

# 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 attach it to the U.FL port as shown in the image at the top of the guide.

## Connect Board

Next, let's 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

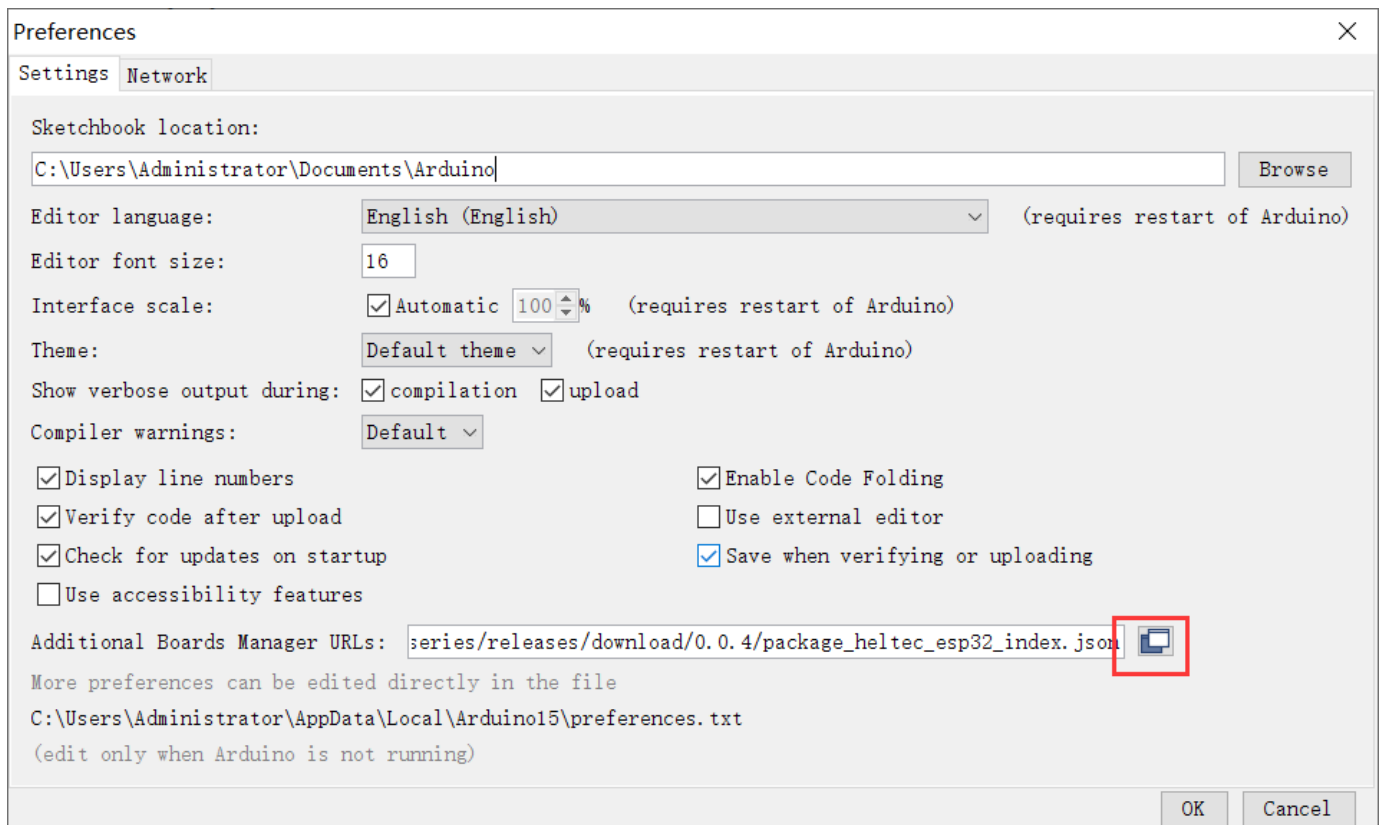
The Heltec WiFi LoRa 32 V2 requires one Arduino board support package. Follow the instructions below to install.

### 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.js
```

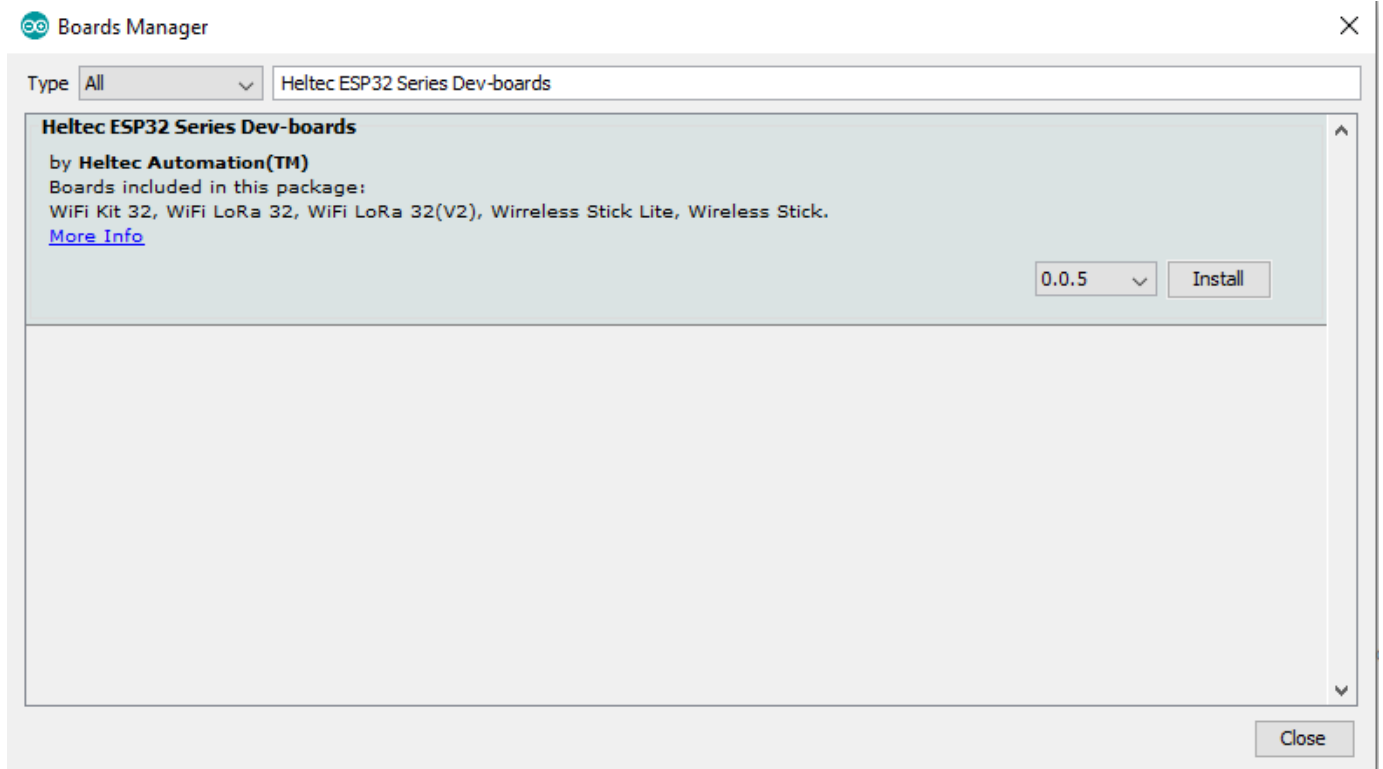
**i NOTE**

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



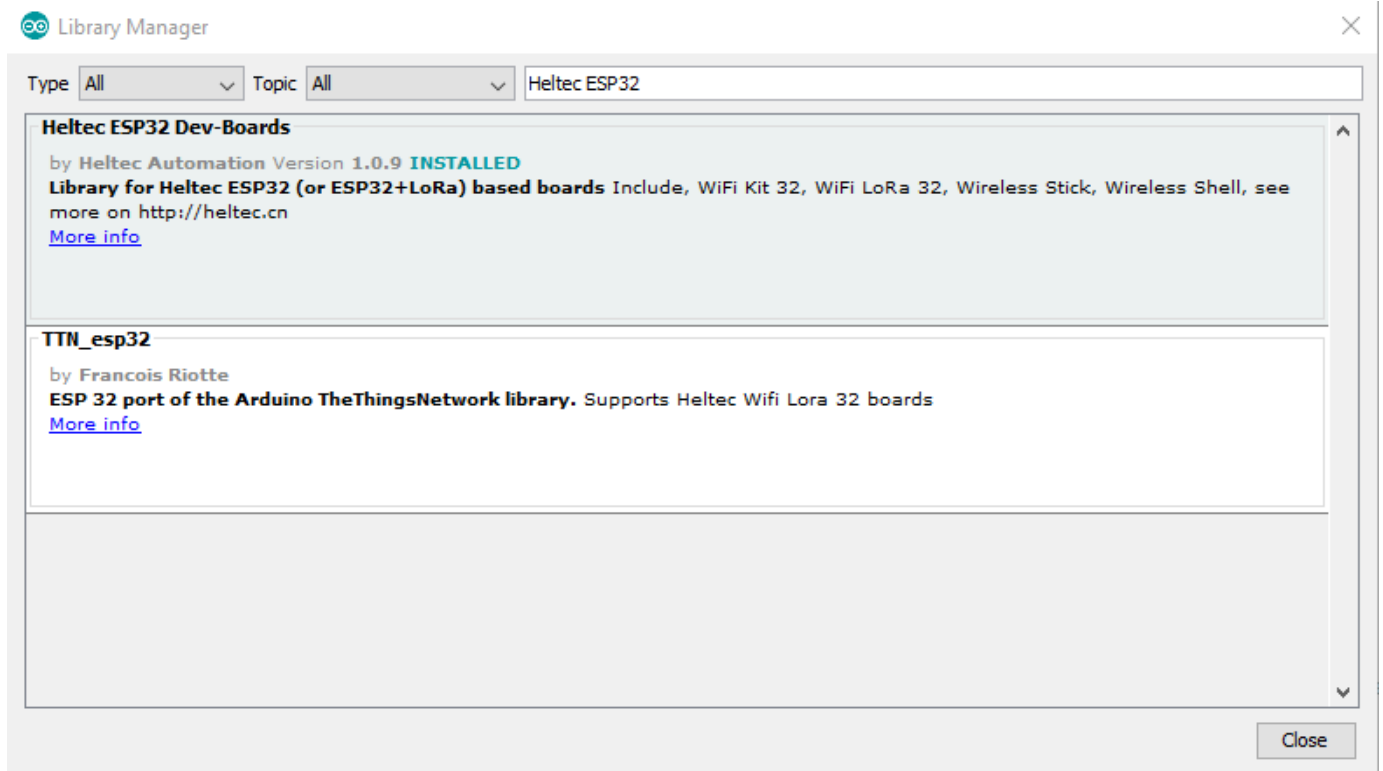
## Arduino Library

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](https://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 endianness and formatting required for the sketch, shown below.



Device Details

Name

Heltec LoRa V2

UUID

6afe172f-915d-4f3c-a8c1-98810cddca44

Device EUI

msb

0xFB, 0xE2, 0xD9, 0x74, 0x80, 0xC4, 0xD0, 0x35

App EUI

msb

0x09, 0xF4, 0x56, 0x88, 0x7B, 0x85, 0xC1, 0xDC

App Key

msb

0x50, 0x13, 0xC6, 0xB2, 0x78, 0xEB, 0x4A, 0xAD, 0x6C, 0xD7, 0x57, 0x1A, 0x18, 0xB8, 0x8F, 0xC8

Activation Method

OTAA

LoRaWAN US Channels

48-55 (sub-band 7)

Attached Labels

+ Add Label

Associated Integrations

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](http://resource.heltec.cn/search)

```

/*license for Heltec ESP32 LoRaWan, quarry 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.





```

Sketch uses 281893 bytes (8%) of program storage space. Maximum is 3342336 bytes.
Global variables use 18180 bytes (5%) of dynamic memory, leaving 309500 bytes for local variables. Maximum is 327680 bytes.
esptool.py v2.6
Serial port COM15
Connecting.....
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
MAC: 24:6f:28:77:83:28
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 8MB
Compressed 8192 bytes to 47...

Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 636.3 kbit/s)...
Hash of data verified.
Flash params set to 0x023f
Compressed 17392 bytes to 11185...

Writing at 0x00001000... (100 %)
Wrote 17392 bytes (11185 compressed) at 0x00001000 in 0.2 seconds (effective 597.2 kbit/s)...
Hash of data verified.
Compressed 284416 bytes to 153439...

Writing at 0x00010000... (10 %)
Writing at 0x00014000... (20 %)
Writing at 0x00018000... (30 %)
Writing at 0x0001c000... (40 %)
Writing at 0x00020000... (50 %)
Writing at 0x00024000... (60 %)
Writing at 0x00028000... (70 %)
Writing at 0x0002c000... (80 %)
Writing at 0x00030000... (90 %)
Writing at 0x00034000... (100 %)
Wrote 284416 bytes (153439 compressed) at 0x00010000 in 2.9 seconds (effective 777.6 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 129...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (129 compressed) at 0x00008000 in 0.1 seconds (effective 227.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

)

## 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.



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

[illegible]

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.

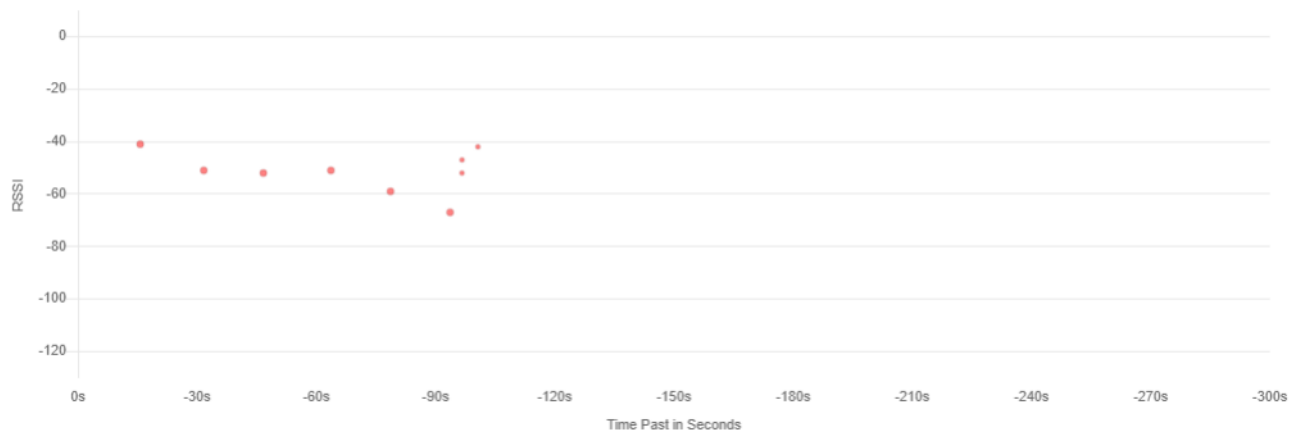
```
uint16_t userChannelsMask[6]={
0xFF00,0x0000,0x0000,0x0000,0x0002,0x0000 };
```

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

## Device Integrations

Debug

● Live Data



### Event Log

		FCnt	Port	DevAddr	Time
+	uplink	5	2	84FC2064	May 5, 2020 3:40 PM
+	uplink	4	2	84FC2064	May 5, 2020 3:39 PM
+	uplink	3	2	84FC2064	May 5, 2020 3:39 PM
+	uplink	2	2	84FC2064	May 5, 2020 3:39 PM
+	uplink	1	2	84FC2064	May 5, 2020 3:39 PM
+	uplink	0	2	84FC2064	May 5, 2020 3:38 PM
+	downlink	2	2	84FC2064	May 5, 2020 3:38 PM
+	downlink	1	2	84FC2064	May 5, 2020 3:38 PM
+	activation	0	0		May 5, 2020 3:38 PM

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](#)