Arduino

Heltec WiFi LoRa 32 V2 Arduino Guide

Introduction to Guide

> INFO
>
> Before we begin, please make sure you've followed the steps from this guide, which goes over some initial setup steps.

Objective and Requirements

In this guide, you will learn:

- How to setup your environment
- How to program a basic application that will send packets over the Helium Network
- Verify real-time packets sent to the Helium Console via Hotspot that's in range

For this example, you will need the following:

Hardware

- Heltec WiFi LoRa 32 V2
- Micro USB Type B Cable - Example

Software

- Arduino software (IDE)
- Helium Console

Hardware Setup

Adding the Antenna

Your board should have come with a U.FL antenna. All you have to do is attache it to the U.FL port as shown in the image at the top of the guide.

Connect Board

Next, lets connect our board to our computer with a USB 2.0 A-Male to Micro B cable.

Software Setup

Getting the Arduino IDE

Download and install the latest version of Arduino IDE for your preferred OS.

- Windows
- Linux
- Mac OSX

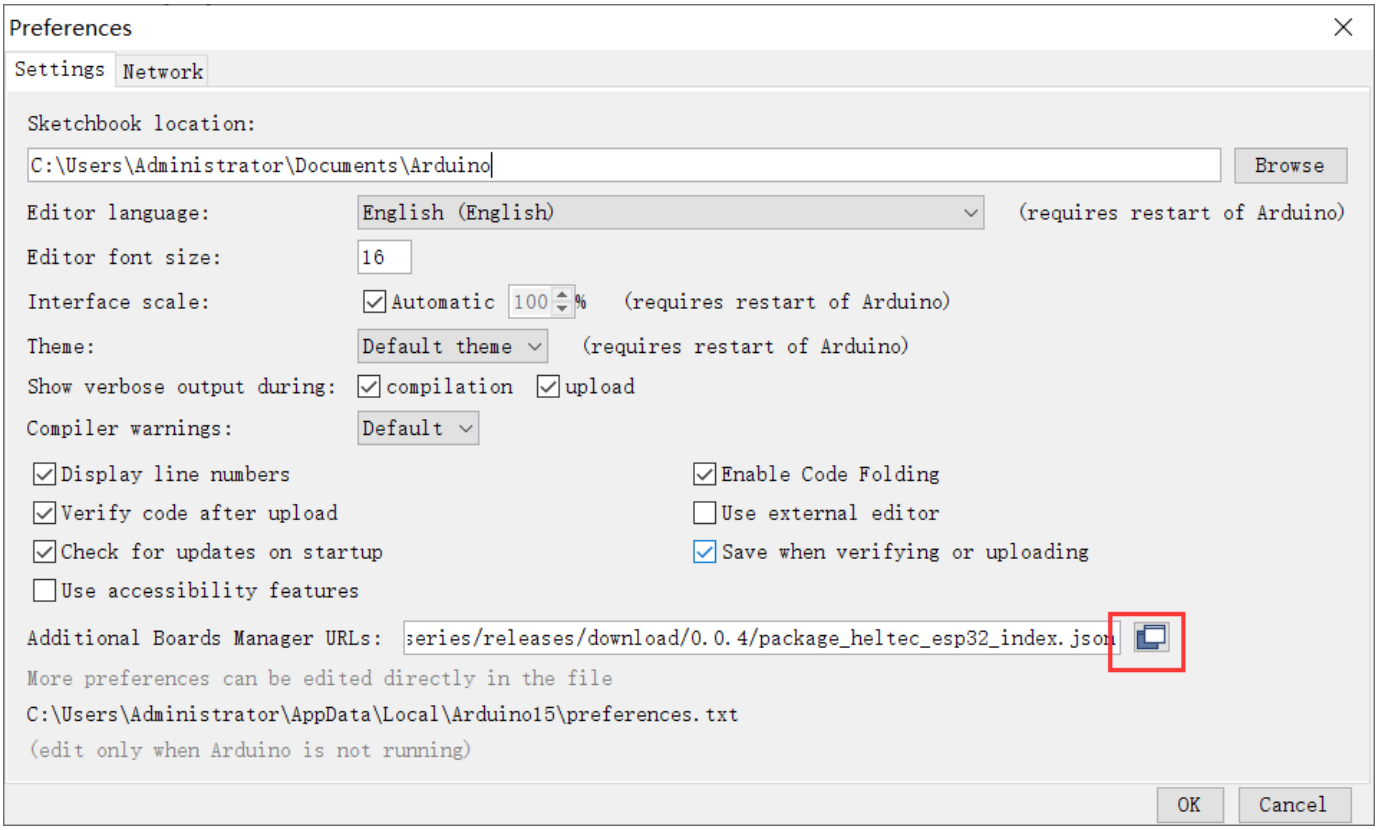
Arduino Board Support

The Heltec WiFi LoRa 32 V2 requires one Arduino board support package. Follow the instructions below to install.

Arduino-ESP32

To install, open your Arduino IDE:

1. Navigate to **(File > Preferences), (Arduino > Preferences) on MacOS.**
2. Find the section at the bottom called **Additional Boards Manager URLs:**



1. Add this URL in the text box:

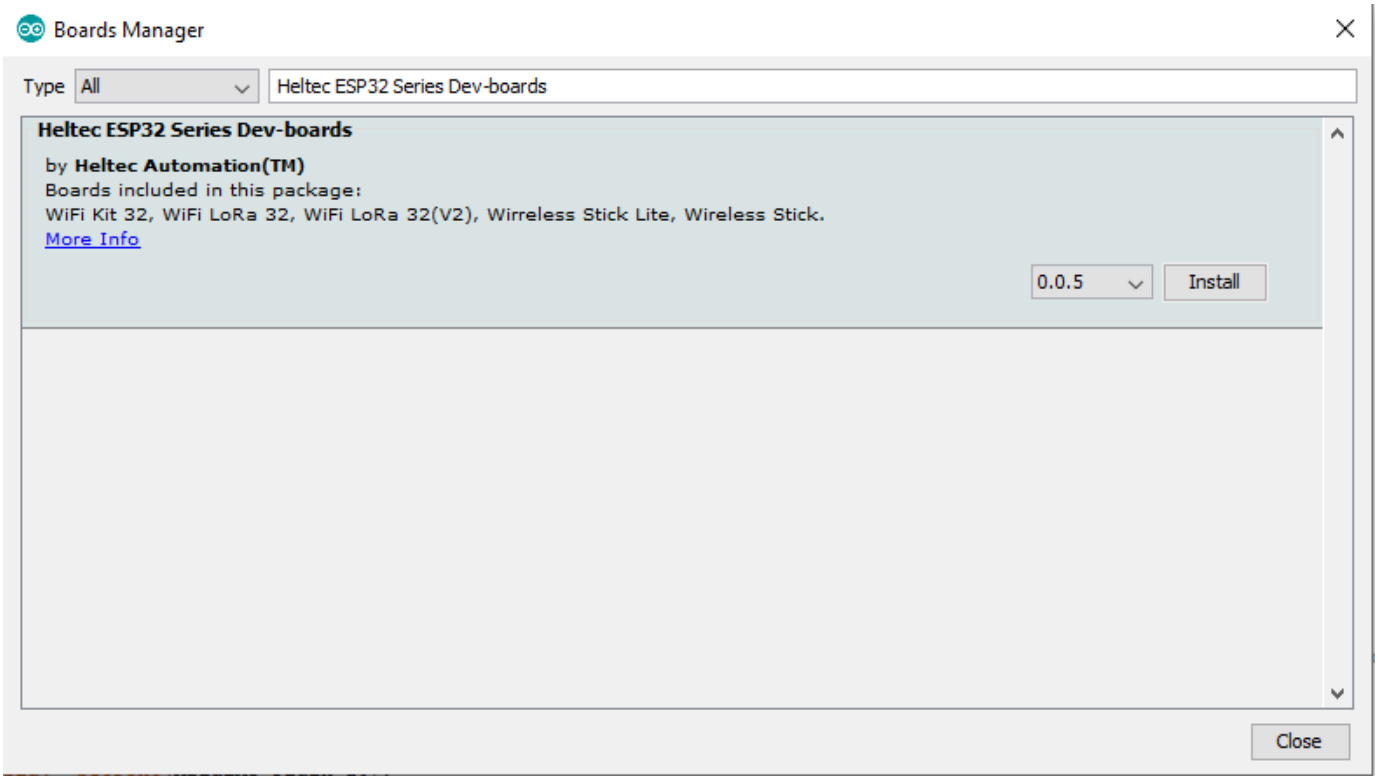
```
https://github.com/Heltec-Aaron-
Lee/WiFi_Kit_series/releases/download/0.0.5/package_heltec_esp32_index.jso
```

> **NOTE**
>
> The latest versions can be found on Heltec Automation Docs website.

1. Close the Preferences windows

Next, to install this board support package:

1. Navigate to (**Tools > Boards > Boards Manager...)**
2. Search for **Heltec ESP32**
3. Select the newest version and click Install



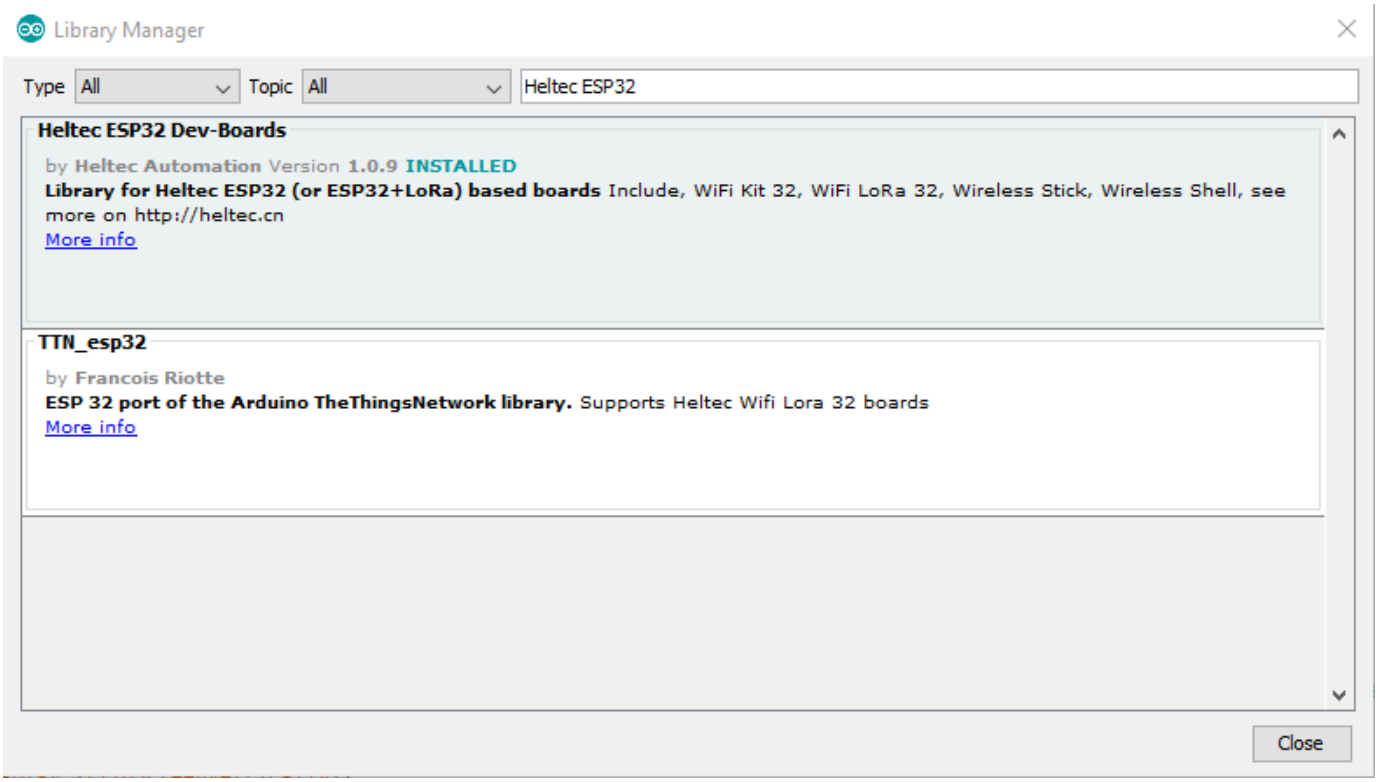
Arduino Library

To communicate with Helium's LoRaWAN network, we'll need to install two Arduino libraries.

Heltec ESP32

To install, open your Arduino IDE:

1. Navigate to Library Manager (**Sketch > Include Library > Manage Libraries**).
2. In the search box, type **Heltec ESP32** into the search, select the version shown below, and click Install.



)

Heltec ESP32 LoRaWAN

Download this find into the directory below and unzip.

```
linux: /home/{user}/Arduino/libraries
windows: Documents\Arduino\libraries
mac os: Documents/Arduino/libraries
```

Install Serial Driver

Find Directions on Heltec's website here.

Select Board

Arduino IDE:

1. Select Tools -> Board: -> WiFi LoRa 32(V2)

Obtain Heltec License Key

Upload GetChipID example

Arduino IDE:

1. Select File -> Examples -> ESP32 -> ChipID -> GetChipID
2. Select Tools -> Port: "COM# or ttyACM#"
3. Select Sketch -> Upload
4. Wait for Done uploading message
5. Select Tools -> Serial Monitor Serial Monitor Window
6. Select 115200 baud from bottom right dropdown.
7. You should see something that looks like this every second `ESP32 Chip ID = ############`
8. Save this Chip ID

Obtain License Key with Chip ID

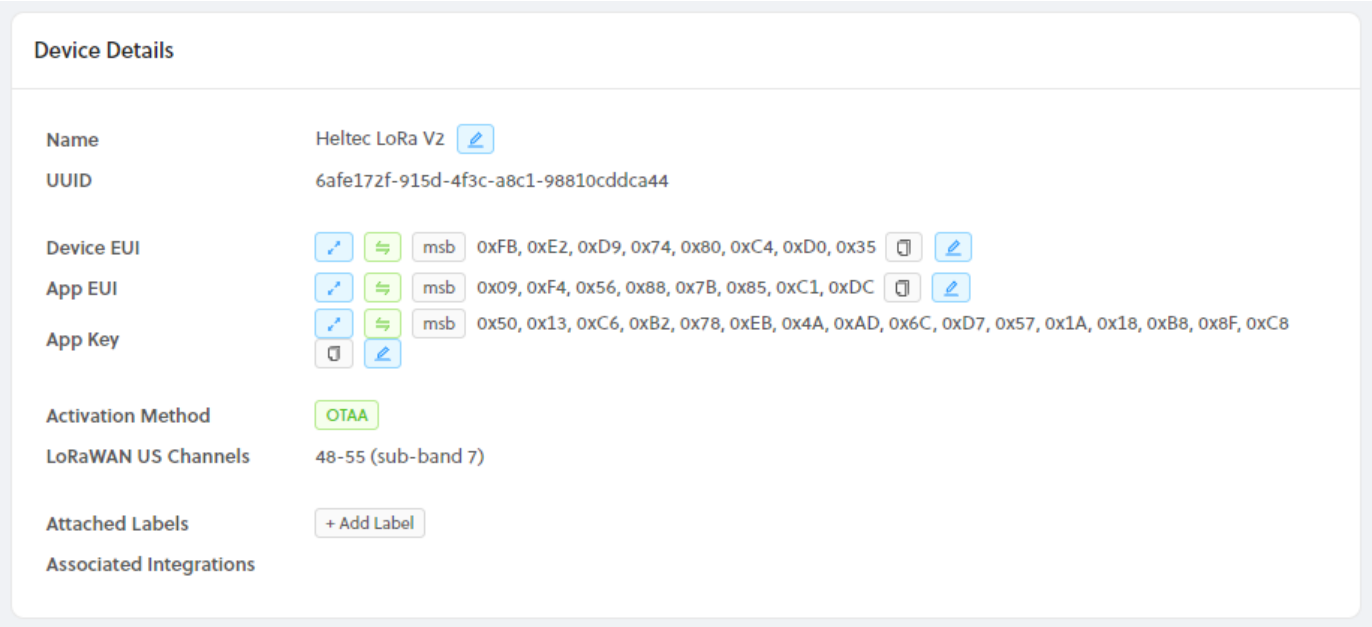
1. Go to resource.heltec.cn/search
2. Enter ChipID
3. Save license field, will look like `0x########,0x#########,0x########,0x########`

Programming Example Sketch

Now that we have the required Arduino board support and library installed, lets program the board with the provided example sketch.

To create a new Arduino sketch, open your Arduino IDE, (**File > New).** Next, replace the template sketch with the sketch found here, copy and paste the entirety of it.

Next we'll need to fill in the AppEUI(msb), DevEUI(msb), and AppKey(msb), in the sketch, which you can find on the device details page on Console. Be sure to use the formatting buttons to match the endianess and formatting required for the sketch, shown below.



At the top of the sketch, replace the three **FILL_ME_IN** fields, with the matching field from Console, example shown below.

```
/* OTAA para*/
uint8_t DevEui[] = { 0xFB, 0xE2, 0xD9, 0x74, 0x80, 0xC4, 0xD0, 0x35 };
uint8_t AppEui[] = { 0x09, 0xF4, 0x56, 0x88, 0x7B, 0x85, 0xC1, 0xDC };
uint8_t AppKey[] = { 0x50, 0x13, 0xC6, 0xB2, 0x78, 0xEB, 0x4A, 0xAD, 0x6C, 0xD7, 0x57, 0x1A, 0x18, 0xB8, 0x8F, 0xC8 };
```

)

Finally, we will paste in the License Key we obtained above to the sketch. Replace the values in license[4] with the key obtained from resource.heltec.cn/search

```
/*license for Heltec ESP32 LoRaWan, quary your ChipID relevant license: http://resource.heltec.cn/search */
uint32_t  license[4] = {0xC1670CF8, 0x19C71AD5, 0x6CE47540, 0x8CF267EC};
```

Selecting Port

Next, we need to select the correct Serial port in the Arduino IDE. Navigate to (**Tools > Port: COM#/ttyACM#**). You will also see either **COM# or /dev/ttyACM#** depending on whether you are on Windows, Mac, or Linux. If you do not see a port, please visit the Drivers section in Heltec's docs to make sure you have what's needed for your operating system.

Select LoRaWAN Region

The last step before we upload our sketch is to select the correct LoRaWAN Region, navigate to **(Tools > LoRaWAN Region: > REGION_US915).**

Required Change to Default DataRate

In this example sketch, ADR is turned off, therefore you must manually set the desired data rate for your payload size manually. This library uses a default data rate that is not supported by the Helium Network so you must change it to one in the following range DR_0 - DR_4. To do this change the default data rate on line 20 in the file found below for your operating system

`linux: /home/{user}/Arduino/libraries/ESP32_LoRaWAN-master/src/ESP32_LoRaWAN.cpp`

`windows: Documents\Arduino\libraries\ESP32_LoRaWAN-master\src\ESP32_LoRaWAN.cpp`

`mac os: Documents/Arduino/librariesESP32_LoRaWAN-master/src/ESP32_LoRaWAN.cpp`

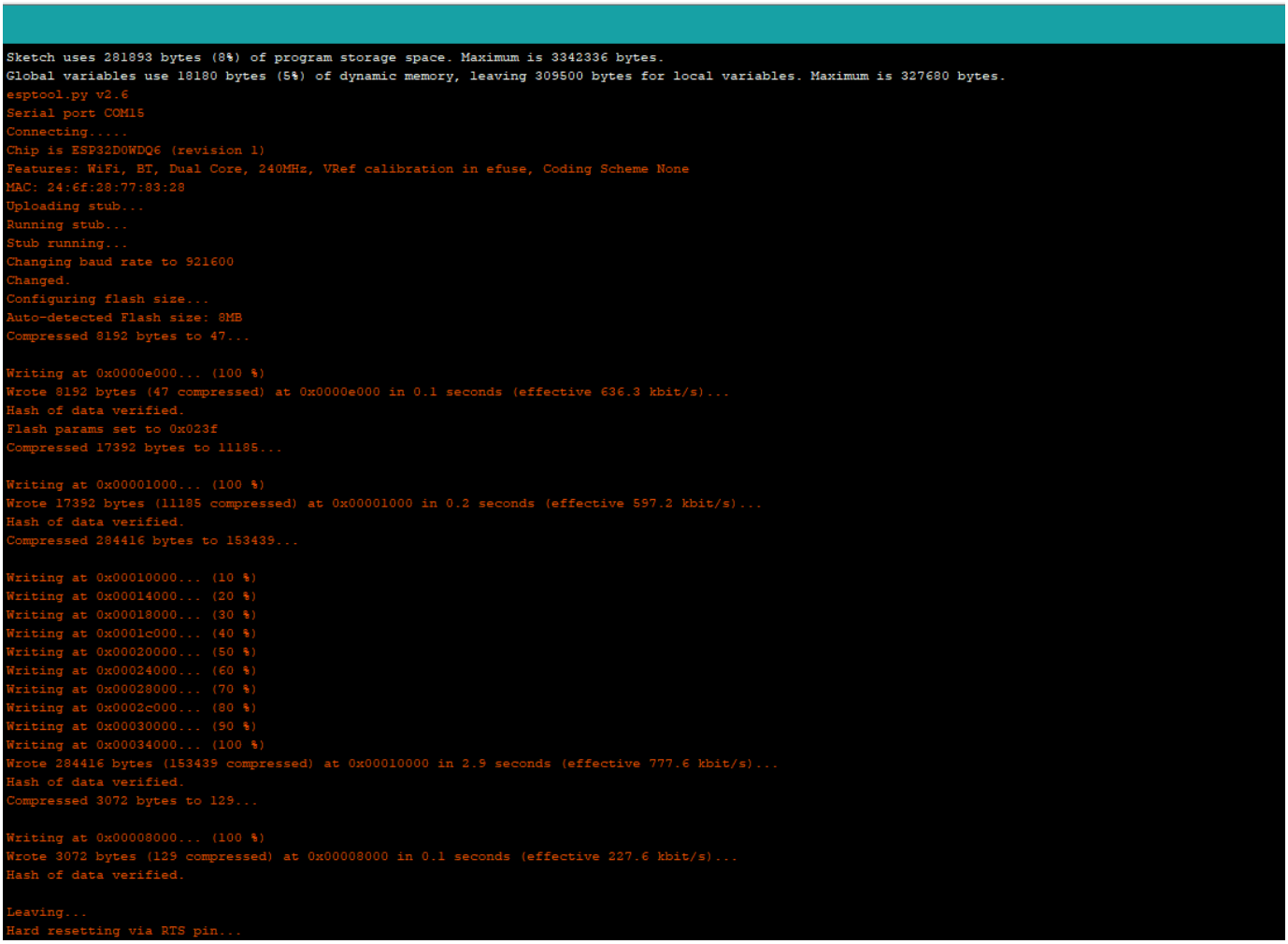
Change line 20 to:

```
#define LORAWAN_DEFAULT_DATARATE                    DR_3
```

> **INFO**
>
> If you don't want to change data rate manually, Helium now supports ADR

Upload Sketch

We're finally ready to upload our sketch to the board. In the Arduino IDE, click the right arrow button, or navigate to (**Sketch > Upload)**, to build and upload your new firmware to the board. You should see something similar to the image below at the bottom of your Arduino IDE, when the upload is successful.



)

Viewing Serial Output

When your firmware update completes, the board will reset, and begin by joining the network. Let's use the Serial Monitor in the Arduino IDE to view the output from the board. We first need to select the serial port again, but this time it will be a **different port** than the one we selected to communicate with the bootloader. Once again, navigate to (**Tools > Port: COM#/ttyACM#**), but make sure the serial device, either COM# or ttyACM#, is different! Next navigate to (**Tools > Serial Monitor**), you should begin to see output similar to below.

```
LoRaWAN US915 Class A start!

joining...
join failed, rejoin at 30000 ms later

LoRaWAN US915 Class A start!

joining...
joined
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
```

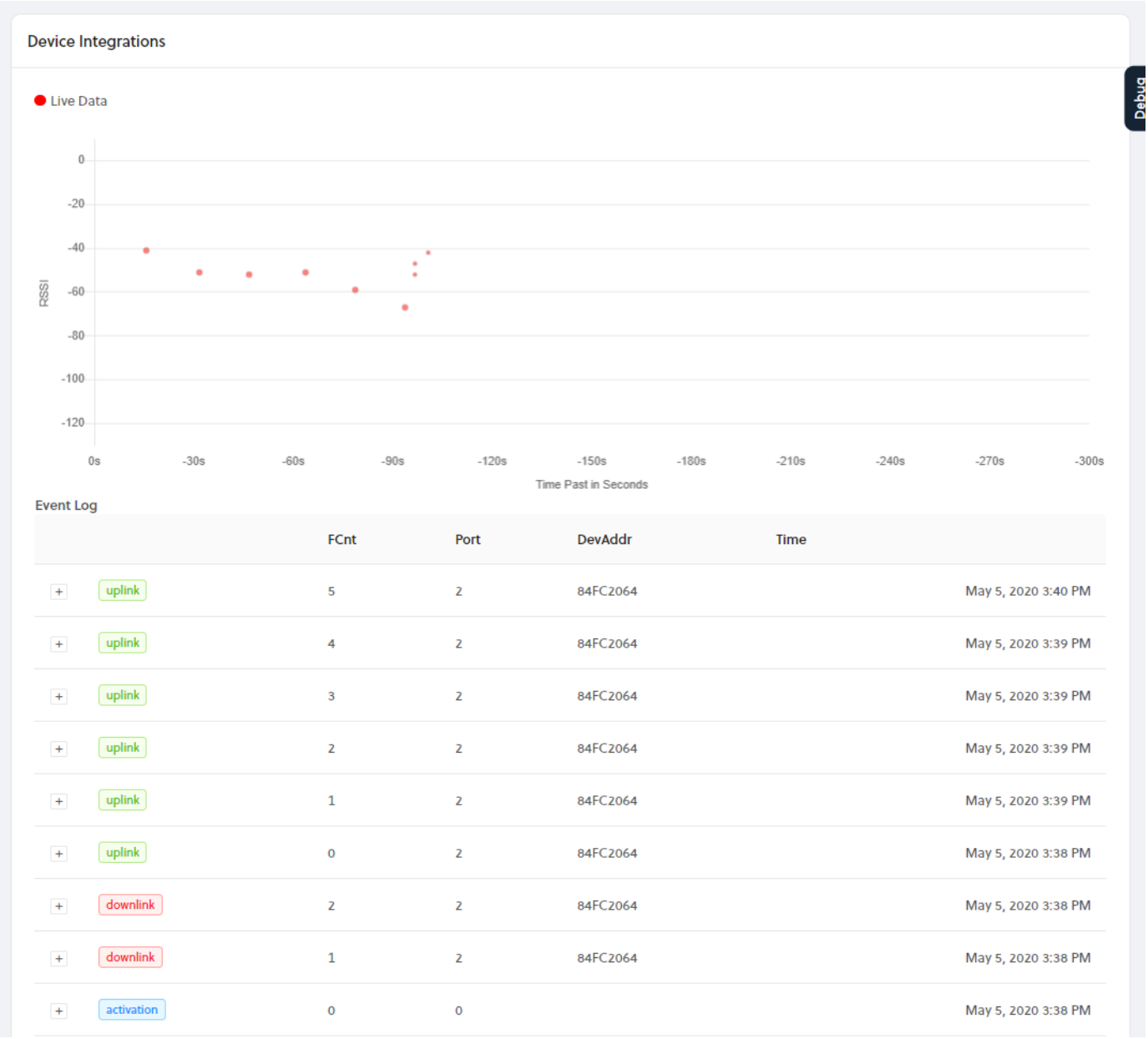
)

Your device may take several minutes to join and begin to send uplink packets because the library is designed to work in several LoRaWAN regions and networks. Because of this, the firmware will attempt different sub-bands until it is successful. If you would like to change the default channel mask in the library to speed up this process you can make the following change below the sketch.

Change line 59 to:

```
uint16_t userChannelsMask[6]={
0xFF00,0x0000,0x0000,0x0000,0x0002,0x0000 };
```

Now let's head back to Helium Console and look at our device page, you should see something similar to the screenshot below.



Congratulations! You have just transmitted data on the Helium network! The next step is to learn how to use your device data to build applications, visit our Integrations docs here.

Edit this page