Red Teaming with LoRA

March 5, 2026 @ SunSecCon

Venky Raju
Maker, Hacker, Engineer, Reverse Engineer

Agenda

1. Intro to LoRa & Meshtastic

B. Hands-on with Meshtastic

iii. Red-teaming with LoRa / Meshtastic

Δ Building your own Meshtastic image

What is LoRa?

What is LoRa (Long Range)?

- A proprietary radio communications technology.
- Based on spread spectrum modulation.
- Developed by a French company, Cycleo, in 2010.
- Cycleo was acquired by Semtech in 2012.
- Operates in the unlicensed, sub-Ghz ISM bands (US 902-928 MHz)

LoRa vs WiFi / Bluetooth

WLAN
Low Gbps
100 m

WPAN
Low Mbps
10 m

LPWAN
Low kbps
2-20 km

Unlicensed Spectrum

Low Power

Applications

Long Range

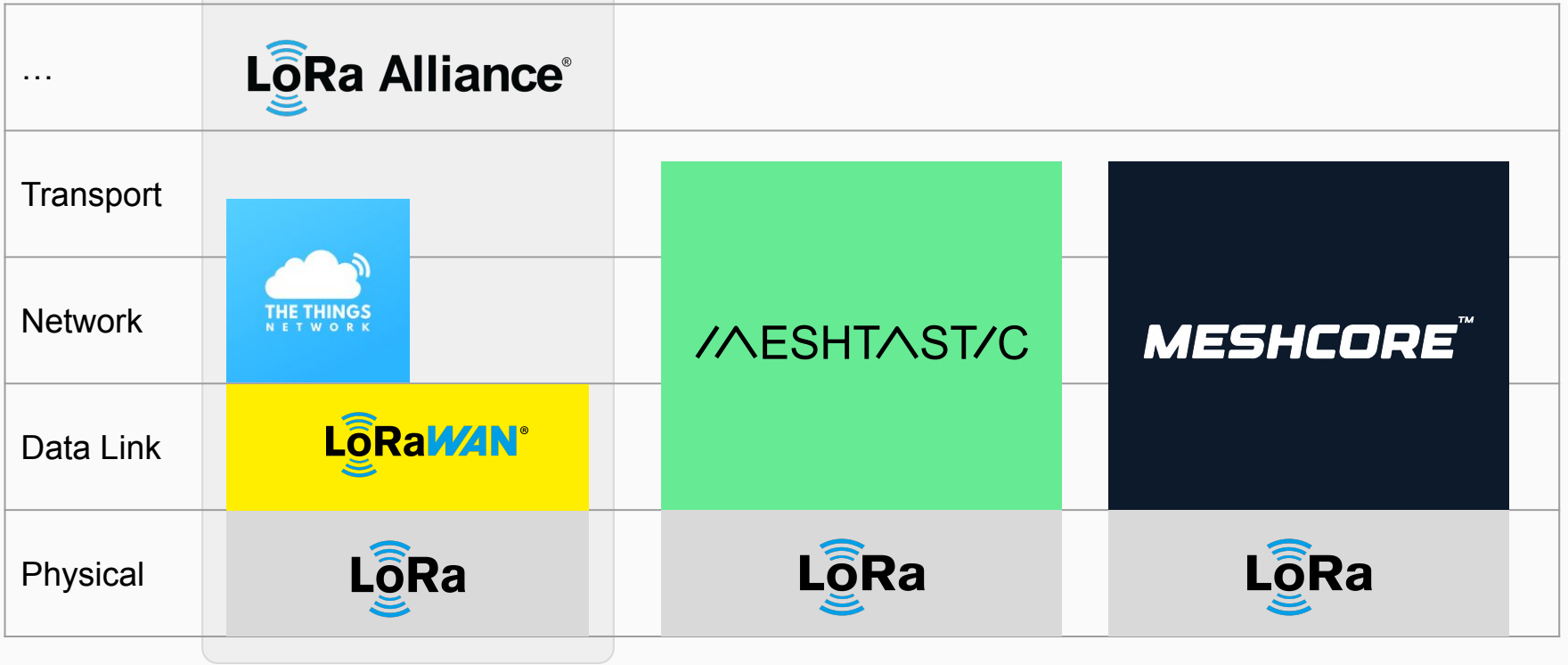
Low Speed

- Sensor networking
  - Industrial IOT
  - Smart Cities
  - Agriculture
- Community networking
  - Public networks
  - Ad hoc / Conferences
  - Hiking / Expeditions

How do we use LoRa?

After all, LoRa is a physical layer protocol

There are quite a few ways to use LoRa!



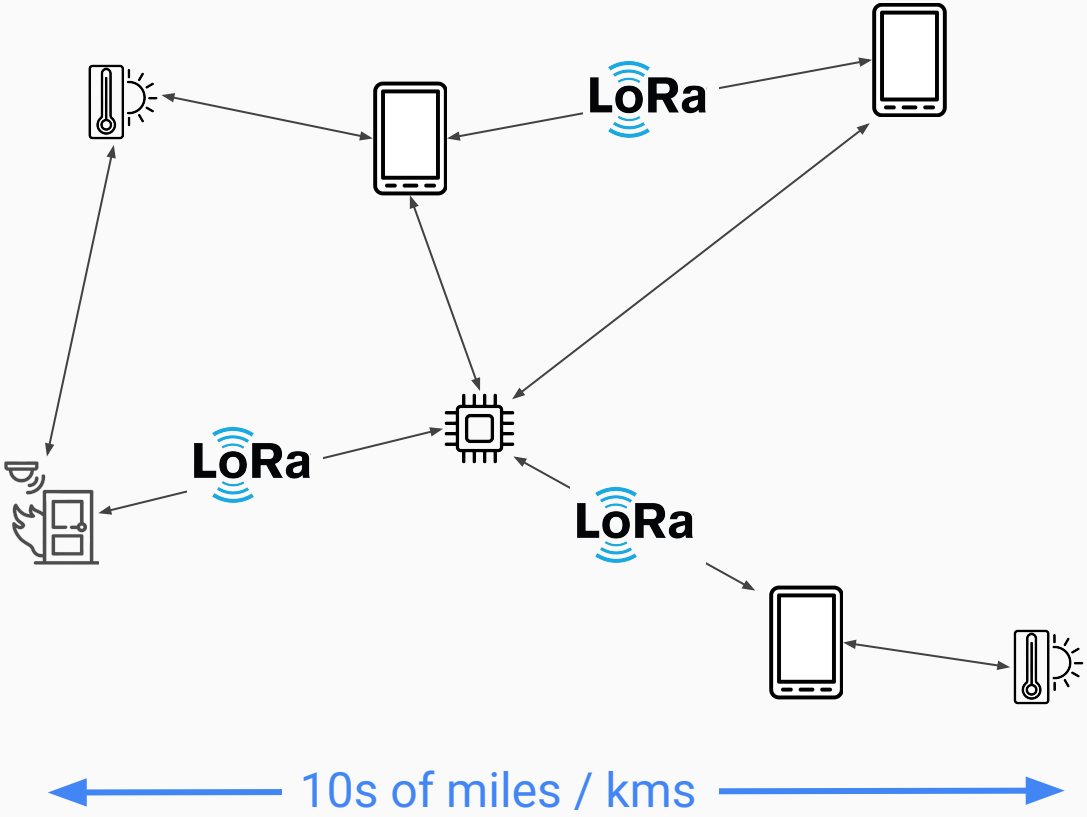
What is Meshtastic?

A mesh network of devices that use LoRa as the wireless link.

Fully decentralized, and every device can send, receive or relay messages.

100% open source and community driven!

https://meshtastic.org



TTN vs Meshtastic vs Meshcore?

TTN - Consortium-driven, focussed on industrial IoT use cases.

Meshtastic - User friendly, community networking and personal IoT.

Meshcore - Similar use cases as Meshtastic, arguably more capable.

Hands-on with Meshtastic

How do I get started?

STEP 1:

Build or purchase a Meshtastic-compatible LoRa device

Handheld

Looks like a RIM Blackberry

LCD screen with QWERTY keyboard

Ideal for messaging

LILYGO T-Deck

LILYGO T-Deck Plus

Pocket | Belt

Pair with a smartphone app (iOS / Android) for screen and keyboard

Messaging and some IOT

LowMesh Pocket-S
Solar / Battery

RAKwireless
WisMesh

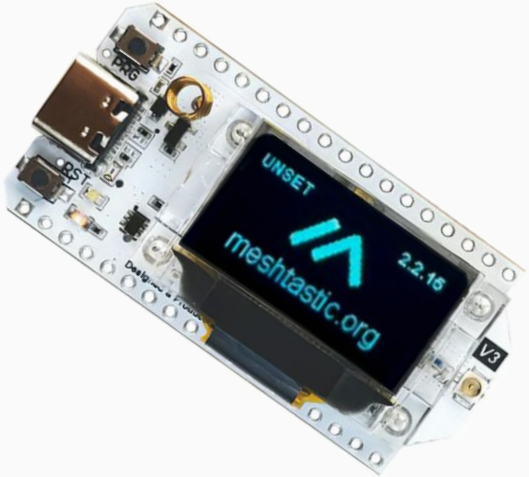
Node | Module

Pair with a smartphone app (iOS / Android) for screen and keyboard

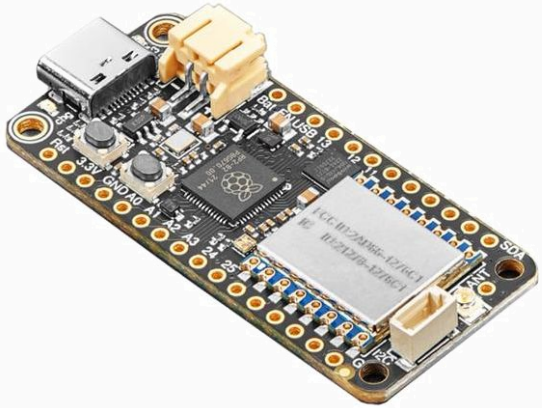
May have a small on-board OLED display

Breakout pins available for some functions

IOT + Hacking



Heltec ESP32 WiFi + BT LoRa Node



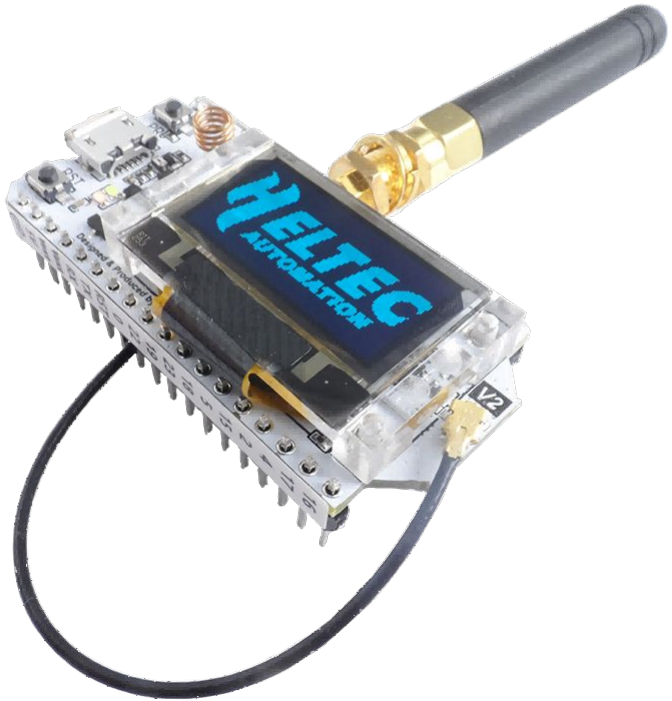
Adafruit RP2040 + RFM95 LORA

How do I get started?

STEP 2:

Flash the Meshtastic firmware onto your shiny new device!

STOP



First, attach the antenna!

Two ways to flash your device

Web Flasher:

- For ESP32-based devices
- Requires USB drivers

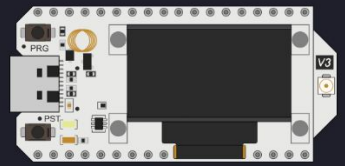
Drag-n-Drop:

- For RP2040 and nRF52 devices
- Typically do not require a driver
- Device shows up as a drive (UF2 bootloader)

https://meshtastic.org/docs/getting-started/serial-drivers/

Web-based Flasher - https://flasher.meshtastic.org/

FLASHER



Heltec V3

2.6.11.60ec05e Beta

Flash

Device

Plug in your device via USB. Please ensure the cable is not a power-only one.

Firmware

Choose from the release options or upload a release zip downloaded from Github.

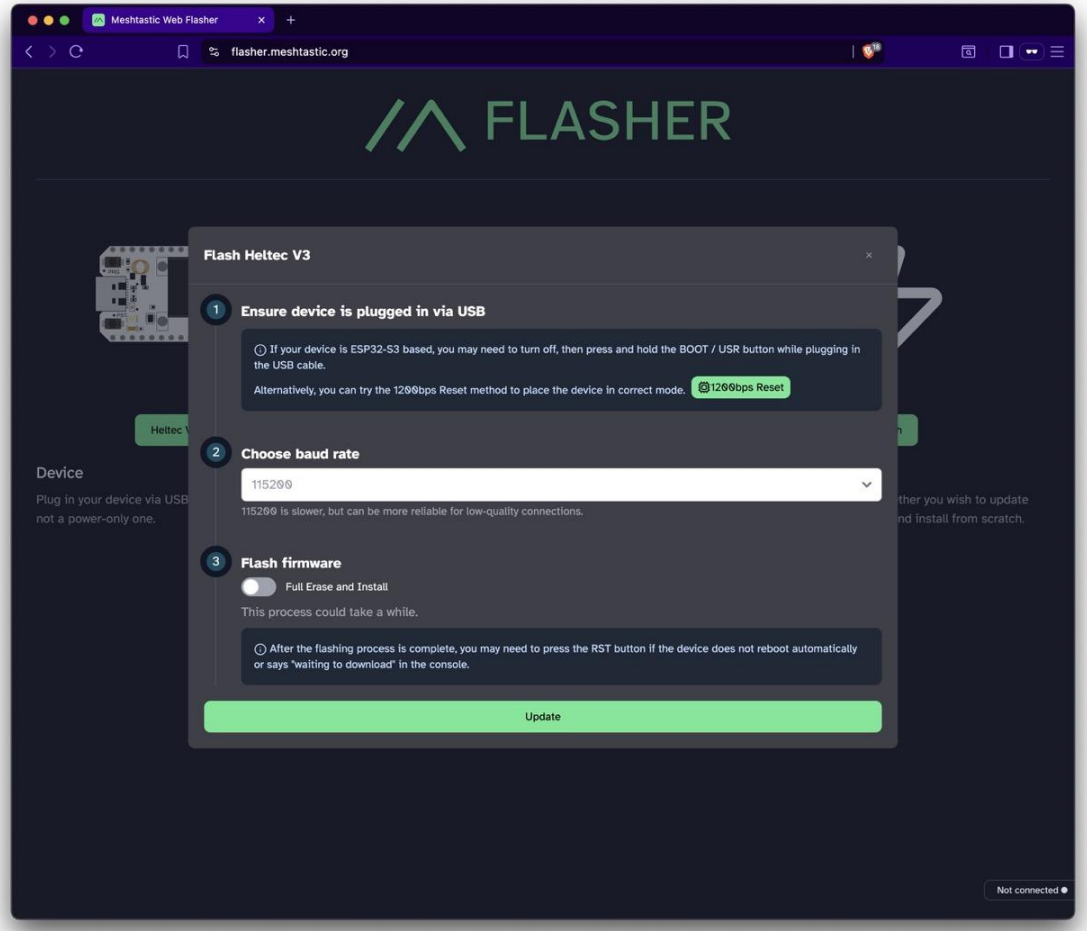
Flash

Flash your device. Choose whether you wish to update your device or wipe the flash and install from scratch.

Open Serial Monitor >_ | Meshtastic Docs | Contribute on GitHub | Language

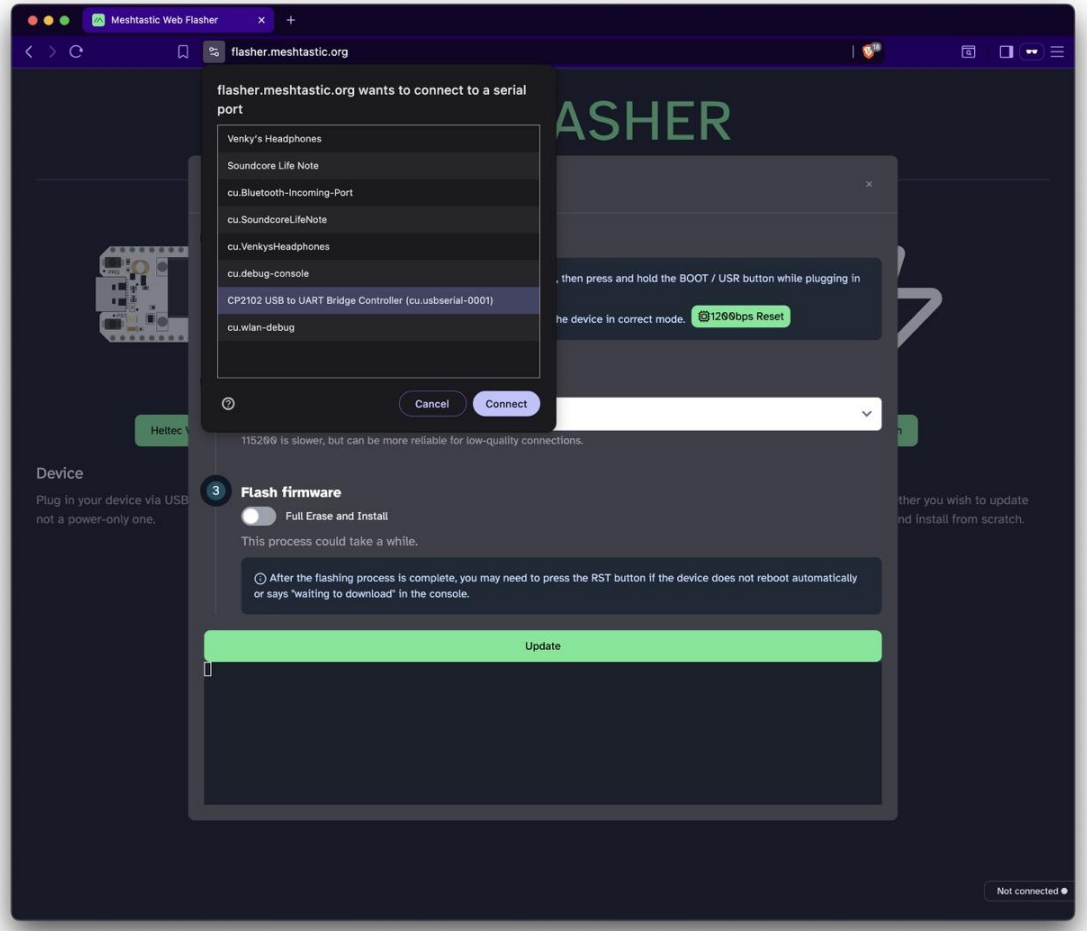
1

Connect your device and start the update wizard.



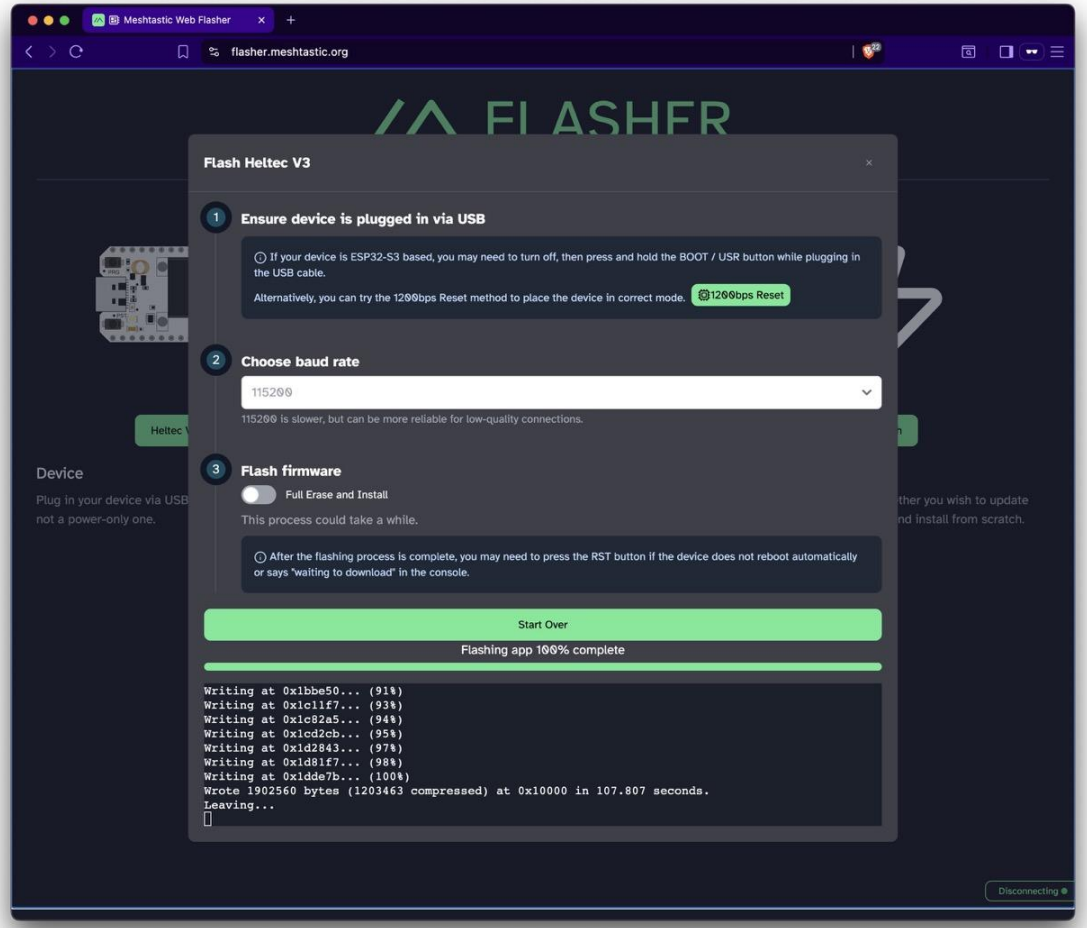
2

Select the virtual COM port and initiate the update process.



3

Wait until you see the “Leaving…” message after the update finishes.



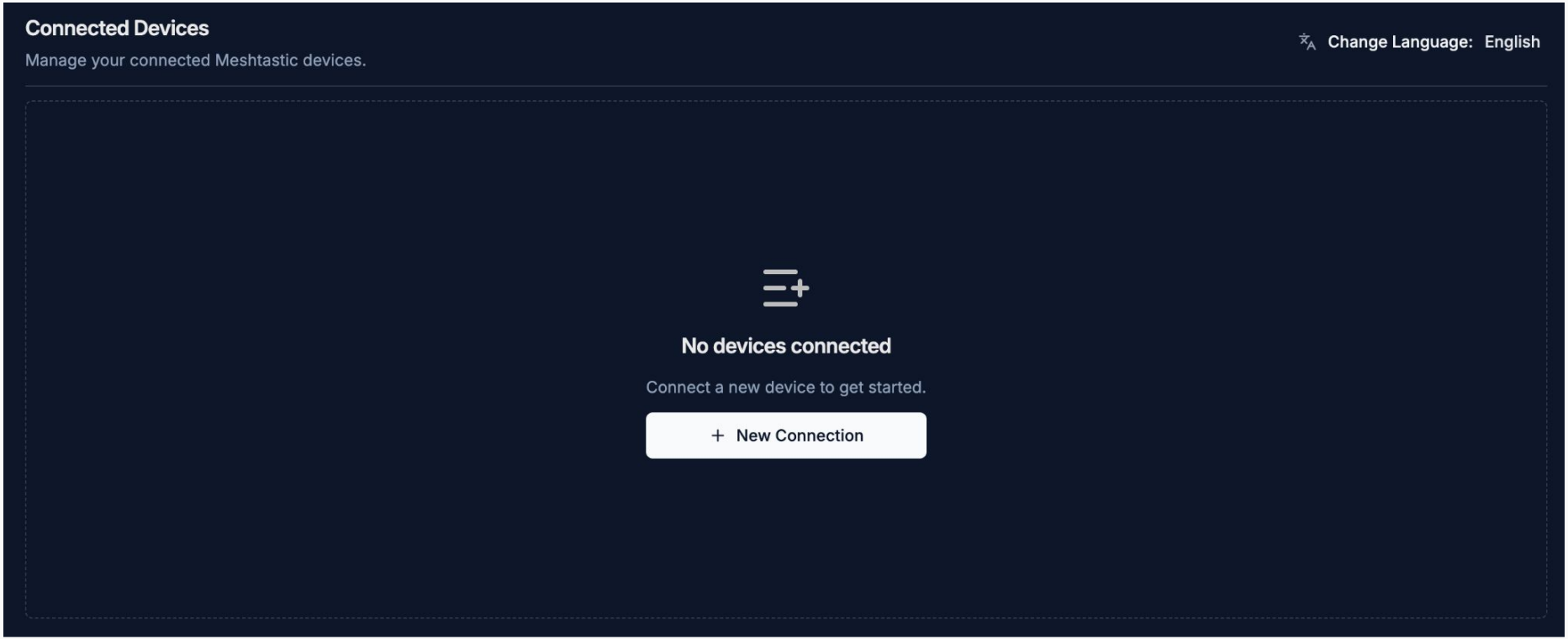
Time for a keyboard and larger display!

Mobile Apps

GET IT ON Google Play

Download on the App Store

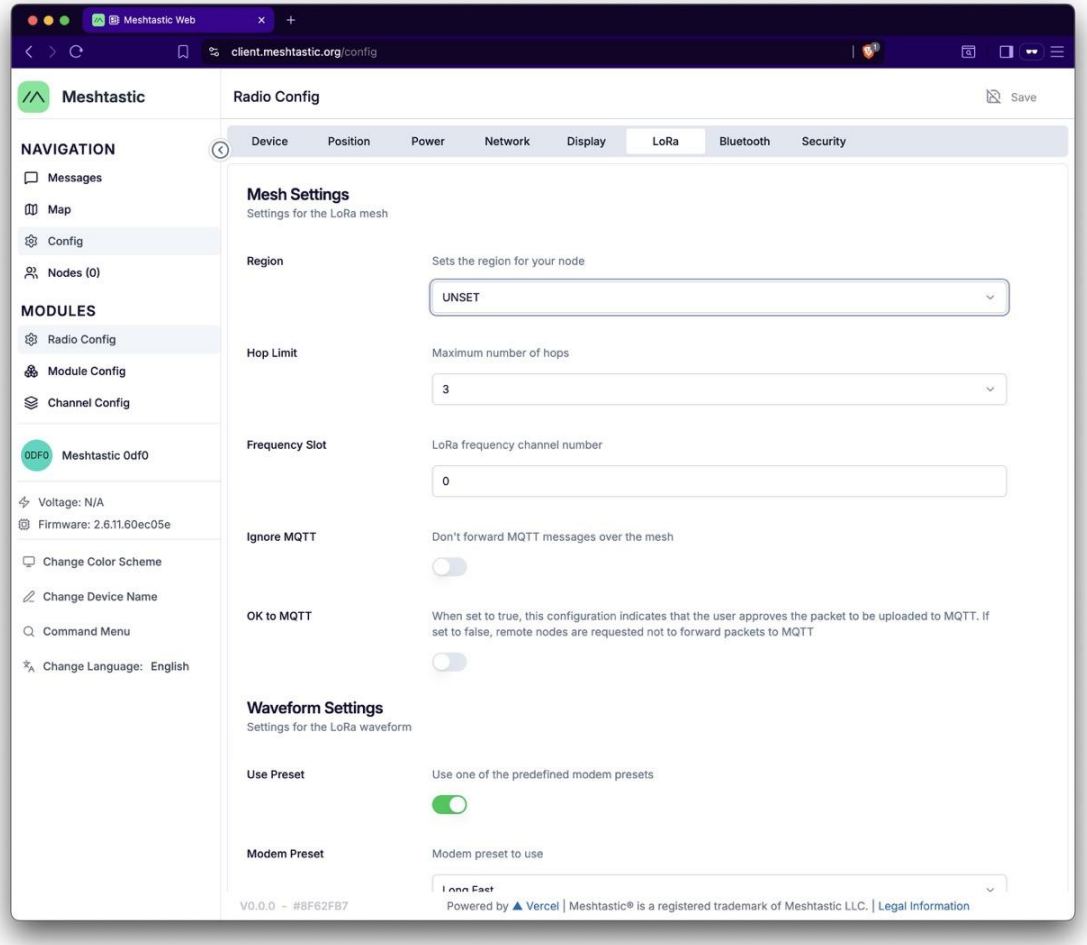
Web-based Client - https://client.meshtastic.org/



1

Select the “LoRa” tab under Modules > Radio Config

Change the region to US

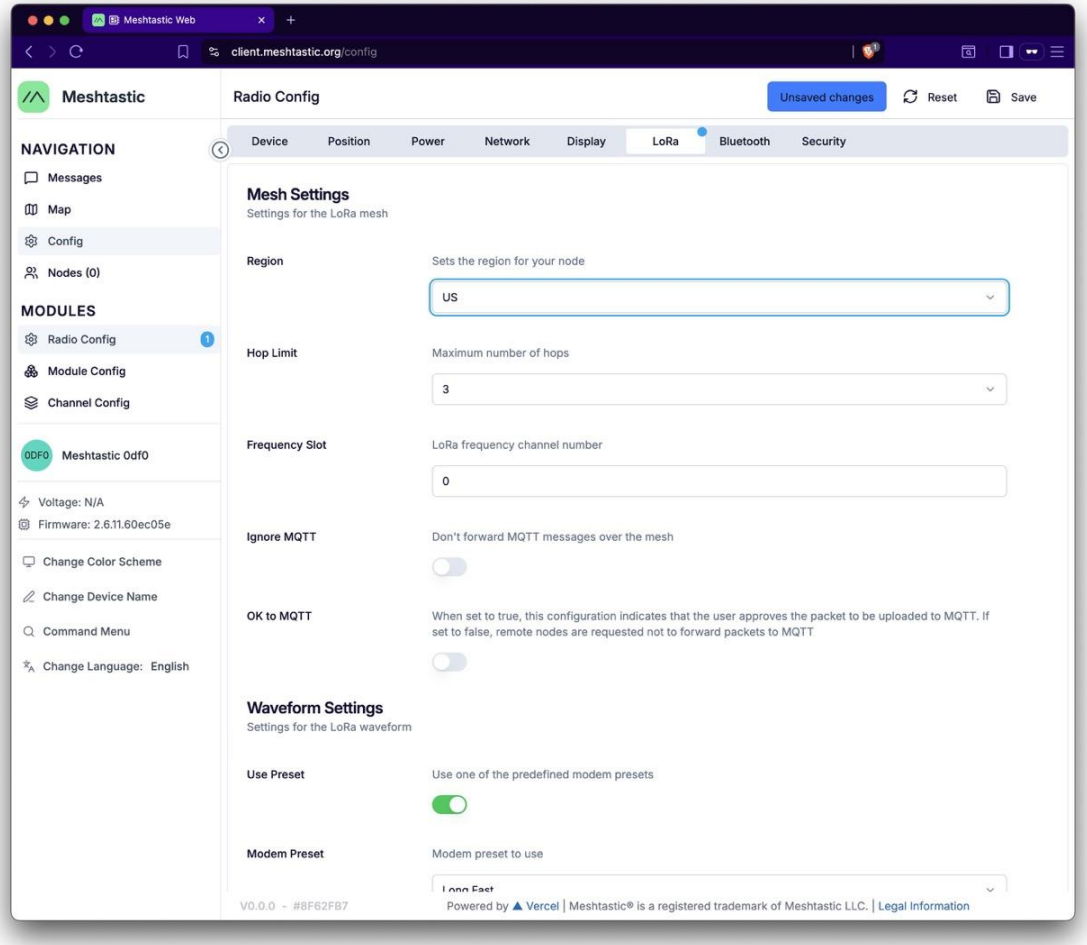


2

Click on “Save”

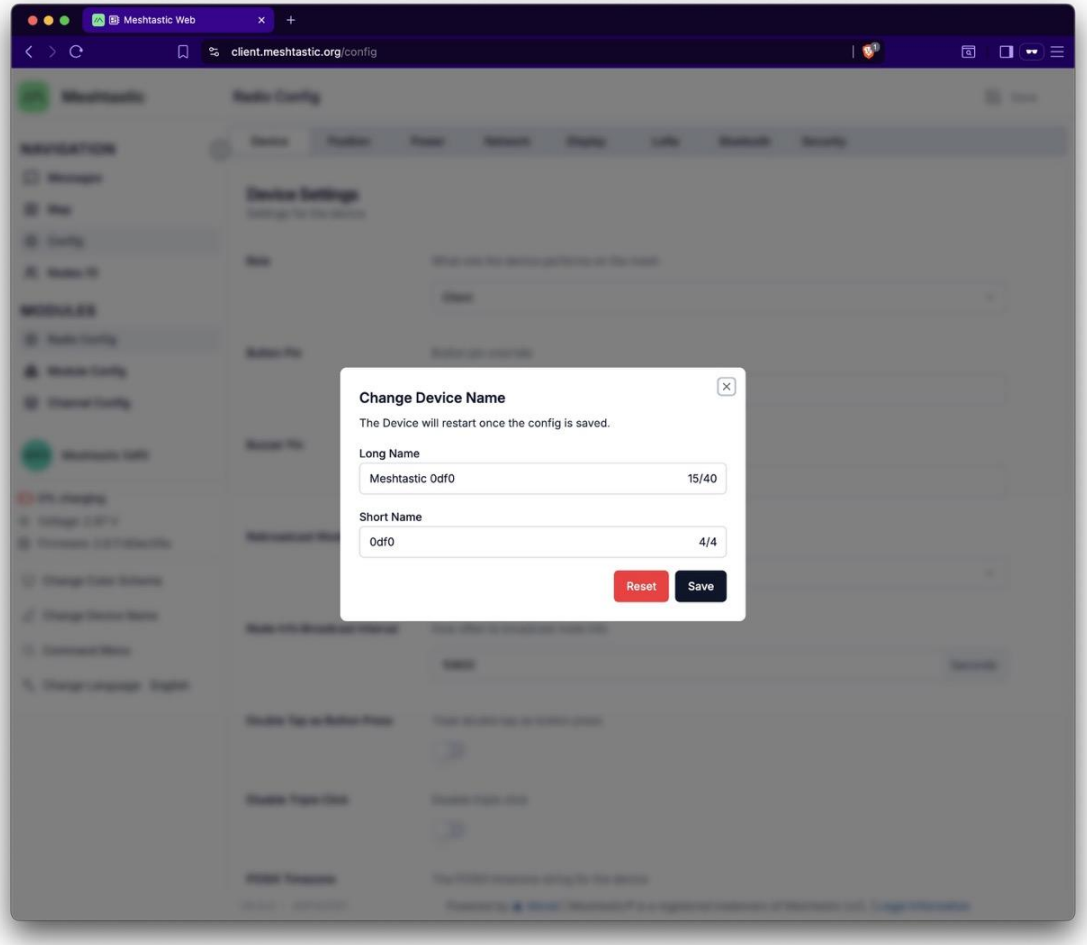
The device should reboot.

Reconnect when the device is ready.

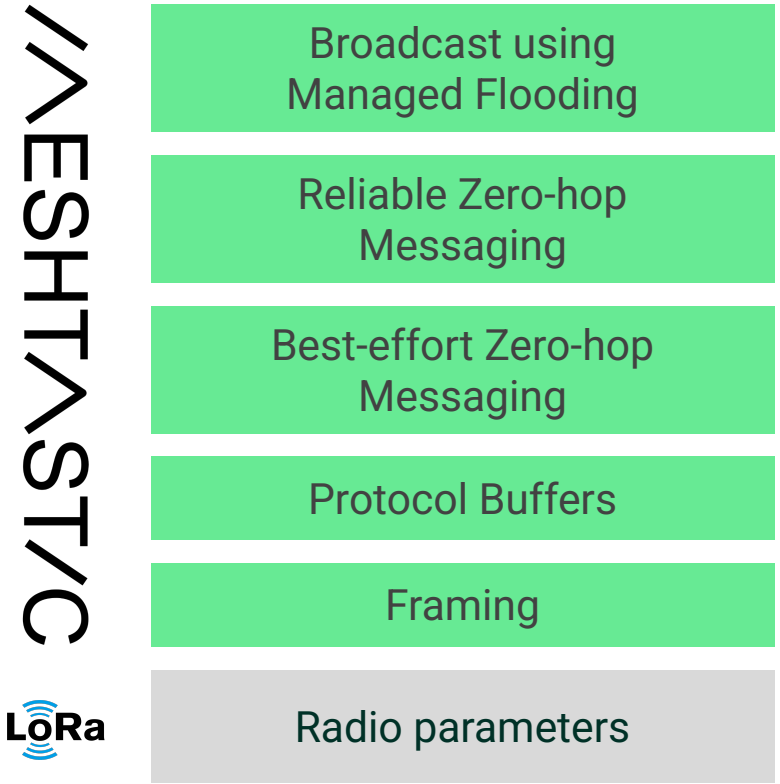


3

Change the device name and save.

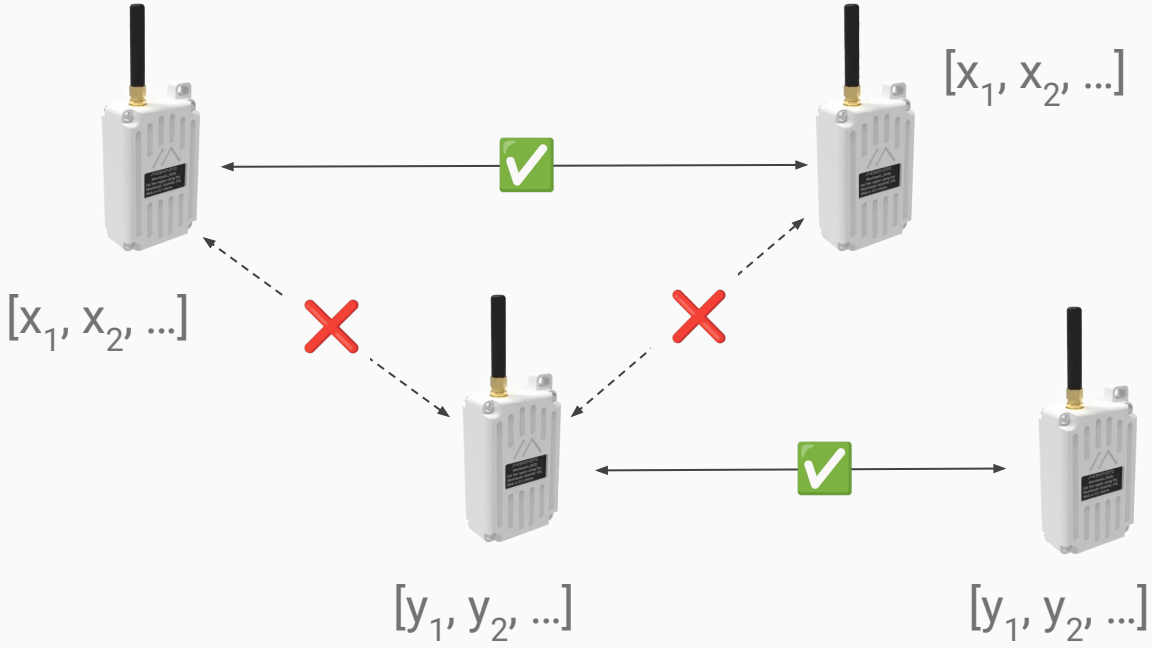


The Meshastic Stack



Radio Parameters

For two LoRa nodes to communicate, their radio parameters must match.



LoRA Parameters

- Spreading factor
  (bits per chirp)
- Bandwidth
  (slice of spectrum)
- Coding Rate
  (amount of redundancy)

Example

SF = 11 bits/chirp

BW = 250 KHz

Coding Rate = 4/5

Chirp time T_c = $(2^{SF}/BW)$ = 2048/250k = 8.2ms

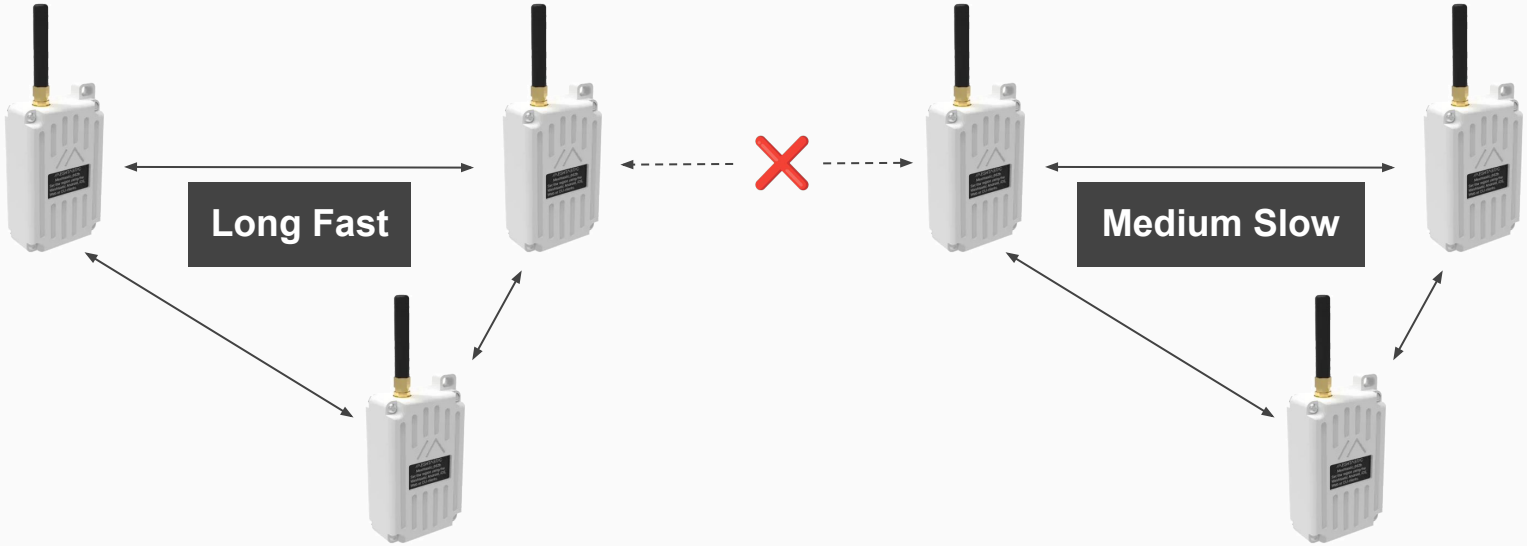
Bit rate = SF/T_c = 11/8.2ms = 1.34 kbps

Data rate = Bit rate * Coding rate = 1.07 kbps

Meshtastic Presets

Preset Name	Data-Rate	SF / Symbols	Coding Rate	Bandwidth
Short Turbo	21.88 kbps	7 / 128	4/5	500 kHz[1]
Short Fast	10.94 kbps	7 / 128	4/5	250 kHz
Short Slow	6.25 kbps	8 / 256	4/5	250 kHz
Medium Fast	3.52 kbps	9 / 512	4/5	250 kHz
Medium Slow	1.95 kbps	10 / 1024	4/5	250 kHz
Long Fast (Default)	**1.07 kbps**	**11 / 2048**	**4/5**	**250 kHz**
Long Moderate	0.34 kbps	11 / 2048	4/8	125 kHz
Long Slow	0.18 kbps	12 / 4096	4/8	125 kHz

If you don’t see your friend, they may be on a different network!

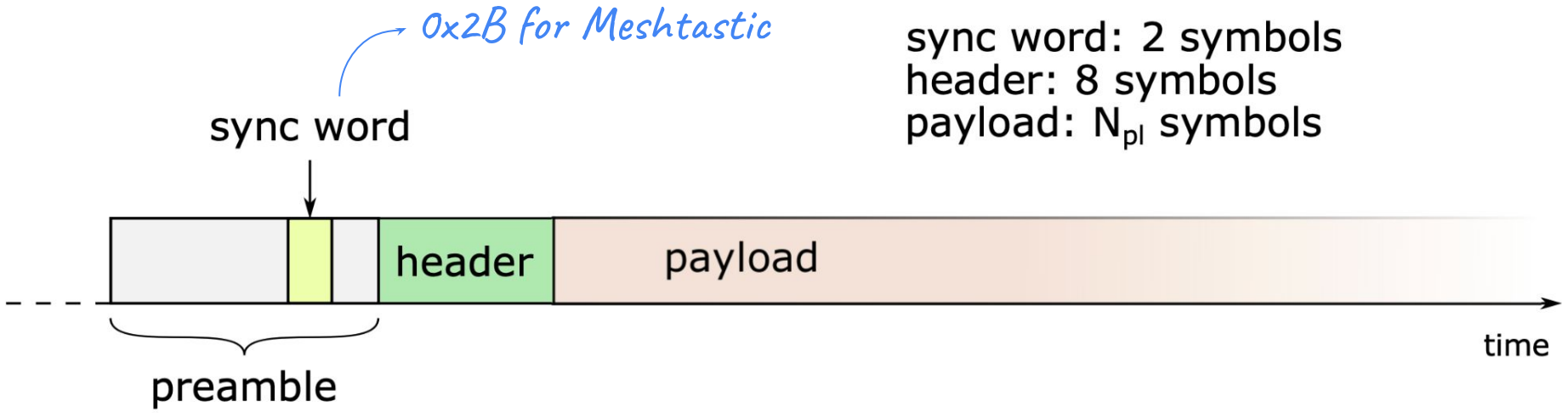


What about interference from other LoRa networks?

Meshcore, TTN and other LoRaWAN networks share the same LoRa physical layer.

How can my Meshtastic node avoid processing packets of those networks?

LoRa Framing and the “Sync word”

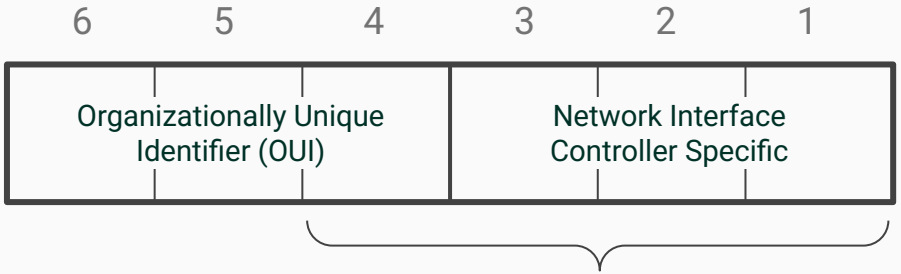


Source: **Vincent Savaux, Christophe Delacourt, Patrick Savelli.** Considering Sync Word and Header Error Rates for Performance Assessment in LoRa System. Wireless telecommunications Symposium, Apr 2021, San Francisco (virtual), United States.

Node Addressing

Lower 4 octets of Bluetooth EUI (aka MAC address)

Broadcast is *0xFFFFFFFF*



Zero-Hop Direct Message

Channel Activity Detection
Transmit message

Source ID = A
Destination ID = B
Want ACK = 1
Data

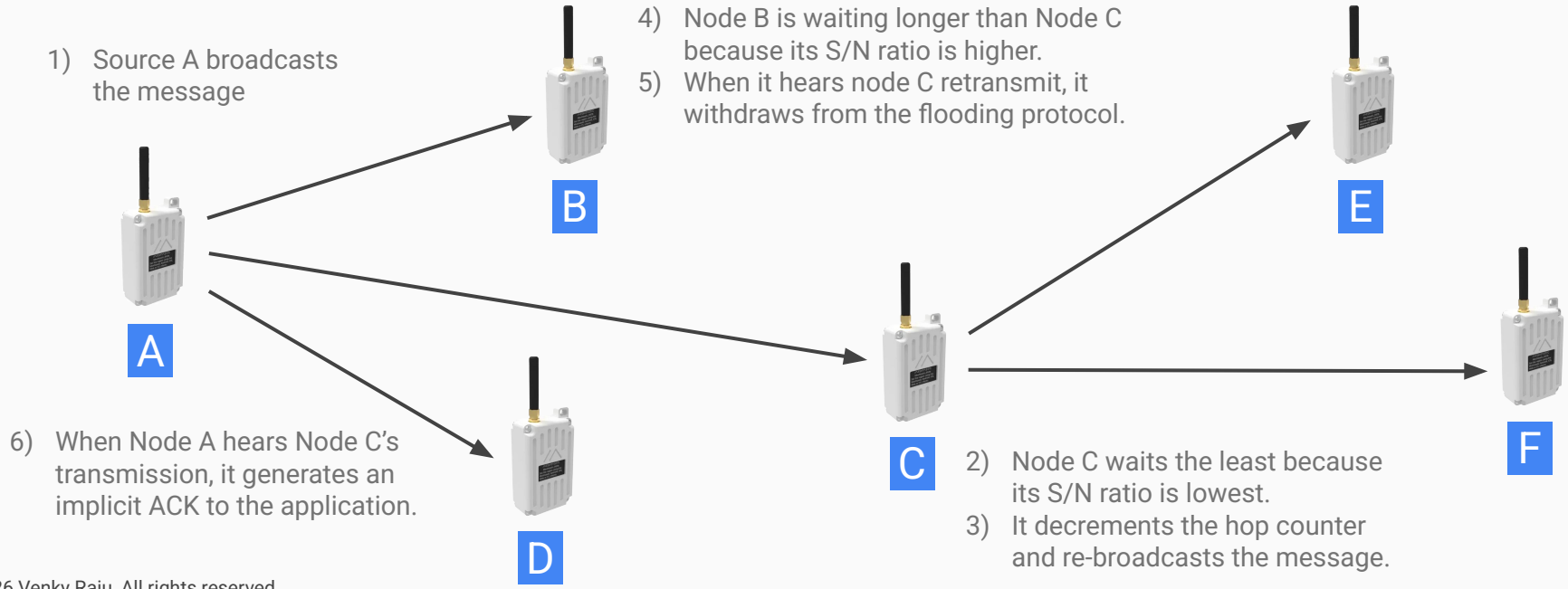
A

B

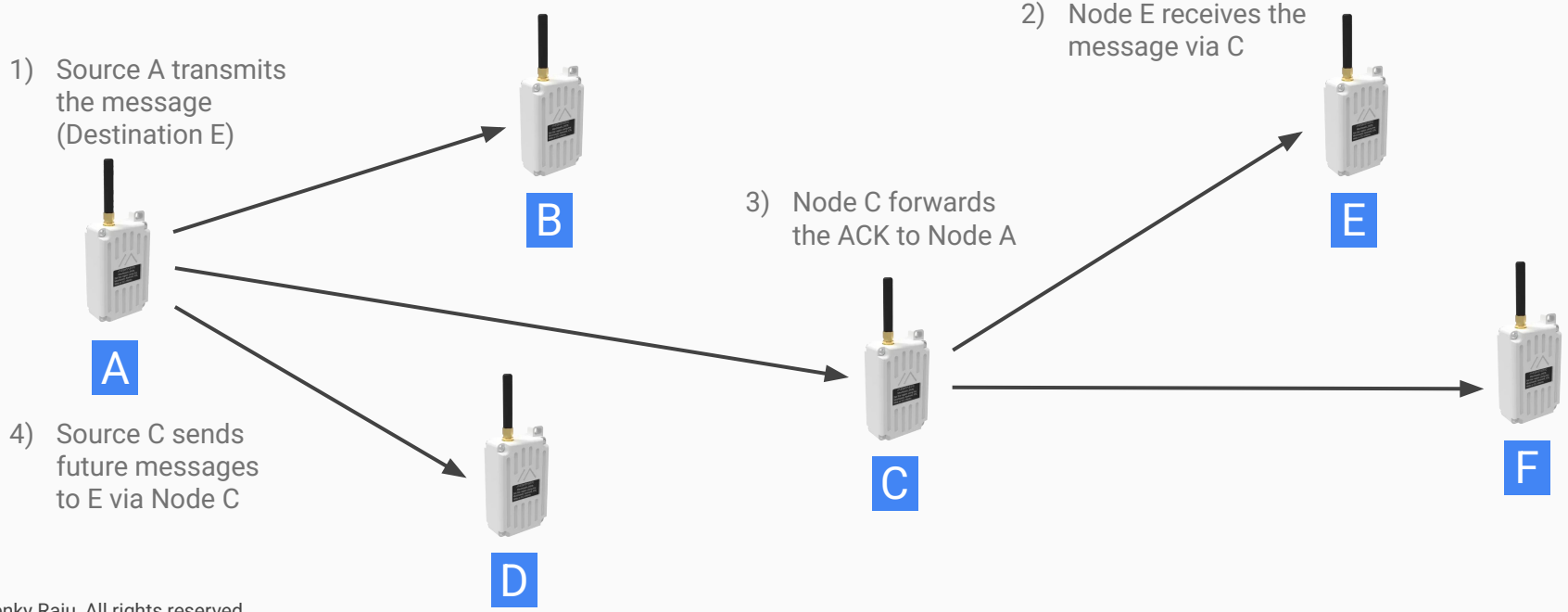
Channel Activity Detection
Transmit message

Source ID = B
Destination ID = A
ACK

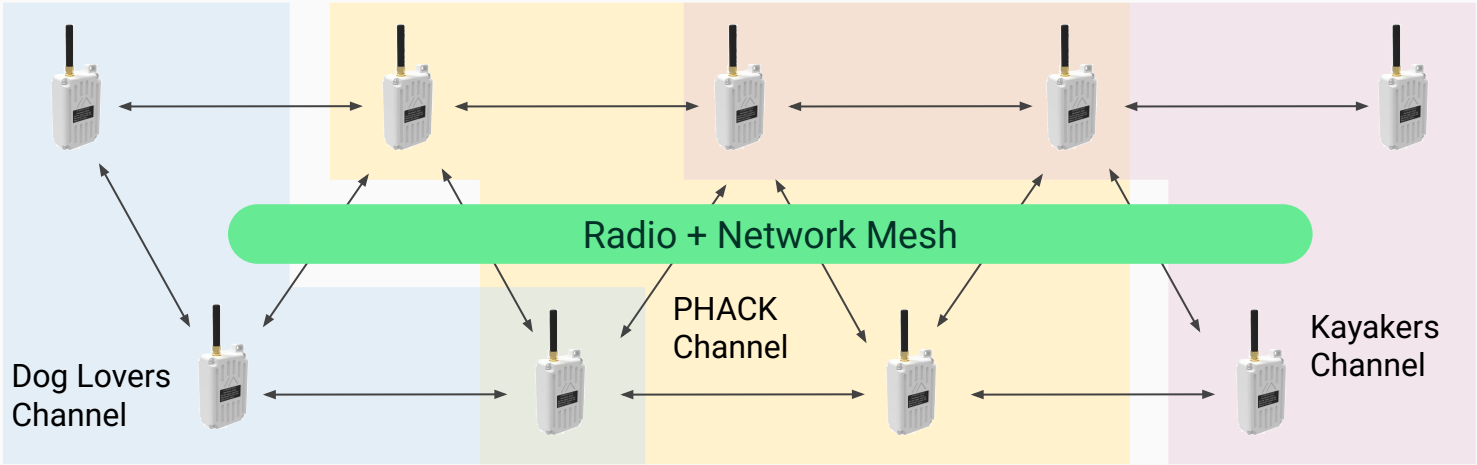
Broadcast Message
[Managed flooding]



Multi-hop Direct Message
[Managed flooding + memory]



Channels
[A sub-mesh, or community within the larger mesh]



About Channels

A channel is identified by a name and an encryption key

A node may belong to a maximum of 8 channels

All nodes are members of the unnamed Channel 0, with encryption key AQ==

Red Teaming

Notice

- This demonstration is strictly for informational purposes only.
- Do not use any of the techniques for malicious or illegal activities.
- The presenter and host organization disclaim all liability.
- Attendees are solely responsible for their own actions.

How can adversaries exploit LoRa?

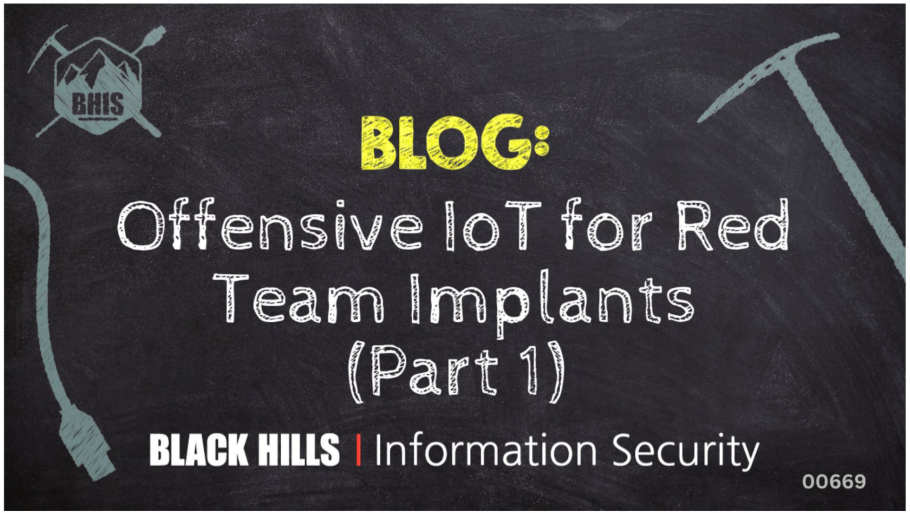
Enablers:

1. Long range
2. Low power
3. Public mesh

Inspirations

Offensive IoT for Red Team Implants – Part 1

| Tim Fowler

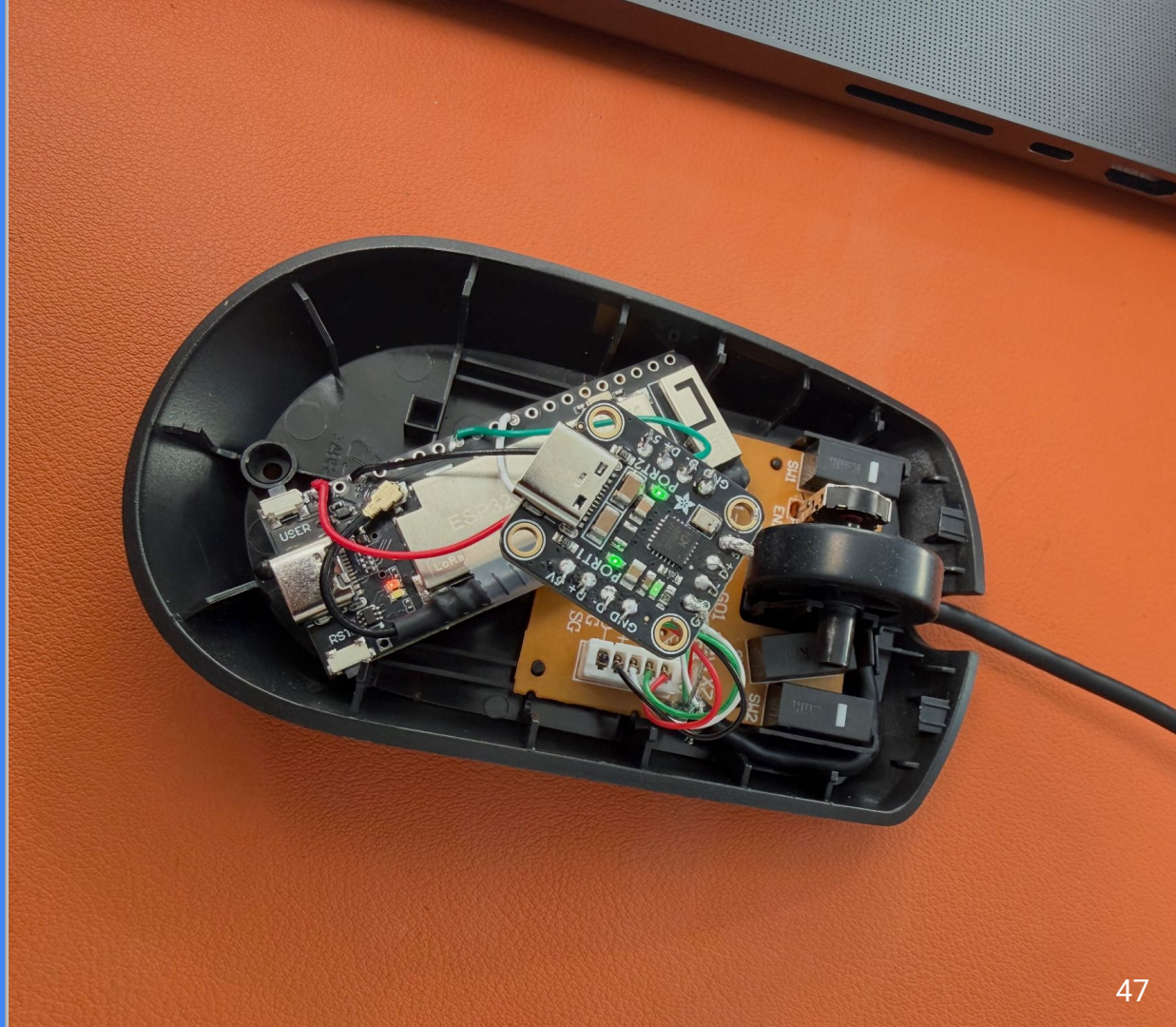


Meet LoKey

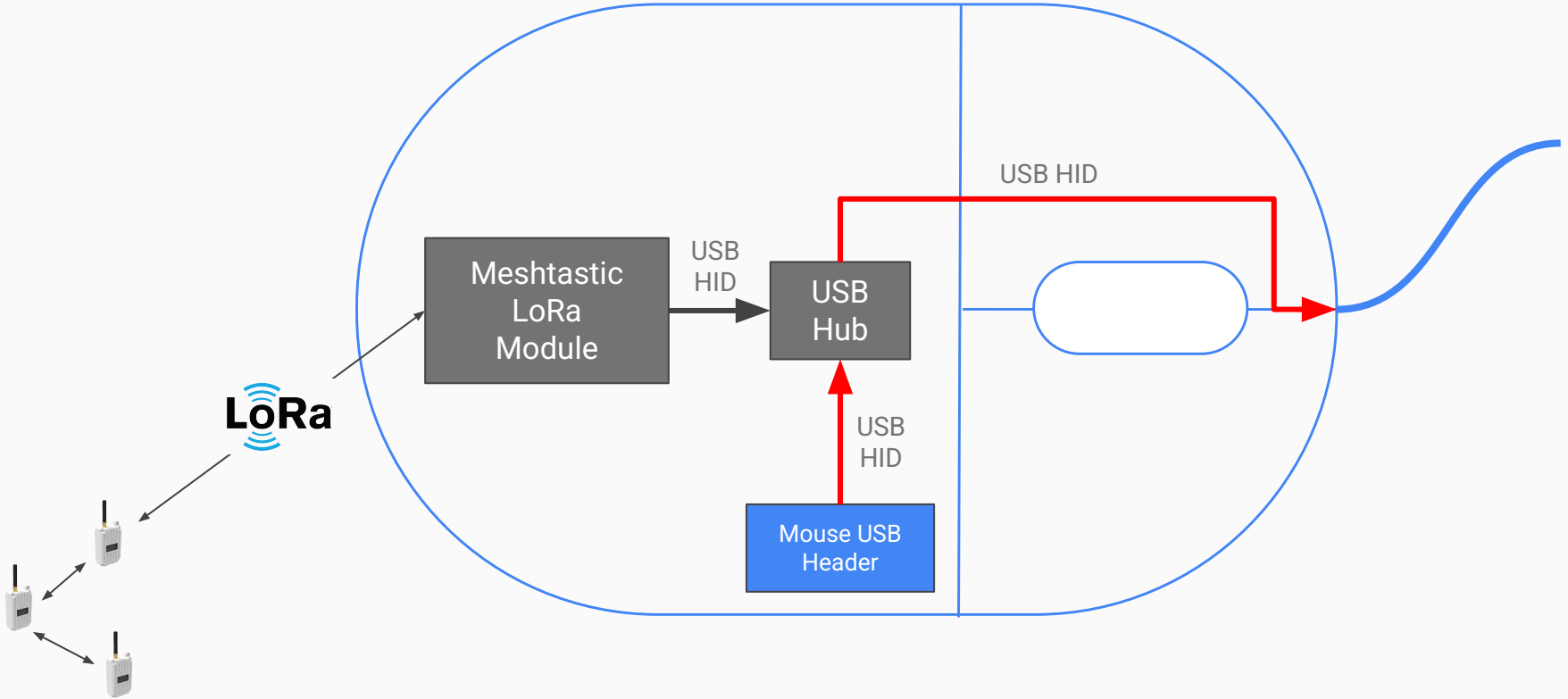
A LoRa-based Implant in an off-the-shelf wired USB mouse.

Emulates a keyboard with Ducky Script™ compatible commands.

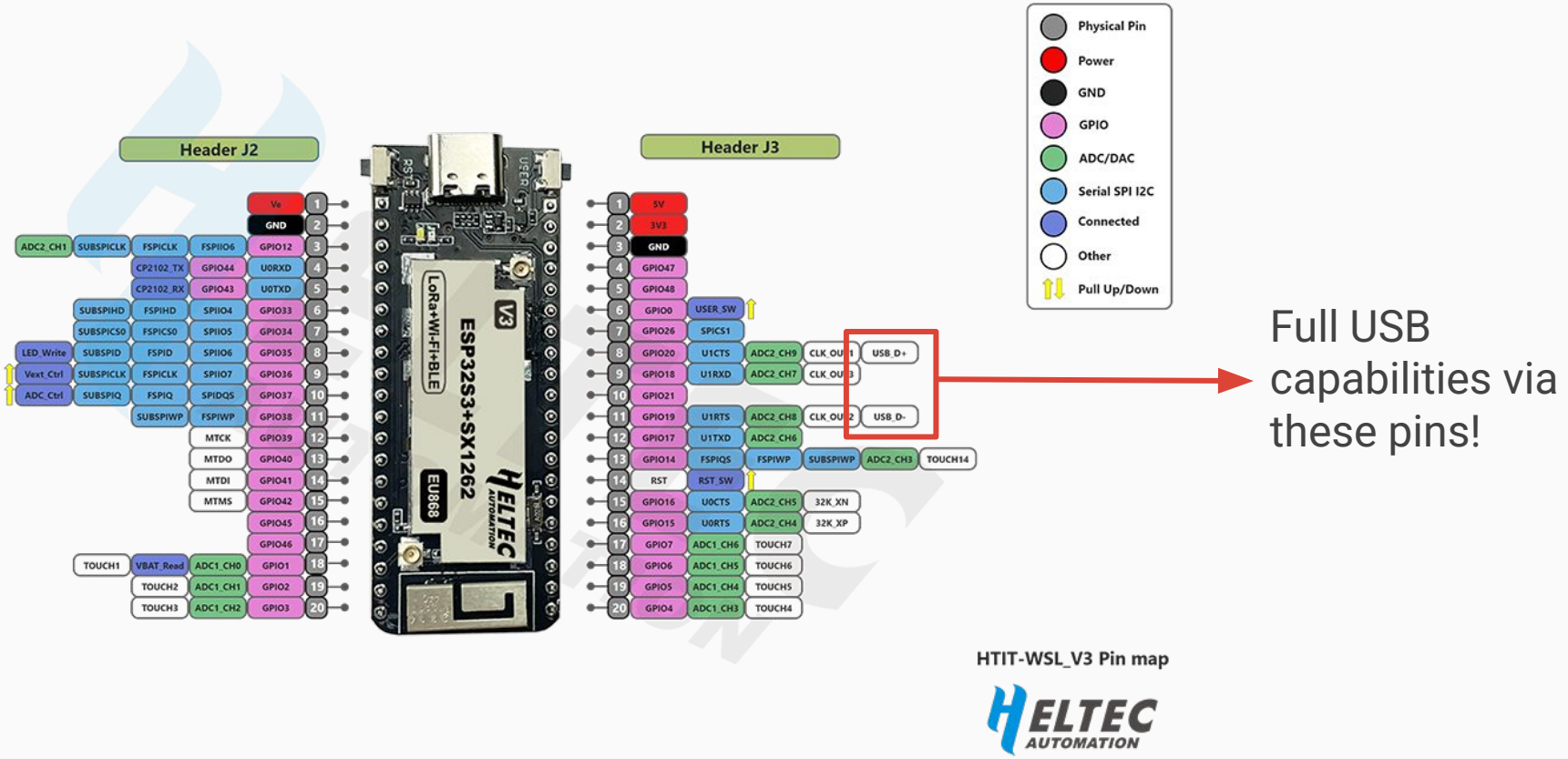
Retains full mouse functionality!



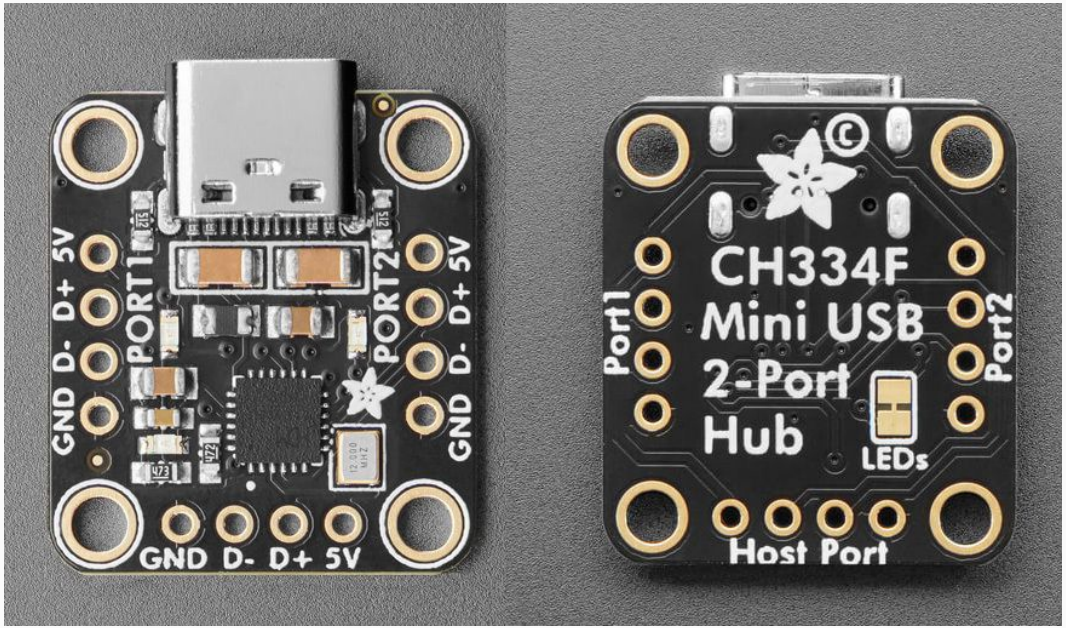
Block Diagram of the Implant



LoRa Module - Heltec WiFi LoRa module (V3)

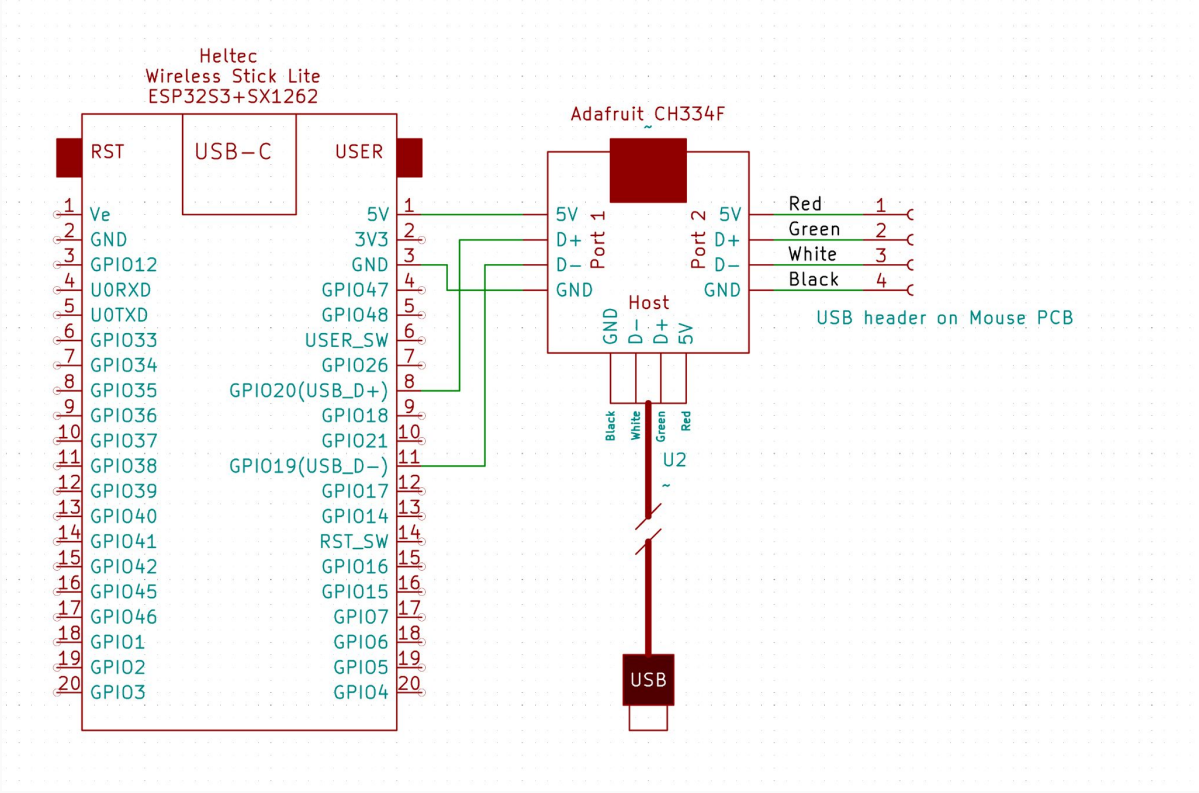


USB Hub Module - Adafruit CH334F Mini USB Hub Breakout

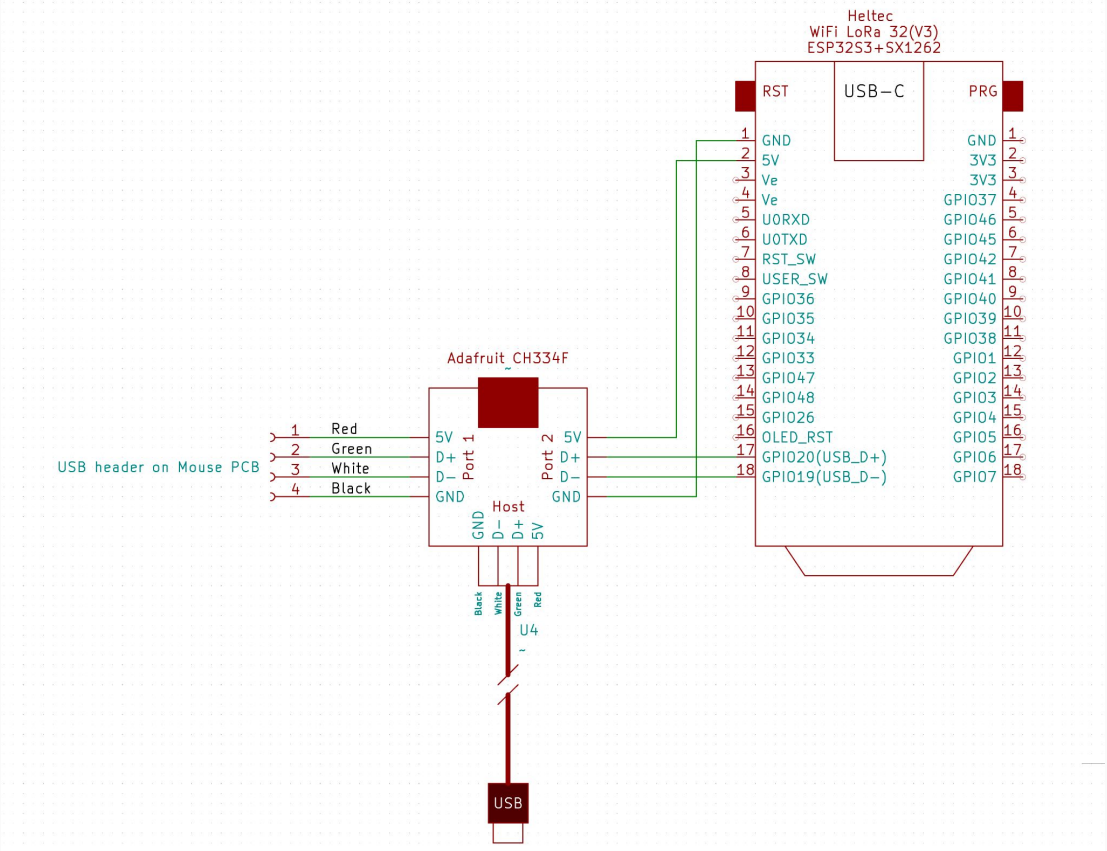


- Two downstream ports
- USB-C or pins for the host port

Schematic for the Heltec Wireless Stick Lite (No OLED)



Schematic for the Heltec WiFi LoRA module with OLED



Software

Fork of official firmware

2 files added

2 files modified

New HID Keyboard Module (.cpp and .h)

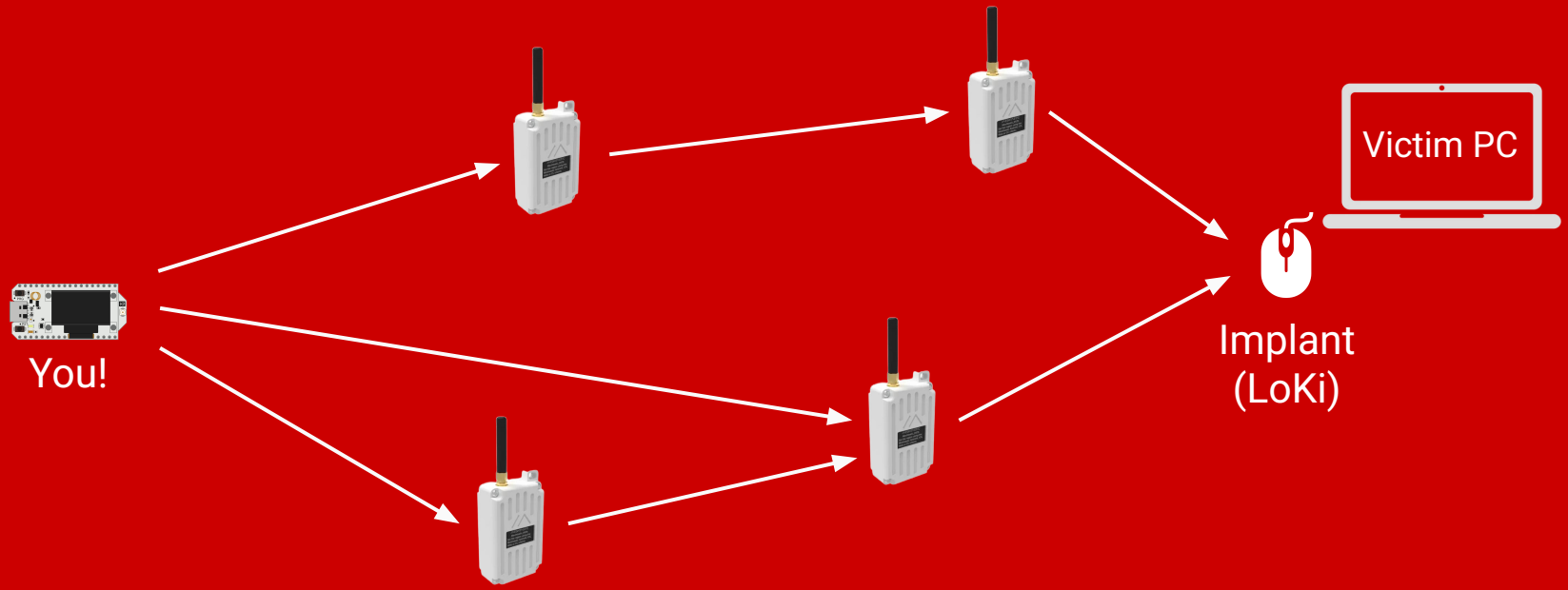
- Meshtastic module that converts LoRa text messages into USB HID keyboard input
- Implements a USB Rubber Ducky–style scripting language with commands: STRING, STRINGLN, ENTER, TAB, BACKSPACE, SPACE, GUI, ALT, CTRL, DELAY, and arrow keys
- Only processes direct messages (ignores broadcasts) and executes explicit LoKey commands, preventing accidental execution of plain text
- Initializes automatically at device boot, ready to receive commands without manual setup. Includes PING/PONG mechanism to signal readiness of USB host.
- Supports single modifiers (GUI/ALT/CTRL + key), dual modifiers (CTRL+SHIFT+key), function keys (F1–F12), and special keys with proper timing delays
- Multi-Line Script Execution — Parses and executes multi-line LoKey scripts line-by-line with automatic trimming and command separation

Changes to platform.ini and Modules.cpp

- Defines HELTEC_V3_HID

- Integrate HID Keyboard into the Meshtastic module system by instantiating it in setupModules() alongside other core modules
- Module initializes only when HELTEC_V3_HID is defined, keeping it disabled for other boards and hardware variants

Today’s Demo Scenario



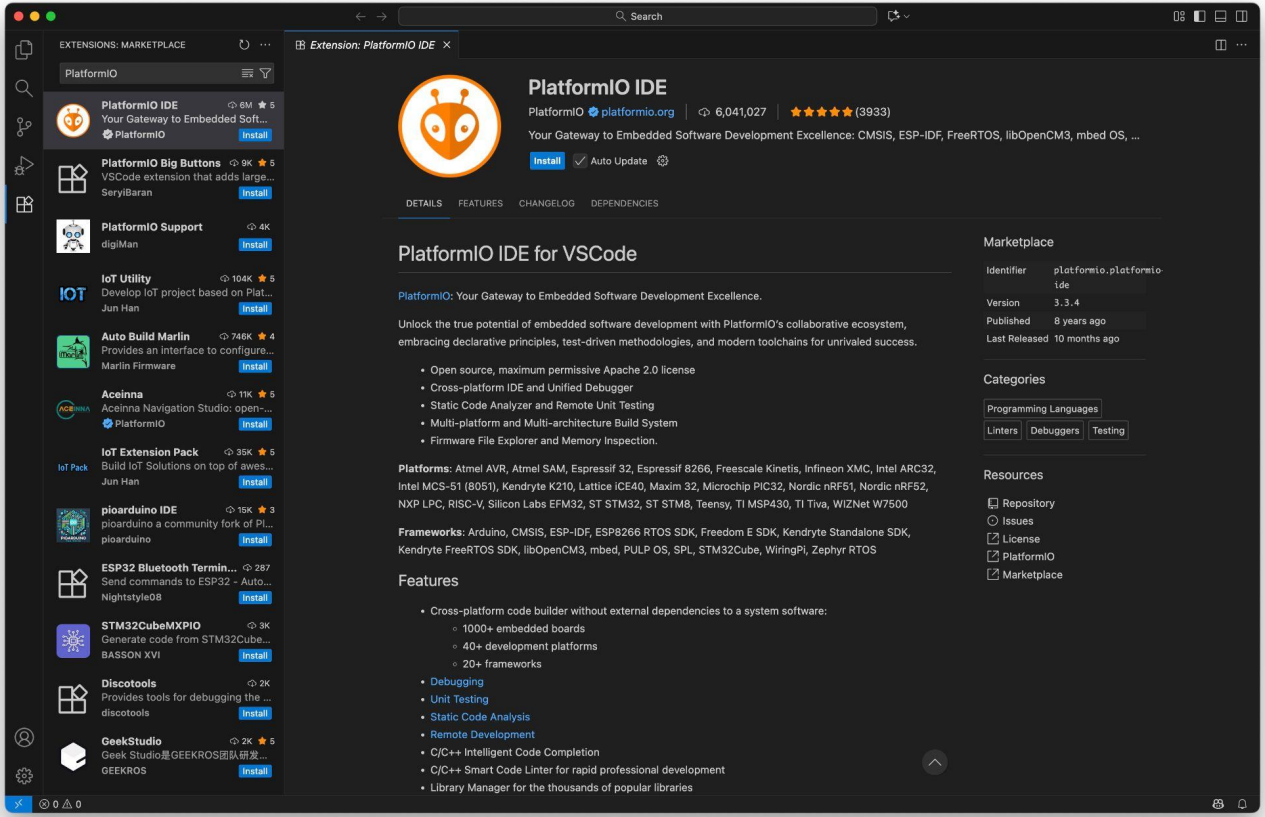
Building your own Meshtastic firmware image

Pre-requisites

- Install git
- Install VSCode / Cursor / Windsurf / …
- Clone the meshtastic firmware repo

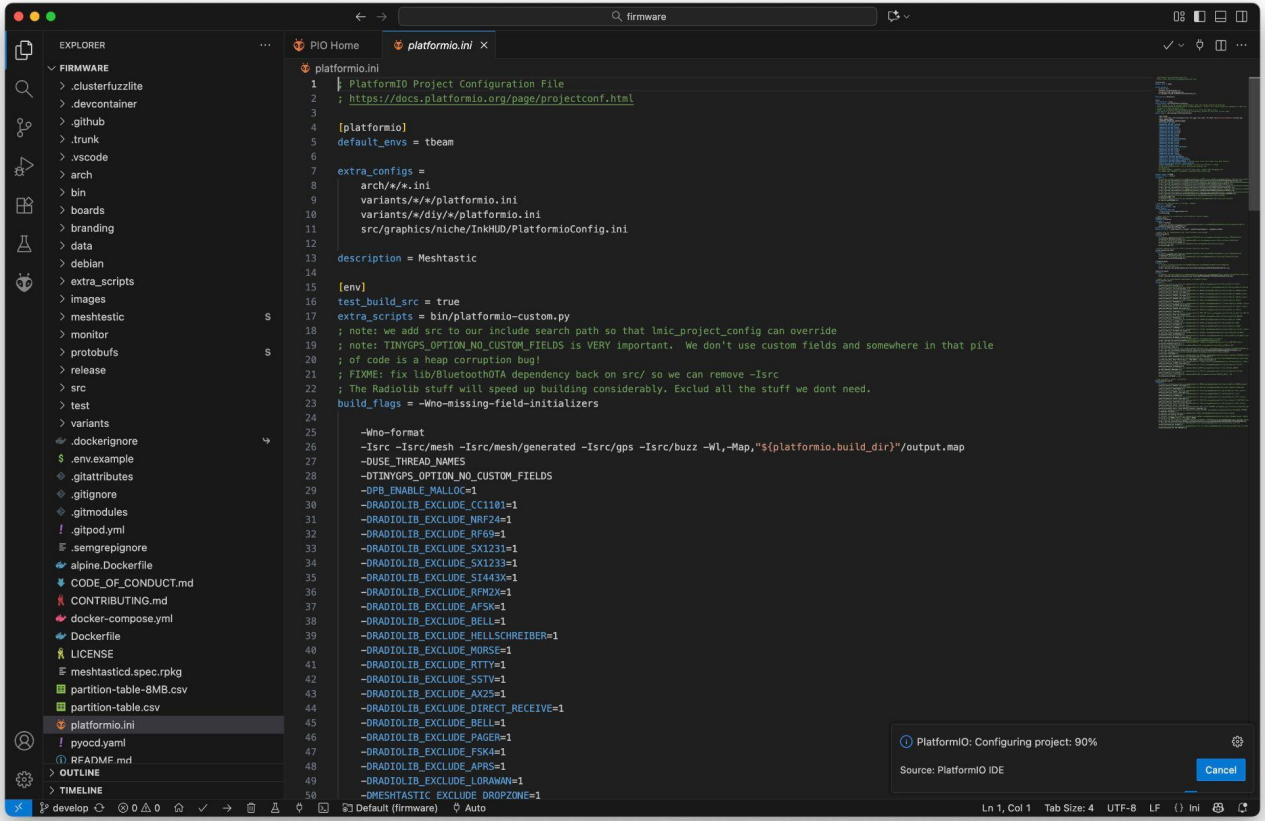
1

Install the PlatformIO IDE extension



1...

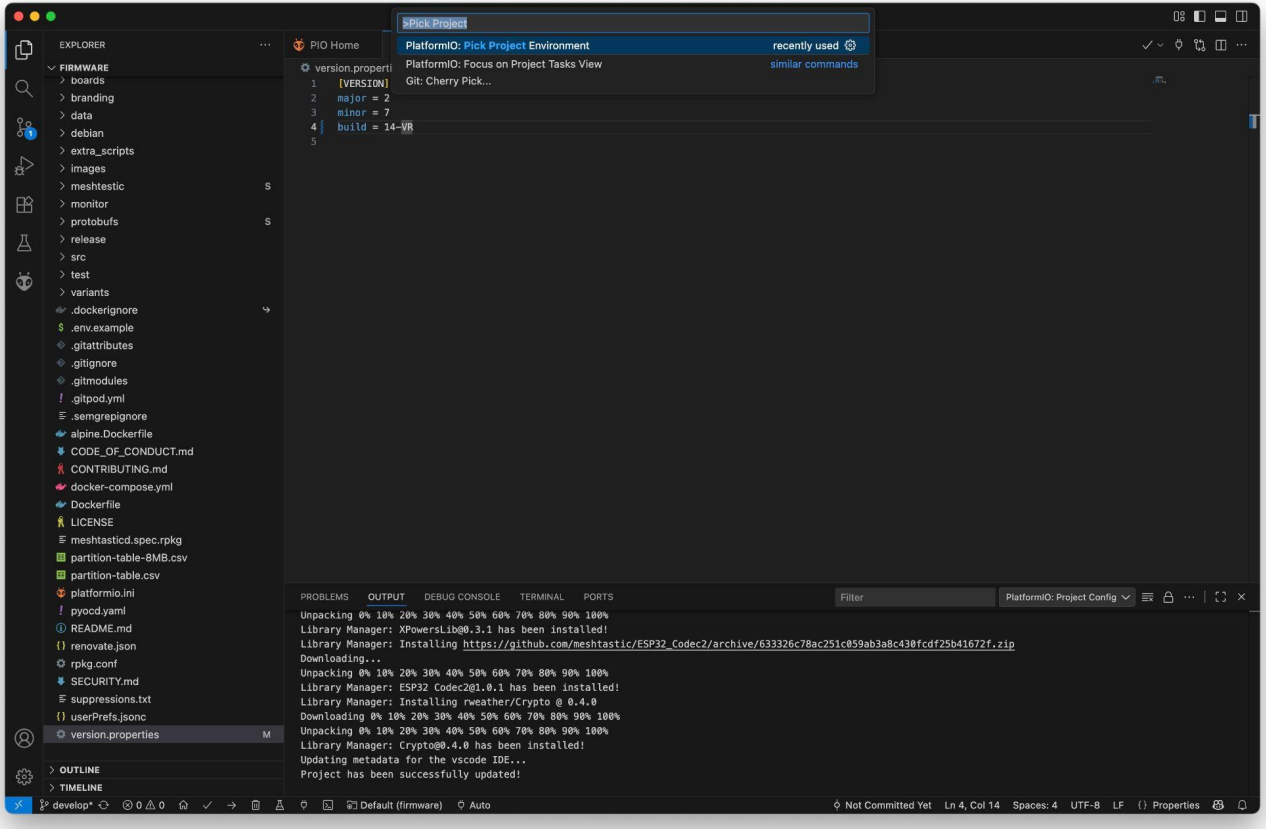
This may take a few minutes to complete.



2

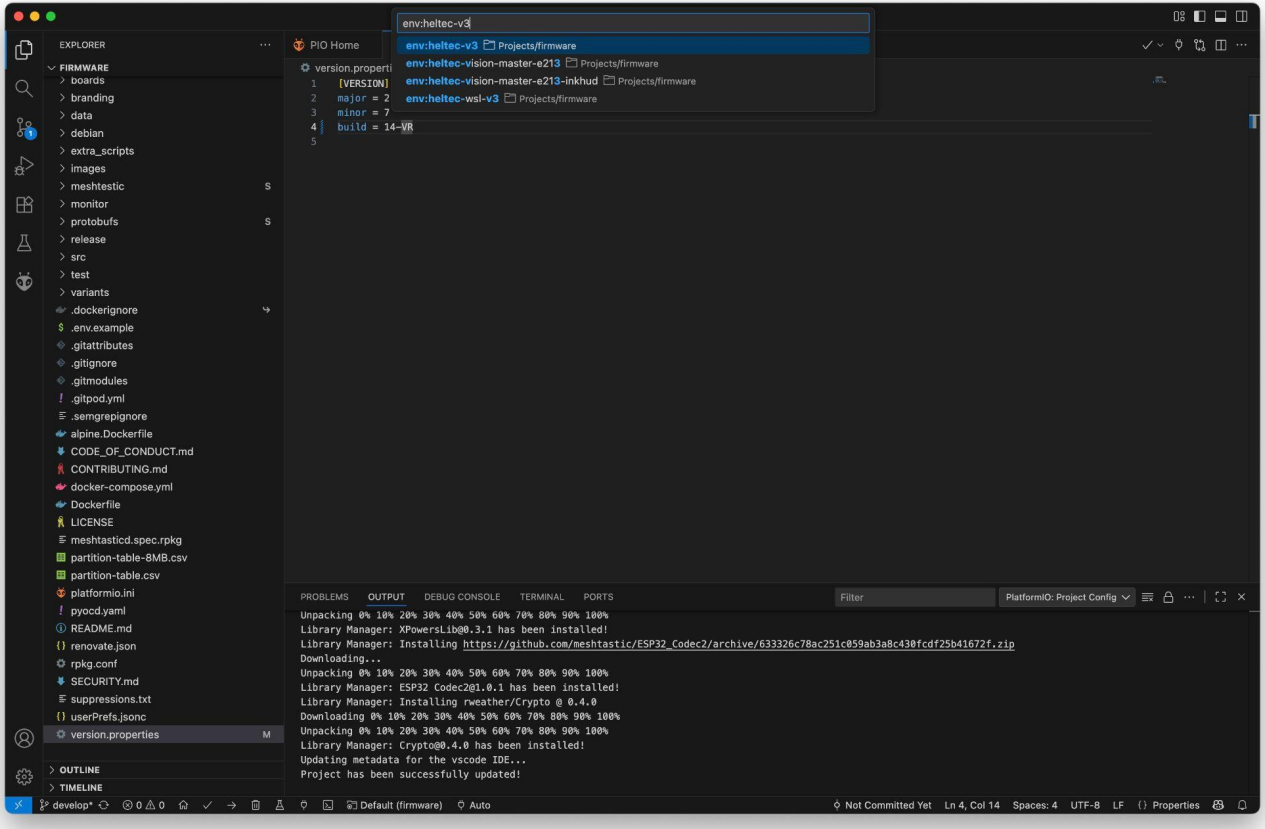
Make a small change in the version.properties file.

From the command palette, select PlatformIO : Pick Project Environment



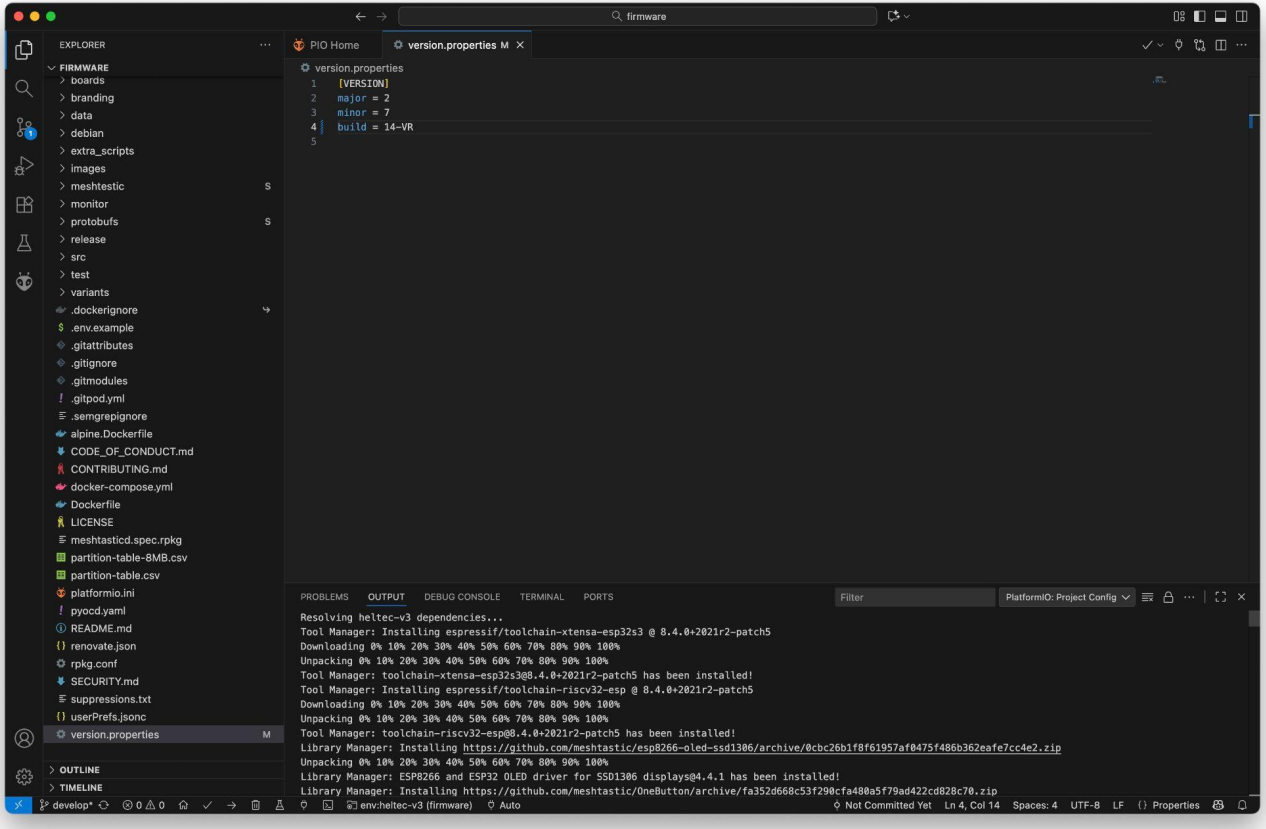
2...

Select Heltec-V3 or your device



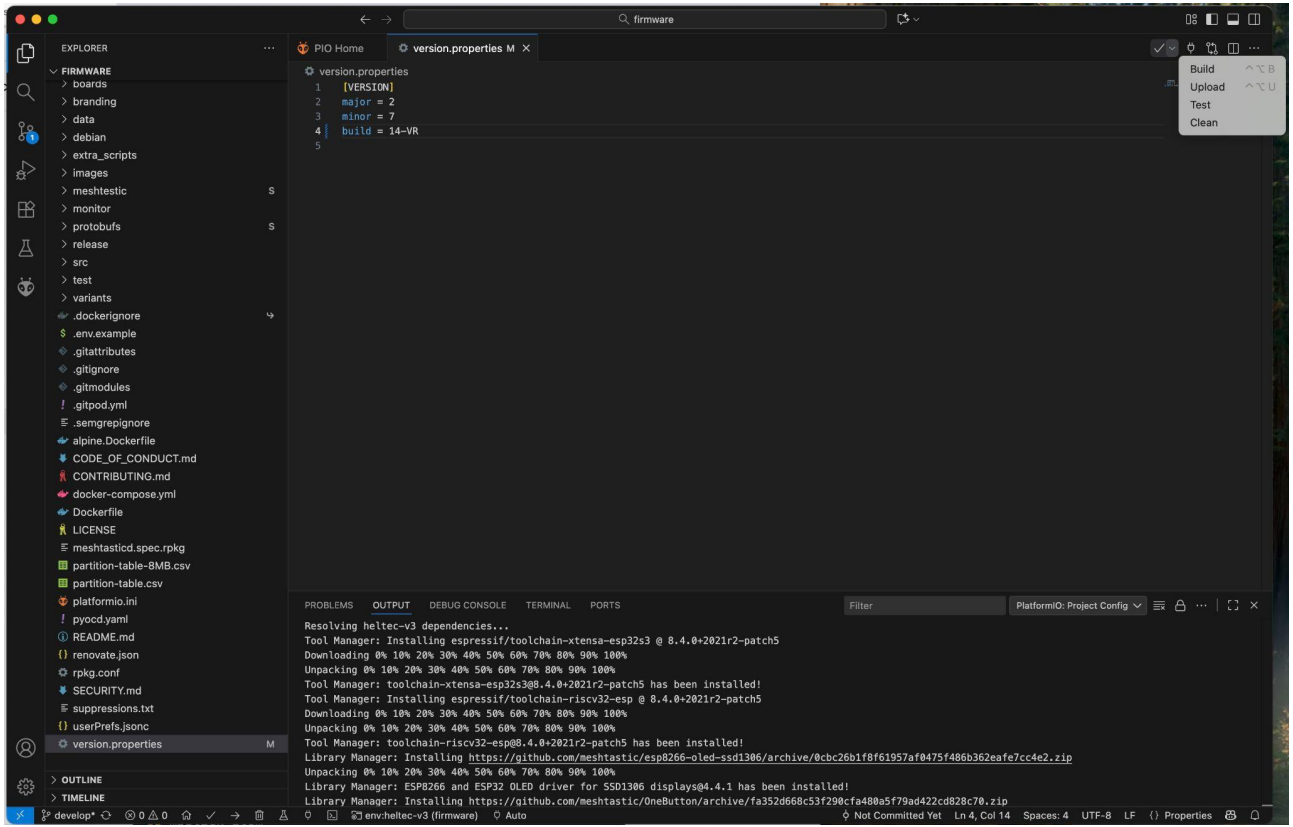
2...

This may take a couple of minutes.



3

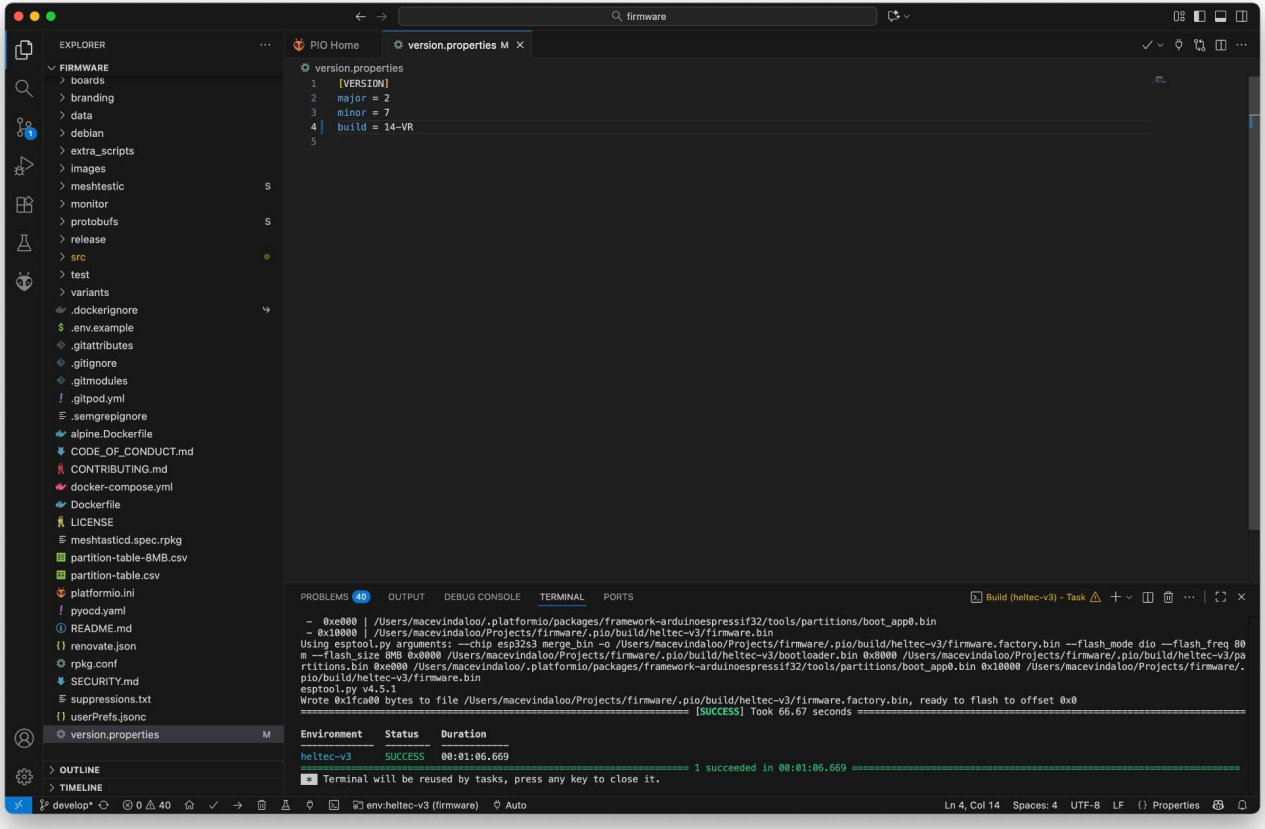
Start the build.



3...

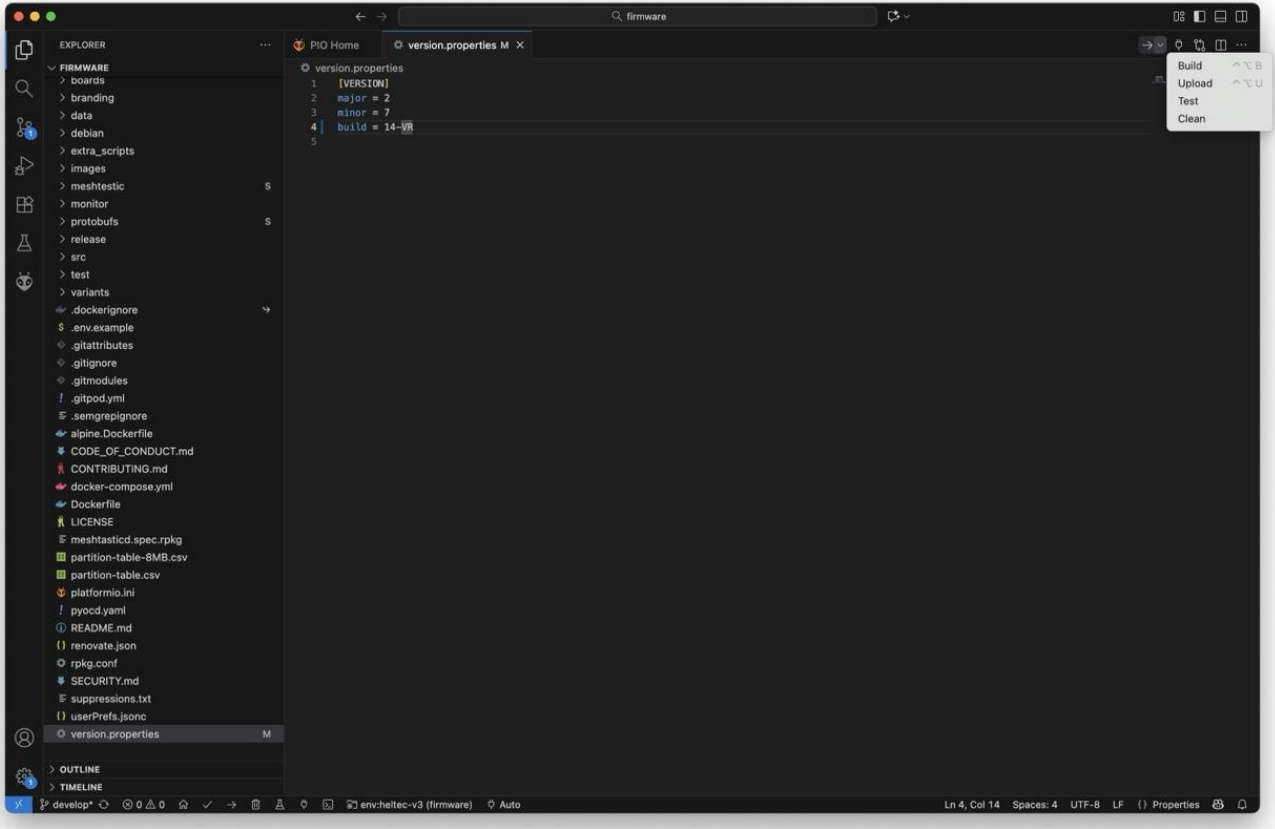
This may take a minute or so.

Congratulations, you have now built your own firmware image!



4

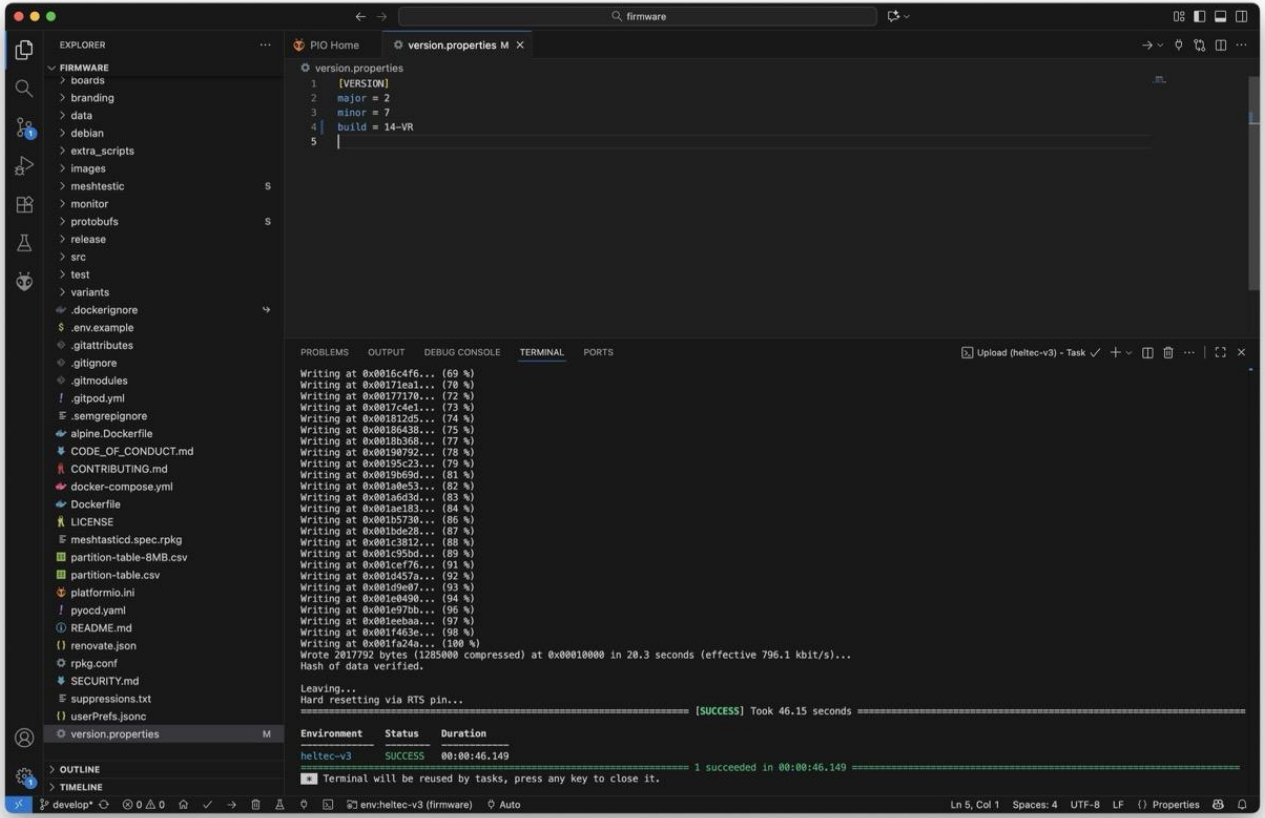
Upload the firmware to your device.



4...

This may take a couple of minutes.

Verify that the display shows your custom version string.



Thank you!

venky.raju@gmail.com