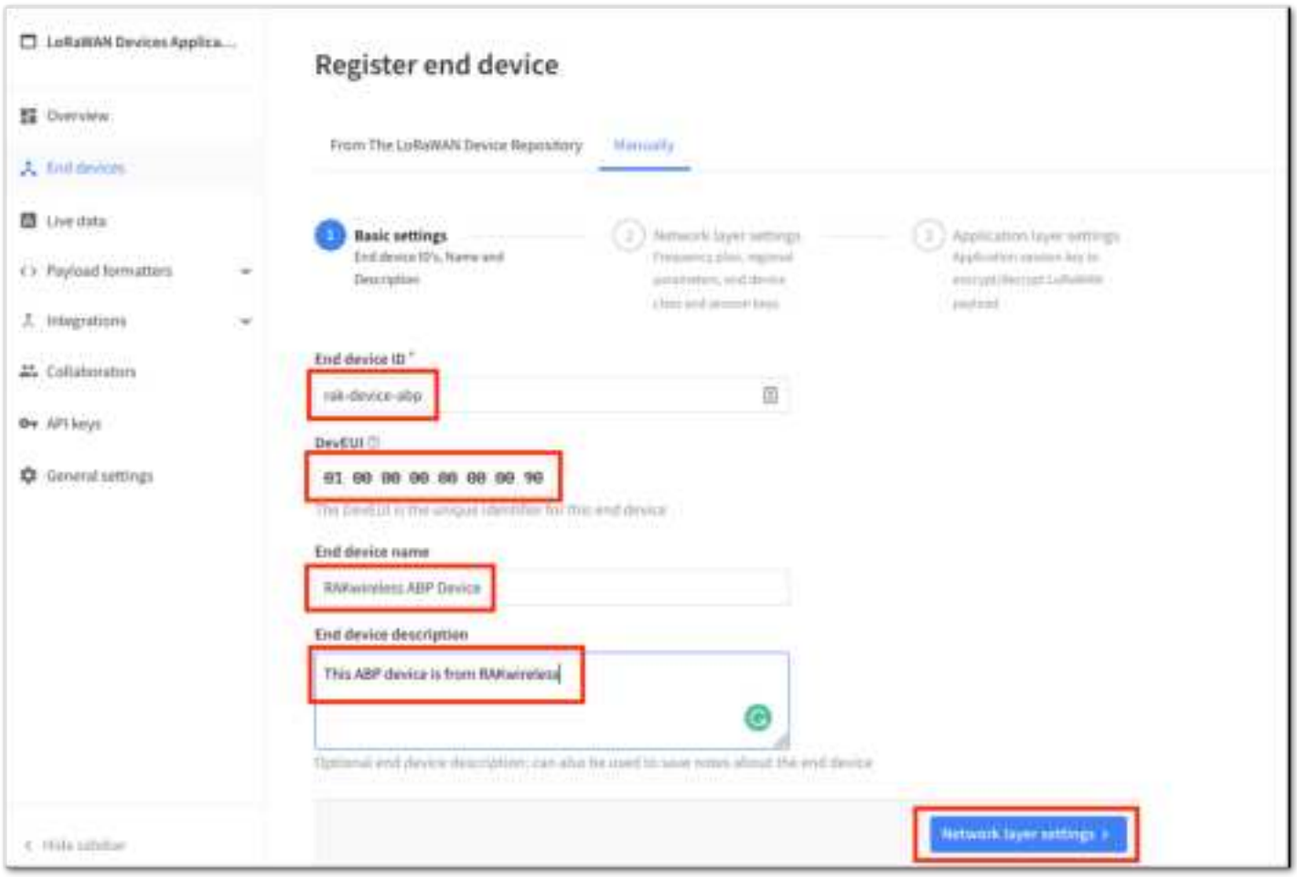

Figure 59: ABP Device Information

5. The next step is to 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**.

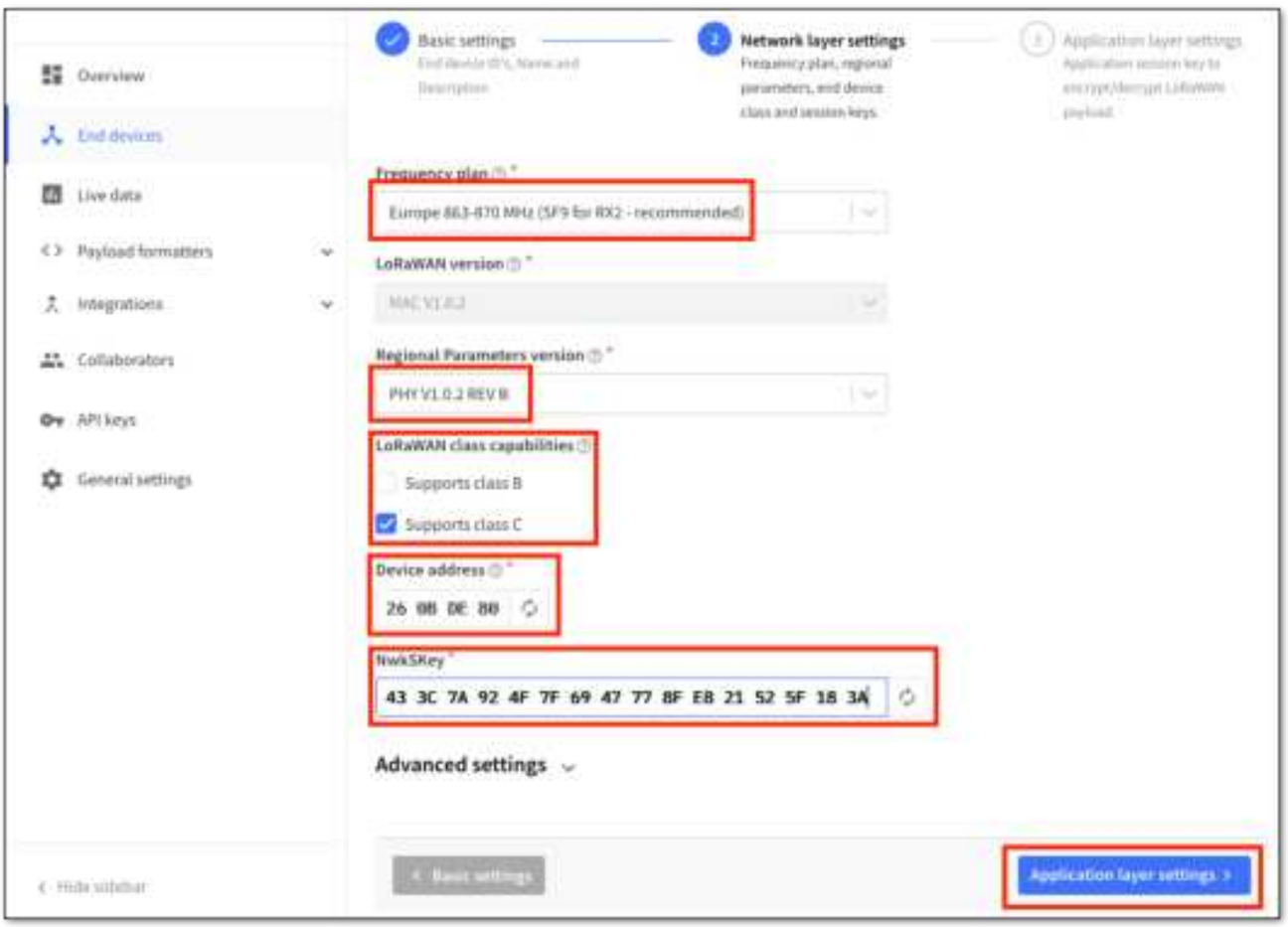

Figure 60: ABP Device Configuration

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 you need to click **Add end device** to finish your new device registration.

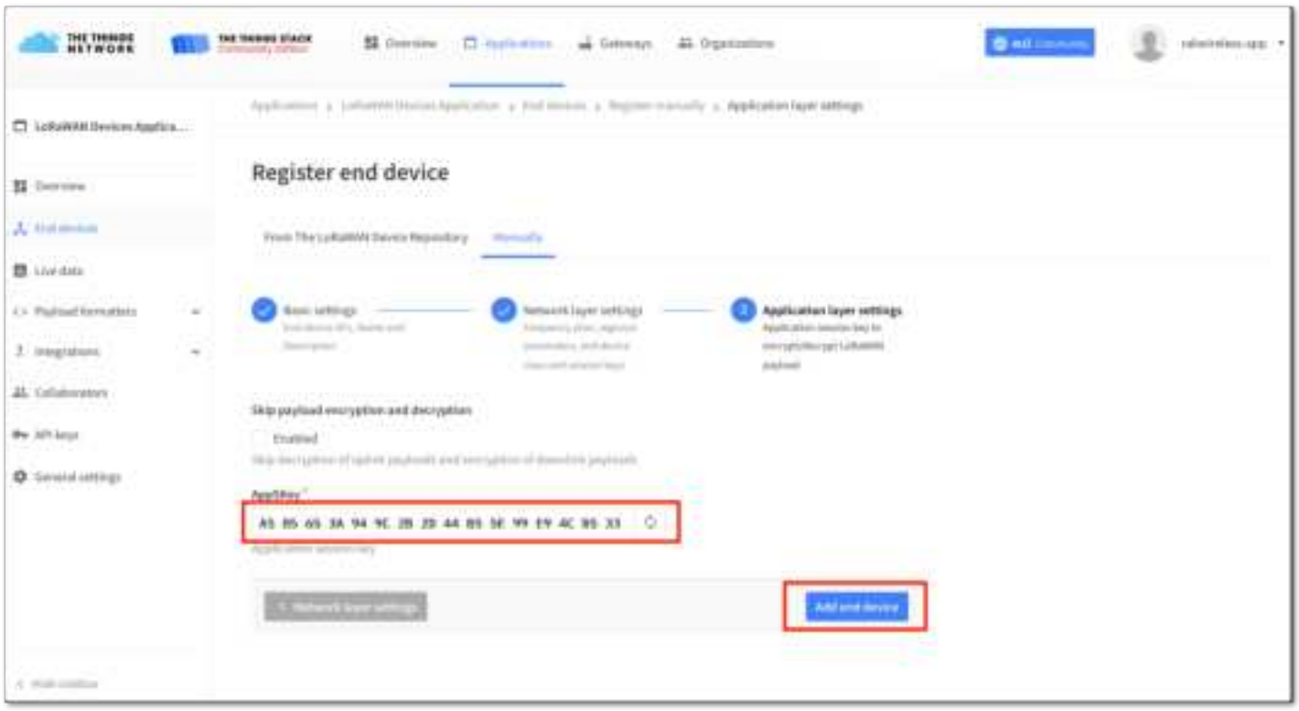

Figure 61: ABP AppSKey generation and device registration

You should now be able to see the device on the TTN console after you fully register your device, as shown in **Figure 57**.

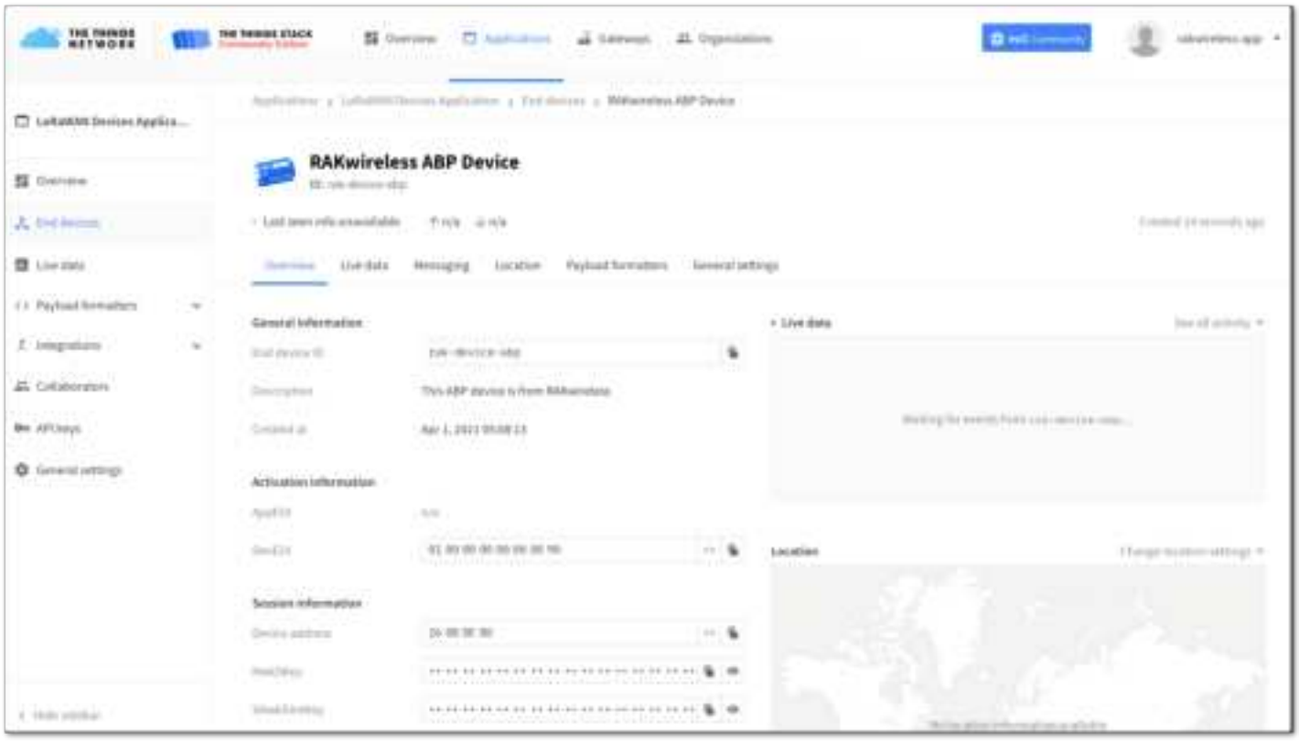

Figure 62: ABP device successfully registered to TTN

ABP Configuration for TTN

1. To set up the RAK3172 module to join the TTN using ABP, start by connecting the RAK3172 module to the computer (see **Figure 30**) 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 45**.

> 📝 **NOTE:**
>
> If there is no `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.

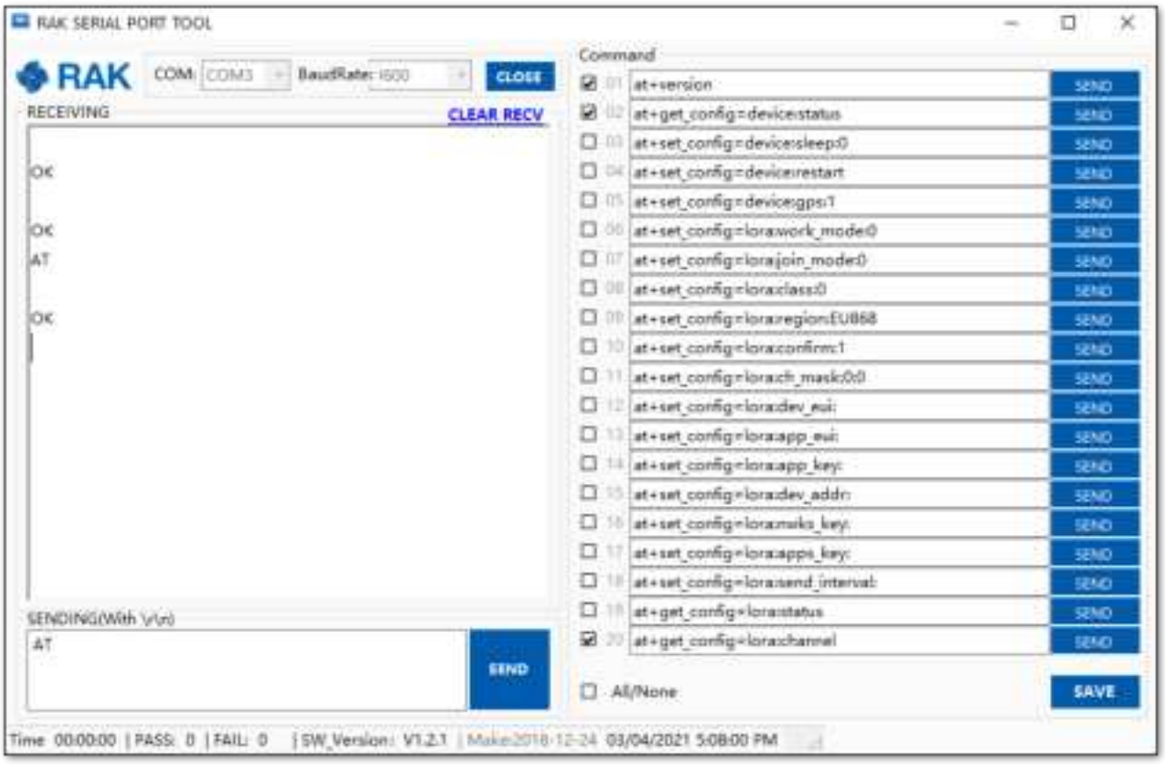

Figure 63: at+version command response

3. The next step is to configure the ABP LoRaWAN parameters in RAK3172:

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

Set the work mode to LoRaWAN.

```
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 RAK3172 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 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	AS923-1
9	AS923-2

Code	Regional Band
10	AS923-3
11	AS923-4

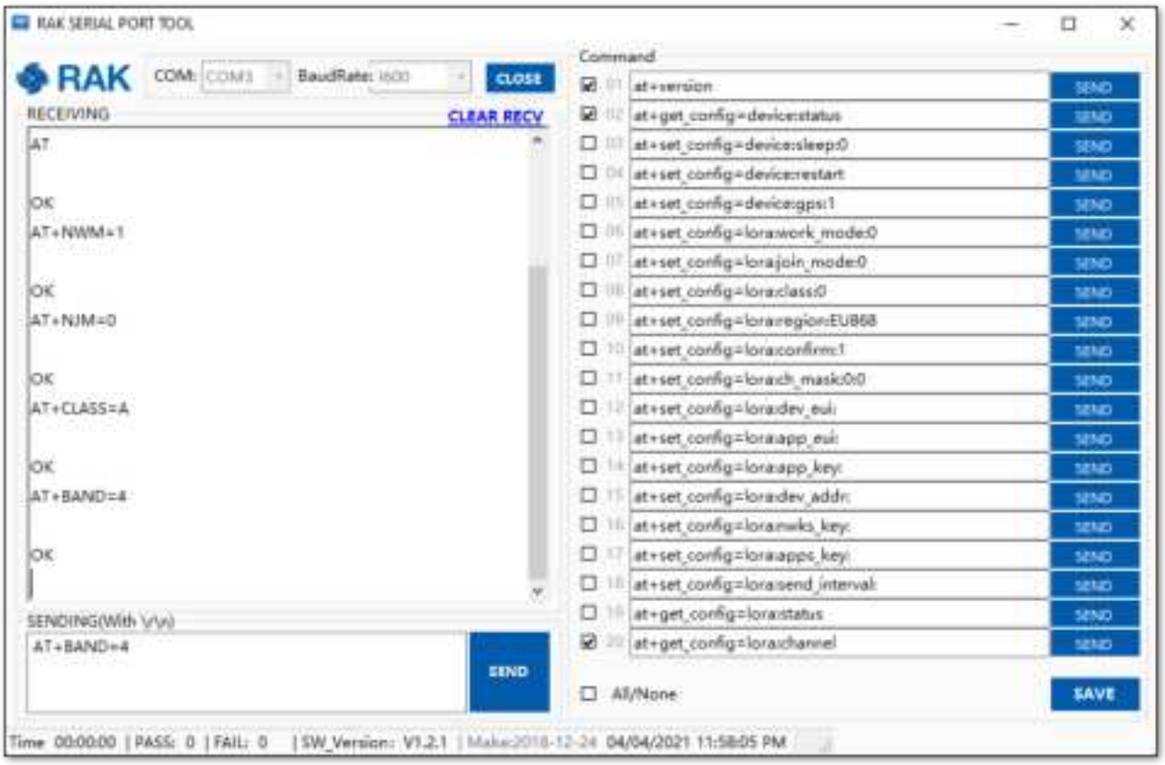

Figure 64: Configuring LoRa Parameters

4. 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 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
```

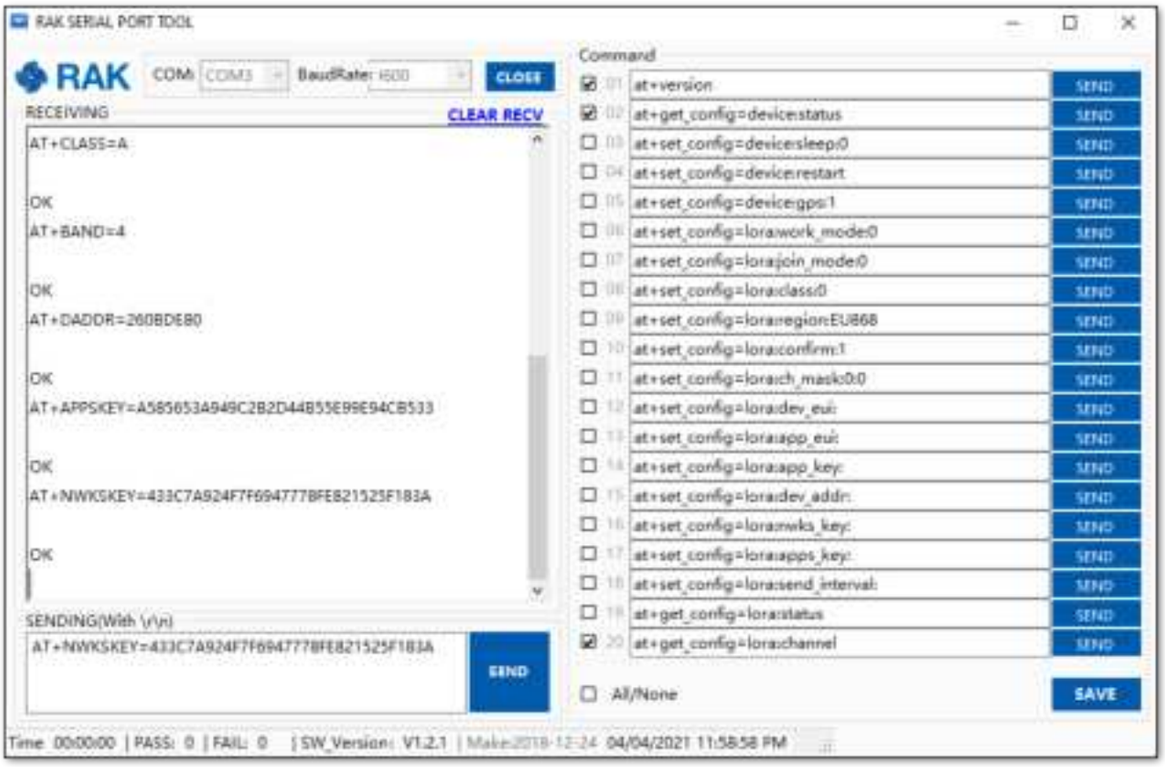

Figure 65: Configuring LoRa Parameters

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.

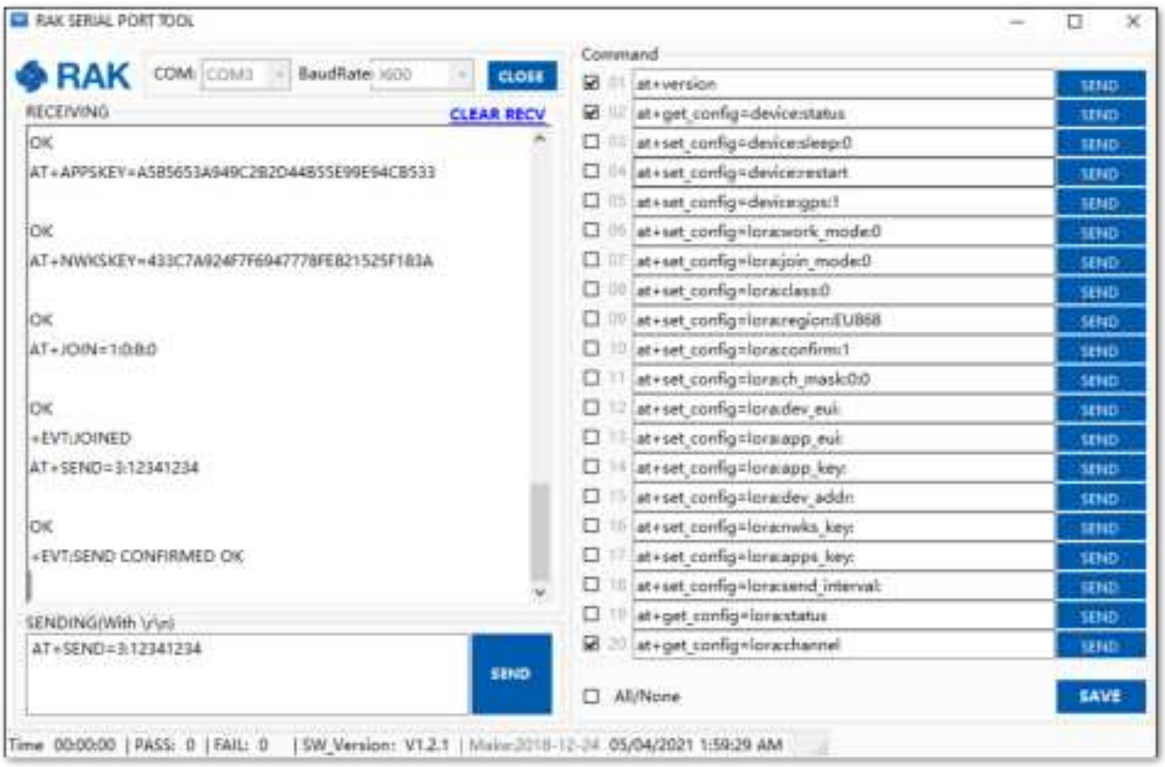

Figure 66: ABP Test Sample Data Sent via RAK Serial Port Tool

7. You can see the data sent by the RAK3172 module on the TTN device console *Live data* section and the *Last seen* info should be a few seconds ago.

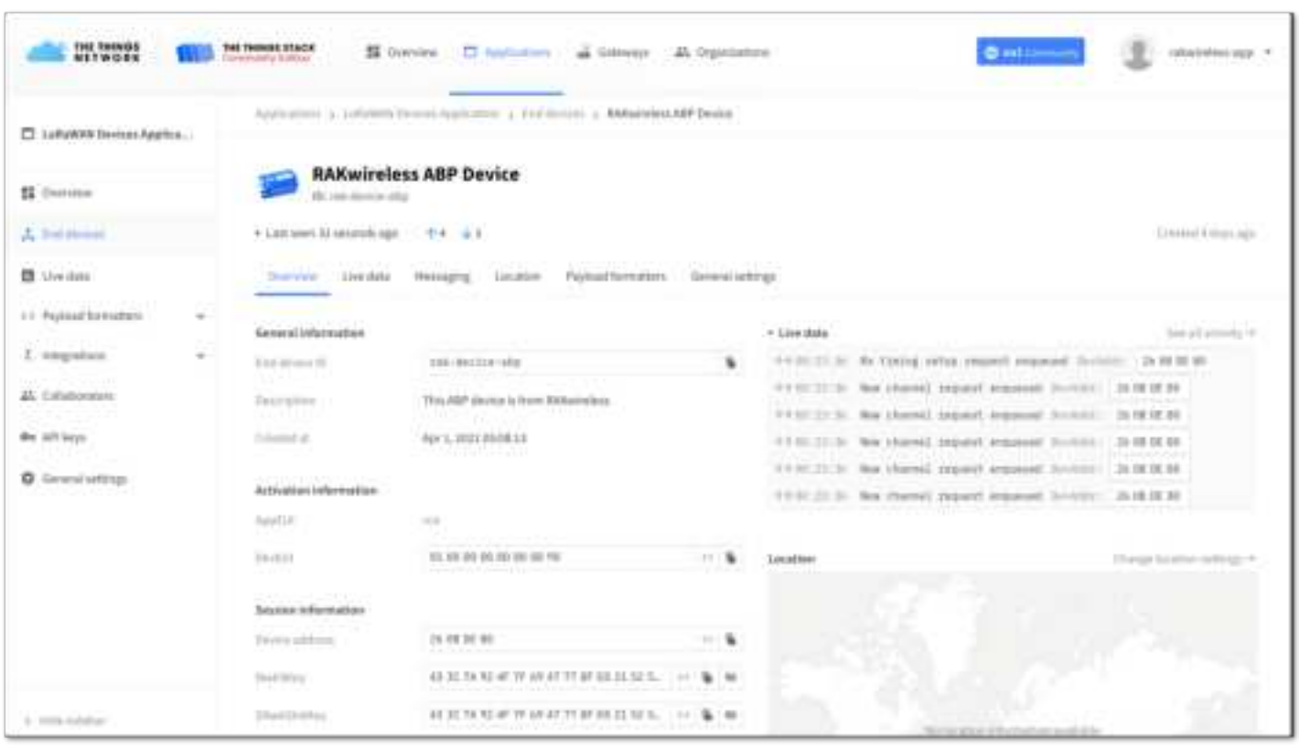

Figure 67: OTAA Test Sample Data Sent Viewed in TTN

Connecting with ChirpStack

This section shows how to connect the RAK3172 module to the ChirpStack platform.

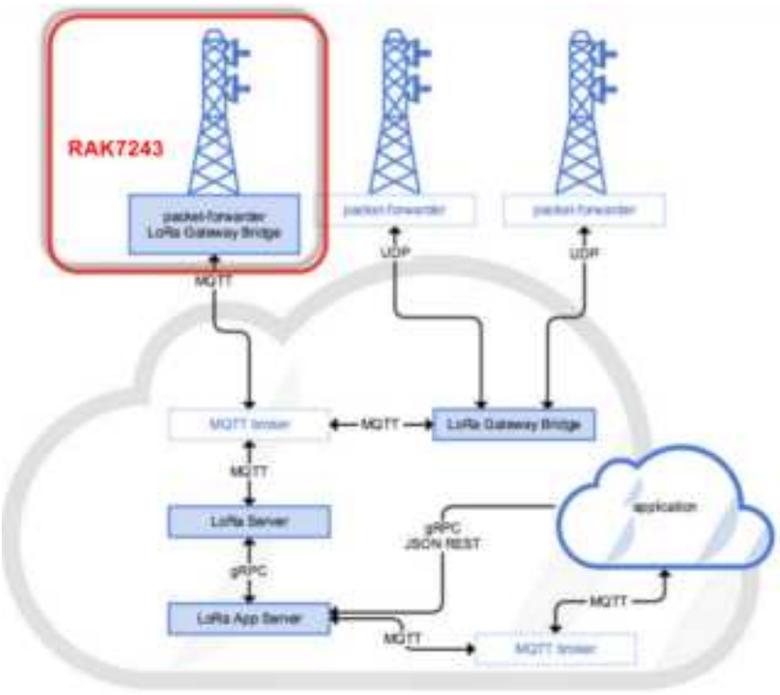

Figure 68: RAK3172 Module in the Context of the ChirpStack Platform

The ChirpStack, previously known as the LoRaServer project, provides open-source components for building LoRaWAN networks. Like in the case of TTN, the RAK3172 module is located in the periphery and will transmit the data to the backend servers through a LoRa gateway. Learn more about ChirpStack .

> 📝 **NOTE:**
>
> 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. The RAK Serial Port Tool provided by RAK
3. RAK3172 module

> 📝 **NOTE:**
>
> The frequency band used in the demonstration is EU868. Use a high-frequency version of RAK3172. The product number should be “**RAK3172 (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 64**.

Figure 69: 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 65** and **Figure 66**.

Figure 70: 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 71: Filling Parameters of an Application

Register a New Device

1. Choose the **Application** created in the previous step, then select the **DEVICES** tab, as shown in **Figure 67** and **Figure 68**.

2. Once done, click "**+ CREATE**".

Figure 72: List of Applications Created

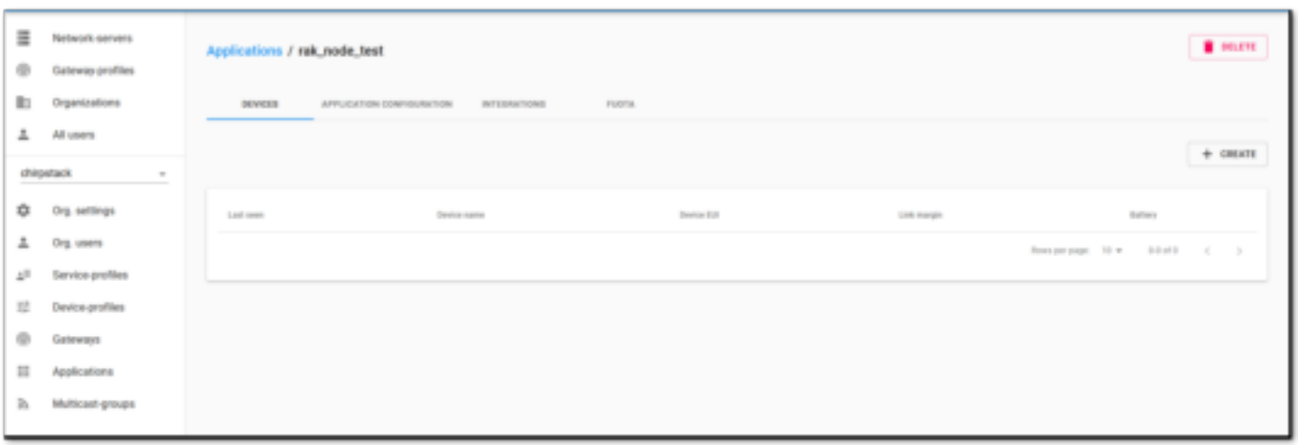

Figure 73: Device Tab of an Application

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

Figure 74: Add a New Device

Figure 75: Chirpstack Adding Node into the RAK3172 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 RAK3172.

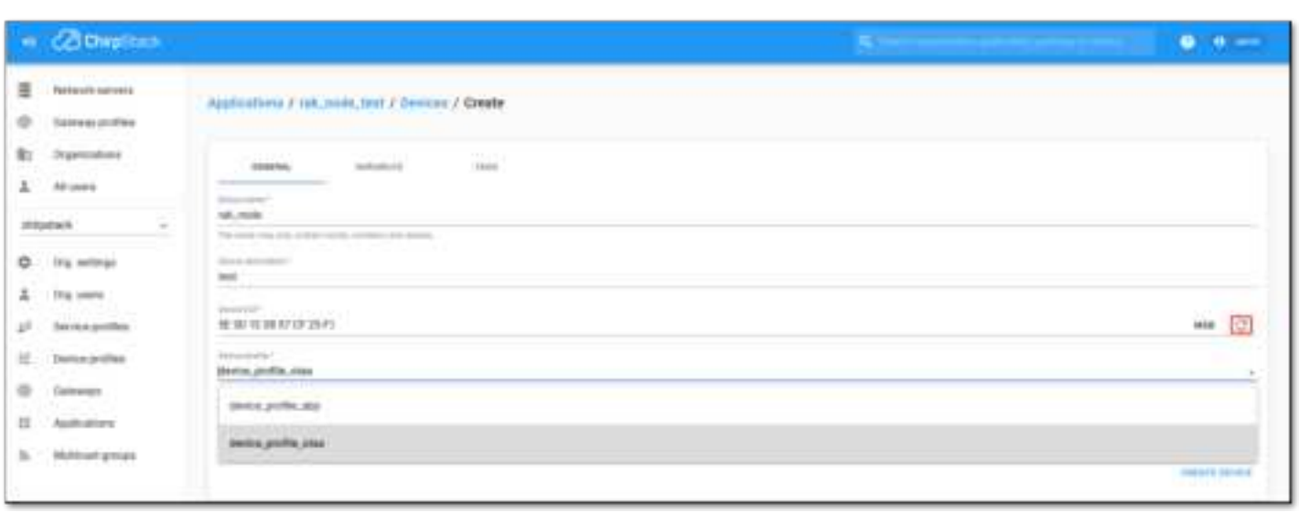

Figure 76: Generate a New Device EUI

Chirpstack OTAA Device Registration

1. If you have selected “**DeviceProfile_OTAA**”, as shown in **Figure 72**, then after the device is created, an Application Key must be also created for this device.

Figure 77: 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 73**.

Figure 78: 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 74**, a new device should be listed in the DEVICES tab. The most important parameters, such as the Device EUI, are shown in the summary.

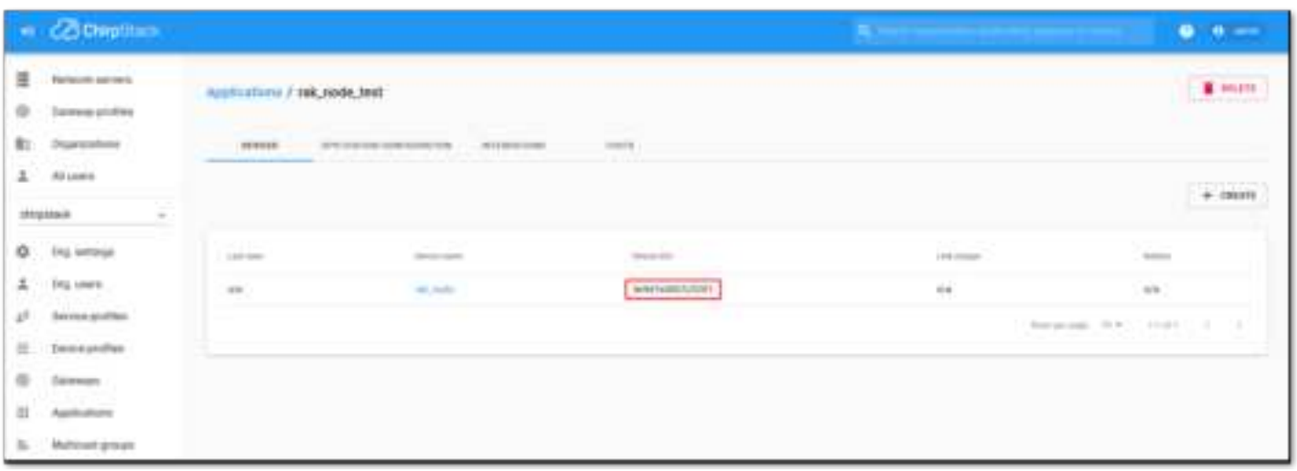

Figure 79: Chirpstack OTAA List of Device in the Device Tab

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 75**.

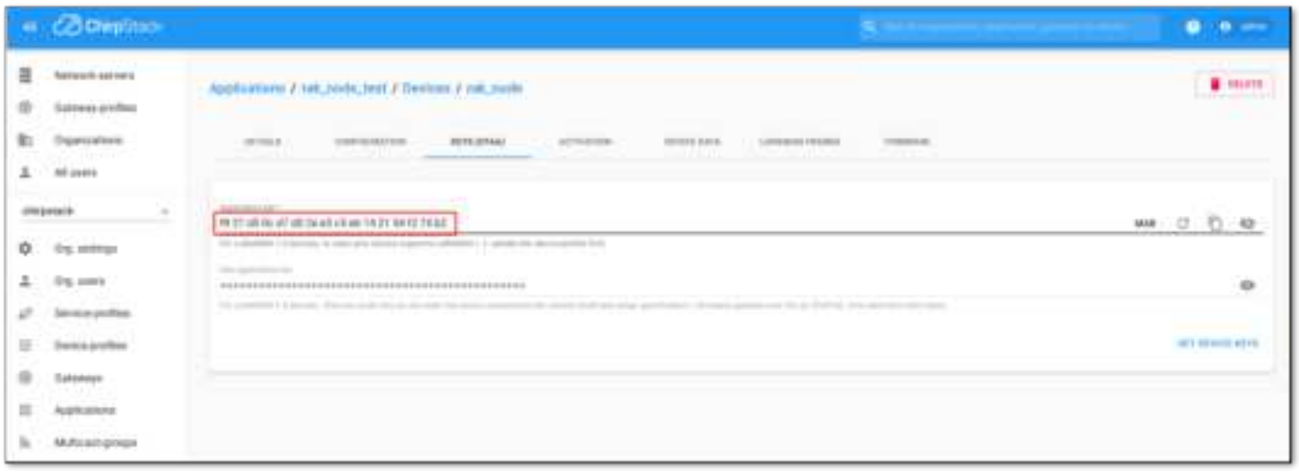

Figure 80: Application Key Associated with the New Device

> 📝 **NOTE:**
>
> Standard OTAA mode requires the **Device EUI**, **Application Key**, and **Application EUI**, but in the ChirpStack's implementation, only the Device EUI and the Application Key are mandatory. The Application EUI is not required and 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 RAK3172 module supports a series of AT commands to configure its internal parameters and control the functionalities of the module.

1. To set up the RAK3172 module to join the Chirpstack using OTAA, start by connecting the RAK3172 module to the Computer (see Figure 30) 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 76**.

> 📝 **NOTE:**
>
> If there is no `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.

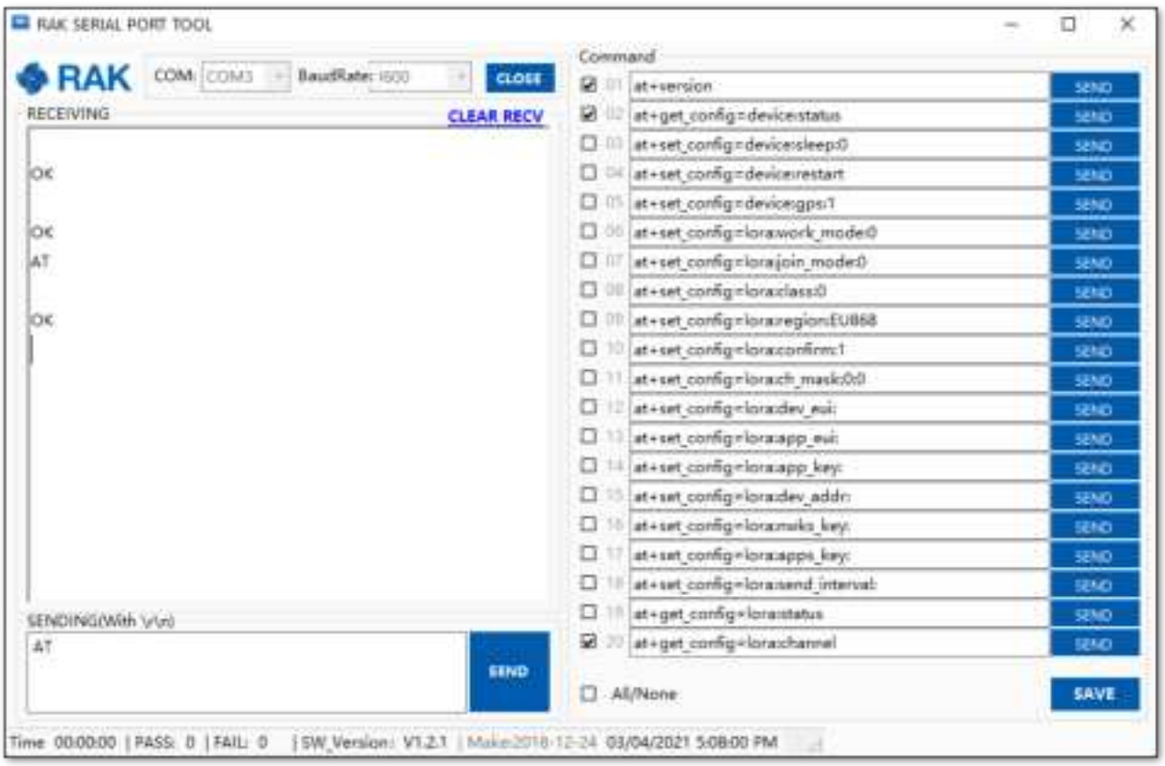

Figure 81: at+version command response

2. The next step is to configure the OTAA LoRaWAN parameters in RAK3172:

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

Set the work mode to LoRaWAN.

```
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 RAK3172 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	AS923-1
9	AS923-2
10	AS923-3
11	AS923-4

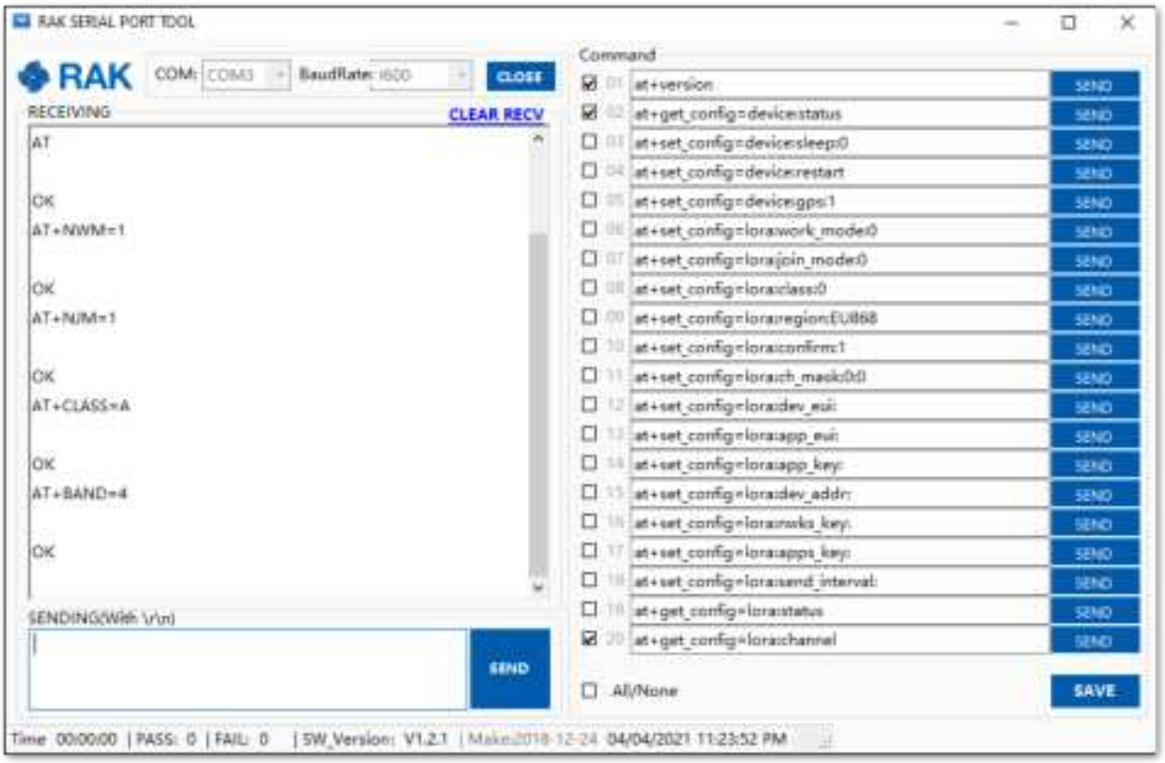

Figure 82: Configuring LoRa Parameters

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
```

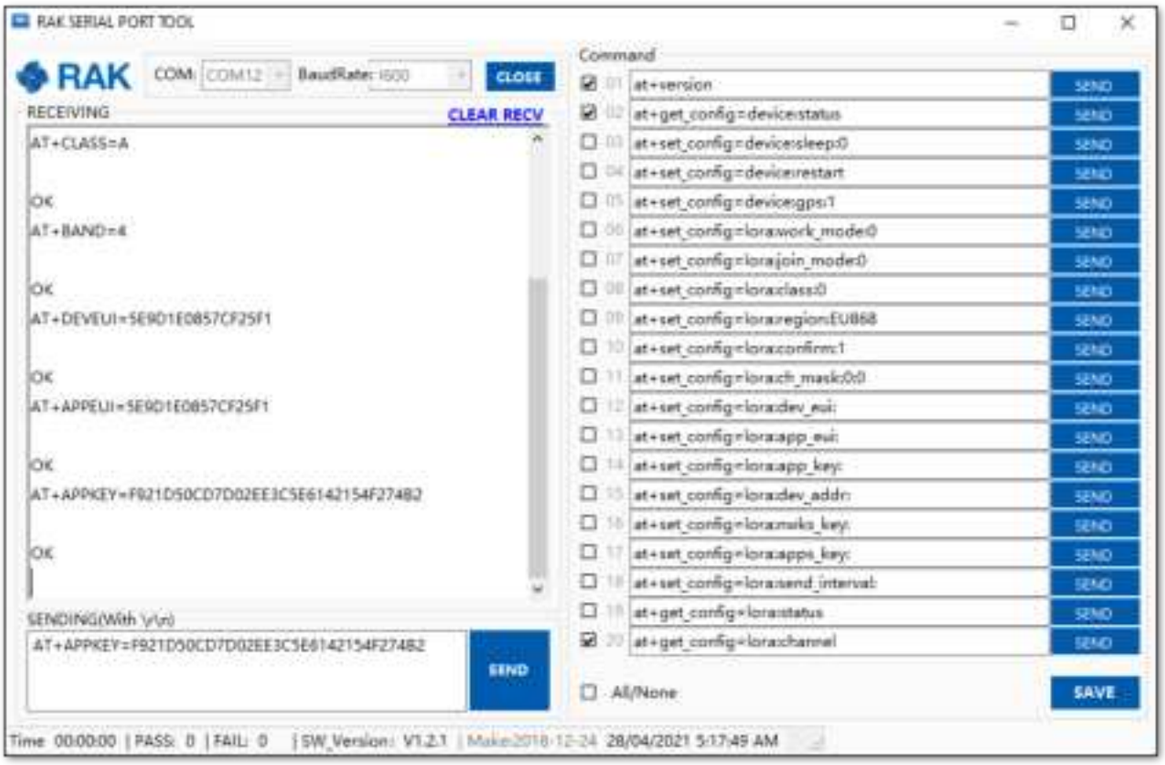

Figure 83: Configuring LoRa Parameters

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

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

> 📝 **NOTE:**
>
> `AT+JOIN` command parameters are optional. You can configure the settings for auto-join, reattempt interval, and the number of join attempts if your application needs it. If not configured, it will use the default parameter values.
>
> `AT+JOIN` and `AT+JOIN=1` also share the common functionality of trying to join the network.

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>`

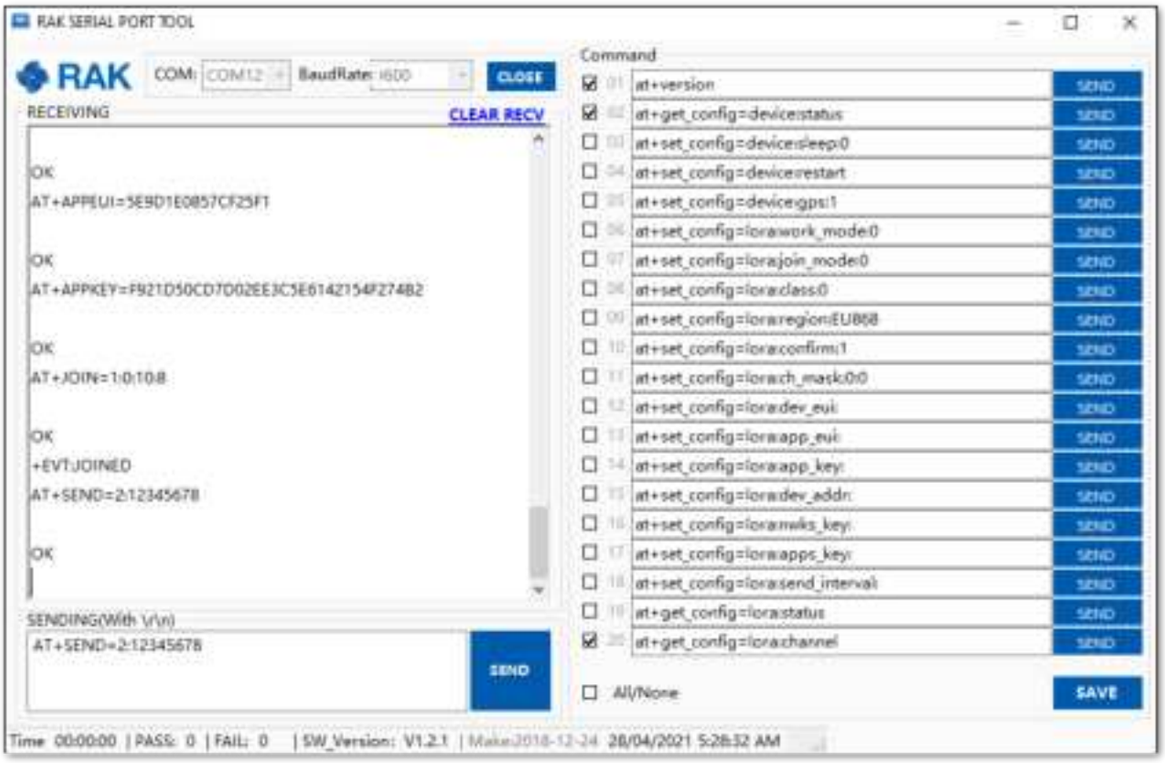

Figure 84: OTAA Test Sample Data Sent via RAK Serial Port Tool

On the ChirpStack platform, you should see the join and uplink messages in the LORAWAN FRAMES tab, as shown in **Figure 80**. By convention, messages sent from nodes to gateways are considered as **Uplinks** while messages sent by gateways to nodes are considered as **Downlinks**.

Figure 85: 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 81**, 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 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 86: 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**

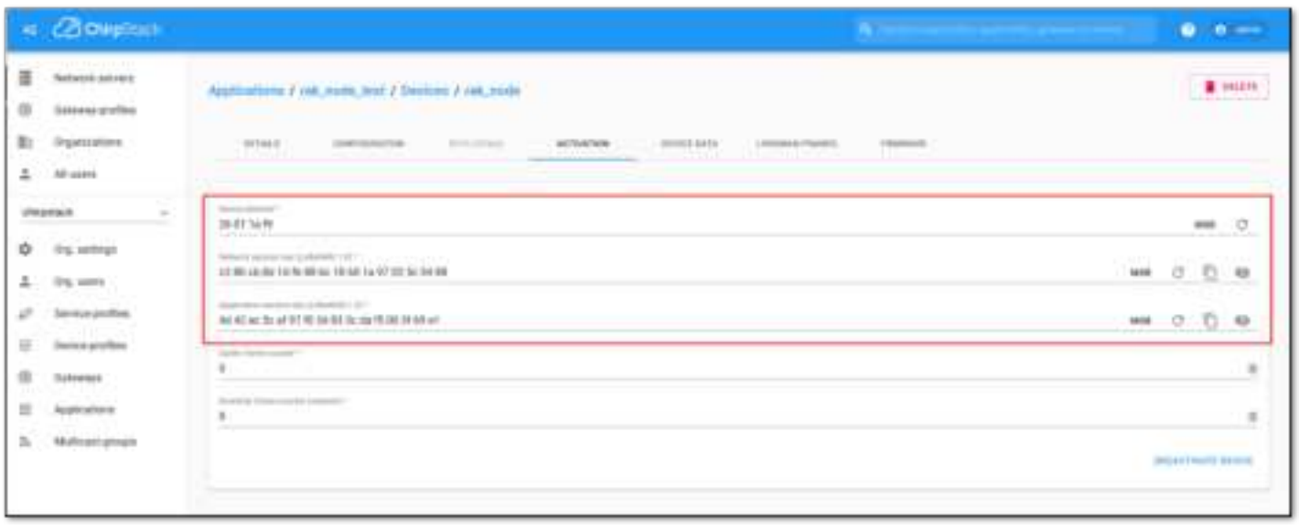

Figure 87: Chirpstack ABP Activation Parameters Needed

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 RAK3172 module to join the Chirpstack using ABP, start by connecting the RAK3172 module to the Computer (see Figure 30) 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 83**.

> 📝 **NOTE:**
>
> If there is no `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.

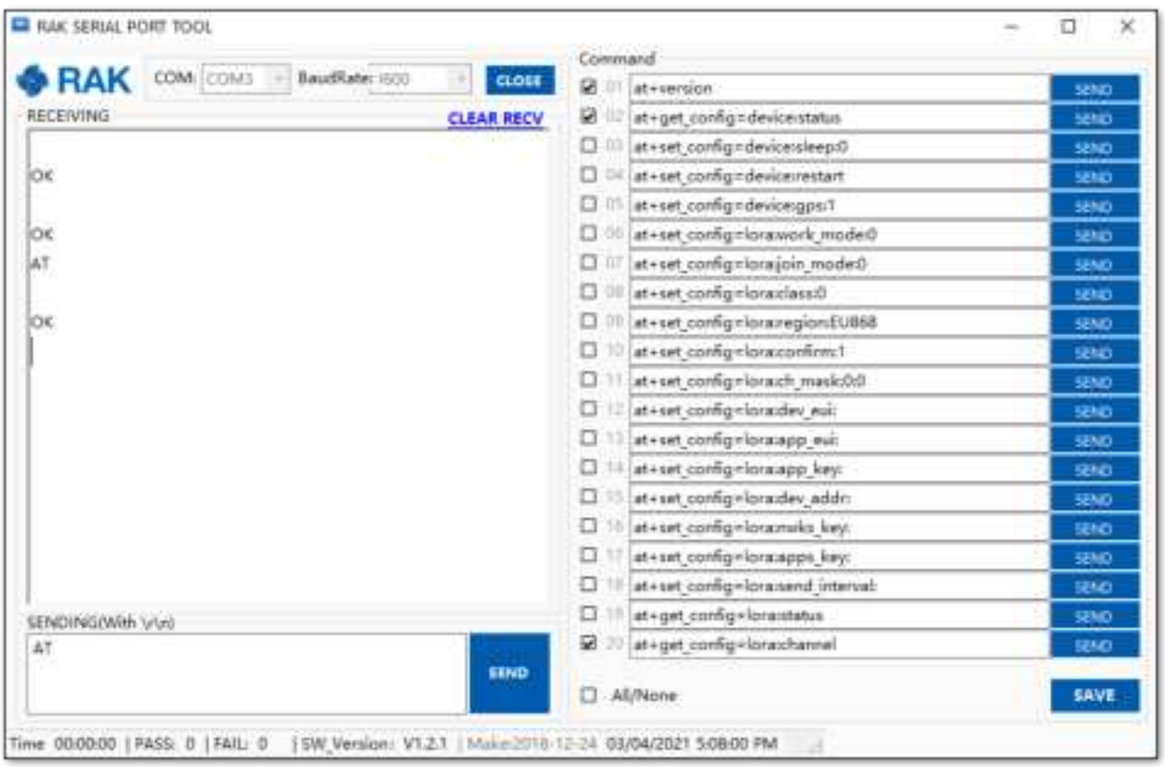

Figure 88: at+version command response

2. The next step is to configure the ABP LoRaWAN parameters in RAK3172:

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

Set the work mode to LoRaWAN. It can be set to P2P as well but by default, the device is in LoRaWAN mode.

```
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 RAK3172 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

Code	Regional Band
7	KR920
8	AS923-1
9	AS923-2
10	AS923-3
11	AS923-4

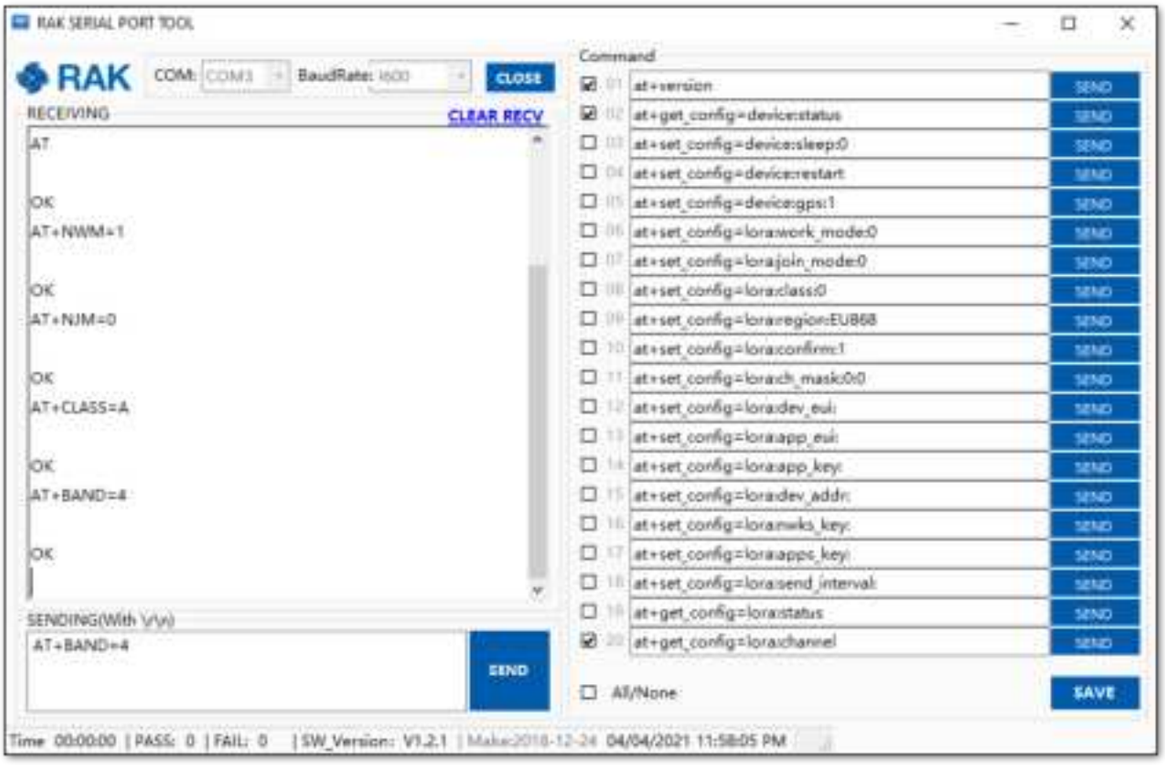

Figure 89: Configuring LoRa Parameters

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 TTN 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
```

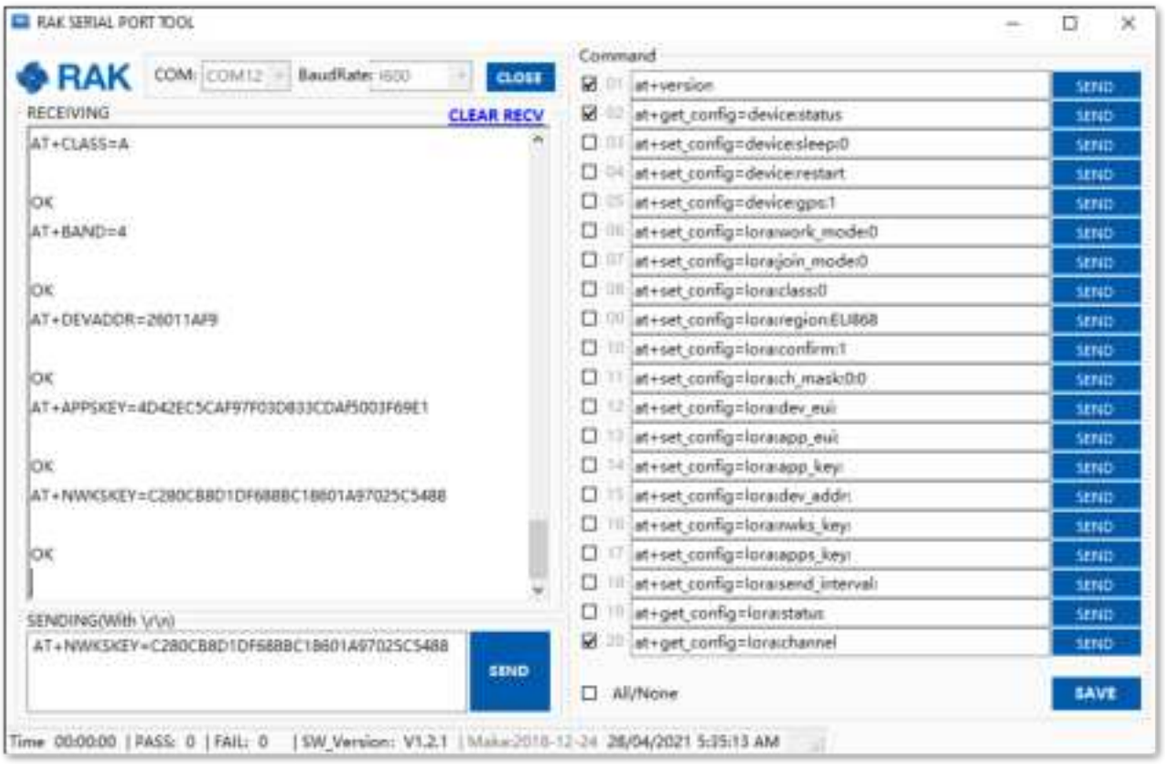

Figure 90: Configuring LoRa Parameters

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

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

> 📝 **NOTE:**
>
> `AT+JOIN` command parameters are optional. You can configure the settings for auto-join, reattempt interval, and the number of join attempts if your application needs it. If not configured, it will use the default parameter values.
>
> `AT+JOIN` and `AT+JOIN=1` also share the common functionality of trying to join the network.

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>`

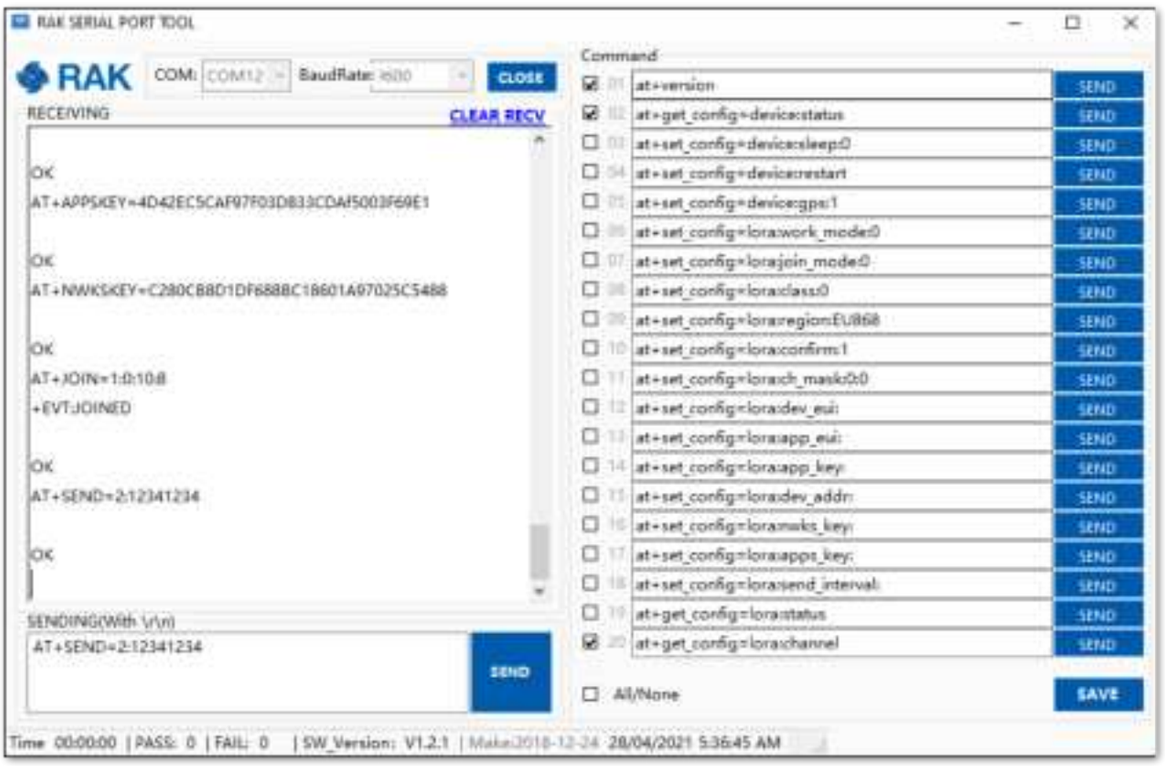

Figure 91: ABP Test Sample Data Sent via RAK Serial Port Tool

LoRa P2P Mode

This section will show you how to set up and connect two RAK3172 units to work in the LoRa P2P mode. The configuration of the RAK3172 units is done by connecting the two modules to a general-purpose computer using a USB-UART converter. The setup of each RAK3172 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`.

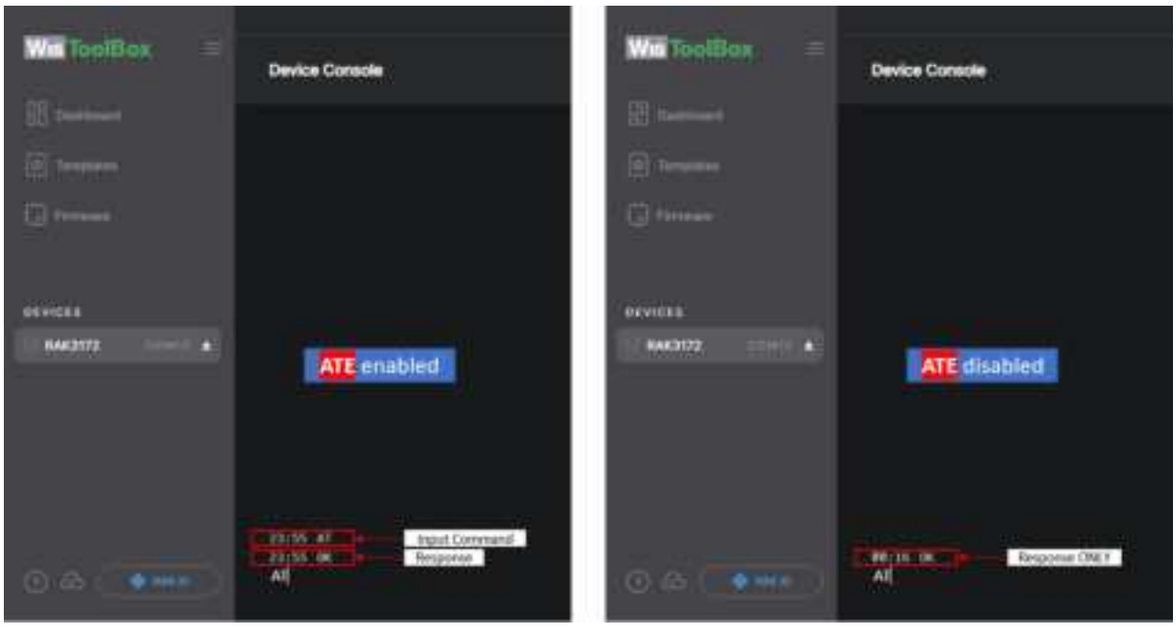

Figure 92: at+version command response

1. In setting up the RAK3172 to work in LoRa P2P mode, you need to change the LoRa network work mode command on both RAK3172 modules.

```
AT+NWM=0
```

`AT+NWM` parameter mode can be either 0=LoRa P2P or 1=LoRaWAN.

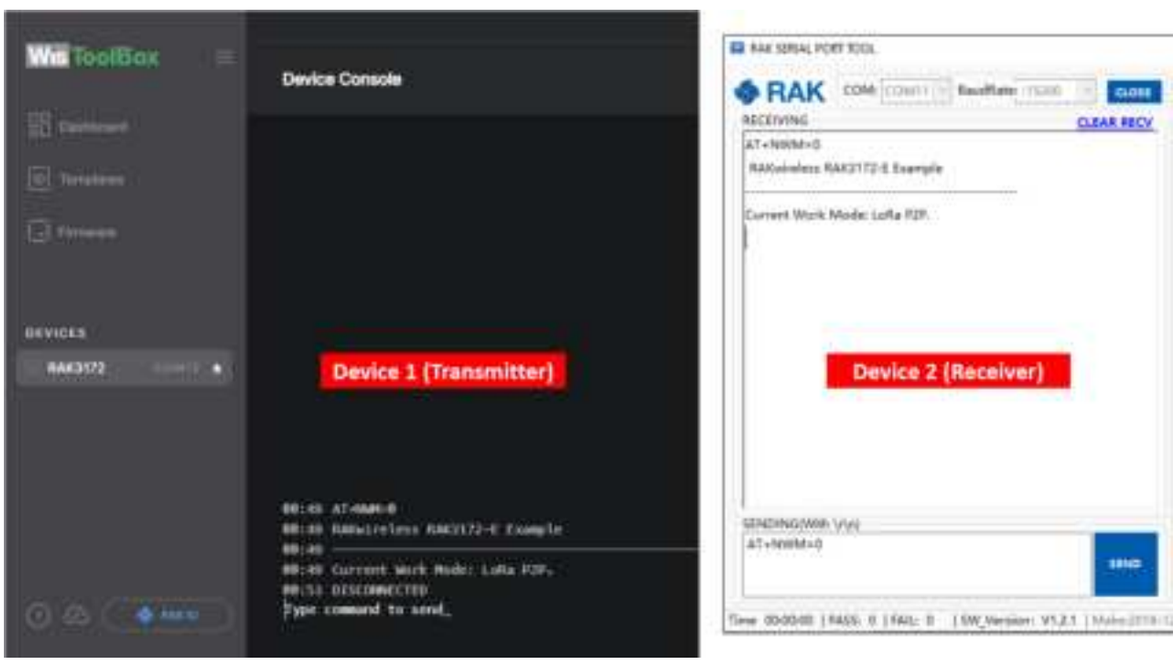

Figure 93: P2P Mode

> 📝 **NOTE:**
>
> - The device will start automatically if you change modes from LoRaWAN to LoRa P2P and vice-versa.
> - You might need to input the `ATE` command again to ensure that your succeeding commands on P2P mode echo on the terminal.

2. You need to input the P2P setup on both RAK3172 modules. The parameters should be exactly the same on the two modules.

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

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**

> **NOTE:**
>
> Refer to the P2P Mode section of the AT command documentation to learn more about the definition of the parameters used and the individual commands if you want specific parameter changed.

Figure 94: Configuring P2P in both RAK3172 Module

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

> 📝 **NOTE:**
>
> LoRa P2P default setting is Transmitter (TX) mode. This consumes lower power compared to Receiver (RX) mode where the radio is always listening for LoRa packets.

a. P2P LoRa RX configurable duration value is from 1 to 65533 ms. In this example, the device will listen and wait for LoRa P2P Packets for 30000 ms or 30 seconds. It will automatically disable RX mode and switch to TX mode after the timeout. If the device did not receive any packets within the time period, then the callback after timeout is `+EVT:RXP2P RECEIVE TIMEOUT`.

```
AT+PRECV=30000
```

b. If the `AT+PRECV` value is set to **65535**, the device will listen to P2P LoRa packets without a timeout, but it will stop listening once a P2P LoRa packet is received. After done receiving the packets, it will disable RX mode and automatically switch to TX mode again.

```
AT+PRECV=65535
```

c. If the `AT+PRECV` value is set to **65534**, the device will continuously listen to P2P LoRa packets without any timeout. The will continuously stay in RX mode until `AT+PRECV` is set to **0**.

```
AT+PRECV=65534
```

d. If the `AT+PRECV` value is set to **0**, the device will stop listening to P2P LoRa packets. It disables LoRa P2P RX mode and switch to TX mode.

```
AT+PRECV=0
```

4. With one module configured as Transmitter (TX) and the other device will be the Receiver (RX), you can now try to send or transmit P2P payload data.

```
AT+PSEND= <payload>
```

> 📝 **NOTE:**
>
> - `AT_PARAM_ERROR` is returned when setting wrong or malformed value.
> - `AT_BUSY_ERROR` is returned if the device is still in RX mode and you try to send or reconfigure RX period. If the `AT+PRECV` command is set to ***65534**, you need to execute first `AT+PRECV=0` to be able to configure again the TX and RX state and avoid `AT_BUSY_ERROR`.
> - `<payload>` : 2~500 digit length, must be an even number of digits and character 0-9, a-f, A-F only, representing 1~256 hexadecimal numbers. For example, if the payload is like `0x03, 0xAA, 0x32`, therefore the AT command should be `AT+PSEND = 03AA32`.

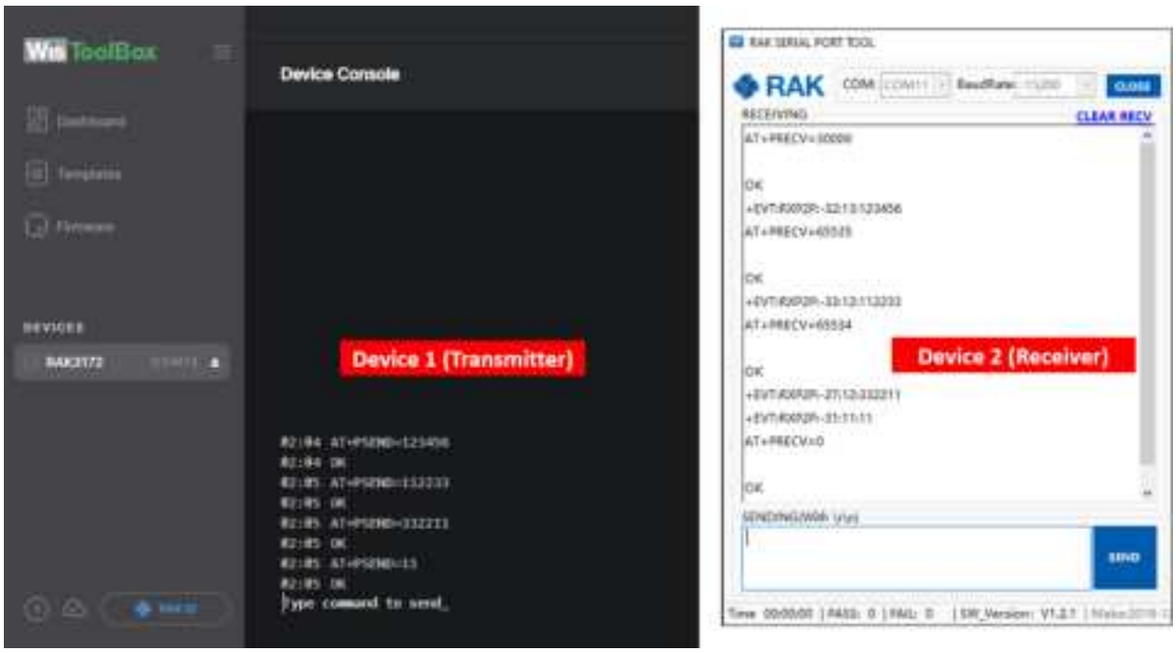

Figure 95: Configuring P2P in both RAK3172 Module

Miscellaneous

Upgrading the Firmware

If you want to upgrade to the latest version of the firmware of the module, you can follow this section. The latest firmware can be found in the software section of RAK3172 Datasheet.

> 📝 **NOTE:**
>
> **What if the RAK3172 module stops responding to AT commands and firmware update?**
>
> You can recover your device by using the `.hex` file in the datasheet and upload it using STM32CubeProgrammer. The guide on updating STM32 firmware using STM32CubeProgrammer can be found in the Learn section.
>
> **WARNING:** Uploading the **.hex** file via STM32CubeProgrammer will erase all configured data on the device.

Firmware Upgrade Through UART2

Minimum Hardware and Software Requirements

Refer to the table for the minimum hardware and software required to perform the firmware upgrade via UART2:

Hardware/Software	Requirement
Computer	A Windows/Ubuntu/Mac computer
Firmware File	Bin firmware file downloaded from the website
Others	A USB to TTL module

Firmware Upgrade Procedure

Execute the following procedure to upgrade the firmware in Device Firmware Upgrade (DFU) mode through the UART2 interface.

> 📝 **NOTE:**
>
> RAK3172 should automatically go to BOOT mode when the firmware is uploaded via RAK DFU Tool or WisToolBox.
>
> If BOOT mode is not initiated, pull to ground the RESET pin twice (or double click the reset button if available) to force BOOT mode.

1. Download the latest application firmware of the RAK3172.
   - RAK3172 Firmware
2. Download the RAK Device Firmware Upgrade (DFU) tool.
   - RAK Device Firmware Upgrade (DFU) Tool
3. Connect the RAK3172 module with a computer through a USB to TTL. Refer to Figure 30.
4. Open the Device Firmware Upgrade tool. Select the serial port and baud rate (115200) of the module and click the "Select Port" button.

> 📝 **NOTE:**
>
> If your firmware upload always fails, check your current baud rate setting using the `AT+BAUD=?` command and use that baud rate value in the RAK DFU Tool. You can also check if you selected the right COM port.

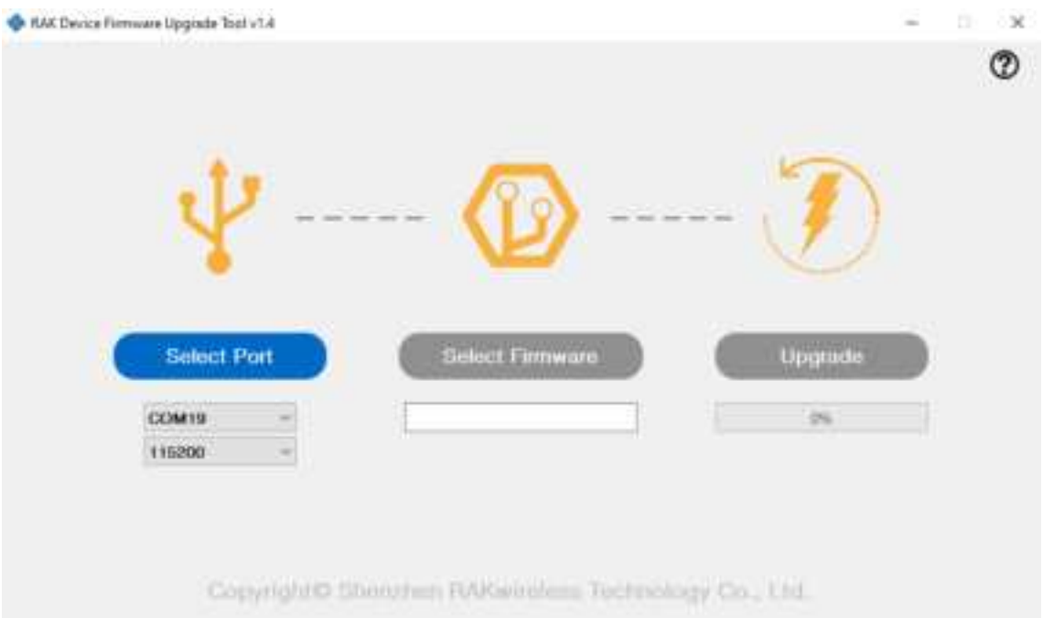

Figure 96: Device Firmware Upgrade Tool

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

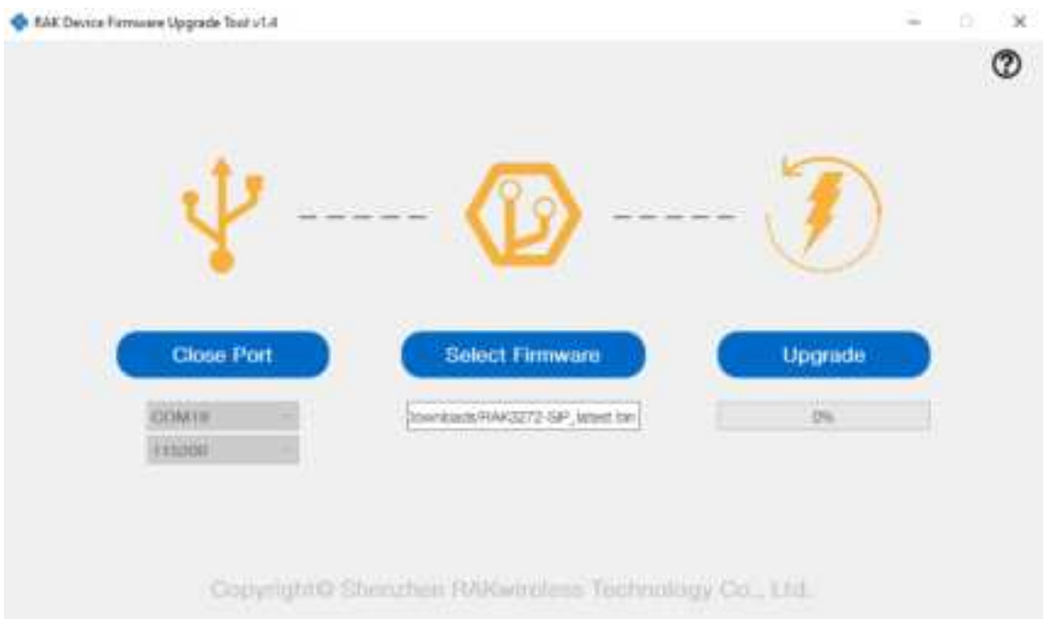

Figure 97: Select firmware

6. Click the "**Upgrade**" button to upgrade the device. After the upgrade is complete, the RAK3172 will be ready to work with the new firmware.

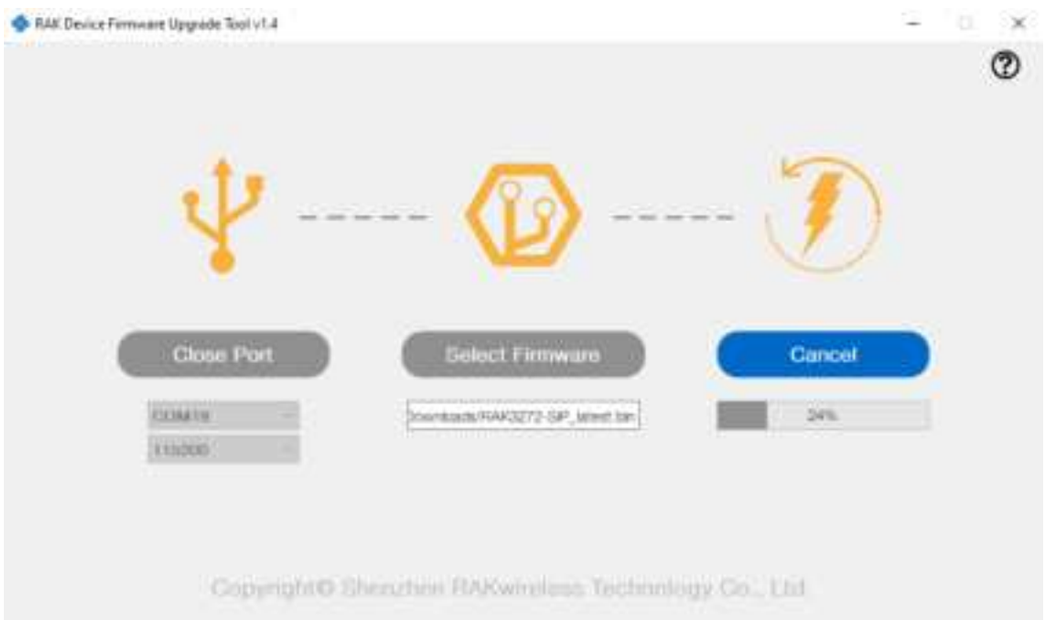

Figure 98: Firmware upgrading

Figure 99: Upgrade successful

Arduino Installation

Refer to Software section.

Last Updated: 9/16/2022, 1:37:35 PM