

# Meshtastic for Beginners

Solder your own Cat-themed Meshtastic node!



# What are we learning today?

- Common & custom types of LoRa/Meshtastic hardware
- Legal & practical considerations
- LoRa & Mesh networking fundamentals
- How to set up new nodes for specific use-cases
- Soldering and hardware assembly
- Meshtastic security & privacy considerations

who is kody kinzie and what is their profession

#### Sources

Kody Kinzie Collection - Eventbrite

Kody Kinzie - Varonis | LinkedIn

Kody Kinzie on LinkedIn: A New HOPE (2022)

eventbrite · 1

linkedin · 2

linkedin · 3



View 2 more

#### Answer

- Kody Kinzie is a cybersecurity researcher based in Kalispell [1](#) [3](#).
- He has expertise in Wi-Fi security and low-cost cyber weapons [1](#).
- Kody Kinzie is a cybersecurity professional, OSINT analyst, and researcher. He also has a background in the USAF [3](#).
- Additionally, he is known for his active engagement in cybersecurity research, as evidenced by his presence on YouTube [5](#).

who is Kody Kinzie?

#### ChatGPT

- Kody Kinzie is a **security researcher** specializing in open-source intelligence and Wi-Fi security.
- He shares his knowledge through **two YouTube channels**, Hak5 and Null Byte, teaching cybersecurity to beginners.
- Organizes **cybersecurity training and outreach** events in Los Angeles.
- Focuses on **low-cost hacking tools** and has a strong background in teaching various security topics.



Fireside Cyber Chat Kody Kinzie 4:30PM EST 23 Feb 2022 Watch 🔍

Fireside Cyber Chat Kody Kinzie 4:30PM EST 23 Feb 2022 Watch 🔍

Search Images +

Generate Image PRO +

Hi! I'm Kody Kinzie.

You can find me at  
[www.hack.gay](http://www.hack.gay)

I'm a security researcher who focuses on Wi-Fi and LoRa radios, digital investigations, and electronics.

I love teaching people how to get started with ethical hacking in person and on my YouTube channels:

- Retia
- Null Byte

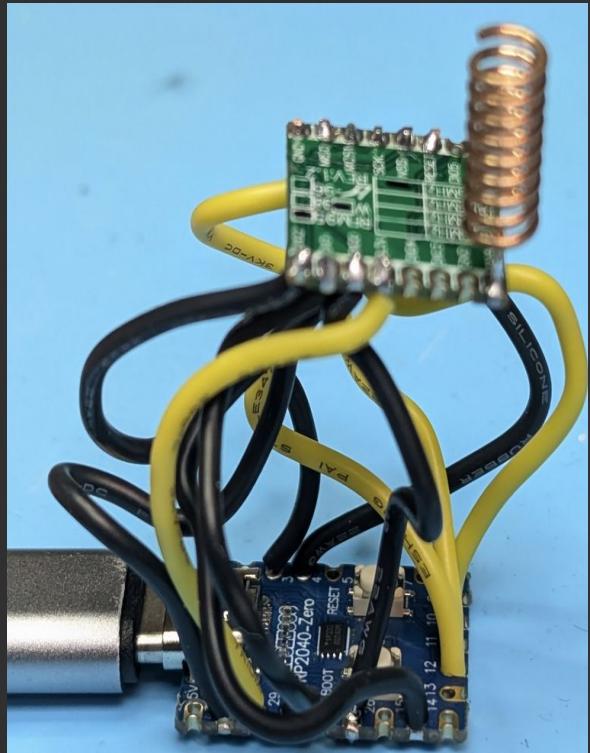
# What is the point of Meshtastic?



- Bidirectional, encrypted off-grid communication over long distances,
- Amplified by nearby nodes, regional networks can span 200+ miles
- Monitor sensors, track objects, control hardware remotely
- Robust communication in areas with congested or nonexistent infrastructure
- Ultra-low cost hardware makes nodes disposable

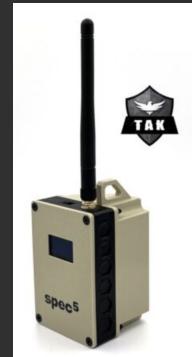
# What is the cost of participating?

- You can make your own horrible node for \$8
- \$20-\$40 for budget nodes
- \$70-\$100 for a fancy node from China
- \$150+ for custom nodes



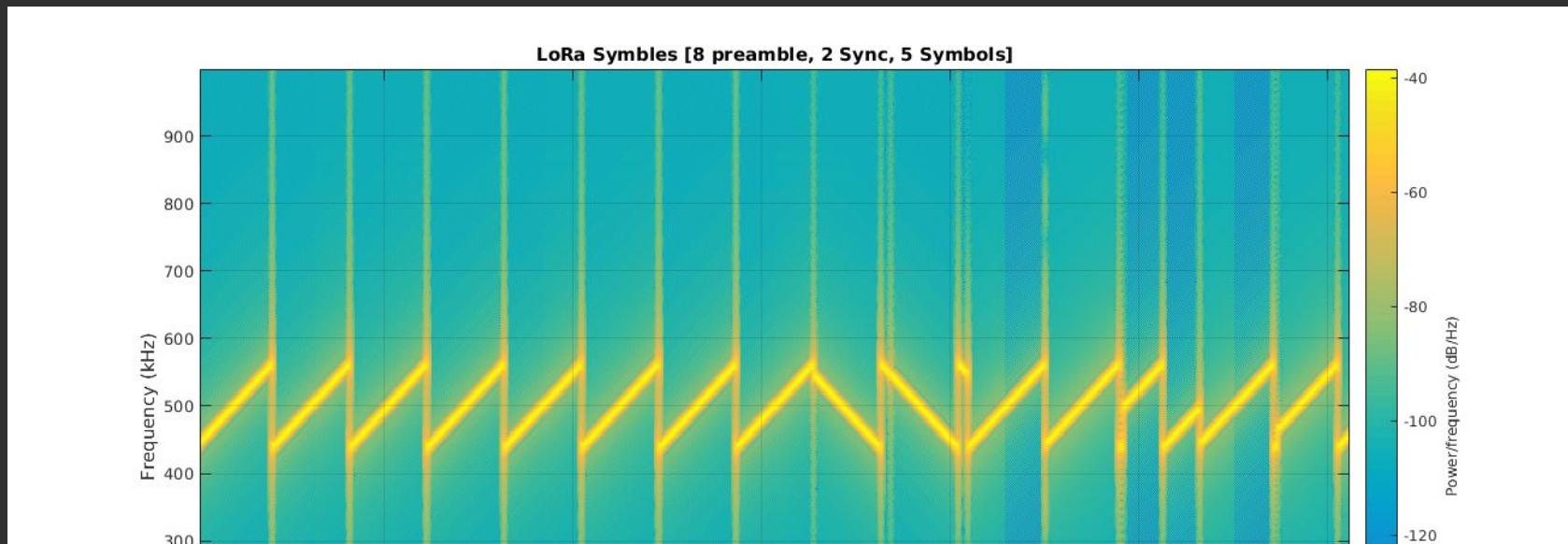
# What can we make with this?

- Off-grid chatting between devices in remote areas (backwoods)
- Communication in areas with congested service (concerts, raves)
- Inter-vehicle communication (vehicles, boats, & aircraft)
- Encrypted disaster/emergency response
- Map and military intelligence sharing
- Remote Sensor telemetry (presence, environmental)
- GPS Tracking of moving nodes
- Remote control of hardware: vehicles, ambient lighting, flashes



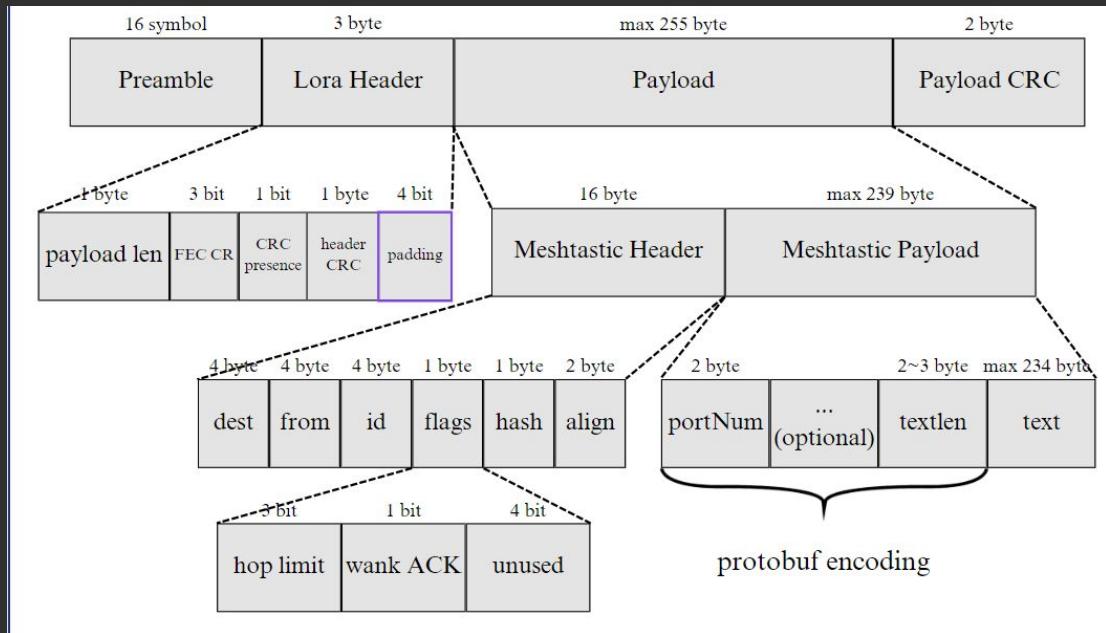
# Defining Terms: LoRa Radio

- LoRa - The long range radio standard we use to transmit, encoding data using up and down chirps



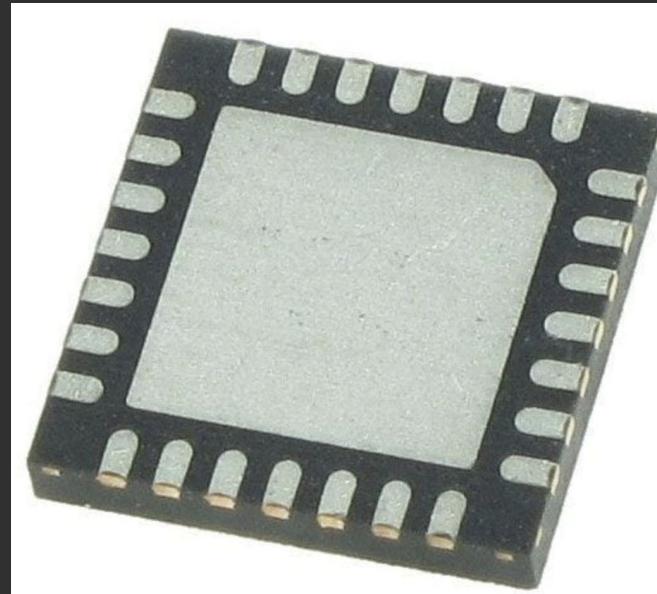
# Defining Terms: Meshtastic

- Meshtastic - A protocol, with packets sent over LoRa, that allows for encrypted mesh networking primarily at Layer 3 (Network Layer) of the OSI model



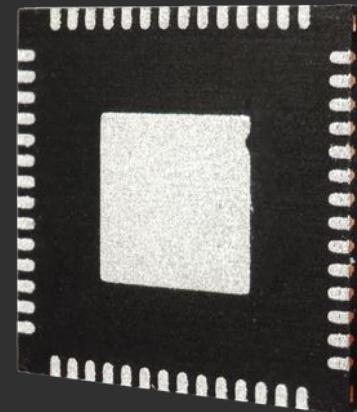
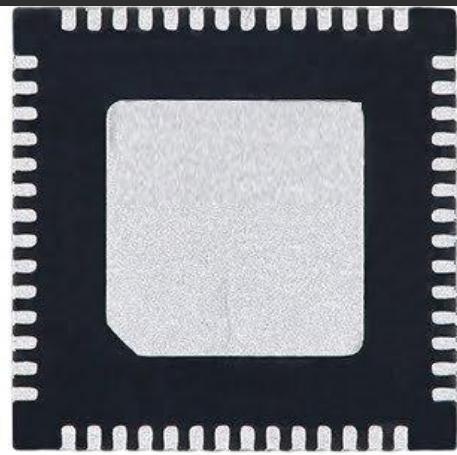
# Defining Terms: LoRa Radio Chips

- **LoRa Chipset** - The LoRa radio chip that controls radio transmission, such as sx1262



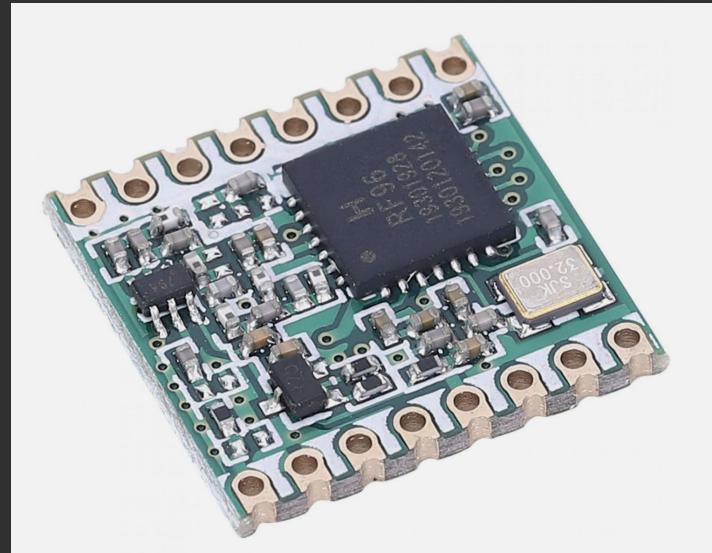
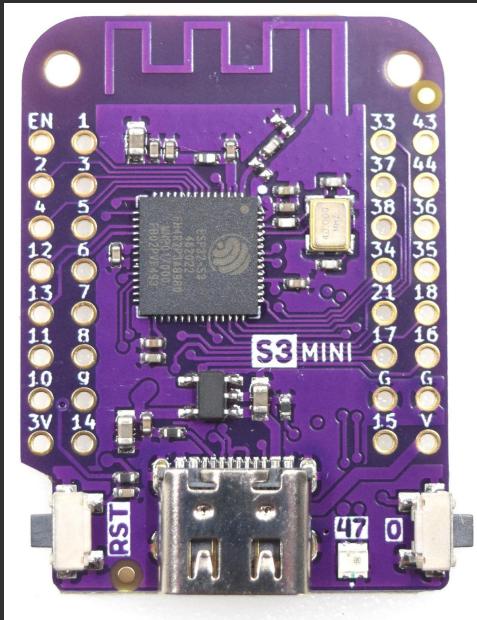
# Defining Terms: Host Microcontroller

- **Host Microcontroller** - The “brains” that run Meshtastic, handle Bluetooth, Wi-Fi, & Serial communication, & interface with external hardware



# Defining Terms: Breakout Boards

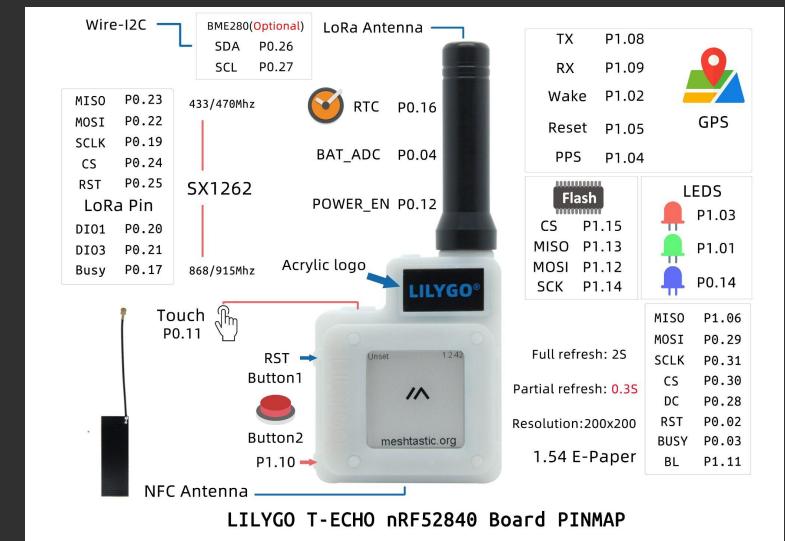
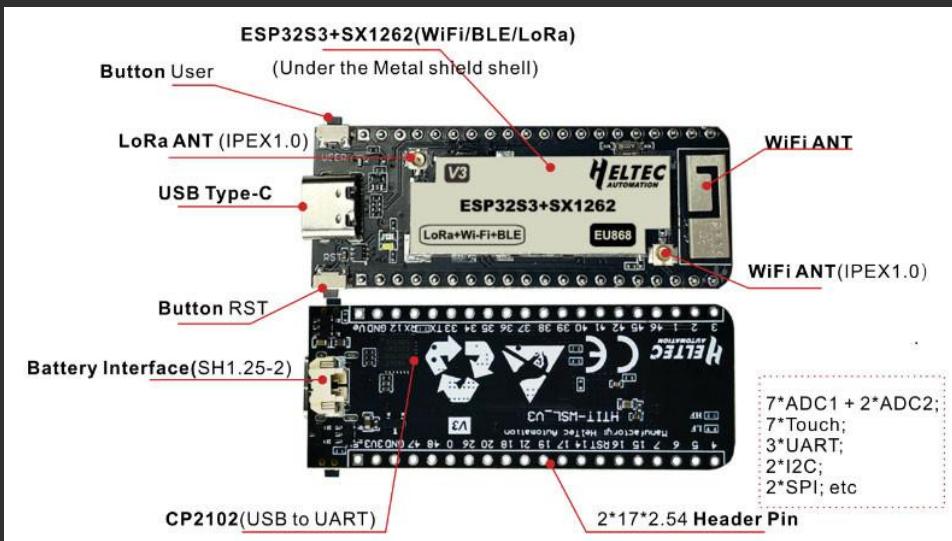
- **Breakout board:** An expansion board designed to make a bare host or radio chip more useful, breaking out pins & sometimes USB & power management



# Defining Terms: Nodes

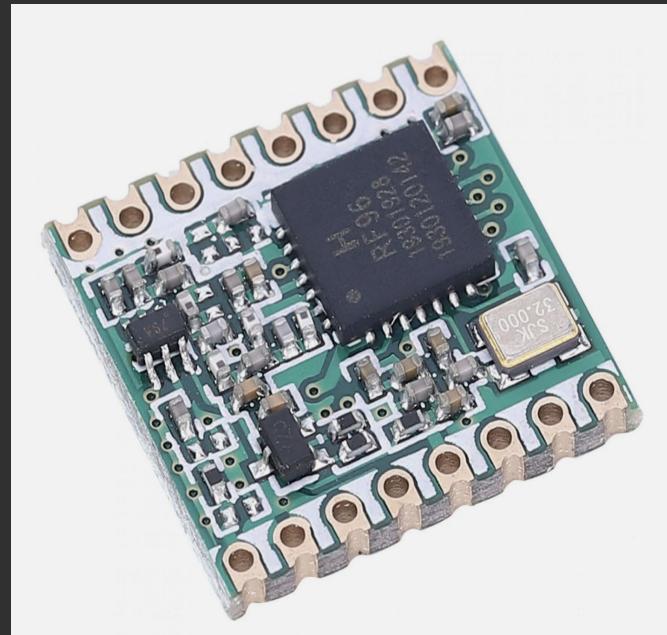
- **LoRa Node:** A combination of LoRa radio & host microcontroller. Together, they can run Meshtastic!

Simple Node <-----> Complex Node

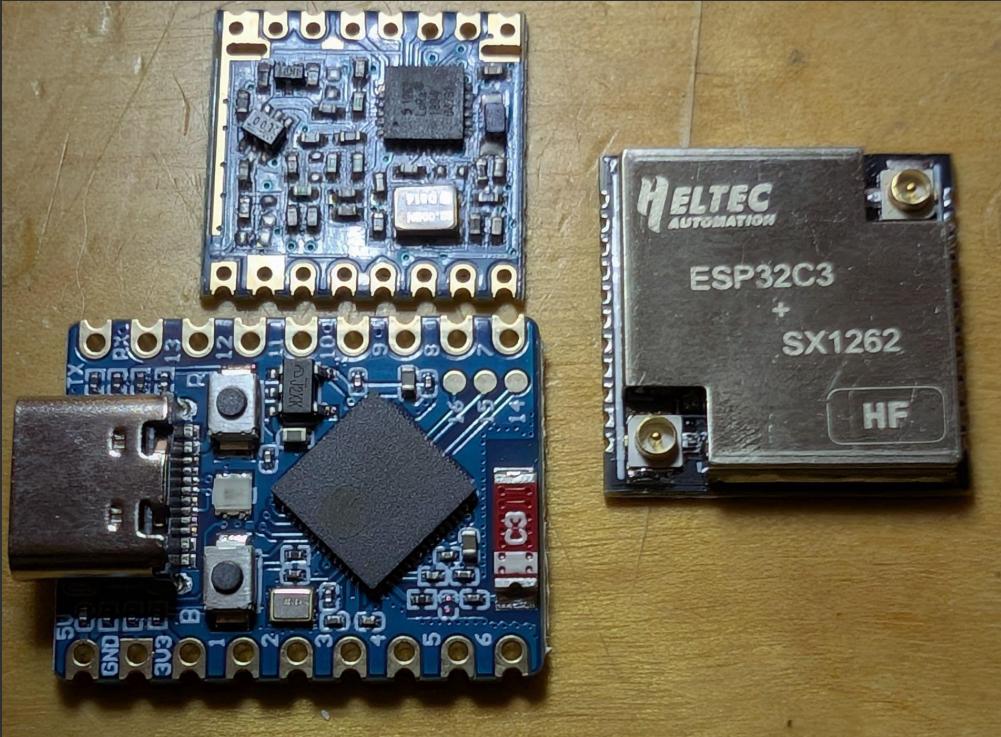


# Host + Radio = Node

**RP2040 Zero (host breakout) + RFM95 (radio breakout) = \$8 Node**



# Future Nodes for \$10



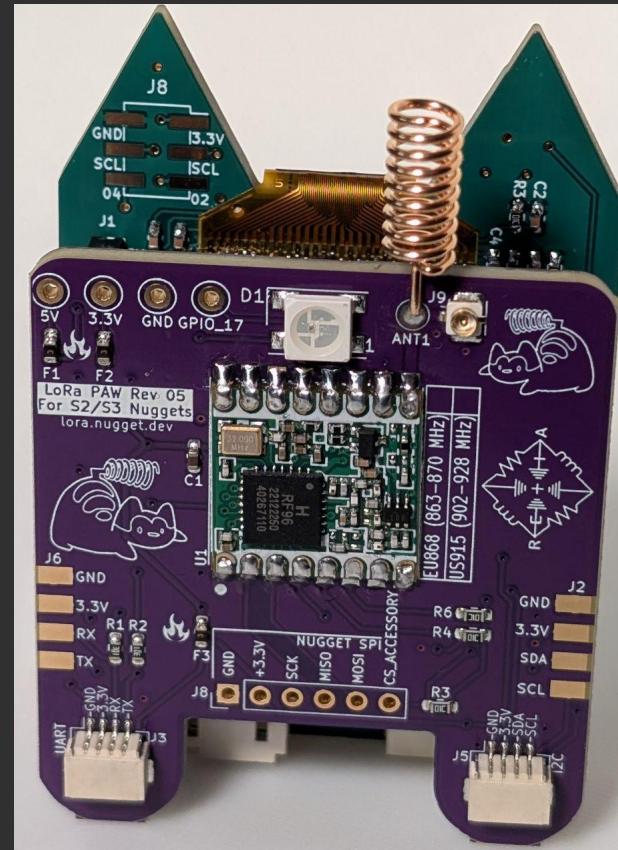
# The Bluetooth Nugget + LoRa Backpack

- The Bluetooth Nugget features an ESP32s3
- Bluetooth, Wi-Fi, native USB, serial
- Quiic connectors for i2c sensors
- 6 buttons for sending and receiving canned messages
- 2 neopixels for status indicators
- OLED i2c screen
- Features Bluetooth, Wi-Fi and support for i2c, UART, and SPI hardware

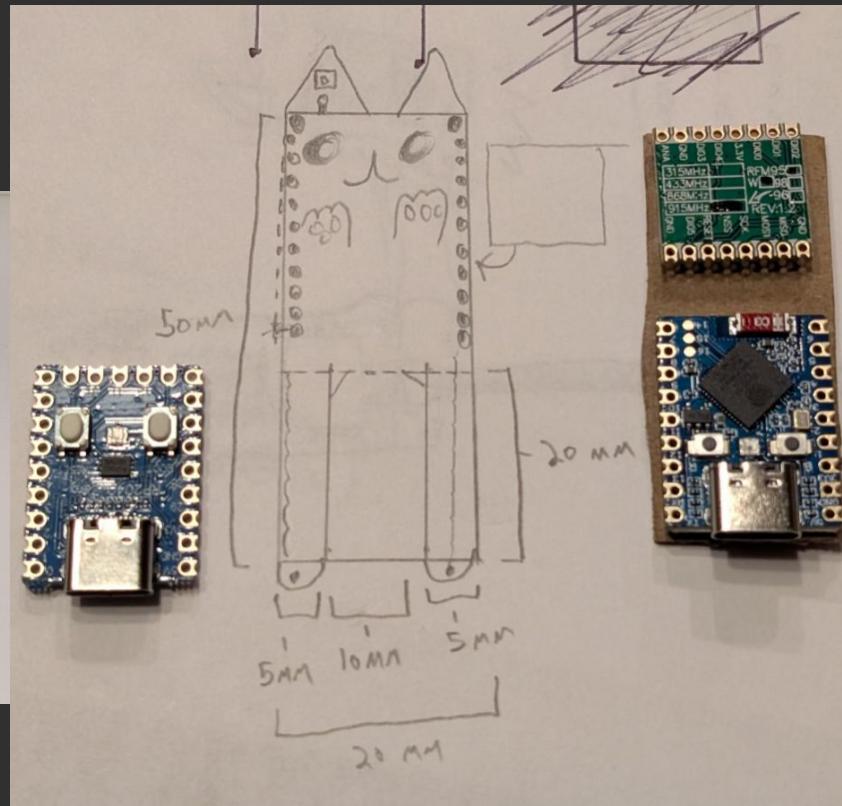
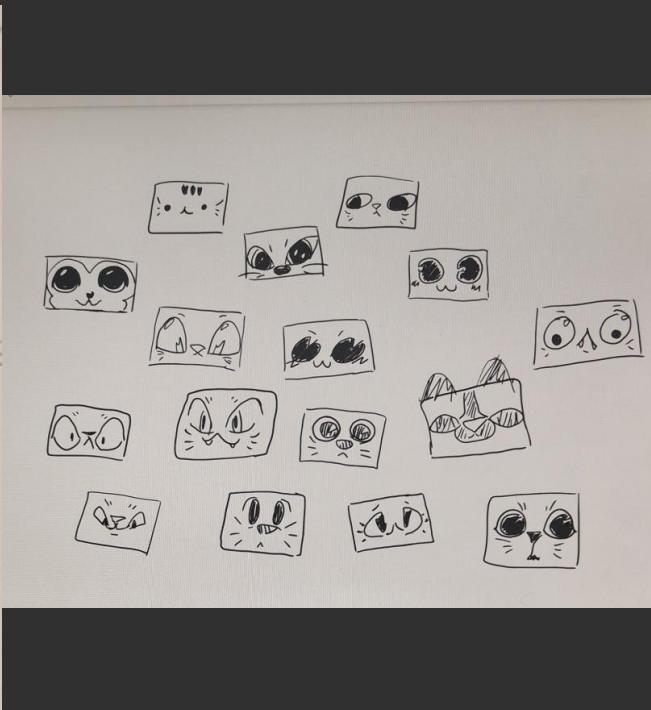
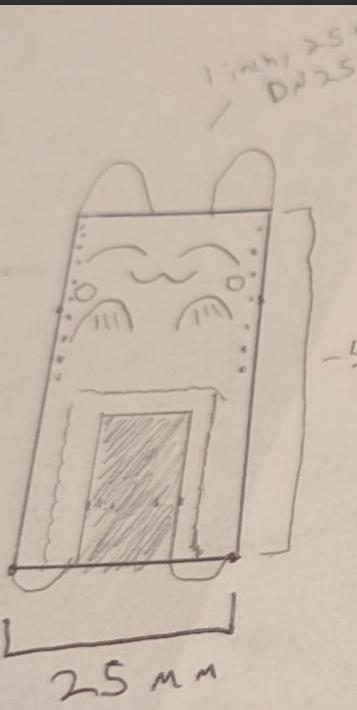


# LoRa Prototyping Powerhouse

- Designed for flexibility in creating custom Meshtastic nodes
- Supports hardware used by Meshtastic's most popular modules
- Powerful and versatile design for learning ethical hacking

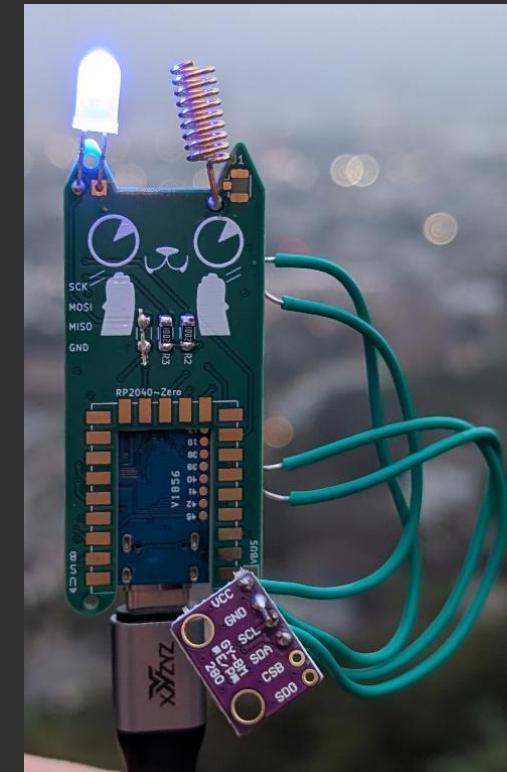


# Making 300 Custom Meshtastic Kits: The Nibble

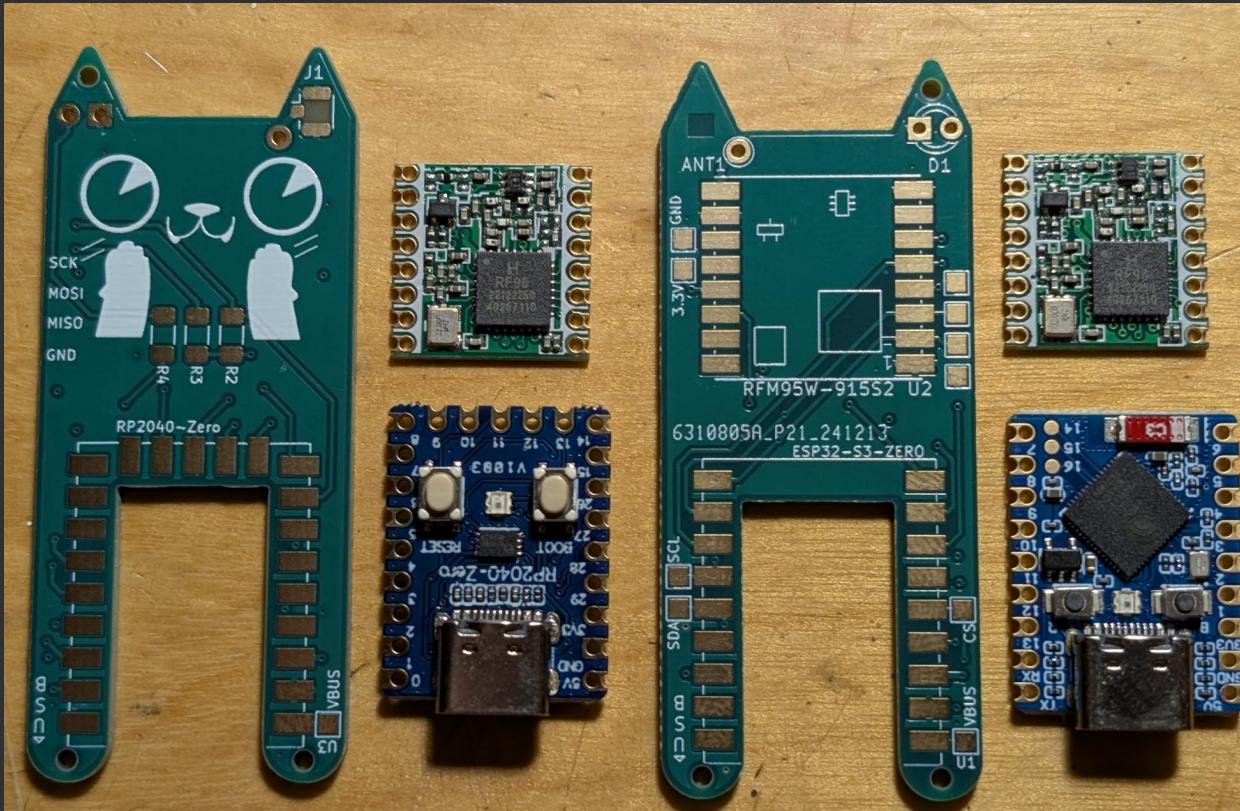


# The Nibble - The Original Cat Meshtastic Node Kit

- Combines RP2040 or ESP32s3 host with sx1276 radio
- Designed to be soldered by beginners
- Pick between RP2040 with serial only, or esp32s3 with Bluetooth & Wi-Fi
- Designed to fit 1 inch PCV pipe for outdoor placement
- Pads for adding SPI or i2c sensor chains like bme280
  
- RP2040 version can't connect to iOS over serial, but serial works on Android, desktop, & webapps
- We made 300 Nibbles at c3!
- Open Sourced at Hackers On Planet Earth



# The Nibble - The Original Cat Meshtastic Node Kit

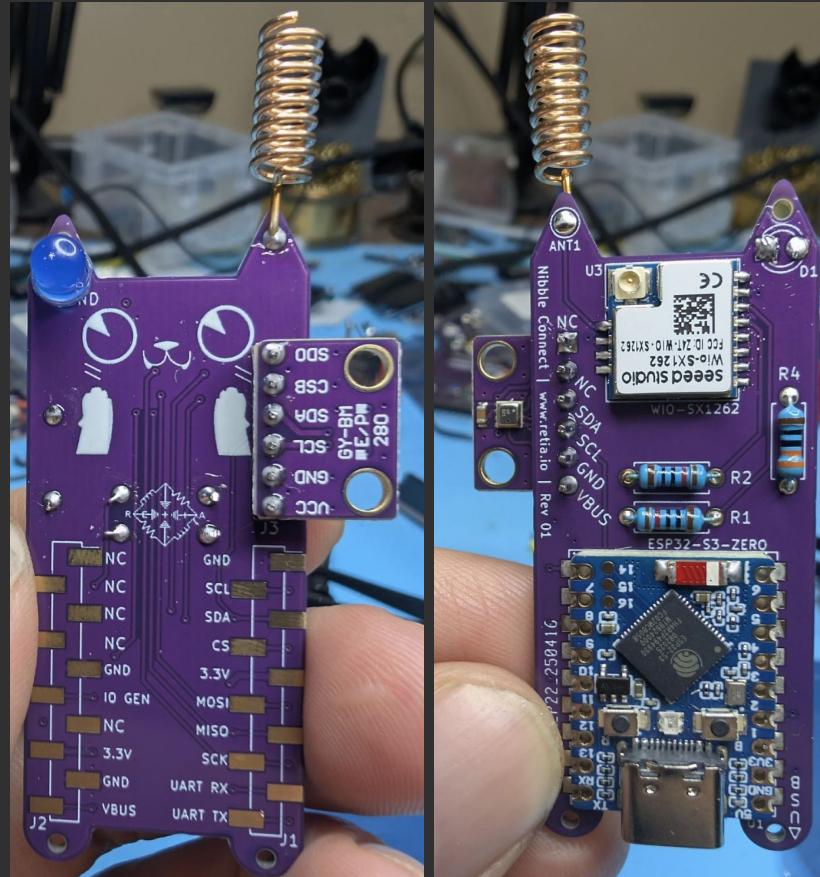


Make your own:  
[github.com/retiallc/nibble](https://github.com/retiallc/nibble)

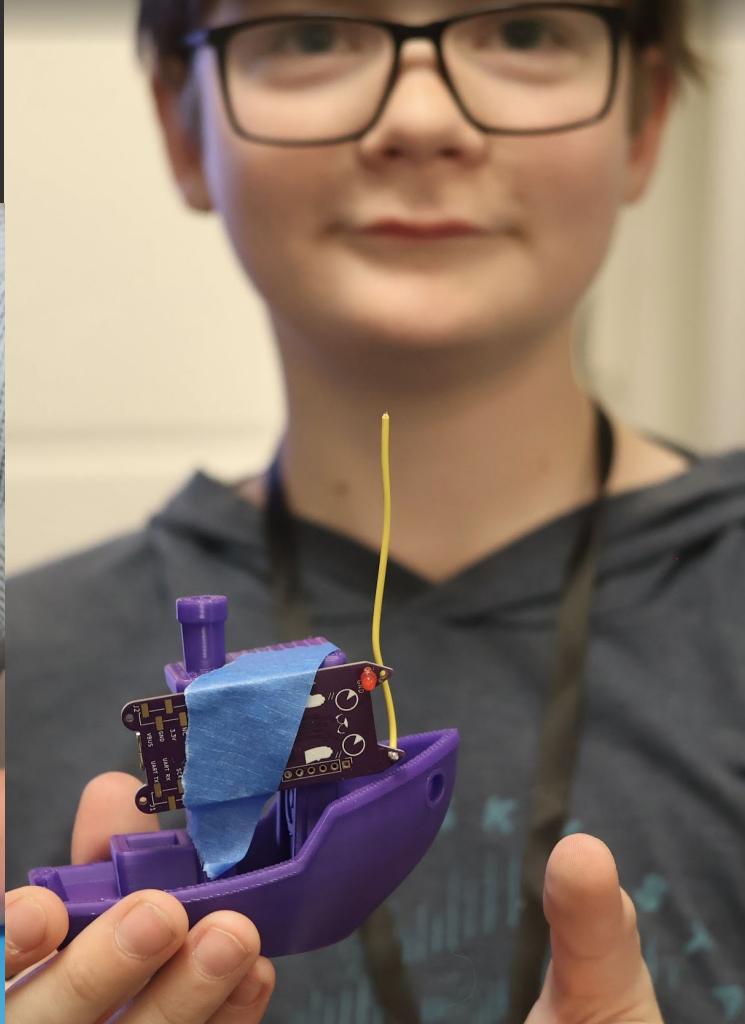
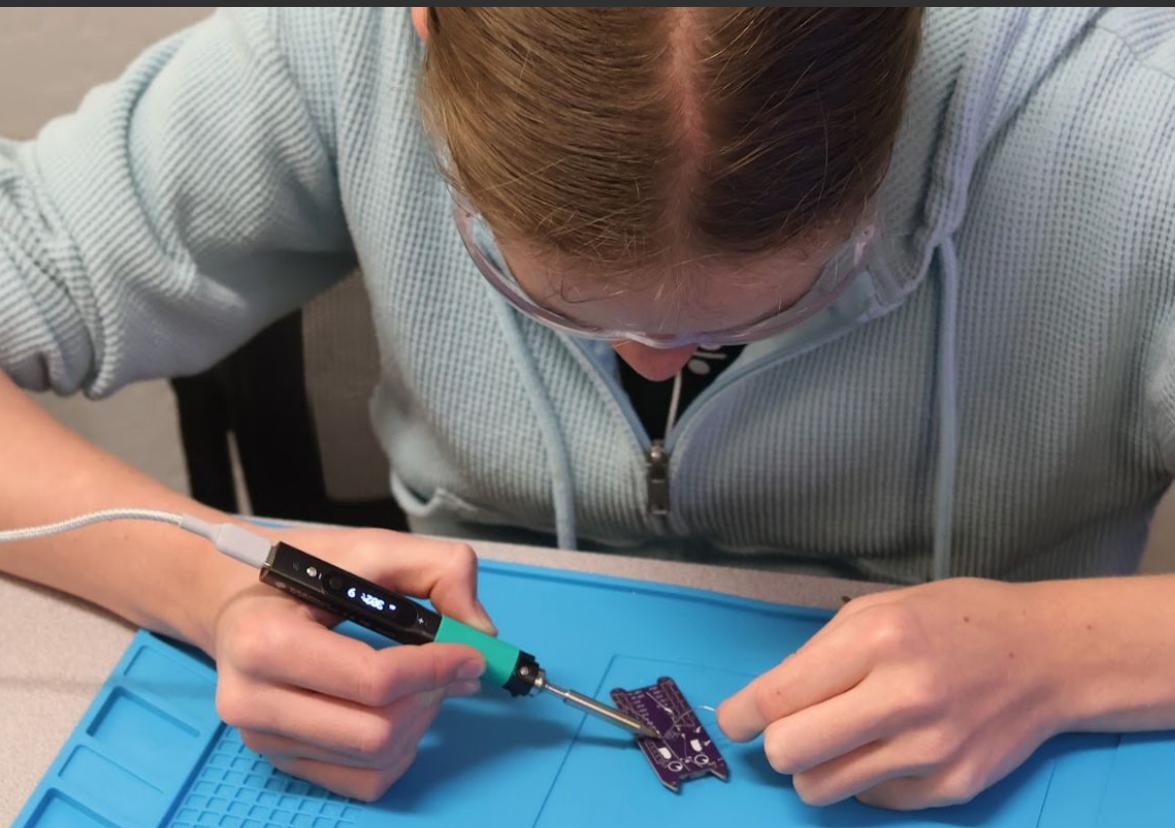
# Update: The Nibble Connect

Based on feedback from C3, we created the Nibble Connect for our kids cyber camps:

- Selected ESP32s3 Microcontroller (no more rp2040)
- Upgraded to Seeed sx1262 LoRa Radio
  - software definable frequency & more sensitive
- SMD Female Pin Headers (not very breadboard friendly)
- Mount for BME280 weather station sensor



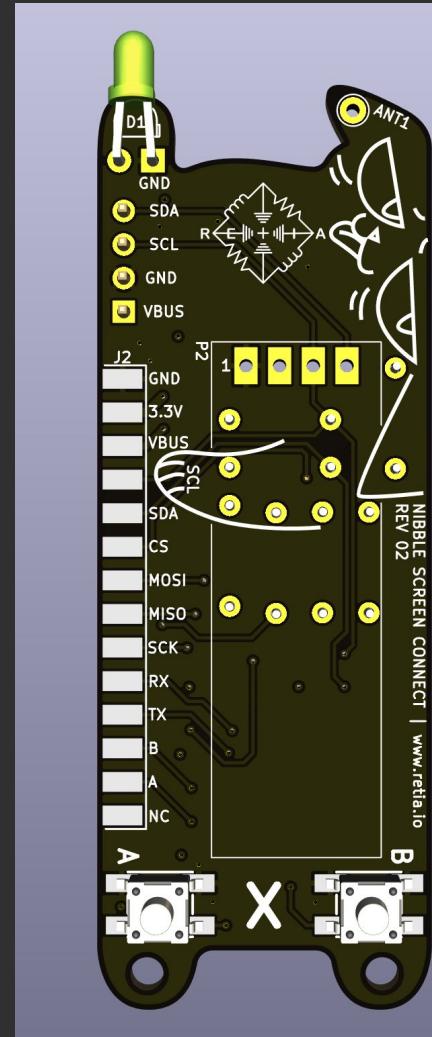
# Update: The Nibble Connect



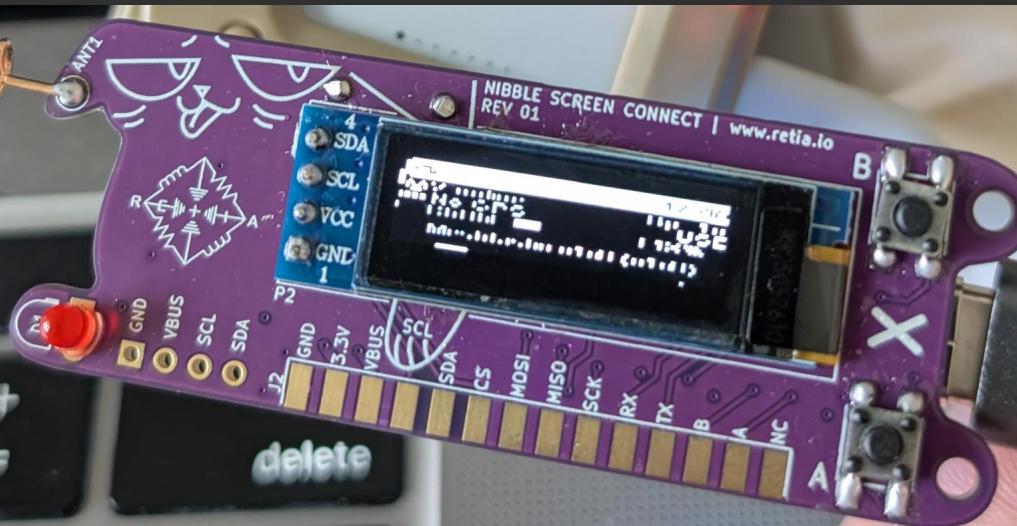
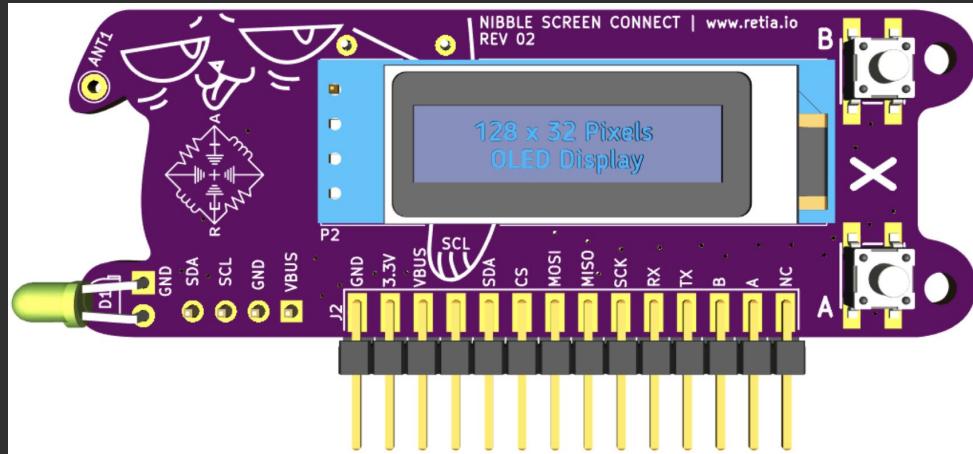
# For Sec-T: New Nibble Screen Connect

We asked - what about a modular DIY node that can be scaled up or down, & use all the features of Meshtastic?

- Also uses ESP32s3 & sx1262 LoRa Radio
- 2 buttons for menu select & sending canned messages
- I2c mount for adding OLED screen
- Sideways pin headers for adding to breadboard
- Minimal mount for BME280 weather station sensor
- **He's a cat lying down on his side for belly scratches**



# Fun With Hardware Design



- Meshtastic doesn't support this tiny screen
- First version i2c pins weren't routed
- No radio keep out area in first revision

# ...But any OLED should do

Meshtastic automatically detects most OLED displays you can add, just make sure the pinout matches.

I brought a limited number for sale!

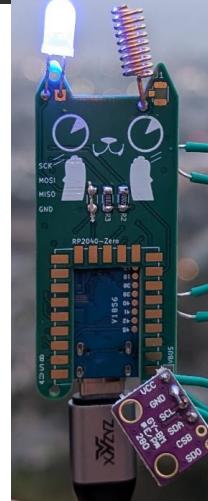
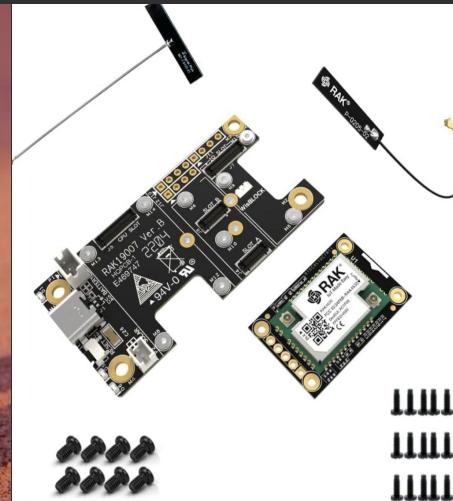


# What other devices can I use?

Here are some of the usual suspects:

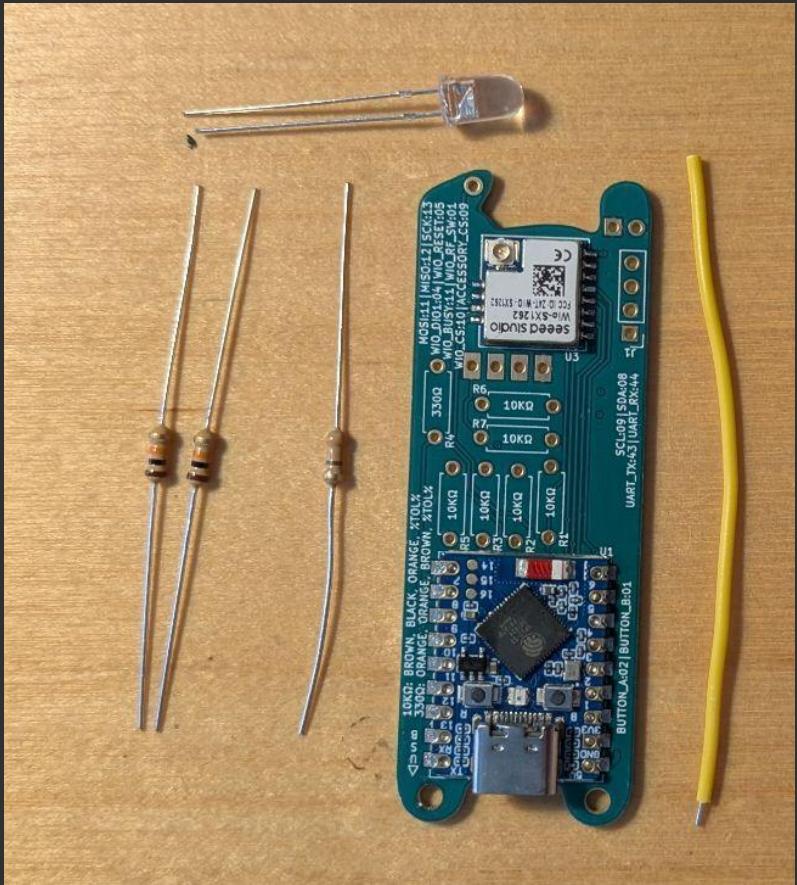
Heltec v3, Lilygo T-Deck, Rak Wireless Wizblock,

Lilygo T-Deck, Nibble, Nugget

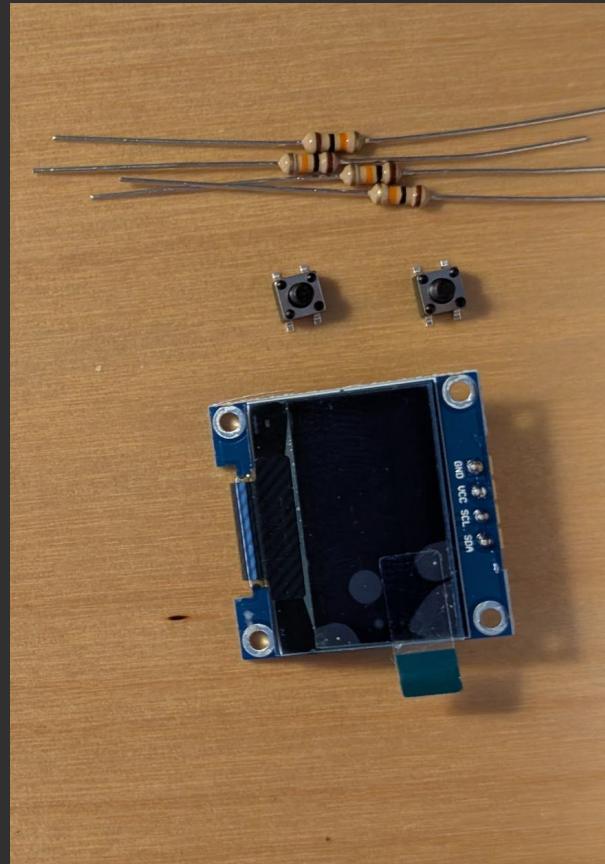


# BREAK - Let's solder!

# The Kit



Mandatory to make the node work



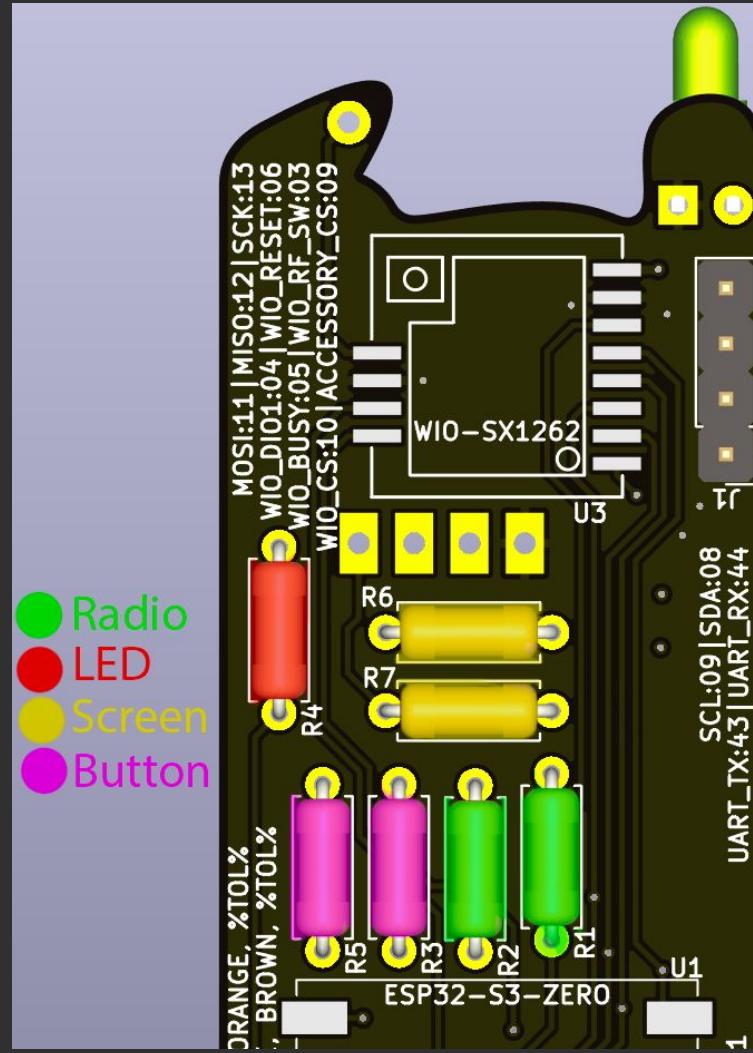
Optional to add screen

# Let's place resistors!

Resistors decrease the voltage in a circuit, we'll be soldering on 7 total!

If you're not adding a screen or buttons, you'll only need 3 resistors total.

You can see here what resistors you must add to enable different features.

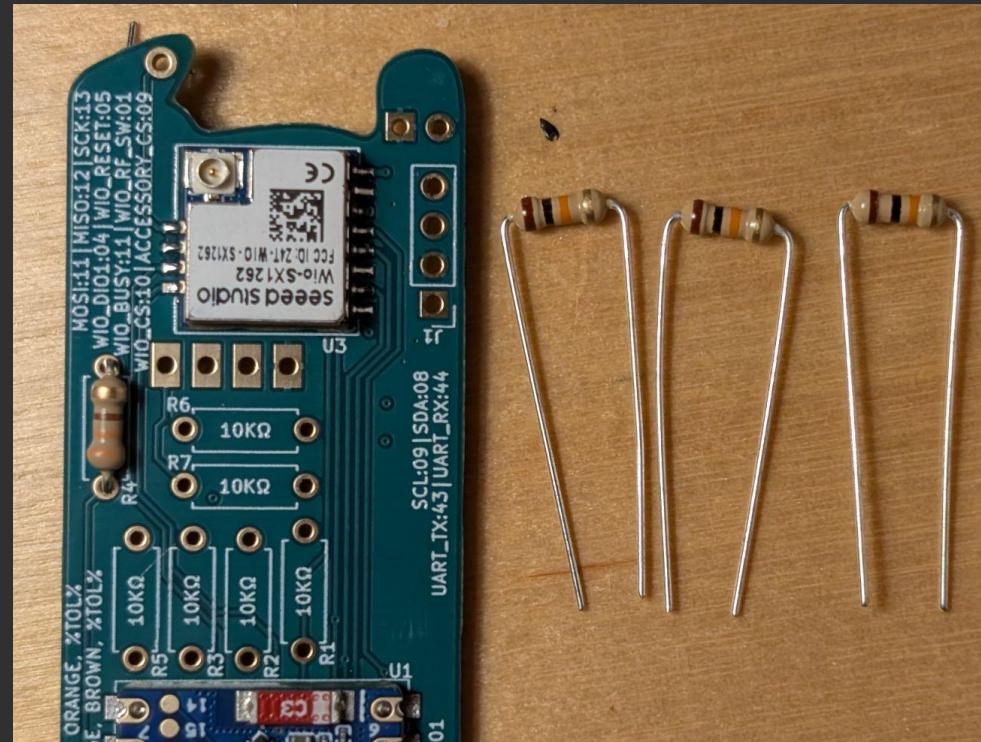


# Solder first resistor

First, place the single 330 ohm resistor in the spot marked R4.

You can bend the leads out on the other side to keep it in place.

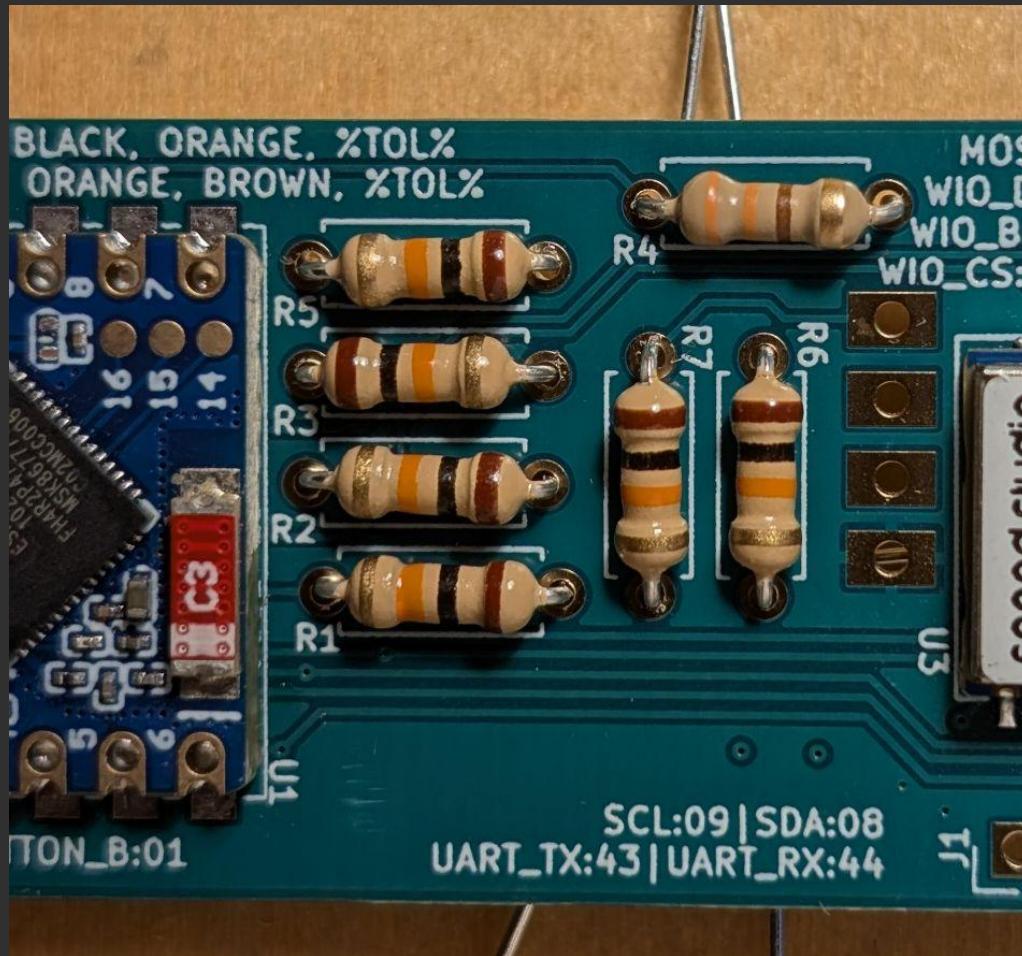
This is omnidirectional - don't worry about placement (just make sure it's on the correct side)



# Add 10k Resistors

Next, we'll place and solder the 10k resistors in the remaining spots.

Not using a screen and buttons? You only need to add 2 10k resistors, in spots R1 and R2.



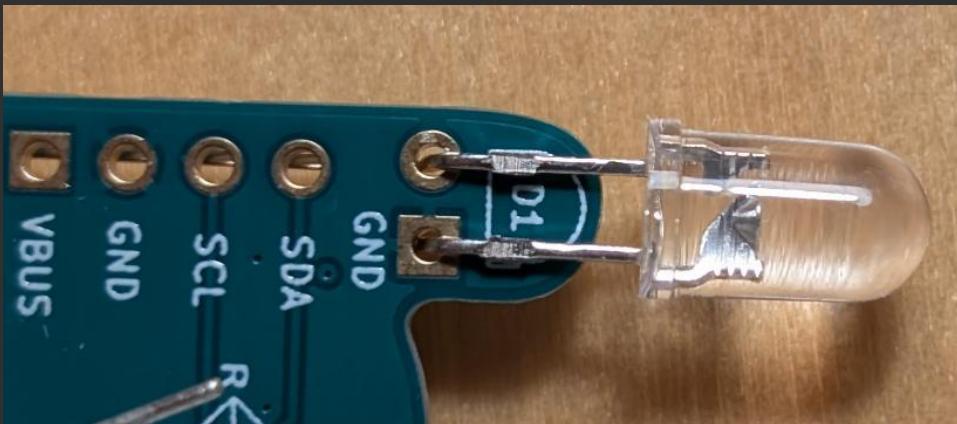
# LED

\*\*\*STOP!!!\*\*\*

This part is directional, do not solder it on backwards!

Identify the GROUND pin of your LED, and place it into the square hole like below.

Here are some ways to find the GROUND pin:

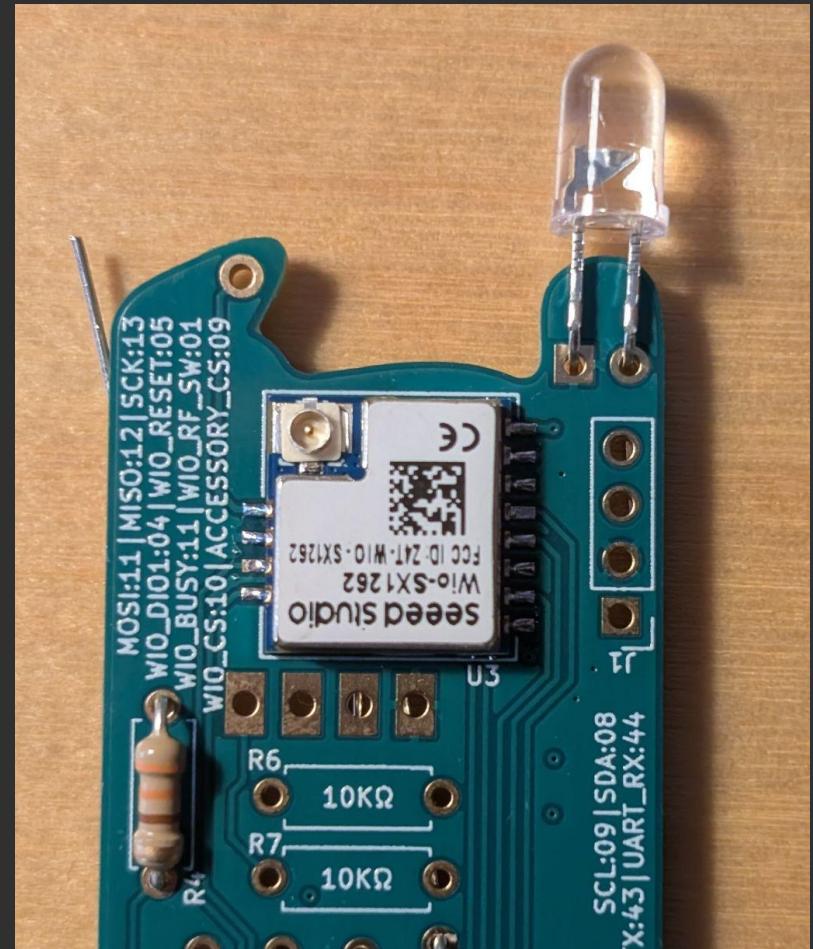


# If you're not sure, ask!

Please ask before soldering in this part if you're not sure!

You can add the LED to the front or back, as long as the ground pin is in the square hole.

Most people leave some space to bend the LED up (do this if you plan on adding a weather sensor)



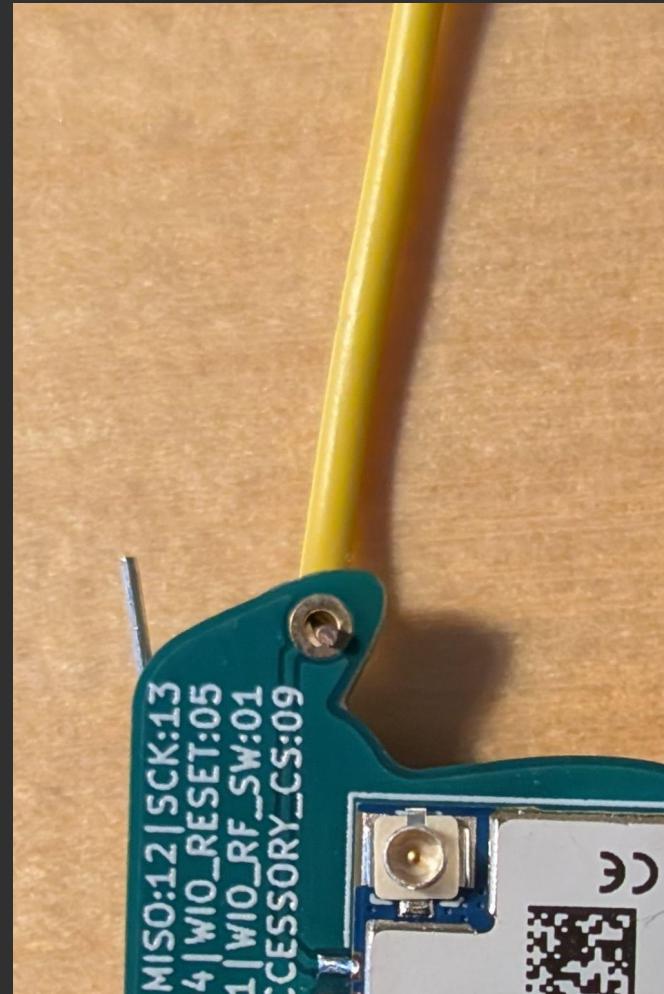
# Add An Antenna

Add your wire antenna to the top hole marked “ANT1”

Bend the end of the antenna so the exposed wire pokes through the other side.

Solder it into place.

If you're not adding buttons and a screen, you're done. Skip the next two steps.

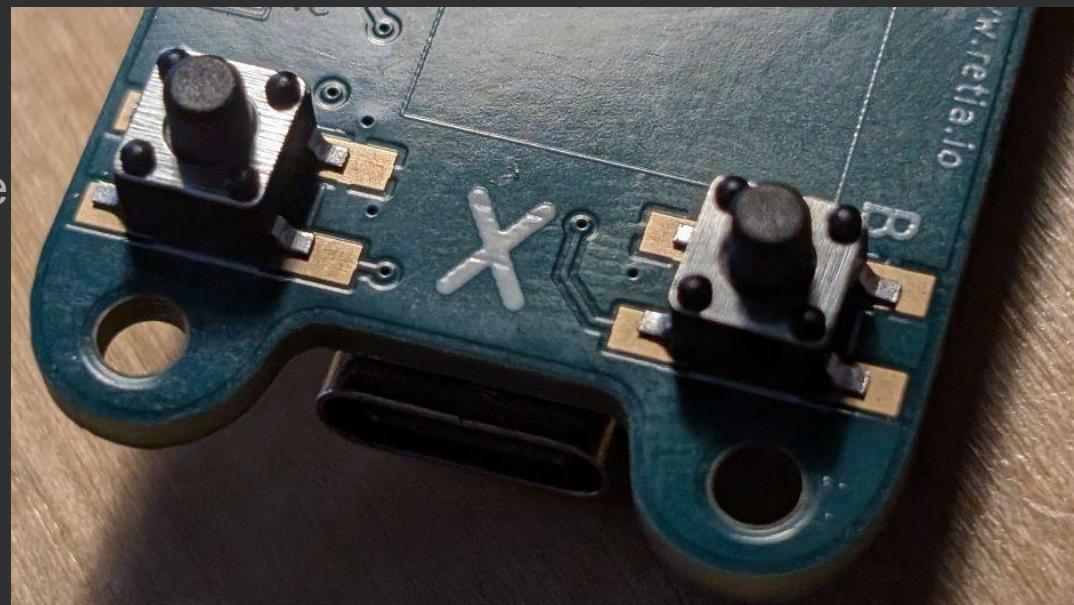


# Optional - Buttons

Solder the two buttons on the front side by the Nibble's feet.

It's easiest to add some solder to one of the pads, place the button, and solder it into place so it doesn't move.

These are omnidirectional, so orientation works.



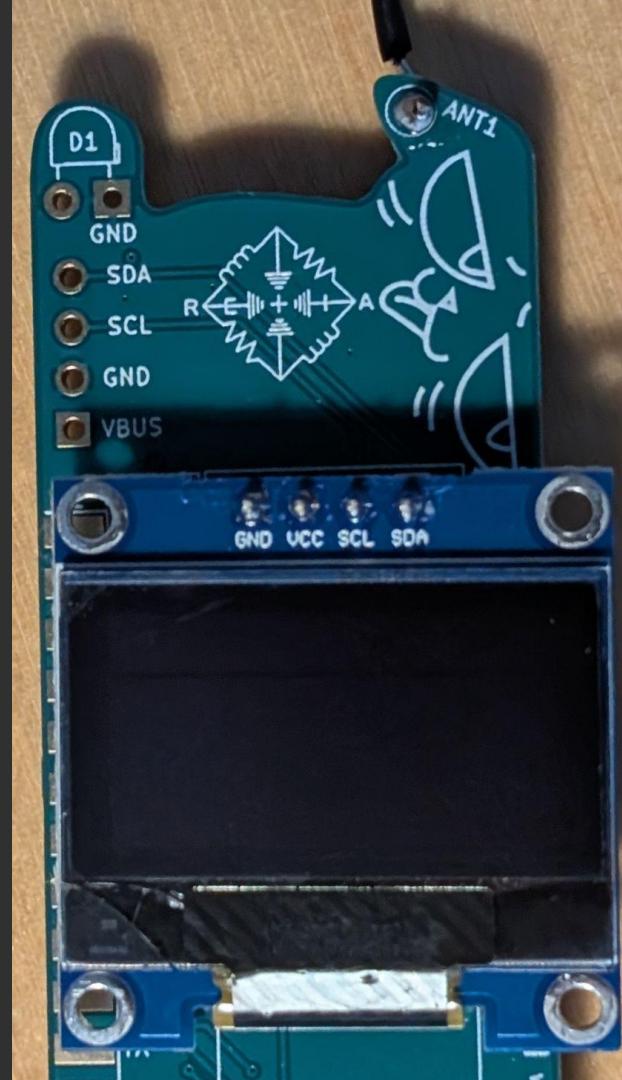
# Optional - Screen

\*\*\*STOP!!!\*\*\* This part is directional and hard to fix.  
Do not solder it on backwards.

Finally, add the screen to the front!

You can add some foam tape under the screen to hold  
it in place and insulate it electronically

Ask before soldering in this last part!



# Optional - Weather Sensor

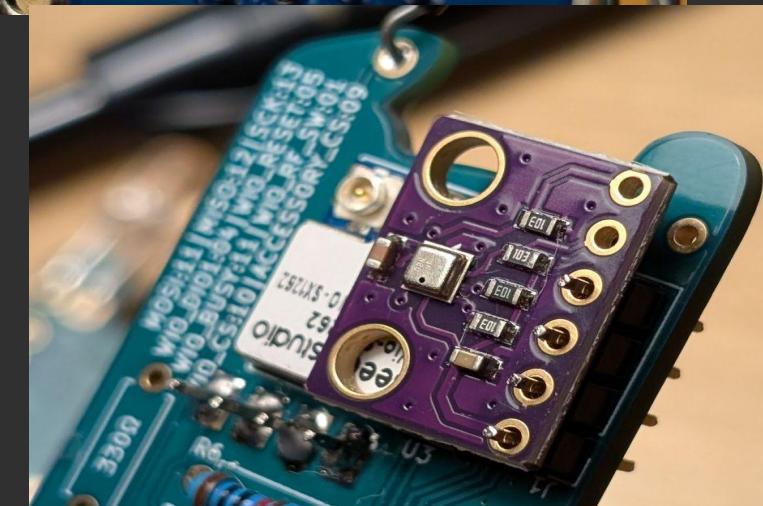
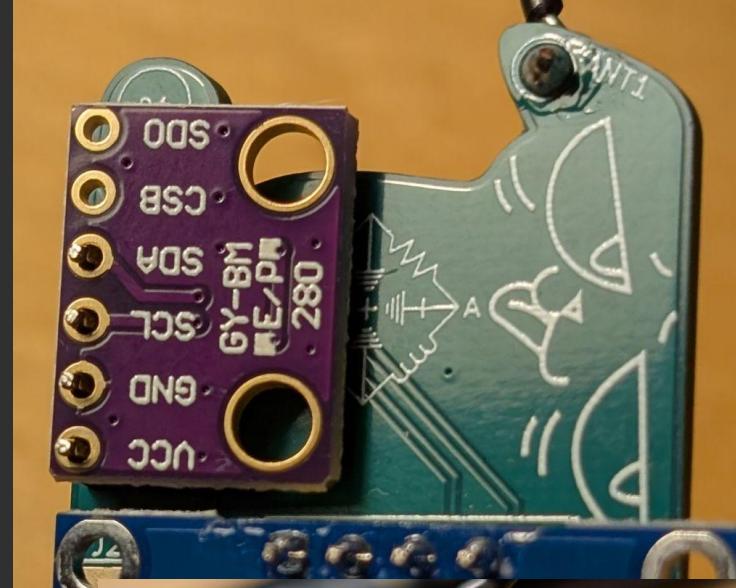
\*\*\*STOP!!!\*\*\* This part is directional.  
Do not solder it on backwards.

If available, solder on the BME280  
temperature, humidity, & pressure sensor.

Make sure to match the VCC, GND, SCL,  
and SDA pins with the board!

The CSB & SDO pins are left unsoldered.

Here is the correct placement on the front  
& back:



# Your Nibble is complete!

Next, we'll flash your Nibble with the latest firmware.

You can download the BIN file here:

<https://github.com/RetiaLLC/sec-t>





## Nugget Dev Suite

Developer Site for the Nugget: A Cat-Shaped Hacking Tool

CONNECT YOUR NUGGET

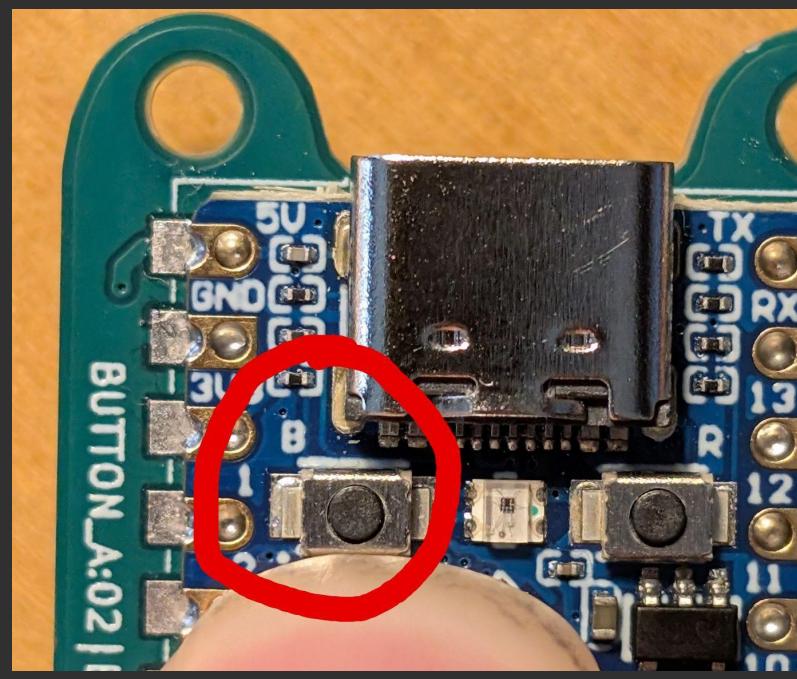
# Chrome-Based Flashing

In a chrome based browser, go to:

NUGGET.DEV

Hold down the “B” button while plugging your Nibble into your computer via USB type C

Once plugged in, click “Connect your Nugget”



# Flashing is easy!

Select your Nibble from the drop down menu of serial devices.

Next, select “**upload a custom binary**”

Select the  
“NIBBLE\_CONNECT\_MESHTASTIC.bin”  
file you downloaded from Github

Click program, unplug when complete!

nugget.dev wants to connect to a serial port

cu.BLTH

cu.Bluetooth-Incoming-Port

cu.JLabStudioProANC

cu.Q158E23E0160036

USB JTAG/serial debug unit (cu.usbmodem14401)



Cancel

Connect



## USB Nugget Flasher

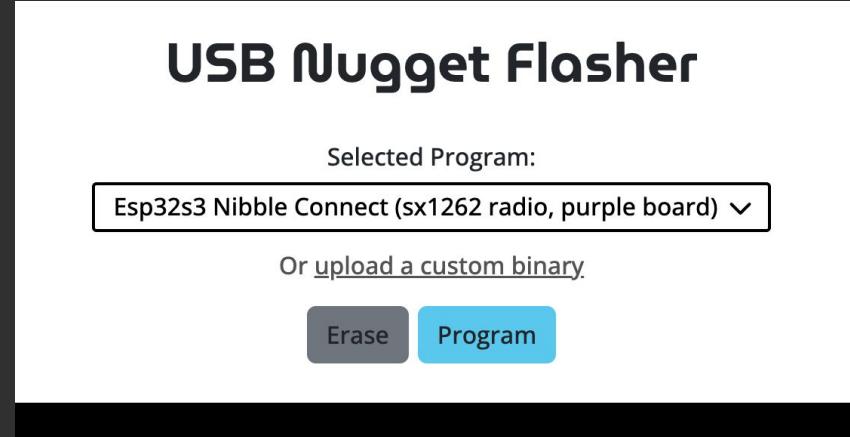
Selected Program:

Esp32s3 Nibble Connect (sx1262 radio, purple board) ▾

Or upload a custom binary.

Erase

Program



# Check for Heartbeat

When you plug your Nibble back in, it should start up and blink its LED after several seconds.

If you have a screen, it should turn on.

It will NOT transmit until you select the correct region, so let's do that!

In a chrome based browser, go to  
<https://client.meshtastic.org/>

**Connected Devices**

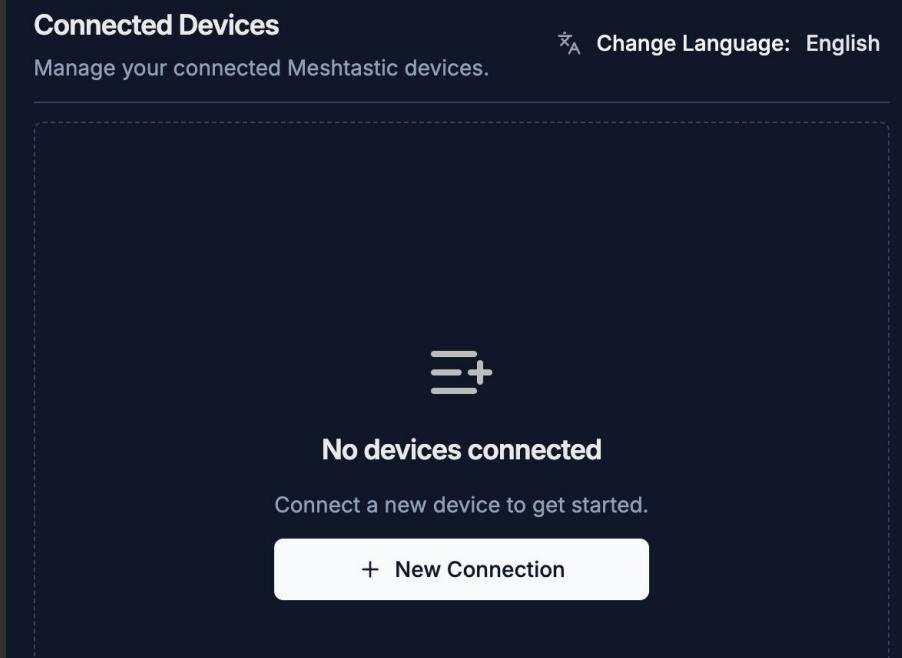
Manage your connected Meshtastic devices.

↗ A Change Language: English

No devices connected

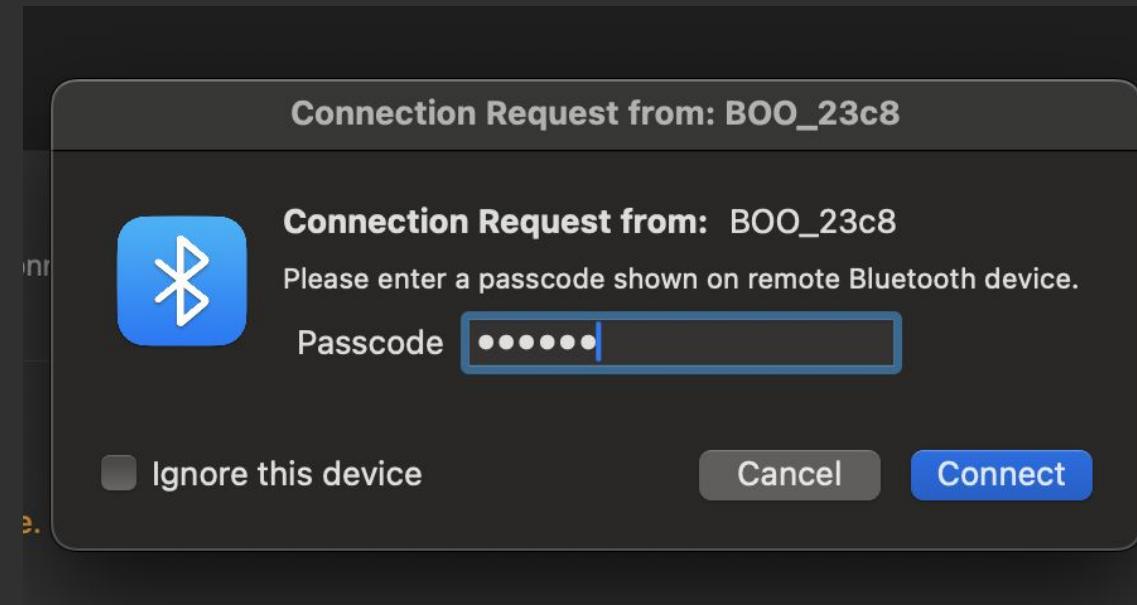
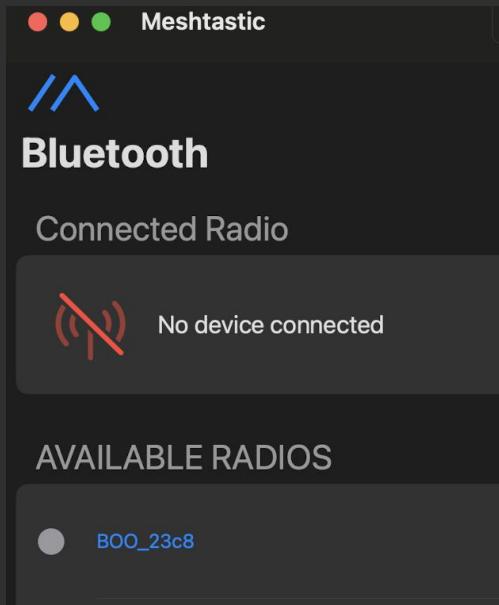
Connect a new device to get started.

+ New Connection



# Alt: Connect via Bluetooth with Smartphone App

- Download the Meshtastic smartphone or Desktop app
- Look for your node short name via Bluetooth
- Pair using the pin: 123456



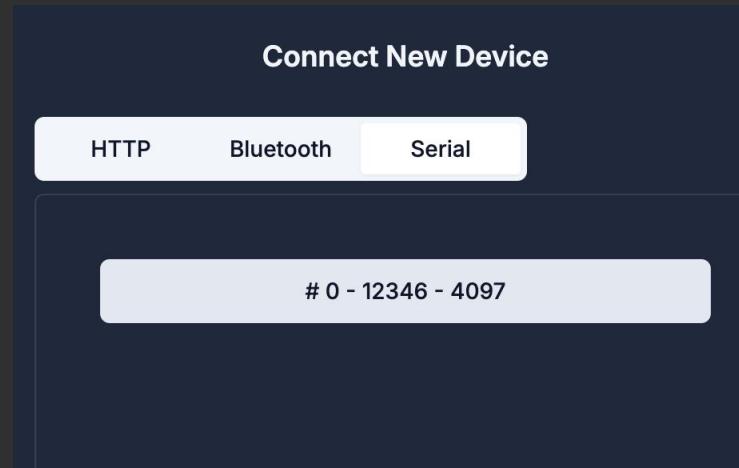
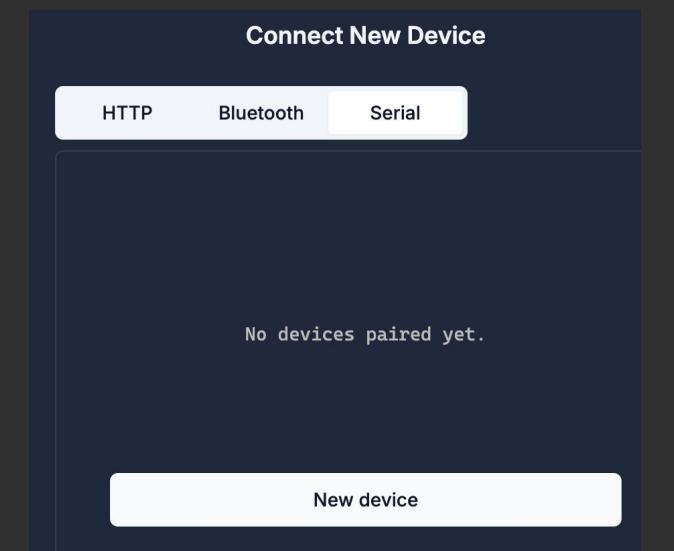
# Connect Via Browser

Click on “New Connection” and select “Serial” from the menu of options.

You can also try Bluetooth here if you want.

Click “New device” and select your Nibble the same way you did to flash it.

It should appear as #0, click it to enter the main menu!



# Change Device Name

At the main menu, click “Change device name” and add a new name.  
You can use emojis!

After you save, it will reboot with the new name.

**Change Device Name**

The Device will restart once the config is saved.

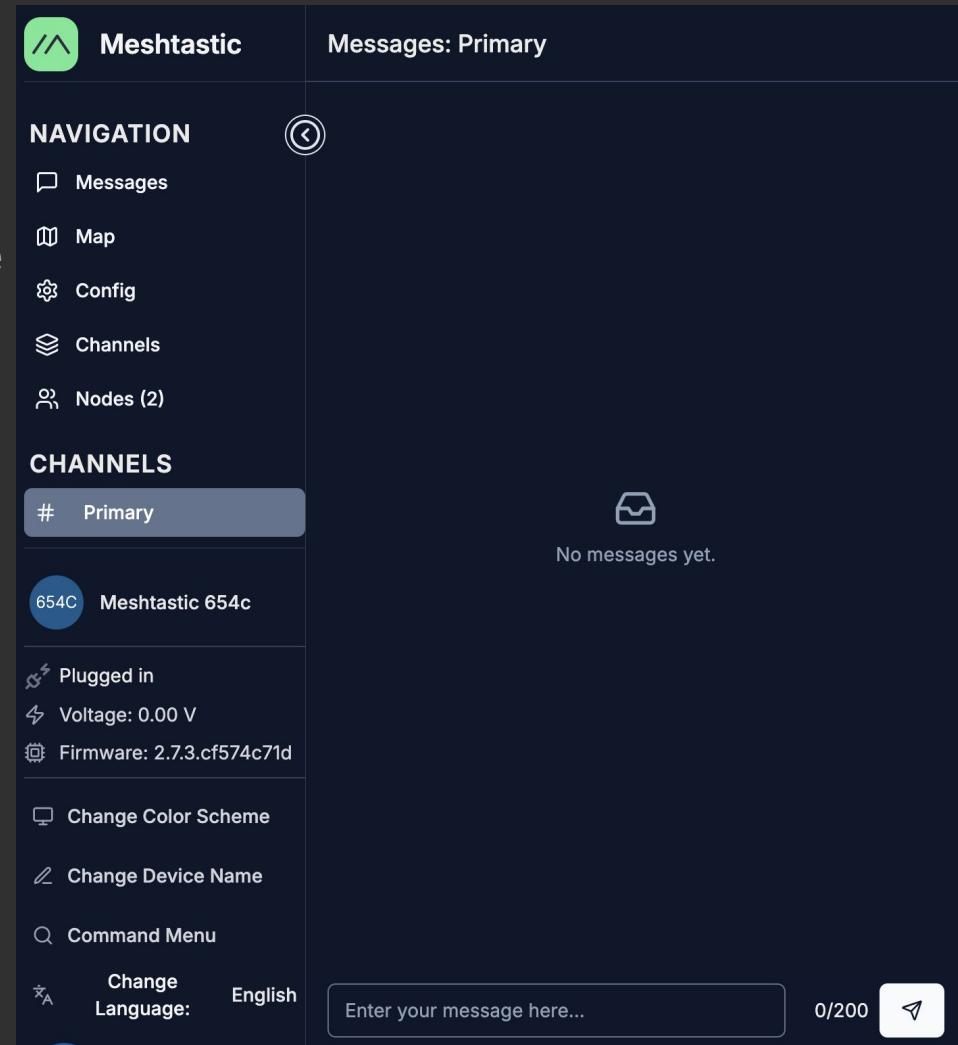
Long Name

 17/40

Short Name

 3/4

**Reset** **Save**



# Let's Enable Our Radio!

 **Connected Radio**  
Shows information for the Lora radio connected via bluetooth.

 **YOUR\_NAME\_HERE**  
BLE Name: BOO\_23c8  
Firmware Version: 2.5.17  
**Subscribed to mesh**

 **Set LoRa Region**

< **LoRa Config**

Configuration for: YOUR\_NAME\_HERE

**OPTIONS**

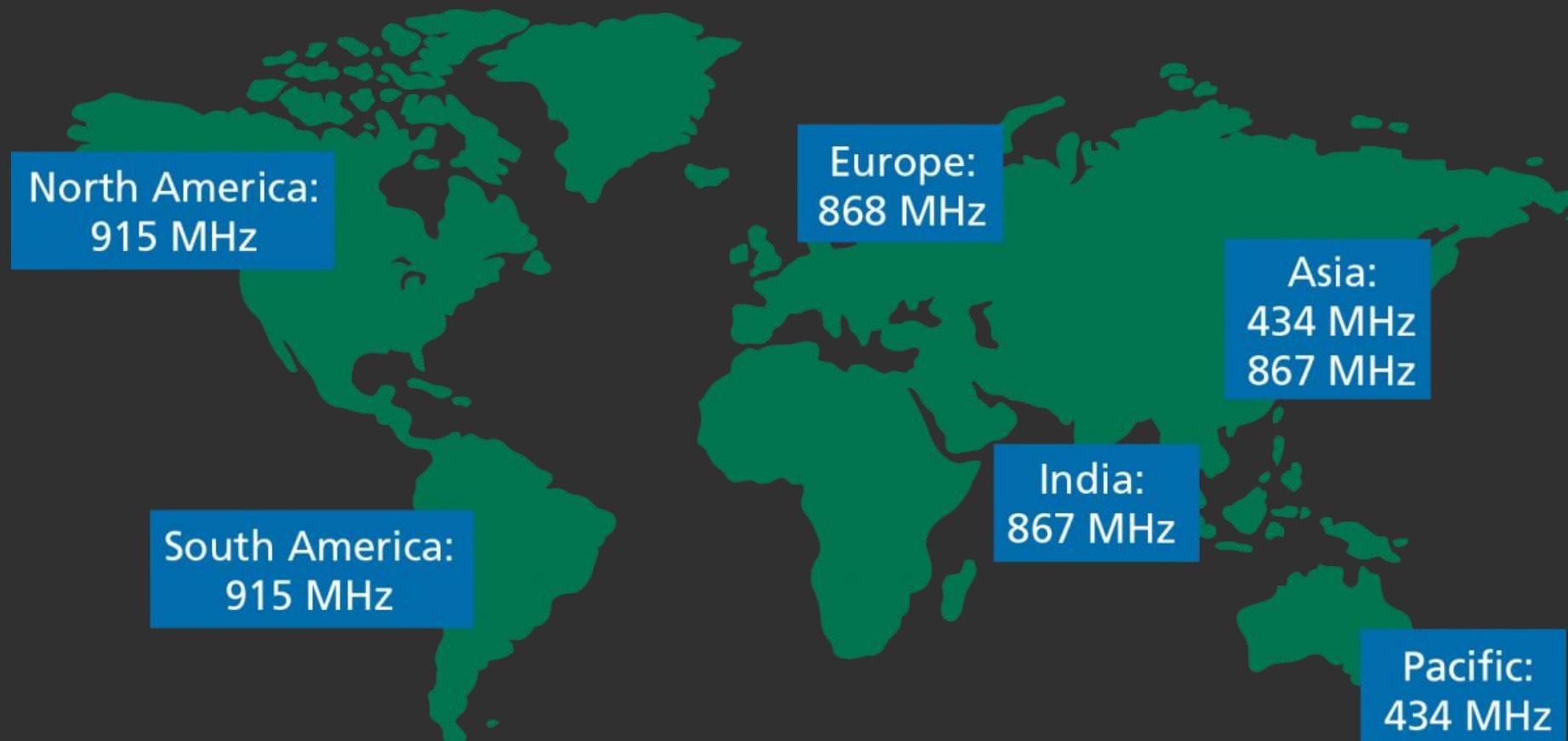
Region Please set a region ▾  
The region is not yet defined.

 United States  
European Union 433mhz  
European Union 868mhz  
China  
Japan  
Australia / New Zealand

 Presets  
Available regions:

 Available regions:  
ADVANCE

# Regional frequencies - Check your radios!



# Set Your Region Via Web

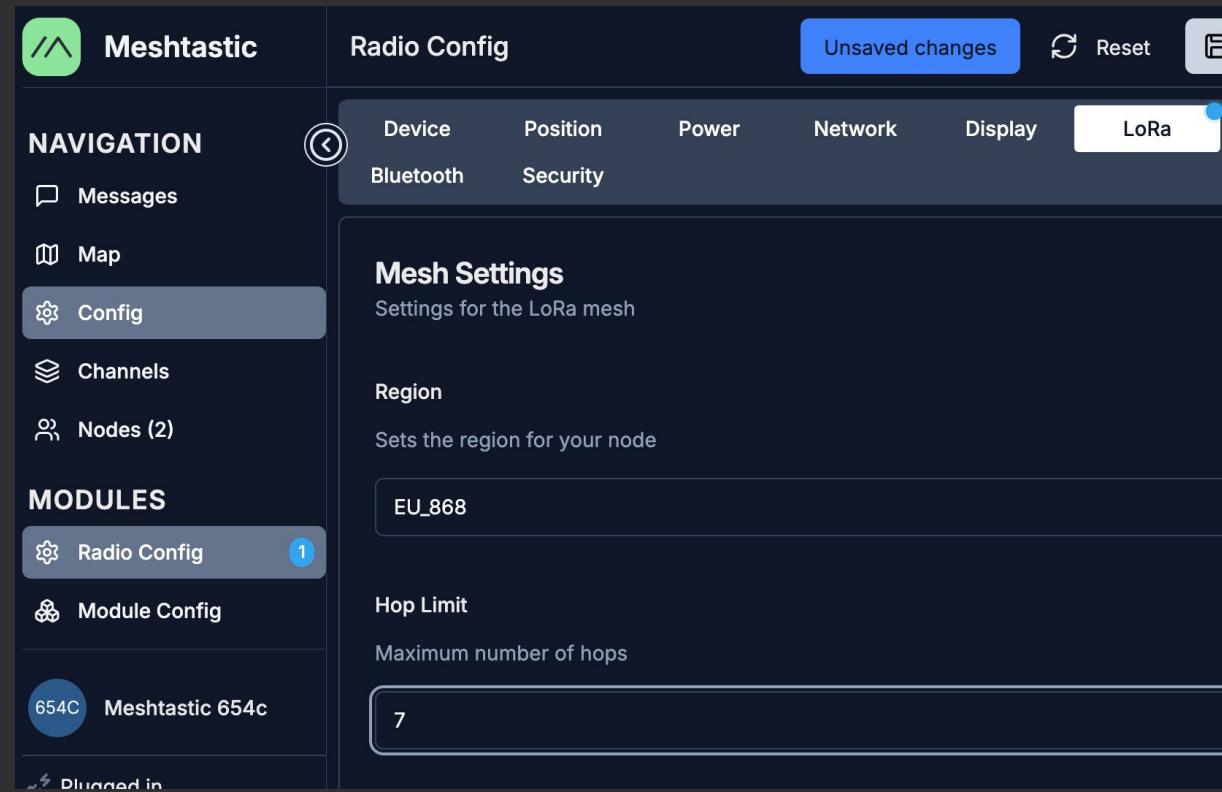
You can use the screen & buttons to set the region, or:

Click on Config, then LoRa.

Select the correct region.

Also set the hop limit to 7.

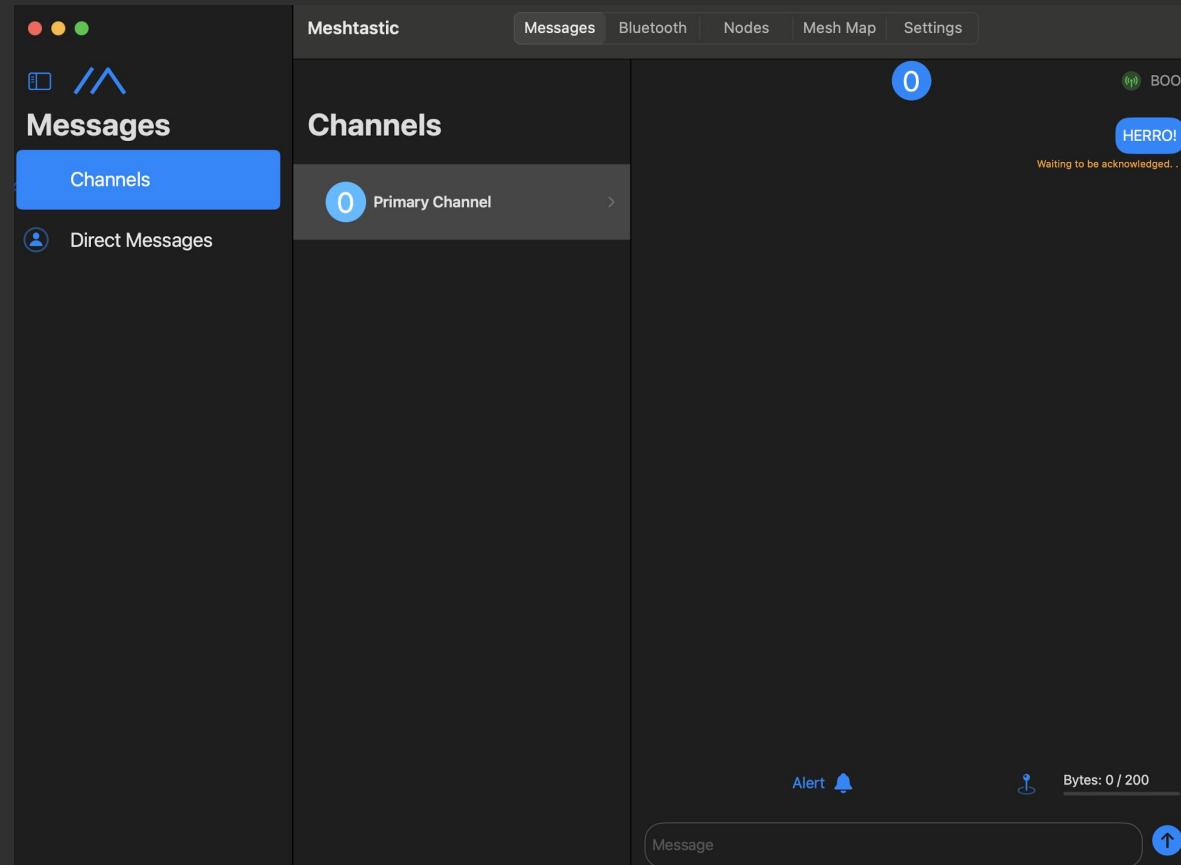
Save, and your Nibble will reboot ready to transmit!



# We are transmitting!

Go to channel, primary channel, and type a message.

If you see “acknowledged,” another node has seen and retransmitted your message!



# Node Discovery

Go to “Nodes” and see if you can see others

Click on them to learn more information

You can run a Traceroute to see the network path

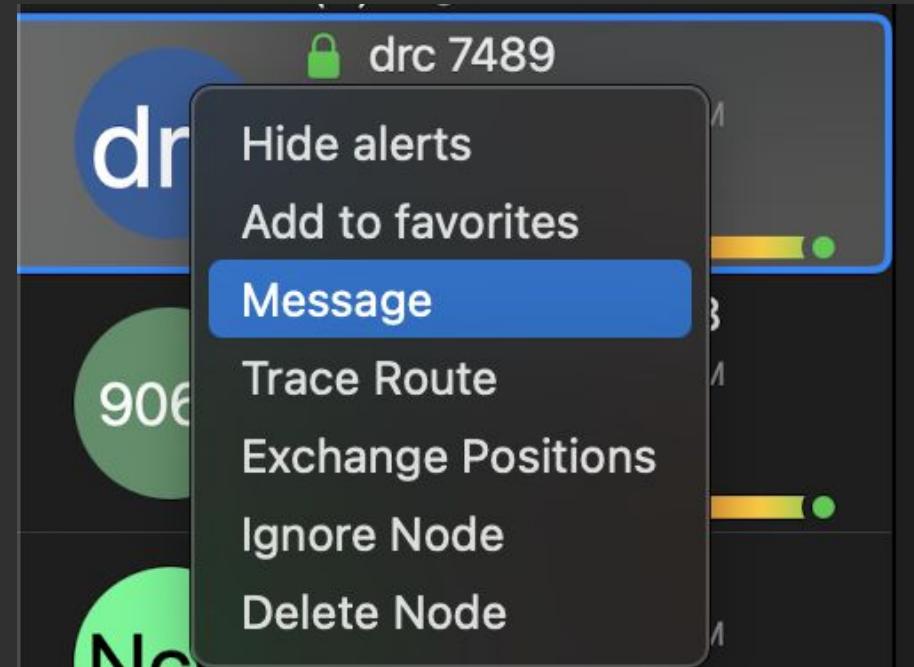
The screenshot shows the Meshtastic mobile application interface. At the top right, the device name "TaMu1" is displayed. Below it, the "HARDWARE" section shows an "Unsupported" device icon with a red "X" and a "Model" section for the "LILYGO T-Echo". The main area is titled "Nodes (11)" and contains a search bar "Find a node". A list of 11 nodes is shown, each with a colored circular icon, the node name, connection status, last heard time, role, and signal strength (indicated by a color bar from green to red). The nodes listed are: YOUR\_-\_NAME\_HERE, drc 7489, Meshtastic 9068, 9068, Nicolas, AAAAAAAA, AAAA, Meshtastic 329c, 329c, ricl, Clem, and TaMu1. The "NODE" section provides detailed information for the selected node, "TaMu1", including Node Number (1724123385), User Id (!66c408f9), Role (Client), First heard (2 minutes ago), and Last heard (2 minutes ago). The "LOGS" section lists various log types: Device Metrics Log, Node Map, Position Log, Environment Metrics Log, Trace Route Log, and Detection Sensor Log.

Node	Role	Last Heard	Signal Good
YOUR_-_NAME_HERE	Client	12/29/2024, 4:05 PM	Green
drc 7489	Client	12/29/2024, 4:06 PM	Green
Meshtastic 9068	Client	12/29/2024, 4:06 PM	Green
9068	Client	12/29/2024, 4:06 PM	Green
Nicolas	Client Mute	12/29/2024, 4:05 PM	Yellow
AAAAAAA	Client	12/29/2024, 4:05 PM	Green
AAAA	Client	12/29/2024, 4:05 PM	Green
Meshtastic 329c	Client	12/29/2024, 4:05 PM	Green
329c	Client	12/29/2024, 4:04 PM	Green
ricl	Client	12/29/2024, 4:04 PM	Green
Clem	Client Mute	12/29/2024, 4:04 PM	Green
TaMu1	Client	12/29/2024, 4:04 PM	Green

# Direct Message a Node

Click on a node you want to message  
and select “message”

Type your DM (it's not encrypted unless  
you exchange keys)



# Let's Explore Settings!

## MODULE CONFIGURATION

Ambient Lighting

Canned Messages

Detection Sensor

External Notification

MQTT

Range Test

PAX Counter

Ringtone

Serial

Store & Forward

Telemetry (Sensors)

## DEVICE CONFIGURATION

User

Bluetooth

Device

Display

Network

Position

Power

## RADIO CONFIGURATION

Hourly Duty Cycle

Your region has a 10% hourly duty cycle, your radio will stop sending packets when it reaches the hourly limit.

Limit all periodic broadcast intervals especially telemetry and position. If you need to increase hops, do it on nodes at the edges, not the ones in the middle. MQTT is not advised when you are

LoRa

Channels

Security

Share QR Code

# Device roles

## ✓ Client

Client Mute

Router

Repeater

Tracker

Sensor

Tak

Client Hidden

Lost And Found

Tak Tracker

Device Role	Description	Best Uses
CLIENT	App connected or stand alone messaging device.	General use for individuals needing to communicate over the Meshtastic network with support for client applications.
CLIENT_MUTE	Device that does not forward packets from other devices.	Situations where a device needs to participate in the network without assisting in packet routing, reducing network load.
CLIENT_HIDDEN	Device that only broadcasts as needed for stealth or power savings.	Use in stealth/hidden deployments or to reduce airtime/power consumption while still participating in the network.
TRACKER	Broadcasts GPS position packets as priority.	Tracking the location of individuals or assets, especially in scenarios where timely and efficient location updates are critical.
LOST_AND_FOUND	Broadcasts location as message to default channel regularly for to assist with device recovery.	Used for recovery efforts of a lost device.
SENSOR	Broadcasts telemetry packets as priority.	Deploying in scenarios where gathering environmental or other sensor data is crucial, with efficient power usage and frequent updates.
TAK	Optimized for ATAK system communication, reduces routine broadcasts.	Integration with ATAK systems (via the Meshtastic ATAK Plugin) for communication in tactical or coordinated operations.
TAK_TRACKER	Enables automatic TAK PLI broadcasts and reduces routine broadcasts.	Standalone PLI integration with ATAK systems for communication in tactical or coordinated operations.
REPEATER	Infrastructure node for extending network coverage by relaying messages with minimal overhead. Not visible in Nodes list.	Best positioned in strategic locations to maximize the network's overall coverage. Device is not shown in topology.
ROUTER	Infrastructure node for extending network coverage by relaying messages. Visible in Nodes list.	Best positioned in strategic locations to maximize the network's overall coverage. Device is shown in topology.

# What device roles mean

Device Role	BLE/WiFi/ Serial	Screen Enabled	Power Consumption	Retransmit	Prioritized Routing	Visible in Nodes List
CLIENT	Yes	Yes	Regular	Yes	No	Yes
CLIENT_MUTE	Yes	Yes	Lowest	No	No	Yes
CLIENT_HIDDEN	Yes	Yes	Lowest	Local Only	No	No
TRACKER	Yes	No	Regular / Low	Awake Only <sup>1</sup>	No	Yes
LOST_AND_FOUND	Yes	No	Regular	Yes	No	Yes
SENSOR	Yes	No	Regular / Low	Awake Only <sup>1</sup>	No	Yes
TAK	Yes	Optional	Regular	Yes	No	Yes
TAK_TRACKER	Yes	Optional	Regular	Yes	No	Yes
ROUTER	No <sup>2</sup>	No	High	Yes	Yes	Yes
REPEATER	Yes	No	High	Yes	Yes	No

Surprise: changing  
to a router will turn  
off BLE!

# Meshtastic LoRa Radio Profiles

8 Profiles: condense complex settings into common use cases

Trade-offs between long range or short range benefits (speed/range)

Often customized for conferences and events

Channel setting	Alt Channel Name	Data-Rate	SF / Symbols	Coding Rate	Bandwidth	Link Budget
Short Range / Turbo	Short Turbo	21.88 kbps	7 / 128	4/5	500 kHz <sup>1</sup>	140dB
Short Range / Fast	Short Fast	10.94 kbps	7 / 128	4/5	250 kHz	143dB
Short Range / Slow	Short Slow	6.25 kbps	8 / 256	4/5	250 kHz	145.5dB
Medium Range / Fast	Medium Fast	3.52 kbps	9 / 512	4/5	250 kHz	148dB
Medium Range / Slow	Medium Slow	1.95 kbps	10 / 1024	4/5	250 kHz	150.5dB
Long Range / Fast	Long Fast	1.07 kbps	11 / 2048	4/5	250 kHz	153dB
Long Range / Moderate	Long Moderate	0.34 kbps	11 / 2048	4/8	125 kHz	156dB
Long Range / Slow	Long Slow	0.18 kbps	12 / 4096	4/8	125 kHz	158.5dB
Very Long Range / Slow	Very Long Slow	0.09 kbps	12 / 4096	4/8	62.5 kHz	161.5dB

# Adding sensors & hardware

Temperature,motion, other sensors  
are easy!

Enable Telemetry module & add  
common i2c sensors!

**Telemetry Settings**  
Settings for the Telemetry module

**Device Metrics**  
Device metrics update interval (seconds)

Seconds

Environment metrics update interval (seconds)

Seconds

**Module Enabled**  
Enable the Environment Telemetry

## Currently Supported Sensor Types

# I2C Sensors Supported



Sensor	I <sup>2</sup> C Address	Data Points
BMP085	0x76, 0x77	Temperature and barometric pressure
BMP180	0x76, 0x77	Temperature and barometric pressure
BMP280	0x76, 0x77	Temperature and barometric pressure
BME280	0x76, 0x77	Temperature, barometric pressure and humidity
BME68x	0x76, 0x77	Temperature, barometric pressure, humidity and air resistance
MCP9808	0x18	Temperature
INA260	0x40, 0x41, 0x43	Current and Voltage
INA219	0x40, 0x41, 0x43	Current and Voltage
INA3221	0x42	3-channel Current and Voltage
LPS22	0x5D, 0x5C	Barometric pressure
SHTC3	0x70	Temperature and humidity
SHT31	0x44	Temperature and humidity
PMSA003I	0x12	Concentration units by size and particle counts by size
DFROBOT_LARK	0x42	Temperature, barometric pressure, humidity, wind direction, wind speed
MAX30102	0x57	Heart Rate, Oxygen Saturation, and body temperature
MLX90614	0x5A	Body temperature

# Modules (Part 1)

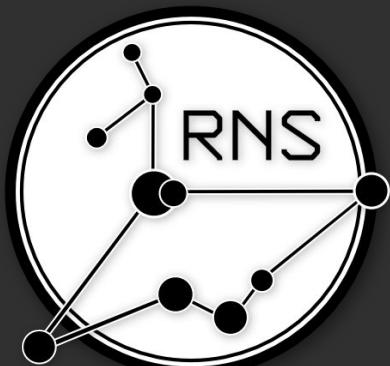
Name	Description
<u>Ambient Lighting</u>	Adjust the brightness of NCP5623 I2C RGB LEDs
<u>Audio</u>	Enable Support for Codec2 Voice Comms on certain devices.
<u>Canned Message</u>	Set a number of predefined messages to send out directly from the device with the use of an input device like a rotary encoder.
<u>Detection Sensor</u>	Configure a GPIO pin to be monitored for specified high/low status and send text alerts.
<u>External Notification</u>	Incoming messages are able to alert you using circuits you attach to the device (LEDs, Buzzers, etc).
<u>MQTT</u>	Forward packets along to an MQTT server. This allows users on the local mesh to communicate with users on another mesh over the internet.
<u>Neighbor Info</u>	Send info on 0-hop neighbors to the mesh.
<u>Paxcounter</u>	Count the number of BLE and Wifi devices passing by a node.

# Modules (Part 2)

<u>Range Test</u>	Send messages with GPS location at an interval to test the distance your devices can communicate. Requires (at least) one device set up as a sender and one as a receiver. The receiver(s) will log all incoming messages to a CSV.
<u>Remote Hardware</u>	Set and read a GPIO status remotely over the mesh.
<u>Serial Module</u>	Send messages across the mesh by sending strings over a serial port.
<u>Store &amp; Forward</u>	Stores messages on a device for delivery after disconnected clients rejoin the mesh.
<u>Telemetry</u>	Attach sensors to the device and transmit readings on a regular interval to the mesh.
<u>Traceroute</u>	Track which nodes are used to hop a message to a certain destination.

# Meshtastic Alternatives

- Meshcore - Routes more like Tor, does not use flood routing and establishes circuits for DM's. Uses less bandwidth for communication.
- Reticulum - A cryptography-based networking stack for building local and wide-area networks with readily available hardware, including Wi-Fi and LoRa



## About MeshCore

MeshCore is a multi platform system for enabling secure text based communications utilising LoRa radio hardware. It can be used for Off-Grid Communication, Emergency Response & Disaster Recovery, Outdoor Activities, Tactical Security including law enforcement, private security and also IoT sensor networks.

### Our Mission

At MeshCore, our passion is building reliable and secure decentralised mesh radio networks for text-based communication. We are driven by the need to empower users in off-grid and disaster-stricken regions with a communication tool that does not depend on the internet or cellular networks.

### Easy to use

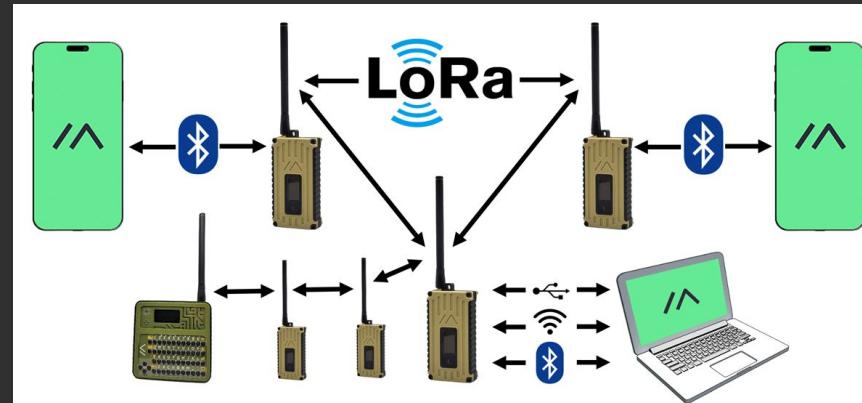
Use our web flasher to install MeshCore firmware on popular devices like the Lilygo T-Deck or Heltec V3 LoRa32, you can also use our web client to communicate securely over radio with other users just using a Chrome web browser - and yes of course it works without the internet!

# What is a mesh network?

Mesh networks assemble themselves! Each device acts to repeat packets across the network.

This allows for convenient, self-deploying networks that allow huge ranges.

Both fixed & moving nodes can make up an active mesh network, often creating temporary network routes.

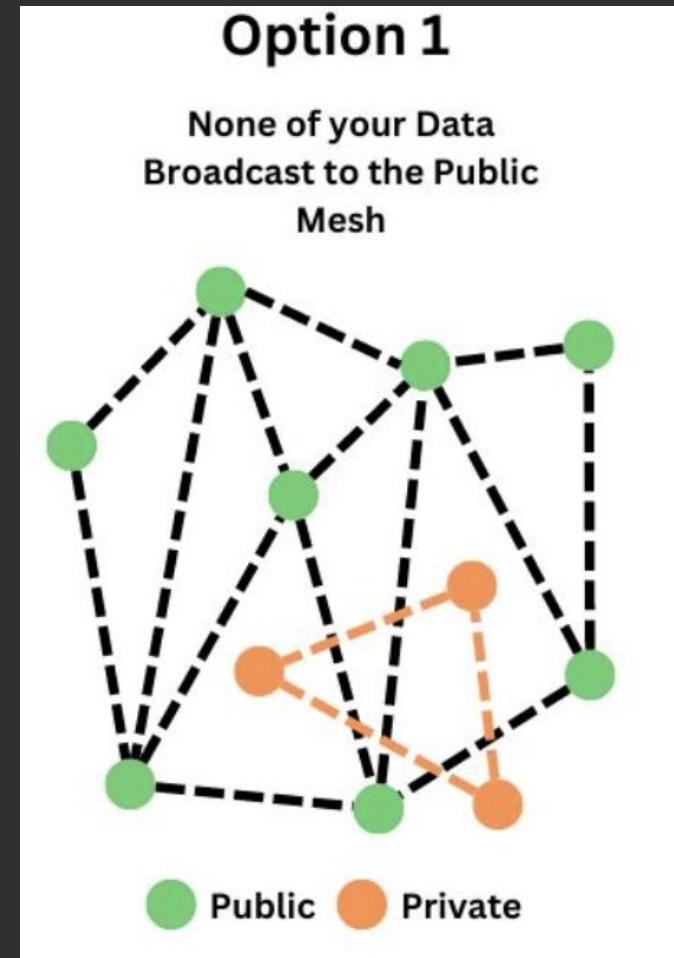


# Public, private, semi-private mesh

## Option 1: Private

Delete default LongFast channel

Create new encrypted channel with different settings and encryption. Only use your own nodes.



# Public, private, semi-private mesh

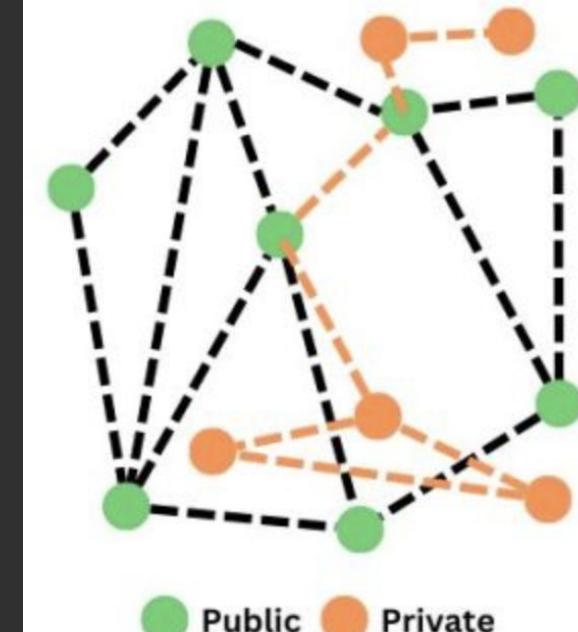
## Option 2: Semi-Private

Keep LongFast, but create secondary encrypted channel only your devices understand.

Your nodes are invisible to the network, but use public nodes to retransmit encrypted packets for extended range.

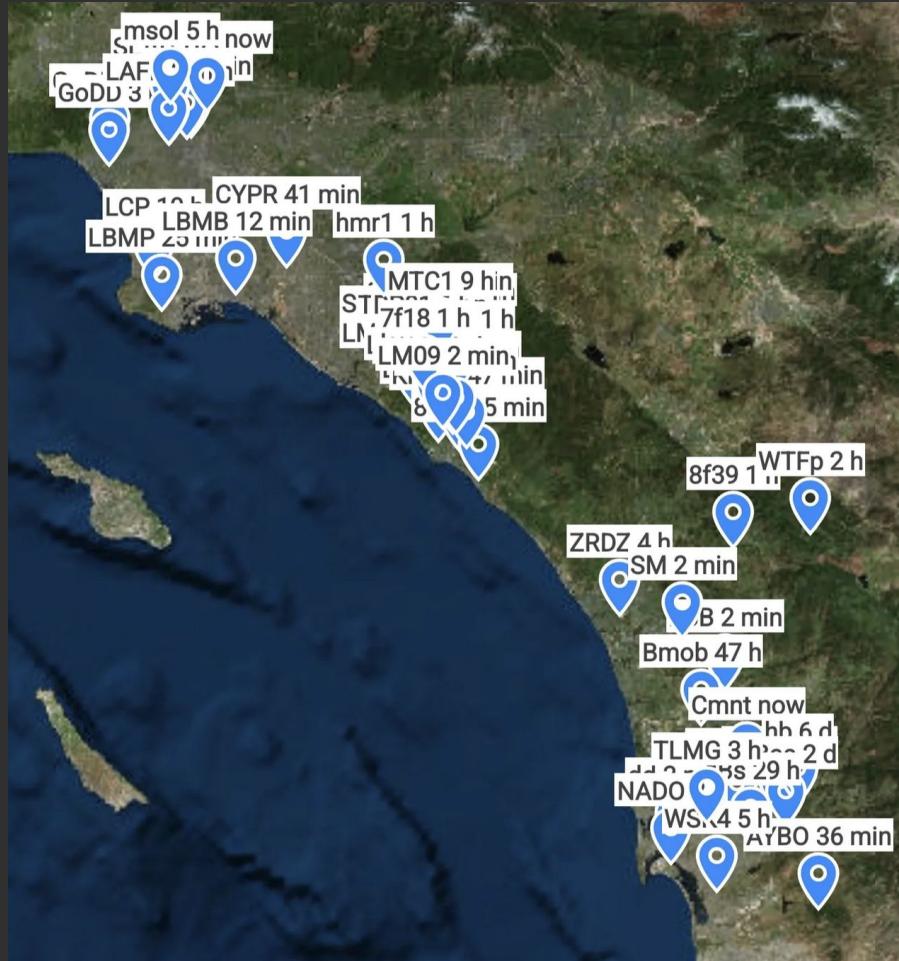
### Option 2

Utilizing the Public Mesh Whilst Remaining Hidden



# Regional Networks

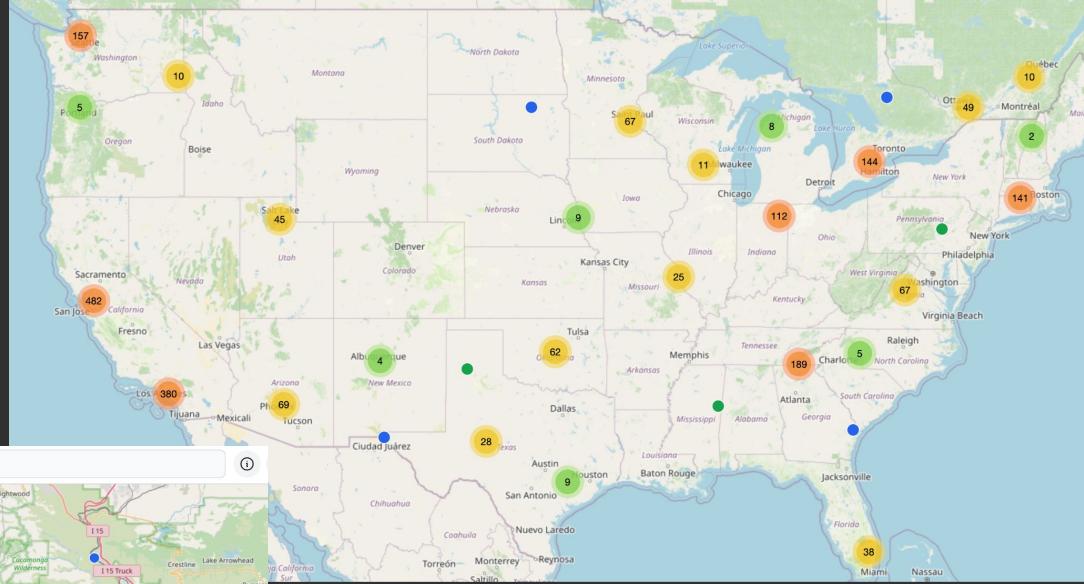
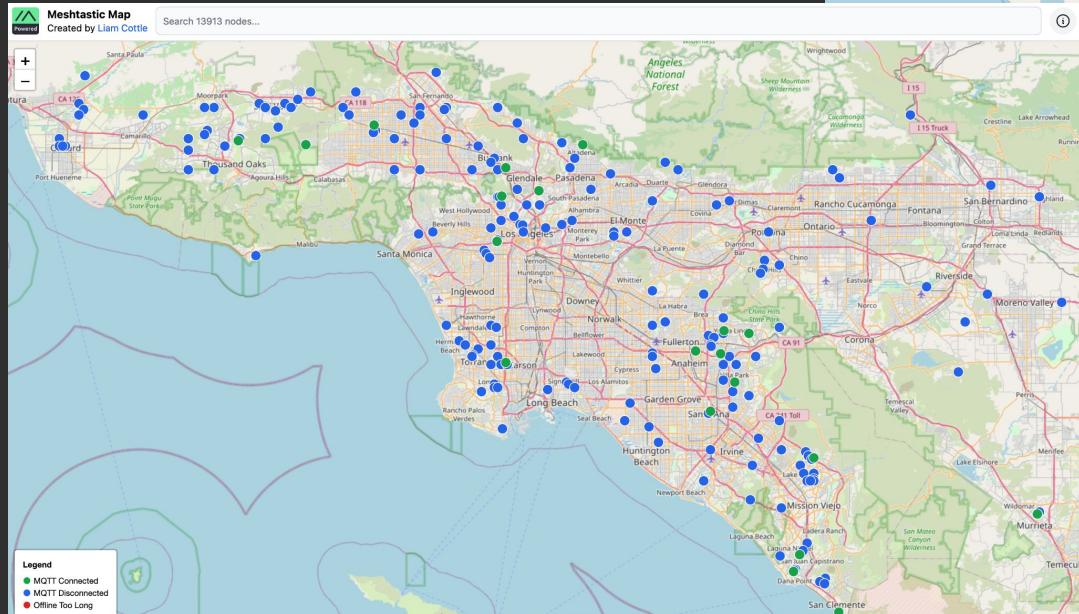
- **SoCal Mesh** - [socalmesh.org](http://socalmesh.org) - San Diego to Santa Clarita
- **Meshtastic Bay Area Group** - [bayme.sh](http://bayme.sh) - San Francisco Bay Area
- **Central Valley Mesh** - [centralvalleymesh.net](http://centralvalleymesh.net) - From Fresno to Santa Barbara



# Real World Mapping

[meshtastic.liamcottle.net](http://meshtastic.liamcottle.net)

Uses local MQTT collectors

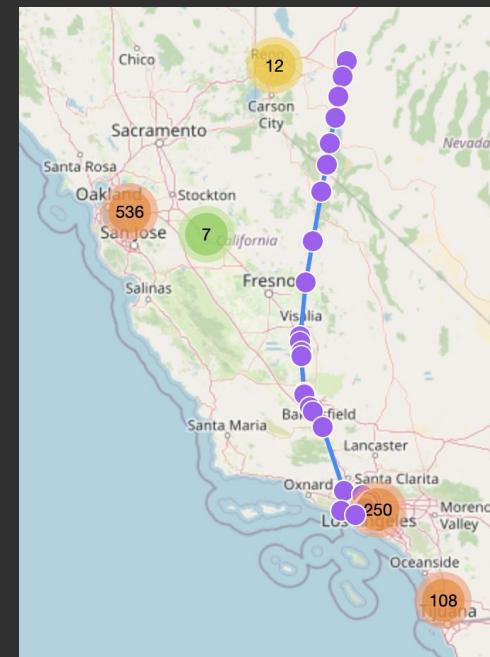
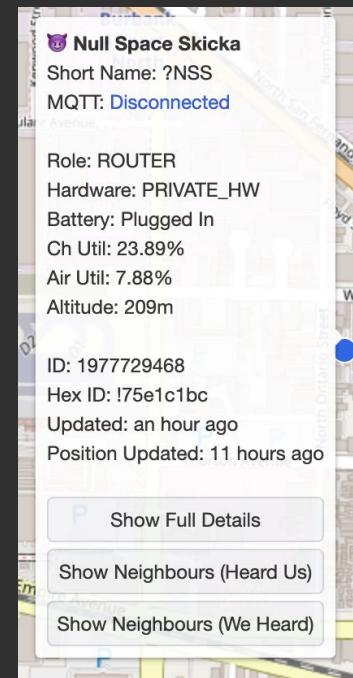
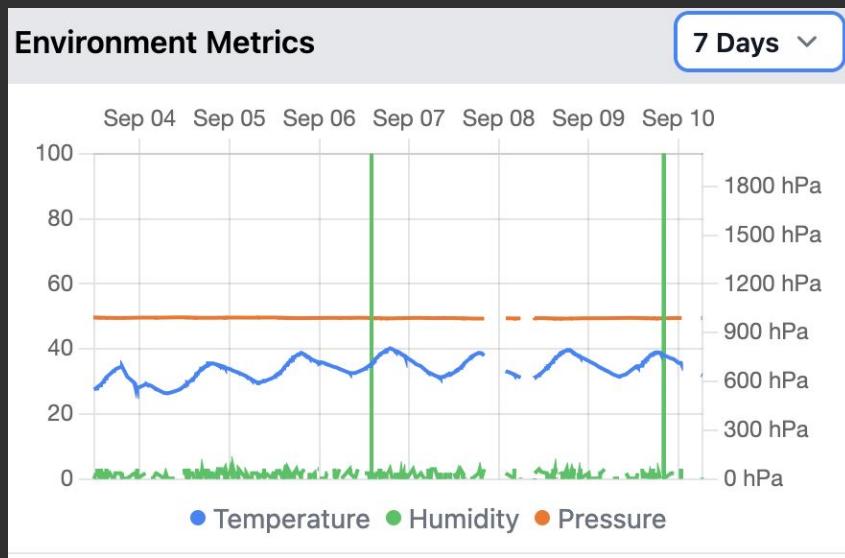


- Collector nodes forward observed nodes over internet to the map via MQTT
- Blind spots, but I can track location of nodes over time
- Tracked my flight!

# Nodes On Default Channel Are Not Private

For nodes in range of collectors:

7 days of sensor, telemetry, and location data



# Encryption Cheat Sheet

- Nodes communicating on LongFast channel with default encryption key:
  - All nodes can see all messages: unless they are DM's encrypted with keys exchanged between nodes
- Nodes communicating on LongFast channel with secret encryption key:
  - All nodes who know the channel key can read all messages, except encrypted DM's
- Nodes who have exchanged encryption keys communicating via DM's:
  - Only another node you've exchanged encryption keys with should be able to read

# Python remote control

We can control a  
Meshtastic node over  
Python!

- Supports serial or Bluetooth connection
- Get info or send commands
- Programmatic Python control for SIGINT

Another example using Python 3 code to send a message to the mesh when WiFi is enabled:

```
import time
import meshtastic
import meshtastic.tcp_interface
from pubsub import pub

def onReceive(packet, interface): # called when a packet arrives
    print(f"Received: {packet}")

def onConnection(interface, topic=pub.AUTO_TOPIC): # called when we (re)connect to the radio
    # defaults to broadcast, specify a destination ID if you wish
    interface.sendText("hello mesh")

pub.subscribe(onReceive, "meshtastic.receive")
pub.subscribe(onConnection, "meshtastic.connection.established")
interface = meshtastic.tcp_interface.TCPIInterface(hostname='192.168.68.74')
while True:
    time.sleep(1000)
interface.close()
```

# Setup scripts

- Turn into bash scripts to config new nodes

```
#!/bin/bash

# Check if a serial port argument is provided
if [ -z "$1" ]; then
    echo "Usage: $0 <serial_port>"
    exit 1
fi

# Assign the first argument to a variable
SERIAL_PORT=$1

# Set the region to US
meshtastic --port "$SERIAL_PORT" --set lora.region US

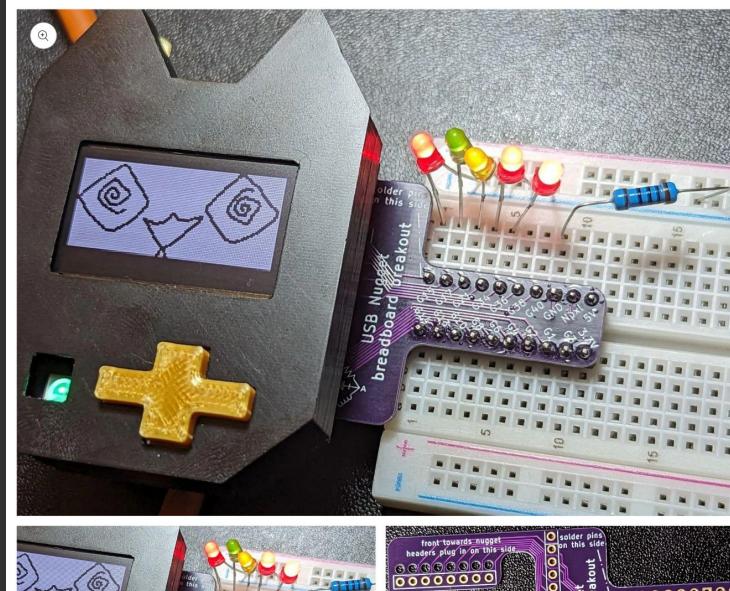
# Wait for 8 seconds to allow the device to reboot
sleep 8

# Set the screen flip option
meshtastic --port "$SERIAL_PORT" --set display.flip_screen true
```

# Teach a friend!

We have kits on Retia.io and discounts for instructors teaching classes!

- Nuggets
- Nibbles
- Add-ons
- Online classes



RETIA.IO

**USB Nugget  
Breadboard Tail  
Breakout**

\$15.00 USD  
Tax included.

Quantity

- 1 +

Add to cart

Buy with **shop Pay**

More payment options

This cute Breadboard tail breakout allows for easy connection to a breadboard for electronics prototyping. It was designed to be used with CircuitPython or Arduino to prototype hardware with the USB Nugget

# Keep in Touch

Want to learn more? You can find us here:

- Discord: <https://discord.gg/rjVJbauAUX>
- Store: [Retia.io](https://Retia.io)
- Nugget Flasher: [Nugget.dev](https://Nugget.dev)
- My work: [www.hack.gay](https://www.hack.gay)

