

LORA + MESHTASTIC: ENCRYPTED MESSAGING FROM YOUR PHONE WITHOUT THE INTERNET

JUNE 2024

IFFY BOOKS + MIKE DANK

You can find this zine w/ future updates on our site:

<https://iffybooks.net/meshtastic-zine-2024>

The Meshastic project, founded in early 2020, uses inexpensive LoRa radios to create an off-grid mesh network for communication. If you've ever been hiking or camping in the wilderness away from cell towers, or at locations like music festivals where infrastructure is often congested from a high volume of use, you may understand the need for a platform that doesn't rely on traditional communication services.

Meshtastic is completely off-grid, and doesn't rely on any existing networks or infrastructure. It does so using LoRa (short for **L**ong **R**ange) radios that are advertised as being able to transmit data at a typical range of 10km (~6 miles) in areas with a clear line of sight, and some have reported communicating over longer distances. In a city environment with many obstructions, your range is more likely to be 1-3 blocks.

LoRa implements chirp spread spectrum (CSS) modulation, meaning that it is robust to channel noise and resistant to multi-path fading, resulting in low interference.

It is important to note that LoRa is not designed for high-speed data transmission, and the hardware is only capable of transmitting between 250 bits/s and 27 kbits/s. At this speed, LoRa is ideal for text communication via realtime chat as well as location beaconing, though it has also been adopted for use in IoT and sensor networks.

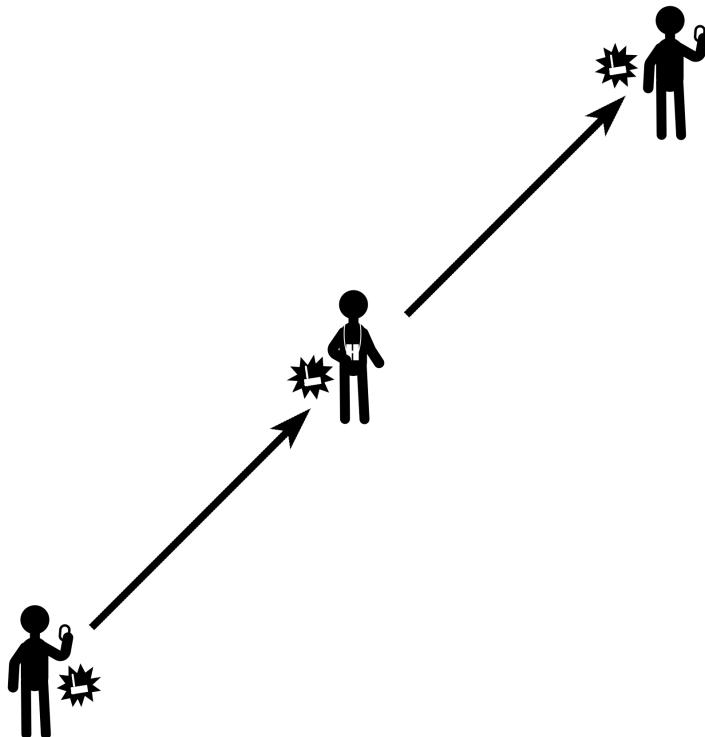
Meshtastic uses LoRa technology to provide a communication platform ideal for small groups of people who wish to have private conversations. This can be ideal for a temporary setup where there is no traditional infrastructure to use, or a permanent setup with fixed and mobile nodes operating in a particular neighborhood.

BUILDING A MESH

Each node in a Meshastic network is composed of a LoRa radio device, typically paired via Bluetooth to a mobile phone. Once paired, the host device can run the Meshtastic application to send and receive messages on the network.

Each node in the Meshtastic network additionally acts as a relay for other nodes. For example, if Alice and Bob want to send messages to one another, but are physically too far apart to make a connection, their data can automatically be routed through a third user, Carlos, who is physically between and in range of both Alice and Bob. This means

that even the most distant nodes on the network have the ability to communicate with all other nodes, even if they are not nearby. As long as Alice and Bob are using a private encryption key, Carlos won't be able to read their messages.



Nodes can also be set up as standalone repeaters that simply route traffic through the network without being connected to a host device. This allows nodes to be operated in an isolated or otherwise fixed position, supplying connection to the greater network for others nearby. This can be an ideal configuration for solar-powered nodes that are elevated for line-of-sight connection, requiring little to no maintenance from the operator.

PRIVACY

By default, a new Meshtastic node is configured to use a general chat channel that everyone has access to. That

makes it easy to get started, but it means your messages can be read by any other Meshtastic user.

To keep your messages secure, you can create a new channel using a custom AES-256 encryption key. As long as you and your friends keep the key private (which means not posting it online), no one else will be able to decipher your chat.

If you want, Meshtastic can share your current GPS coordinates with everyone else on the channel you're using. Sharing your location can be helpful if you're in a remote area or navigating a large crowd. If you're using Meshtastic from home, we recommend turning off GPS sharing.

SUPPLIES

Setting up a Meshtastic node is relatively inexpensive, assuming you already have a computer you can use to interact with the radio. You may want to get materials for two nodes so you can test an end-to-end connection.

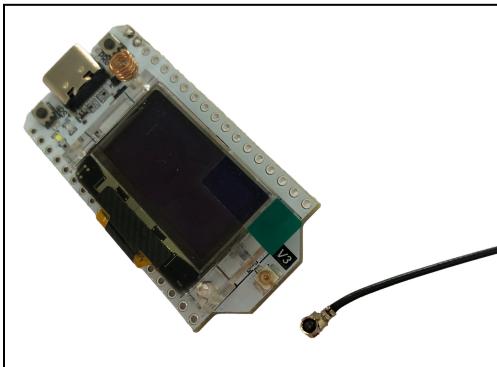
Currently, there are Meshtastic applications for both Android and iOS, though the iOS application is in beta. Alternatively, there is a Meshtastic-python library you can use to interface with a LoRa radio via PC, but you will likely need some prior Python language experience to make full use of it. Further, there is a serial interface and web interfaces to access Meshtastic from a browser, though the latter may currently be unstable. There is no current turnkey application for PC use.

When you purchase your board, make sure to get one that operates in a frequency you are allowed to transmit on in the country you are operating in. In the United States we're using 915MHz.

HELTEC LORA32 SETUP

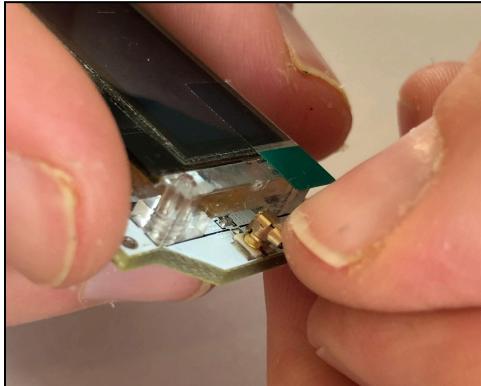
Your LoRa radio comes with an antenna, which you'll want to connect before going further. If you power on your LoRa

radio without an antenna attached, you may damage the circuitry.



If you're using a Heltec LoRa32, your antenna uses an IPEX-1 connector. The port you'll connect it to is on the front of the device, to the right of the LCD screen.

- Line up your antenna's IPEX-1 connector with the port side on your LoRa32 and press it on with your thumb. It may take more pressure than you expect.



INSTALL PYTHON AND PIP

Before you flash the Meshtastic firmware to your LoRa device you'll need **Python 3** and the **pip** package manager. To check whether you have them, first open a terminal window. (If you're using macOS, open the Terminal app. In Windows you can use PowerShell.)

- ❑ Type the following command and press enter:

```
python3 -m pip -V
```

If Python 3 and pip are installed, you'll see a version number for both. If you get an error message, you'll need to install them.

Install Python 3 with pip on Debian-based Linux

- ❑ Open a terminal window and run the following commands to install **Python 3** and the **pip** package manager:

```
sudo apt-get update
sudo apt-get install python3
sudo apt-get install python3-pip
python3 -m pip install -U pip
```

- ↳ If Python and pip are already installed, you can run the following command to update them:

```
sudo apt-get update
sudo apt-get upgrade
python3 -m pip install --upgrade pip
```

Install Python 3 with pip on macOS

- ❑ Go to this page and follow the instructions to install **Python 3** on macOS:

<https://docs.python-guide.org/starting/install3/osx>

- ❑ Once Python is installed, run this command to upgrade the **pip** package manager:

```
python3 -m pip install --upgrade pip
```

Install Python 3 with pip on Windows

- ❑ You may want to install **Git Bash**, which will let you run Linux-like commands instead of using PowerShell. (This is optional.) Here's the download page:

<https://git-scm.com/downloads>

- ❑ Go to the following URL and follow the instructions to install **Python 3** on Windows:

<https://realpython.com/installing-python/#how-to-install-python-on-windows>

- ❑ Once Python is installed, run this command to upgrade the **pip** package manager:

```
python3 -m pip install --upgrade pip
```

INSTALL USB SERIAL DRIVER (WINDOWS ONLY)

- ❑ Go to the following URL:

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

- ❑ Click **CP210x Windows Drivers** to download a **.zip** file containing the driver.

- ❑ Extract the **.zip** file you just downloaded and install the driver.

INSTALL HELTEC LORA32 DRIVER (PROBABLY OK TO SKIP THIS)

- ❑ Go to the following URL and download the Arduino IDE:

<https://arduino.cc/en/software>

- ❑ Follow the following page and follow the instructions under **Use Arduino Library Manager**:

https://github.com/HelTecAutomation/Heltec_ESP32

DOWNLOAD MESHTASTIC FIRMWARE

- ❑ Go to the following URL to find the latest Meshtastic firmware releases:

<https://github.com/meshtastic/firmware/releases>

- ❑ Scroll down until you find the latest release with **Beta** in the title. In the image below, it's **Meshtastic Firmware 2.1.5.23272da Beta**. Click on the title of the release.

The screenshot shows a GitHub release page for the Meshtastic Firmware 2.1.5.23272da Beta. The page has a sidebar on the left showing commit history and a main content area on the right. The main content includes the release title, a 'Latest' button, sections for 'Enhancements' and 'Bug fixes', and a 'Full Changelog' link. At the bottom, there is a 'Contributors' section with two profile icons and their names.

Jun 3

github-actions

v2.1.15.cd787...
cd78723

Compare

Meshtastic Firmware 2.1.15.cd78723 Beta

Latest

Enhancements

- Wio-e5 support progress by @thebentern in #2265
- Open up Serial Module to T-Echo by @thebentern in #2533
- Enable range test module (sending only) on NRF by @thebentern in #2534
- Use INA for device battery level by @thebentern in #2536

Bug fixes

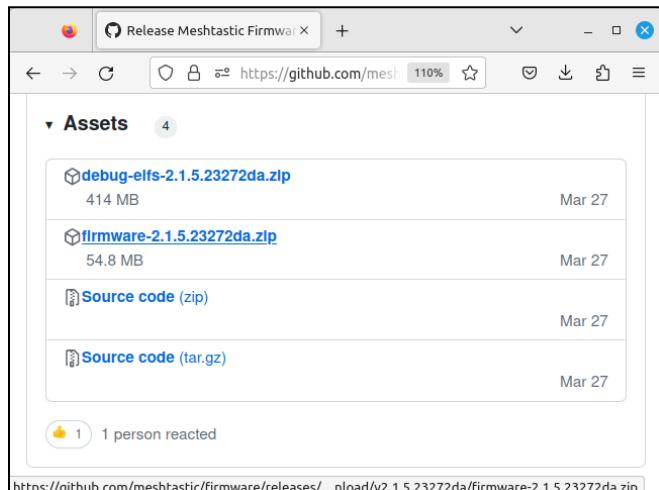
- Cancel rebroadcast in Tx queue upon receiving another rebroadcast by @GUVWAF in #2538

Full Changelog: [v2.1.14.99a31c1...v2.1.15.cd78723](#)

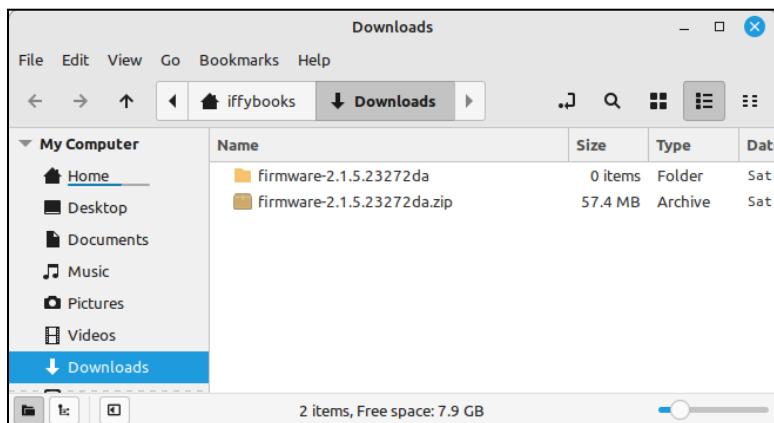
Contributors

thebentern and GUVWAF

- Scroll down and look for a link to a **.zip** file with "firmware" at the beginning of the filename (**firmware-2.1.5.23272da.zip** in the example below). Click the filename to download the firmware release.

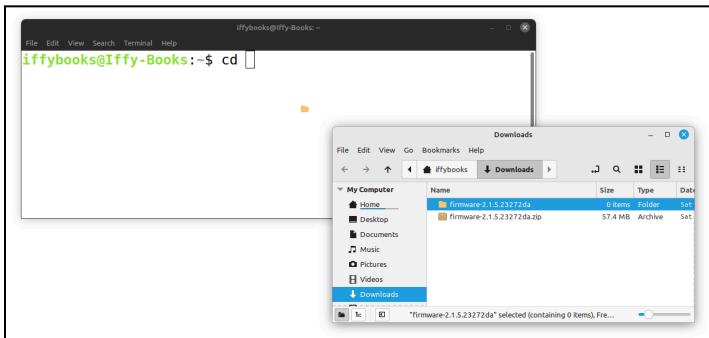


- Find the **.zip** file you just downloaded. Right-click it and extract to the current directory.

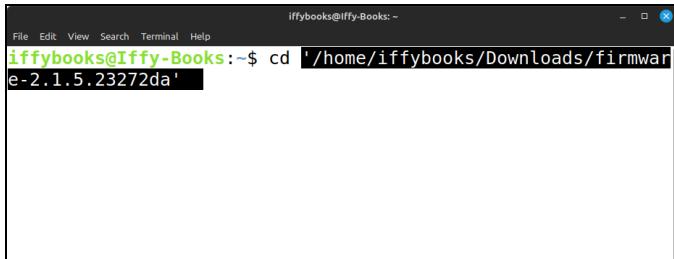


INSTALL ESPTOOL

- ❑ Open a terminal window and type **cd** followed by a space. Click on the firmware directory you just created and drag it onto the terminal window.



- ❑ The directory's full pathname will appear in the terminal window. Press **enter** to change your working directory to the new firmware directory.



- ❑ Type **ls** and press **enter** to see a list of files in the current directory.
- ❑ Run the following command to install the Python module **virtualenv**.

```
python3 -m pip install -U virtualenv
```

- ❑ Now run the command below to set up a Python virtual environment in the current directory.

```
python3 -m venv ./
```

- ❑ Run this command to activate the virtual environment.

```
source ./bin/activate
```

- ❑ Now that the virtual environment is active, run the following command to install the **esptool** module:

```
python3 -m pip install esptool
```

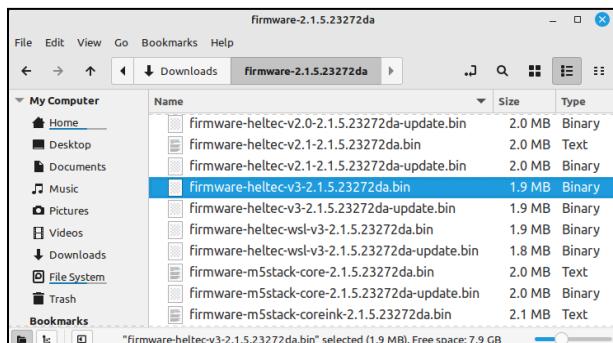
FLASH MESHTASTIC TO YOUR LORA32 DEVICE

- ❑ Type the following command in the terminal window, followed by a space. (Windows PowerShell users should type **./device-install.bat** instead.) Don't press enter yet.

```
./device-install.sh -f
```



- ❑ Open the firmware directory in your File Explorer/Finder and find the file that starts with **firmware-heltec-v3** and ends with **.bin**. The filename below is **firmware-heltec-v3-2.1.5.23272da.bin**.



- ❑ Drag the file into your terminal window to add the path to your command.

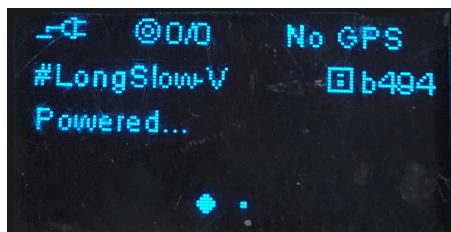


```
iffybooks@Iffy-Books: ~/Downloads/firmware-2.1.5.23272da$ ./device-install.sh -f '/home/iffybooks/Downloads/firmware-2.1.5.23272da/firmware-heltec-v3-2.1.5.23272da.bin'
```

Here's the full command in the example above:

```
./device-install.sh -f  
'/home/iffybooks/Downloads/firmware-2.1.5.23272da/firmware-heltec-v3-2.1.5.23272da.bin'
```

- ❑ Press **enter** to run the command and start flashing your LoRa radio. It may take several minutes to finish.
 - ↳ If you're using Linux, you may need to add **sudo** before the command above.
 - ↳ If something goes wrong, refer to the "Flashing with the CLI" page on the Meshtastic website:
<https://meshtastic.org/docs/getting-started/flashing-firmware/esp32/cli-script>
- ❑ Connect a USB-C cable to your LoRa radio and plug the other end into a USB power source, such as a computer. You'll see the Meshtastic logo on the LCD screen, followed by a screen that looks like this:



At the top right of the screen you'll see a 4-digit ID code that identifies your device. In this case the device's ID is **b494**.

INSTALL THE MESHTASTIC APP ON YOUR PHONE

iOS: Search for "Meshtastic" in the App Store and install it.

Android: Download the .apk from GitHub and sideload it:

<https://github.com/meshtastic/Meshtastic-Android/>

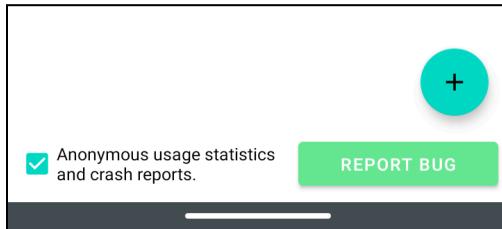
If you use the F-droid app store, follow the instructions on this page:

<https://meshtastic.org/docs/software/android/installation>

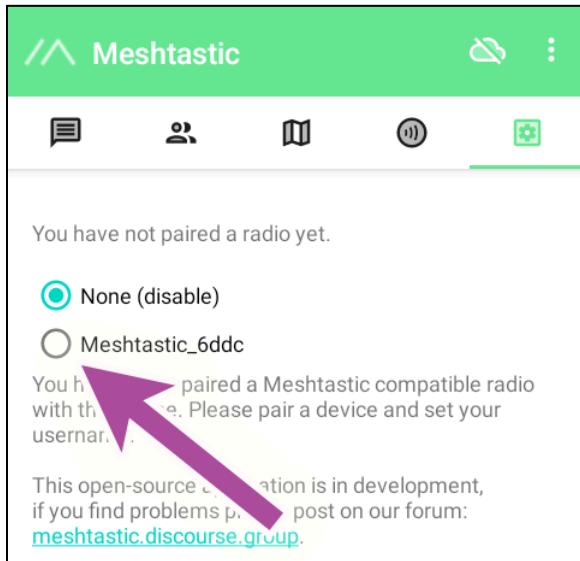
Or go to the Google Play Store and search for "Meshtastic."

PAIR WITH YOUR LORA RADIO (ANDROID)

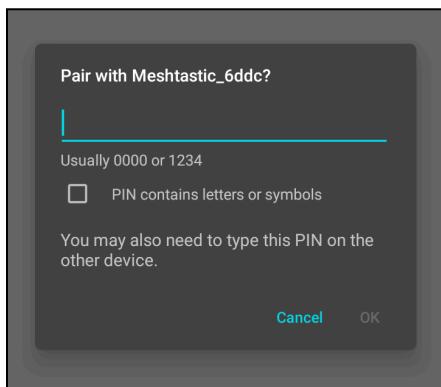
□ Open the Meshtastic app and tap the **plus sign** (+) in the bottom right corner to start scanning for LoRa devices.



- ❑ Find your device using its four-character ID. Tap the radio button next to it to start pairing.

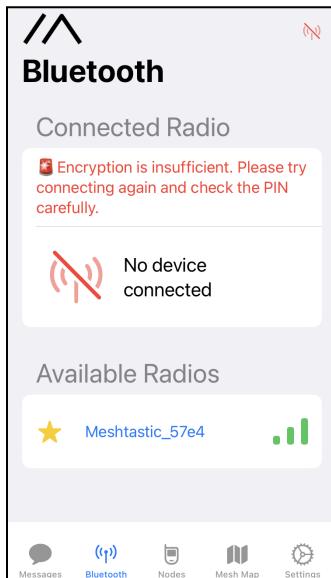


- ❑ Your LoRa device will display a six-digit code on its screen. Enter the code in the app, then tap **OK**.

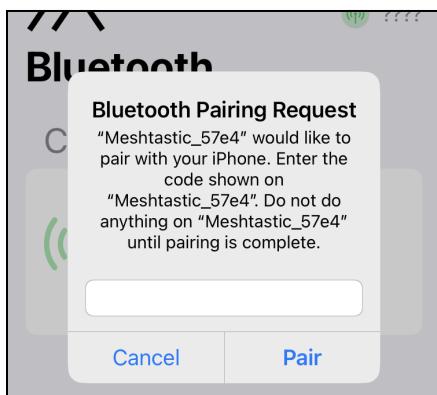


PAIR WITH YOUR LORA RADIO (iOS)

- Tap the **Bluetooth** icon at the bottom of the screen. Find your LoRa device under **Available Radios** and tap it.

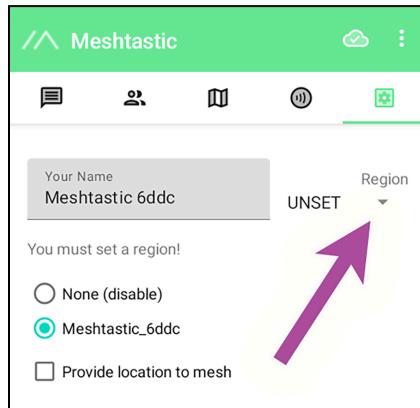


- Your LoRa device will display a six-digit code on its screen. Enter the code in the app, then tap **Pair**.

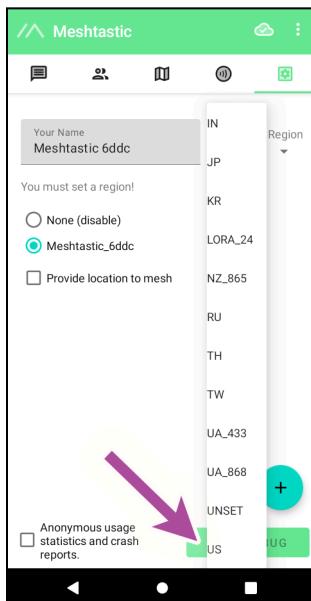


SET YOUR REGION (ANDROID)

- ❑ Tap **Region** at the top right of your screen. (It's in tiny gray type with a triangle underneath.)



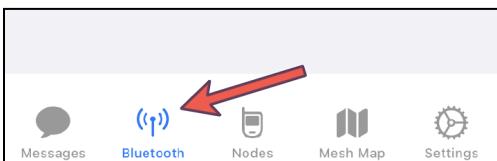
- ❑ Select your region from the dropdown menu.



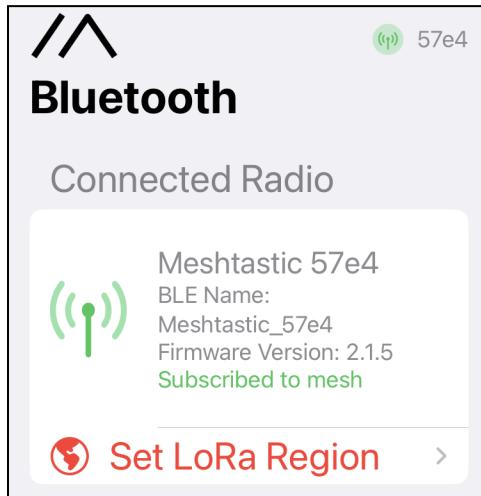
- ❑ Tap **Save**, which will reboot your LoRa device.

SET YOUR REGION (iOS)

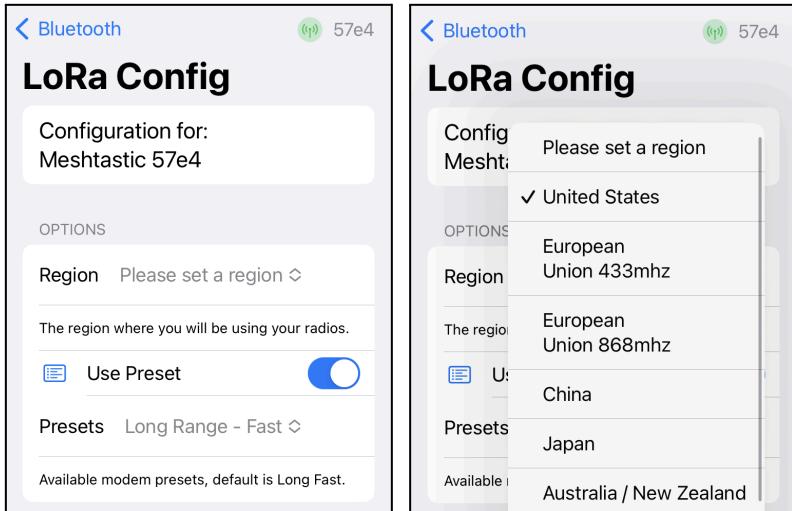
- Tap the **Bluetooth** icon.



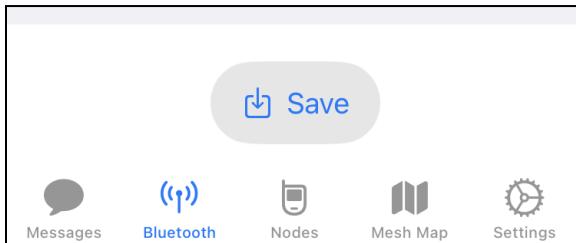
- Tap Set LoRa Region.



- ☐ Tap **Please set a region** and select your region from the dropdown menu.

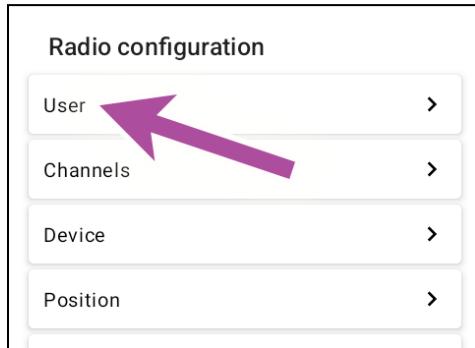


- ☐ Tap **Save**, which will reboot your LoRa device.

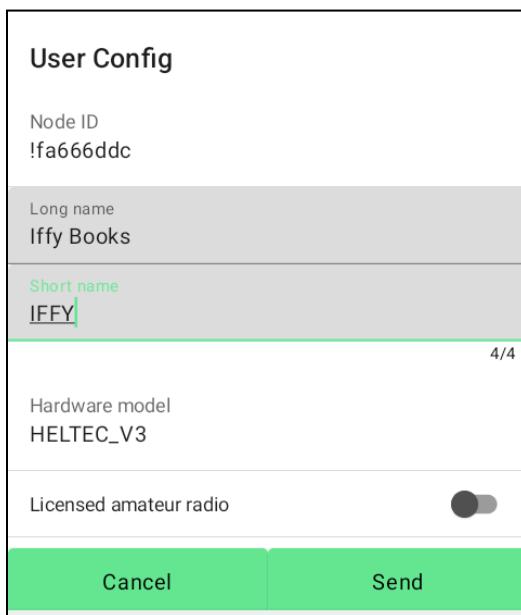


NAME YOUR LORA RADIO (ANDROID)

- Tap the three dots at the top right corner of the screen and select **Radio configuration**. Then tap **User**.



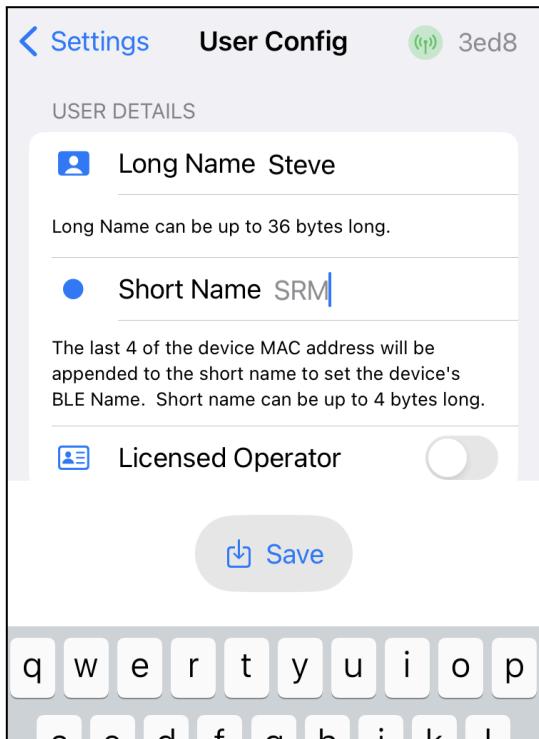
- Enter a short name (<=4 characters) and long name for your LoRa radio.



- Tap **Send** to save and reboot.

NAME YOUR LORA RADIO (iOS)

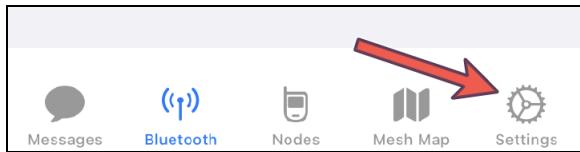
- ❑ Go to **Settings** in the Meshtastic app, then tap **User**.
- ❑ Enter a short name for your device (<= 4 characters). You have to change the short name first, for unknown reasons.
- ❑ Tap **Save**, which will reboot your device.
- ❑ Go to the **Settings**, then tap **User**. Enter a long name for your device.
- ❑ Tap **Save**, which will reboot your device.



UPDATE LORA RADIO SETTINGS

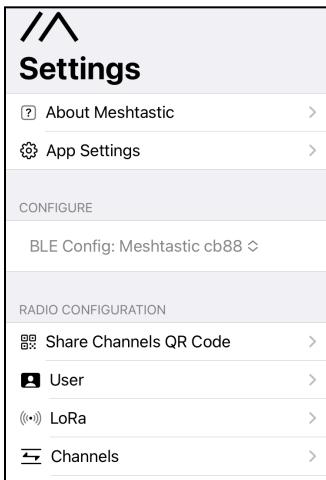
The screen captures below are from the iOS app, but the steps are the same for Android.

- ❑ In the iOS Meshtastic app, open **Settings**. In the Android app, tap the three dots at the top right corner of the screen and select **Radio configuration**.



- ❑ *iOS only:* Under **Configure** in the **Settings** menu, make sure your LoRa device is selected. If it isn't, select it from the dropdown menu.

- ❑ Tap **LoRa** to open the LoRa radio settings.

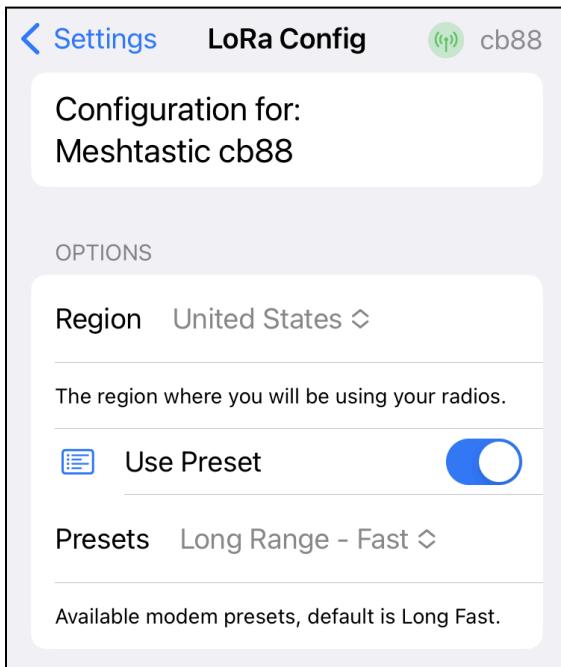


Meshtastic has eight built-in presets for the way it transmits data, which are called **modem presets**. The modem preset you choose will determine the maximum distance you'll be

able to transmit and receive, as well as how long it takes to send each message.

It's important to note that you and your friends will need to use the same modem preset. If you're using different presets, you won't be able to see each other's messages at all.

❑ The default modem preset is called **Long Range / Fast**. Tap the dropdown menu next to **Presets** to see the other options. For now, leave **Long Range / Fast** selected.

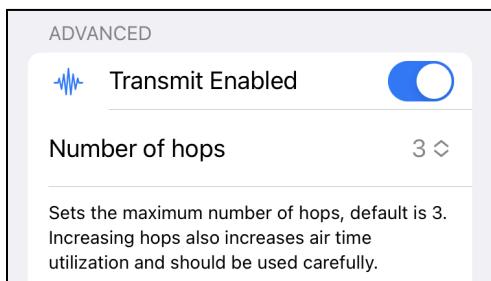


Here's a table with details on all eight of Meshtastic's modem presets:

Channel setting	Alt Channel Name	Data-Rate	SF / Symbols	Coding Rate	Bandwidth	Link Budget
Short Range / Fast	Short Fast	6.8 kbps	7 / 128	4/8	250	137dB
Short Range / Slow	Short Slow	3.9 kbps	8 / 256	4/8	250	140dB
Medium Range / Fast	Medium Fast	2.2 kbps	9 / 512	4/8	250	143dB
Medium Range / Slow	Medium Slow	1.2 kbps	10 / 1024	4/8	250	146dB
Long Range / Fast	Long Fast	0.67 kbps (default)	11 / 2048	4/8	250	148.5dB
Long Range / Moderate	Long Moderate	0.335 kbps	11 / 2048	4/8	125	151dB
Long Range / Slow	Long Slow	0.18 kbps	12 / 4096	4/8	125	154dB
Very Long Range / Slow	Very Long Slow	0.09 kbps	12 / 4096	4/8	62.5	157dB

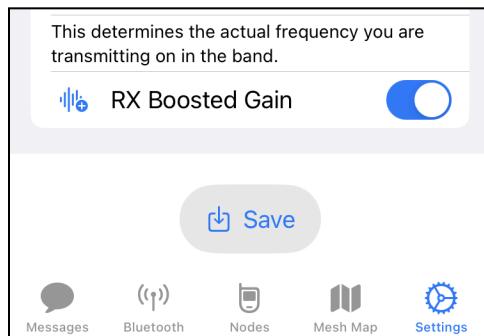
Source (with details on terms & abbreviations in the table above): <https://meshtastic.org/docs/overview/radio-settings>

- ❑ Scroll down in the LoRa settings and you'll see an option called **Number of hops**. By default, each message you send will make three hops across the mesh network before nodes stop retransmitting it. Increasing the number of hops risks overloading the network, so you should leave it on **3** unless you have a good reason to change it.



- ❑ Keep scrolling down and you'll see the **RX Boosted Gain** option, which makes your LoRa radio more sensitive to weak signals. If you're near other LoRa devices, leave it unchecked for now.

If you're trying to connect to a Meshtastic network from a distance (especially in a dense urban area), you can turn this option **on**. Then tap **Save**, which will reboot your device.



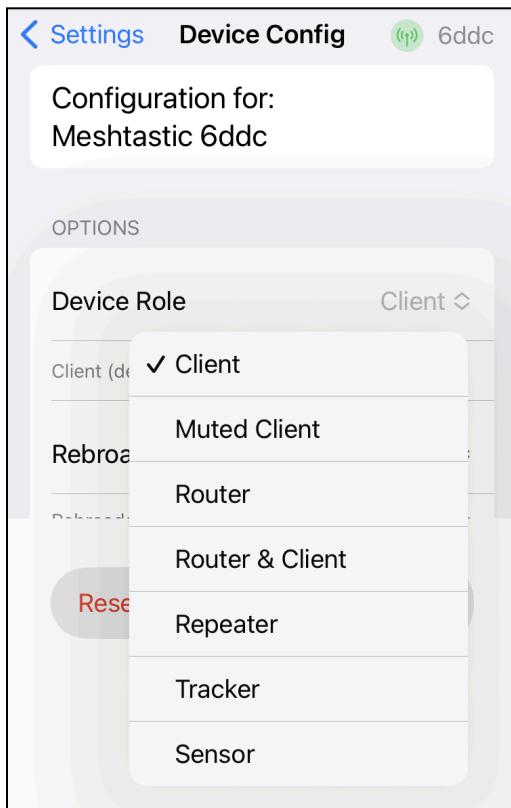
SET YOUR DEVICE ROLE

The screen captures below are from iOS, but the steps are the same for Android.

- ❑ In the iOS Meshtastic app, go to **Settings**. In the Android app, tap the three dots in the top right corner and go to **Radio Configuration**.
- ❑ Tap **Device**.

Meshtastic has seven device configuration presets (a.k.a. device roles), which you can change depending on how you plan to use your Meshtastic node.

Tap the drop down menu next to **Device Role** to see the options. The default device role is called **Client**.



Here's a table with details on each device role:

Device Configuration Preset (a.k.a. Device Role)	Description
CLIENT	Client (default) - This role will follow the standard routing rules while also allowing the device to interact with client applications via BLE/Wi-Fi (Android/Apple/Web).
CLIENT_MUTE	Client Mute - Same as a client except packets will not hop over this node, does not contribute to routing packets for mesh.
ROUTER	Router - Mesh packets will prefer to be routed over this node. The assumption is that Router-type devices will be placed in locations with a height/range/antenna advantage, and therefore have better overall coverage. This node will not be used by client apps. The BLE/Wi-Fi radios and the OLED screen will be put to sleep. Please note: Due to the preferred routing, this role may cause higher power usage due to more frequent transmission.
ROUTER_CLIENT	Router Client - Hybrid of the Client and Router roles. Similar to Router, except the Router Client can be used as both a Router and an app connected Client. BLE/Wi-Fi and OLED screen will not be put to sleep.
REPEATER	Repeater - Mesh packets will prefer to be routed over this node. This role eliminates unnecessary overhead such as NodeInfo, DeviceTelemetry, and any other mesh packet, resulting in the device not appearing as part of the network. As such, direct messaging this node is not available, as it will not appear in your nodes list which results in a cleaner mesh network. Channel and modem settings of the mesh packets being repeated must be identical to the repeater's configuration. Please see

	Rebroadcast Mode for additional settings specific to this role.
TRACKER	Tracker - For use with devices intended as a GPS tracker. Position packets sent from this device will be higher priority. Smart Position Broadcast will default to the currently configured settings.
SENSOR	Sensor - For use with devices intended to primarily collect sensor readings. Telemetry packets sent from this device will be higher priority, broadcasting every five minutes.

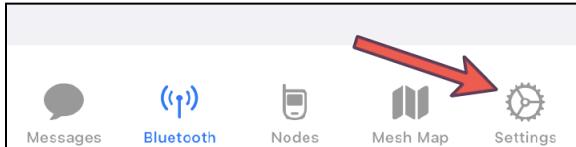
Source: <https://meshtastic.org/docs/settings/config/device>

- If you're planning to put your LoRa radio in a high place to support the Meshtastic network, select **Router & Client**. Otherwise, leave **Client** selected.
- If you've changed your device role, tap **Save**. Your device will reboot.

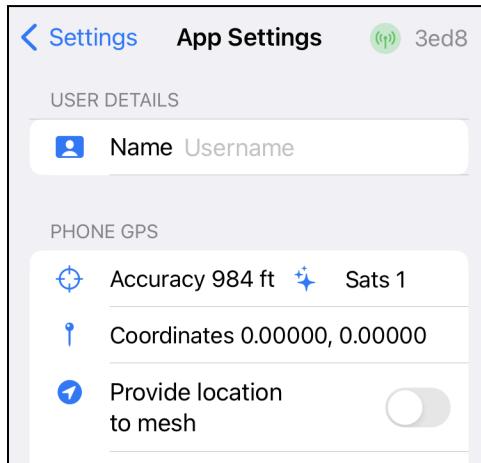
MAKE SURE GPS LOCATION SHARING IS TURNED OFF (iOS)

The screen captures below are from iOS, but the steps are the same for Android.

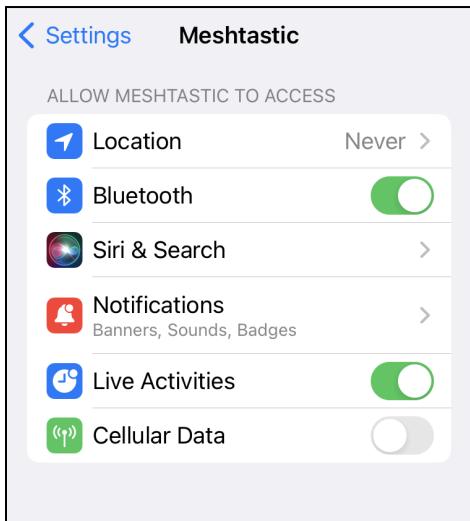
- ❑ Open **Settings**.



- ❑ Tap **App Settings**. If the option **Provide location to mesh** is selected, deselect it.



- ❑ *iOS only:* Go to **Settings** in iOS, scroll down, and tap **Meshtastic**. Then make sure Location sharing is set to **Never**.

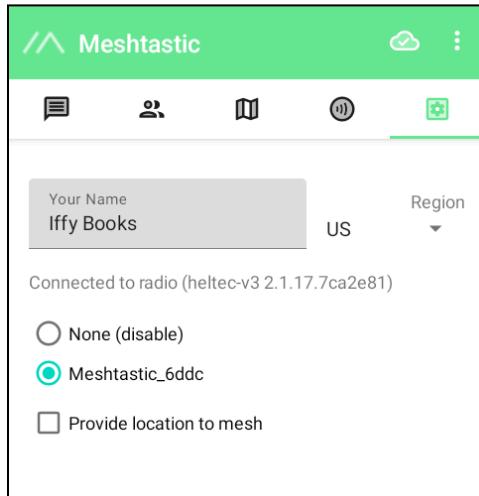


While you're here, tap on **Siri & Search**. To avoid leaking data needlessly, deselect **Learn from this App** and **Show Content in Search**.



MAKE SURE GPS LOCATION SHARING IS TURNED OFF (ANDROID)

- ❑ Tap the **Settings** icon in the top right corner. If "Provide location to mesh" is checked, uncheck it.



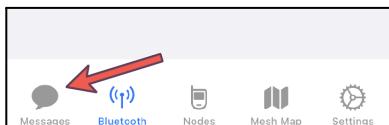
READ MORE ABOUT DEVICE CONFIG OPTIONS

- ❑ Go to the following URL to read more about the config options for Meshtastic:

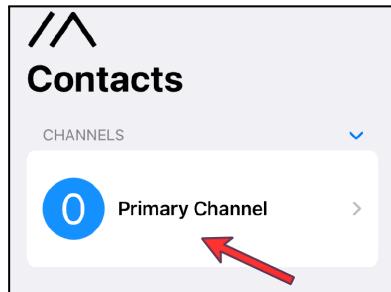
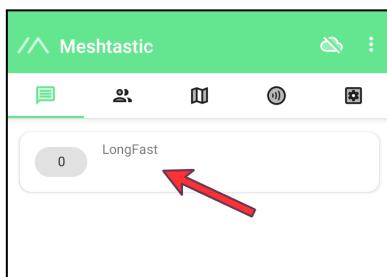
<https://meshtastic.org/docs/settings/config>

SEND A MESSAGE TO THE GROUP CHAT

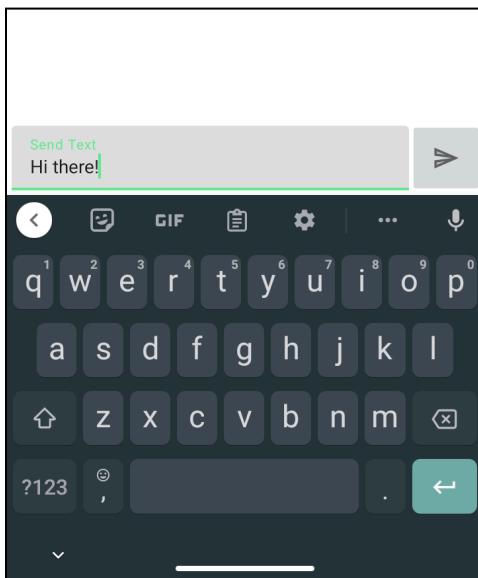
- ❑ Tap the **Messages** icon.



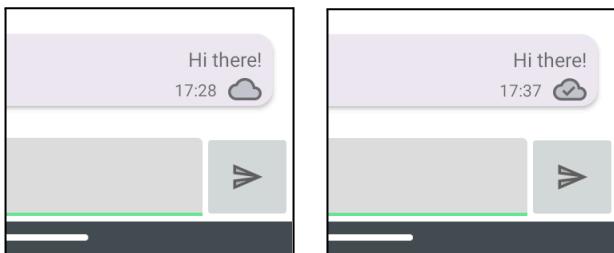
- ❑ Then tap the icon for **channel 0**, which will take you to the primary channel (a.k.a. the group chat).



- ❑ Type a message and tap the **Send** button.



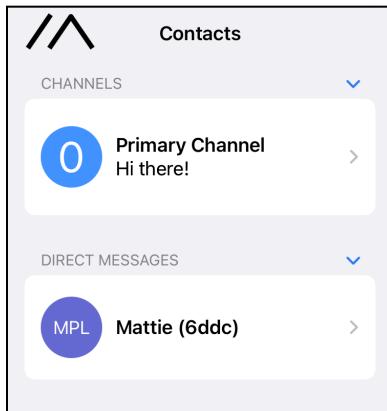
If your message is received and acknowledged, on Android you'll see a check mark appear in the cloud icon under your message. In the iOS app you'll see the word "Acknowledged" appear under the message.



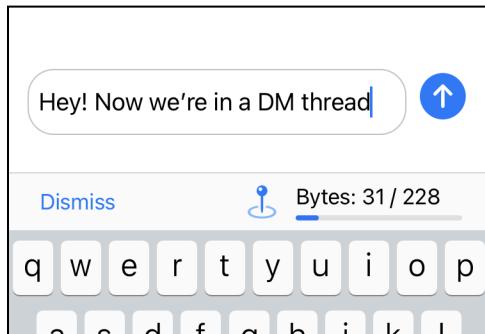
SEND A DIRECT MESSAGE

Once you've connected with someone in the primary channel, you can send a direct message (DM).

- ❑ Tap the **Messages** icon. Under **Direct Messages**, tap the device you want to DM.



- Type a message and tap **Send**.



You're texting without the internet or cell network! Pretty neat.

It's important to note that Meshtastic DMs are encrypted using the same shared key as the channel marked **PRIMARY**. That means anyone else on the same channel can (in theory) read your direct messages.

If you want to chat privately with a friend, create a new shared encryption key, set the channel to PRIMARY, and don't reveal it to anyone else. (Details in the next section.)

CREATE A NEW PRIVATE KEY

When you first set up Meshtastic, your messages are encrypted using a weak default key, which is widely known. Many Meshtastic users stick with the default key because they aren't concerned with privacy. To chat privately, you'll need to generate a new encryption key and share it with the person/people you're planning to chat with.

We recommend against creating and updating encryption keys using the iOS and Android apps. We've gotten unexpected results during testing, and it isn't worth the hassle.

Instead, we'll generate an encryption key using the command line. Then we'll use the Meshtastic Web Client to update the settings on your LoRa device.

❑ Meshtastic uses AES-256 encryption, so you'll need a 256-bit key in Base64 format. To generate one, open a terminal window and run the following command. (To type the pipe character, hold shift and press backslash.)

```
cat /dev/urandom | head -c 32 | base64
```

The command above generates 32 bytes of random characters (equivalent to 256 bits), then converts it to the Base64 encoding format. The output will look something like this:

dqP9LPdBkE/bF5lsYCwsXfnNavx3IkI5LsUoCu91F/4=

❑ Highlight your new key in the terminal window, then right-click and select **Copy**. Now you can share the key with your friends!

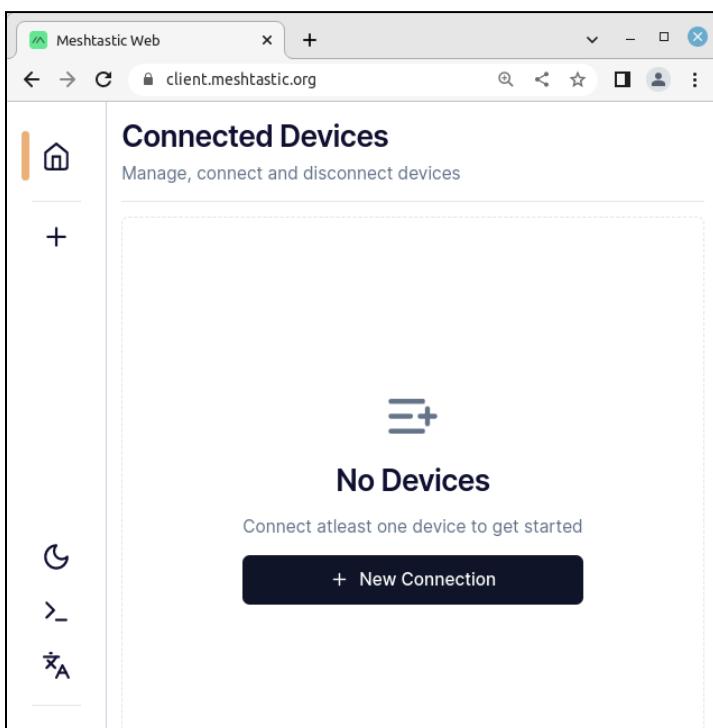
Note: When distributing your shared encryption key with friends, it's best to use an encrypted channel such as Signal. Or post the key on CryptPad.fr and share the link via Signal.

LAUNCH THE MESHTASTIC WEB CLIENT

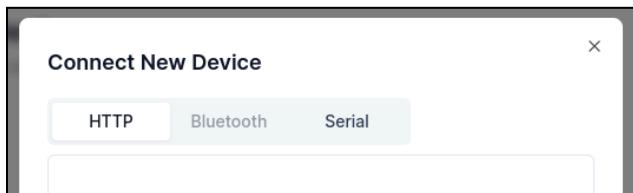
The quickest way to update your Meshtastic settings from a computer is to use the Meshtastic Web Client. Unfortunately you'll need to use a Chromium-based browser, such as Google Chrome. If you don't already have it, you can download it here:

<https://www.google.com/chrome>

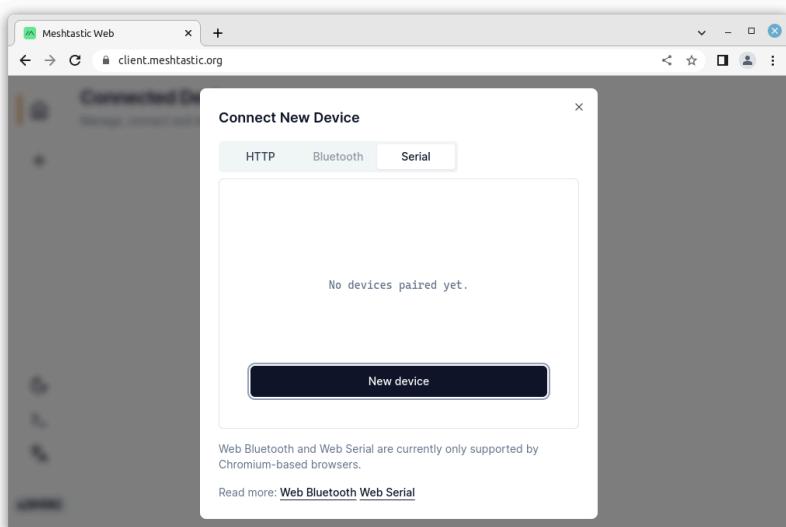
- ❑ Connect your LoRa device to your computer with a USB-C cable that supports data.
- ❑ Open Chrome and go to the following URL:
<https://client.meshtastic.org>
- ❑ Click **New Connection**.



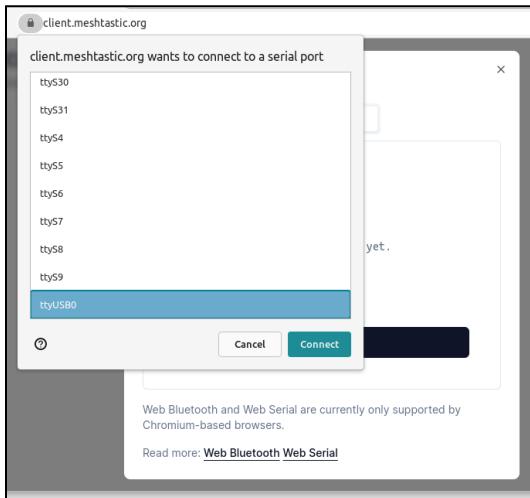
- ❑ At the top of the **Connect New Device** window, click the **Serial** tab.



- ❑ Click the **New device** button.

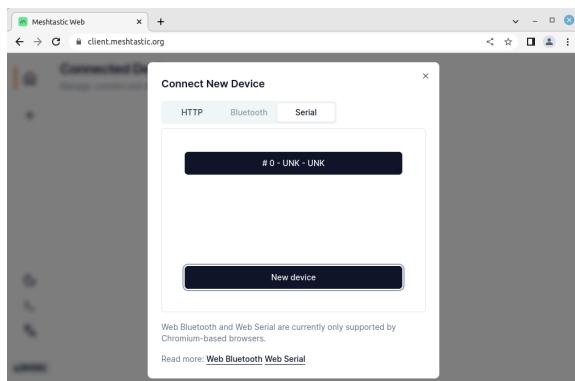


- ❑ You'll see a prompt that reads "client.meshtastic.org wants to connect to a serial port." Scroll through the list and look for the USB serial port that corresponds to your LoRa device. Click the port to highlight, then click **Connect**.



In the example above, the port is called **ttyUSB0**.

- ❑ You'll see a new rectangular button at the top of the window. Click on it to connect to your LoRa device.



- ❑ Click **X** to close the **Connect New Device** window.

UPDATE YOUR SHARED ENCRYPTION KEY

- ❑ In the **Navigation** column, click on **Channels**.

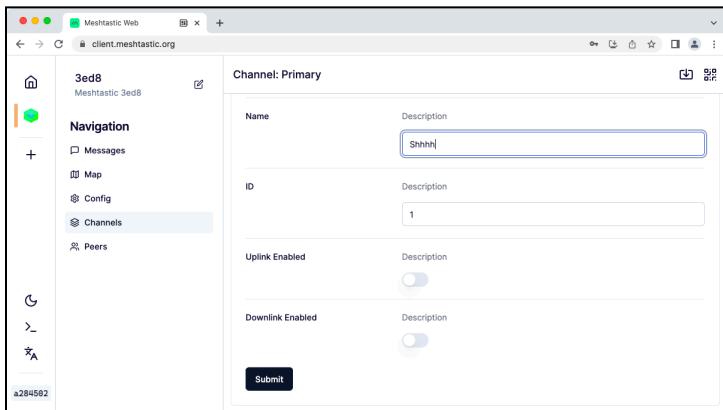
The screenshot shows the Meshastic Web interface with the URL `client.meshastic.org`. On the left, there's a sidebar titled "Navigation" with icons for Home, Messages, Map, Config, Channels (which is selected and highlighted in grey), and Peers. The main content area is titled "Channel: Primary". It has tabs for Primary, Ch 1, Ch 2, Ch 3, Ch 4, Ch 5, Ch 6, and Ch 7. Below the tabs, there's a section for "Channel Settings" with a note about "Crypto, MQTT & misc settings". A table follows, with "Role" set to "PRIMARY" and "Description" as "PRIMARY". Under "pre-Shared Key", the value is "....." and the "Description" is also ".....". At the bottom, there's a "Channel Number" field with "0" and a "Description" field.

Let's keep using the default key on the primary channel and add a shared key on channel 1.

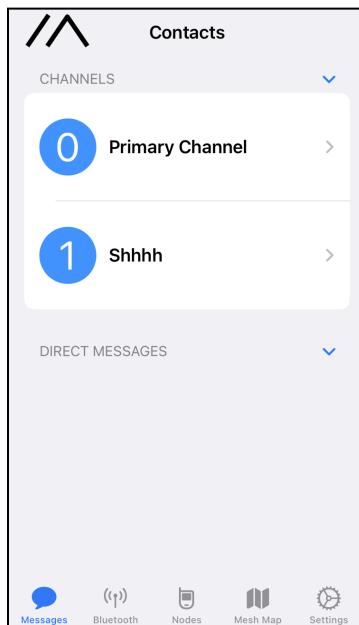
- ❑ Click the **Ch 1** tab to switch to channel 1. Highlight the text in the **pre-Shared Key** field and delete it, then paste in your new shared key.

This screenshot is similar to the previous one, but the "Ch 1" tab is now selected, indicated by a blue border around its tab. The "Role" dropdown is now set to "SECONDARY" and the "Description" is "SECONDARY". The "pre-Shared Key" field contains a long string of dots (...). The "Channel Number" field is still "0" and the "Description" field is empty.

- ❑ Scroll down and give your channel a name (optional). When you're done, click **Submit** and your device will reboot.



❑ Go to **Messages** and you'll see your new channel.



SET YOUR PRIVATE CHANNEL TO PRIMARY

If you want direct messages to be encrypted with your private key, you'll need to set your private channel to **PRIMARY**.

- ❑ Go to **Channels** in the Meshtastic web app.
- ❑ Set Channel 0 from **PRIMARY** to **SECONDARY**. Click **Submit** to save.
- ❑ Set Channel 1 from **SECONDARY** to **PRIMARY**. Click **Submit** to save.

INSTALL THE MESHTASTIC CLI

- ❑ Open a terminal window and run the following commands to install the Meshtastic CLI (command-line interface). The same commands should work for Linux, macOS, and Windows.

```
python3 -m pip install --upgrade pytap2
```

```
python3 -m pip install --upgrade meshtastic
```

- ❑ Run the following command to see if the CLI is installed properly. You should see a list of options.

```
meshtastic --help
```

↳ If the command above doesn't work, you'll need to add the meshtastic application file to your PATH environment variable. If you're using Linux, you can do that by modifying your **~/.profile** file. (On macOS it'll be **~/.bash** instead.)

First, find the **meshtastic** application file. Here's where it's located in Linux Mint:

```
/home/iffybooks/.local/bin/meshtastic
```

Run the following command to open **~/.profile** using the **nano** text editor.

```
sudo nano ~/.profile
```

Create a new line at the end of the file and type the following line, swapping in the path to your **meshtastic** application file:

```
export PATH="/home/iffybooks/.local/bin:$PATH"
```

- ❑ Press **ctrl+o** to save the file. You'll be prompted to check the filename. Press **enter**.
- ❑ Close your terminal window and open a new one.
- ❑ Run the following command and you should see a list of options for the Meshtastic CLI:

```
meshtastic --help
```

GET A LIST OF MESHTASTIC NODES

- ❑ Run the following command to see a list of Meshtastic nodes on the network:

```
meshtastic --nodes
```

SEND A MESSAGE FROM THE COMMAND LINE

- ❑ Connect your LoRa radio to your computer with a USB cable and run the following command to send a message to the primary channel:

```
meshtastic --sendtext "Hello out there\!"
```

SEND A MESSAGE EVERY 10 MINUTES

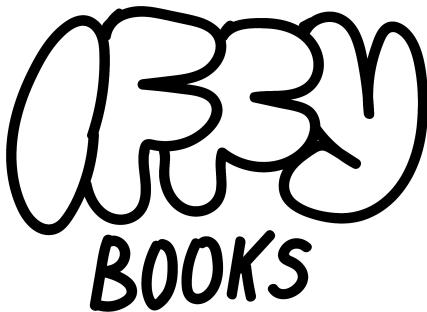
Here's a brief shell script you can use to send the same message every 10 minutes (600 seconds):

```
while :  
do  
    meshtastic --sendtext "Hello out there\!"  
    sleep 600  
done
```

LEARN MORE ABOUT THE MESHTASTIC BROADCAST ALGORITHM

You can learn more about Meshtastic's broadcast algorithm here:

<https://meshtastic.org/docs/overview/mesh-algo>



404 SOUTH 20TH ST., PHL
IFFYBOOKS.NET

(LAST UPDATED 6/29/2024)