

# Meshtastic for Beginners

Long Range Mesh Networks For All!



# 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
- Configuring nodes for specific use cases
- 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 [⑤](#).
- Additionally, he is known for his active engagement in cybersecurity research, as evidenced by his presence on YouTube [⑥](#).

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

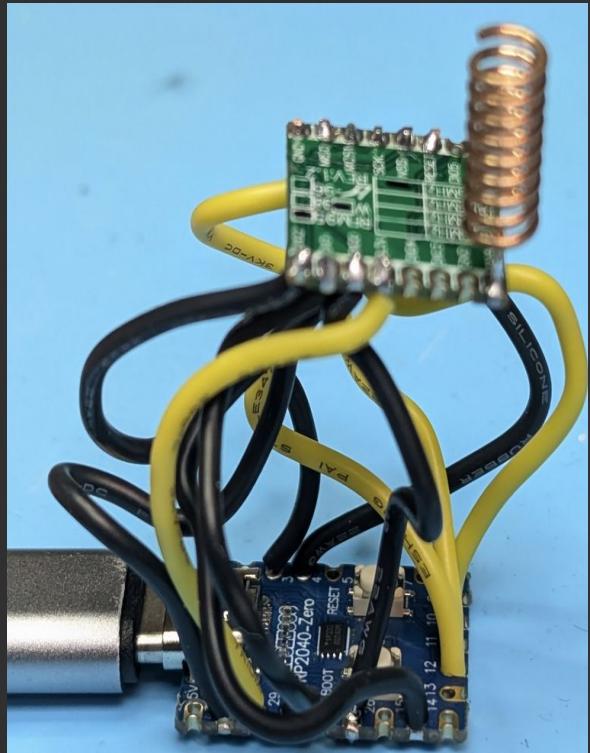
# What is the point of Meshtastic?



- 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

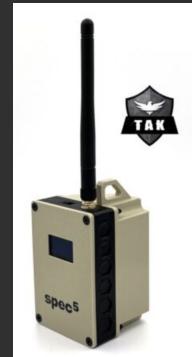
# What is the cost of participating?

- You can make your own horrible node for \$5
- 30-40 euro at this conference
- \$60-\$80 for a fancy node from China
- \$100+ for custom nodes



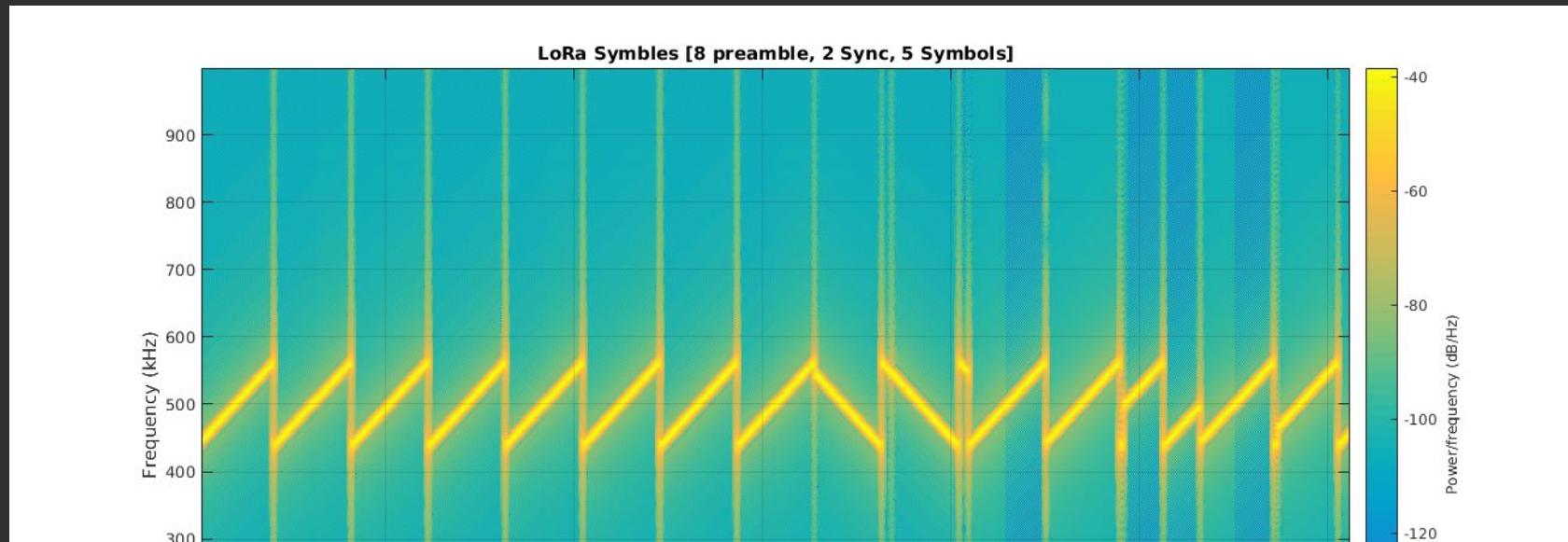
# What can we make with this?

- 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



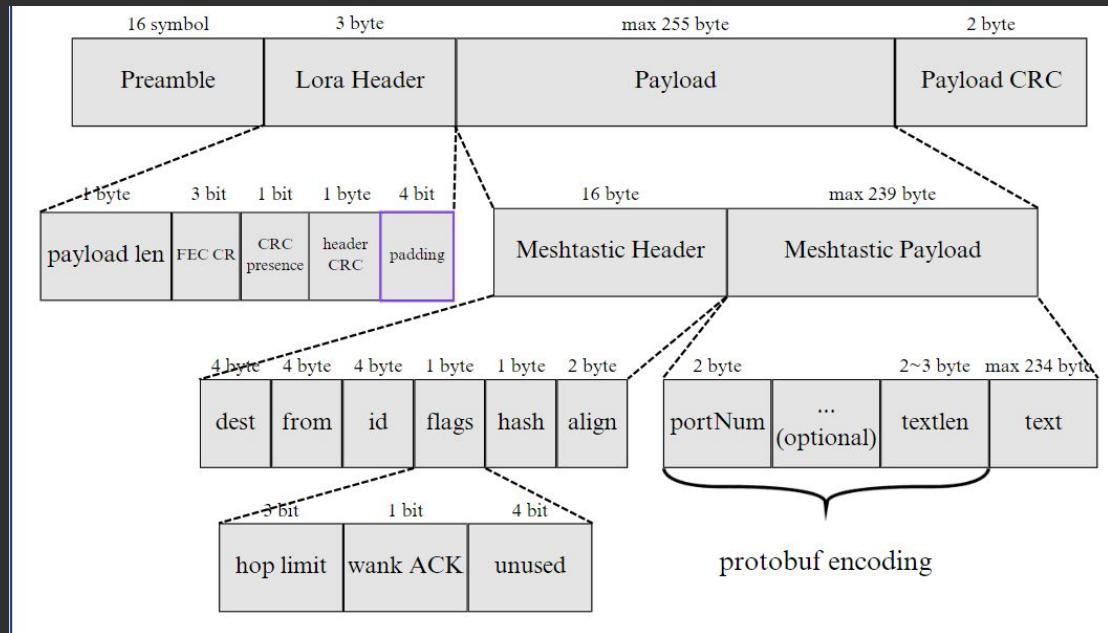
# Defining Terms: LoRa Radio

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



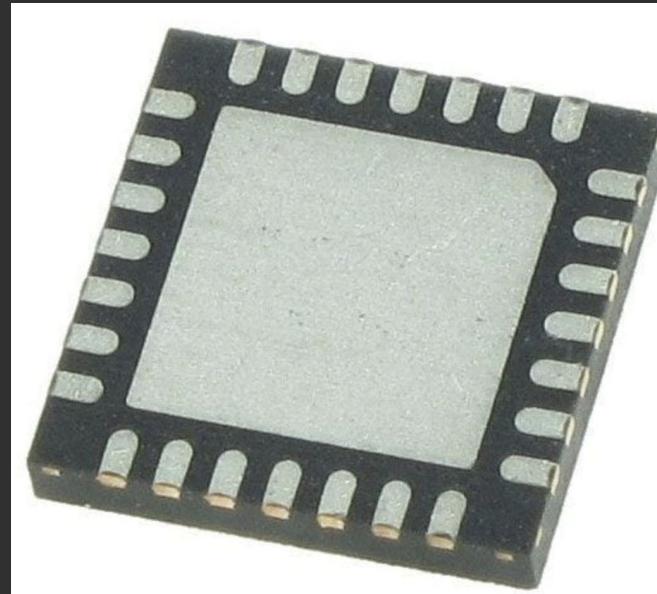
# Defining Terms: Meshtastic

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



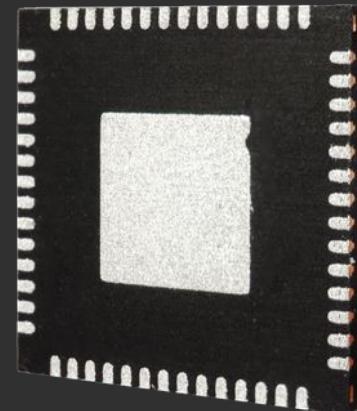
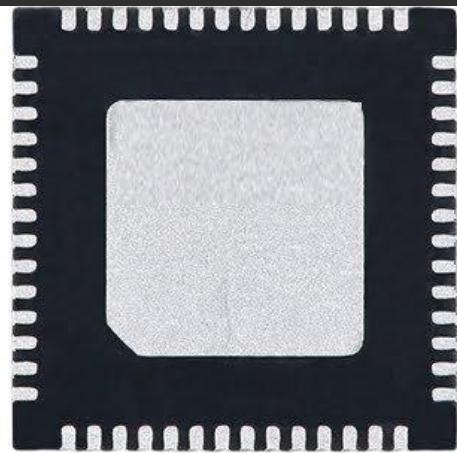
# Defining Terms: LoRa Radio Chips

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



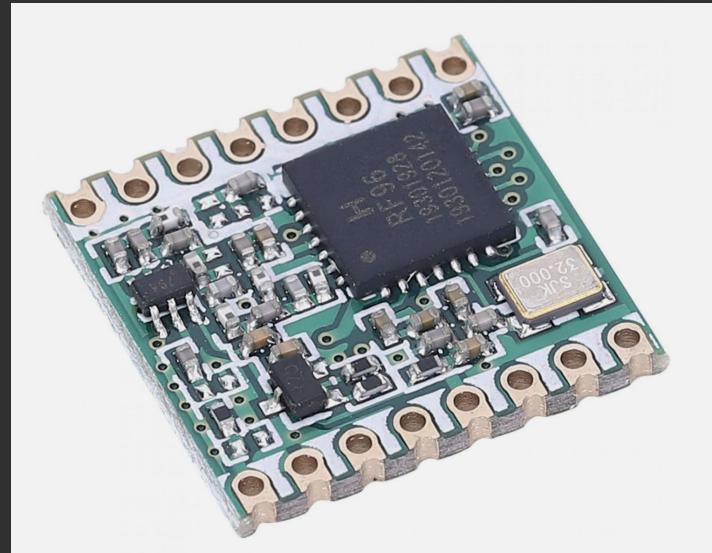
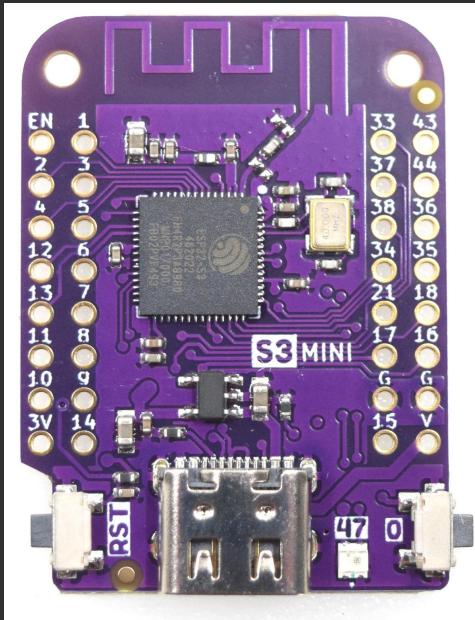
# Defining Terms: Host Microcontroller

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



# Defining Terms: Breakout Boards

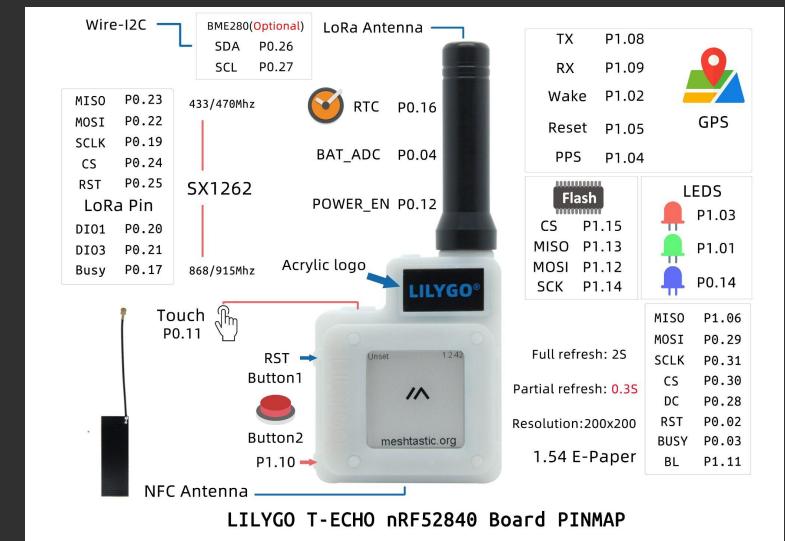
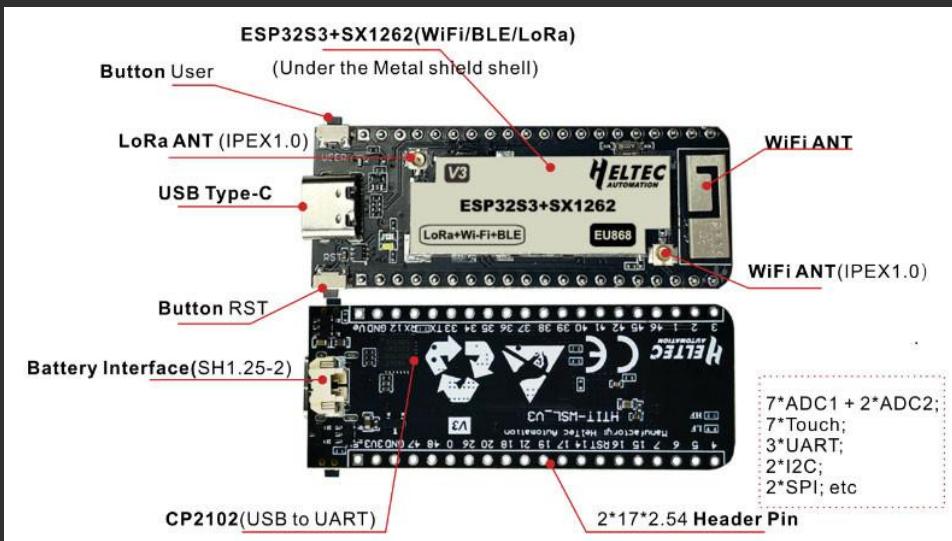
- **Breakout board:** An expansion board designed to make a bare host or radio chip more useful, breaking out pins & sometimes USB & power management



# Defining Terms: Nodes

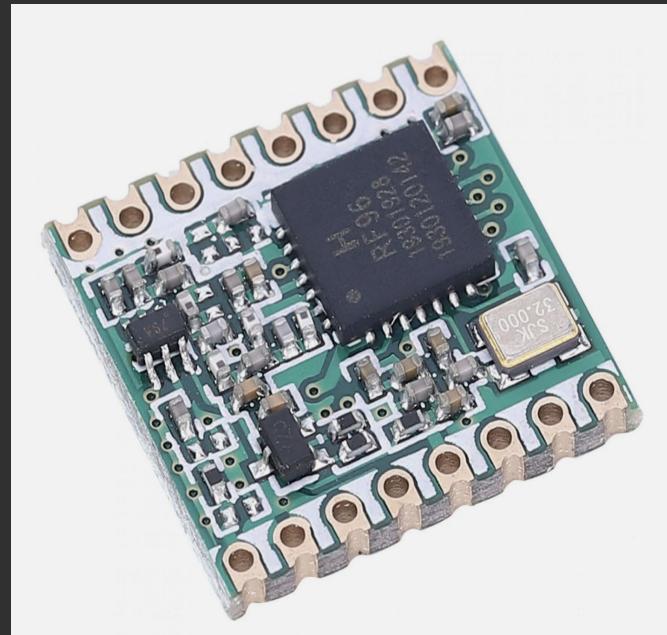
- **LoRa Node:** A combination of LoRa radio & host microcontroller. Together, they can run Meshtastic!

Simple Node <-----> Complex Node

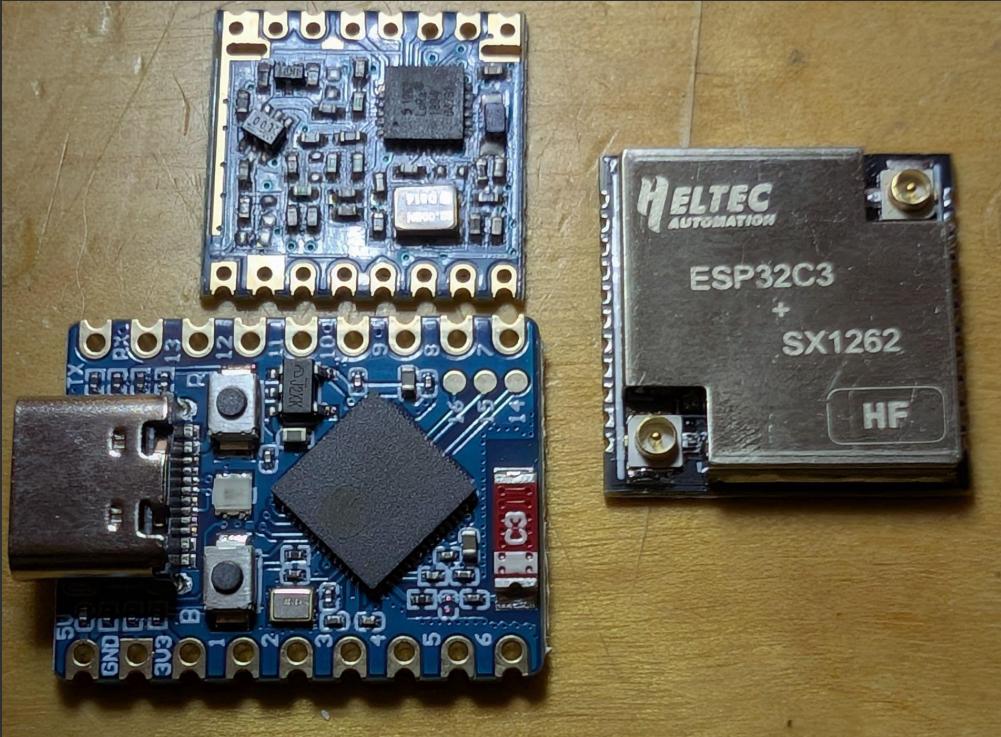


# Host + Radio = Node

**RP2040 Zero (host breakout) + RFM95 (radio breakout) = \$5 Node**

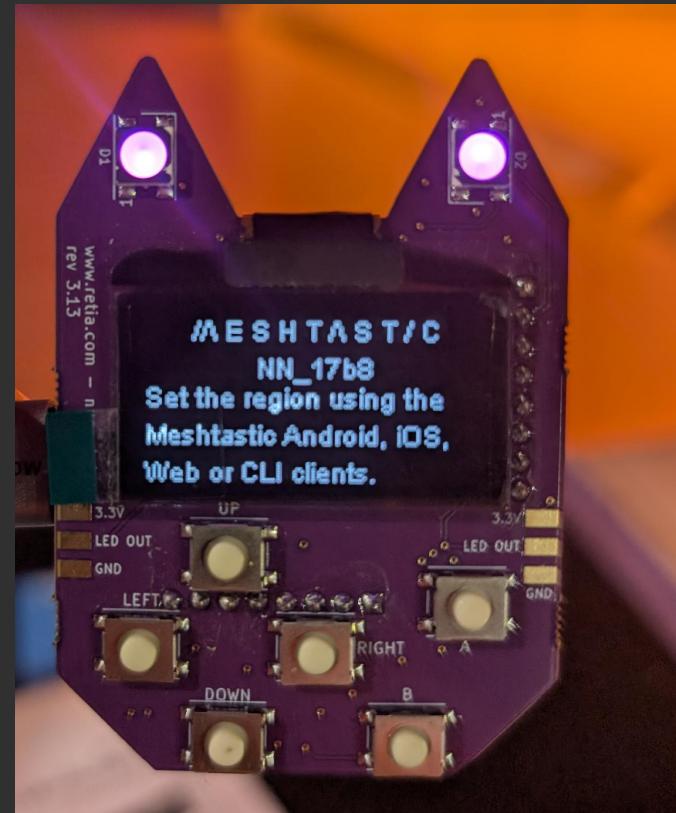


# Future Nodes for \$10



# The Bluetooth Nugget + LoRa Backpack

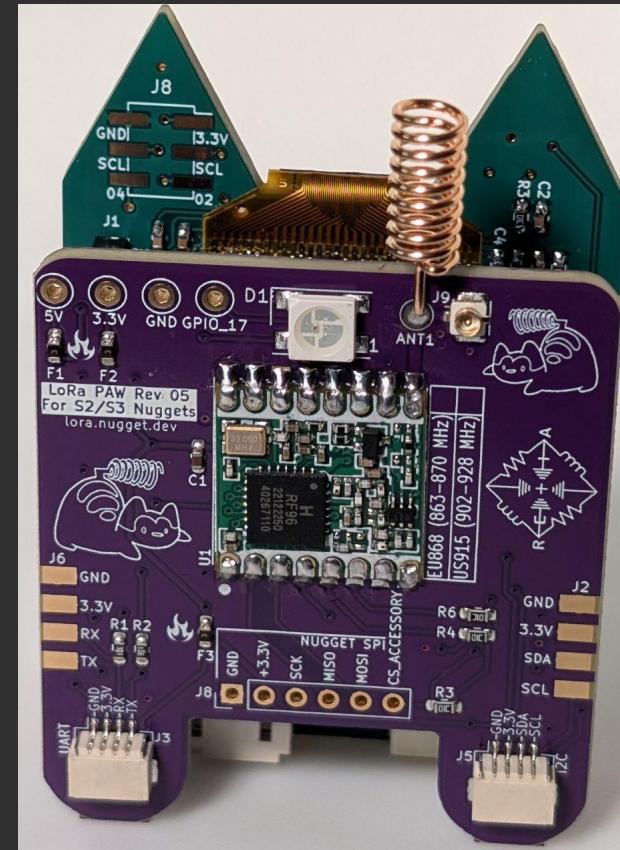
- The Bluetooth Nugget features an ESP32s3
- Bluetooth, Wi-Fi, native USB, serial
- Quiic connectors for i2c sensors
- 6 buttons for sending and receiving canned messages
- 2 neopixels for status indicators
- OLED i2c screen
- Features Bluetooth, Wi-Fi and support for i2c, UART, and SPI hardware



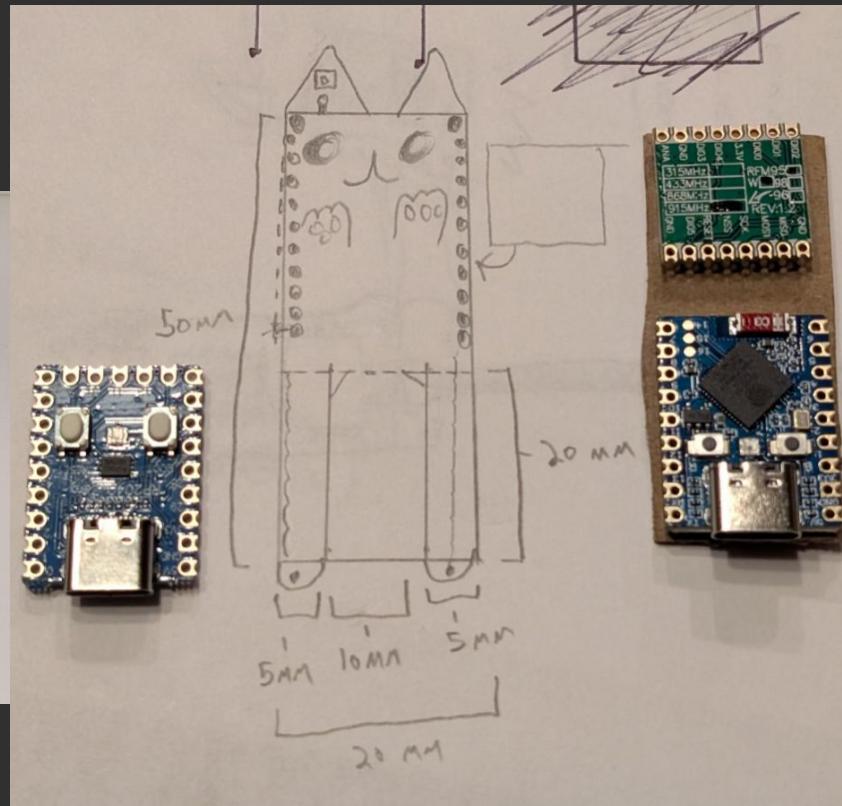
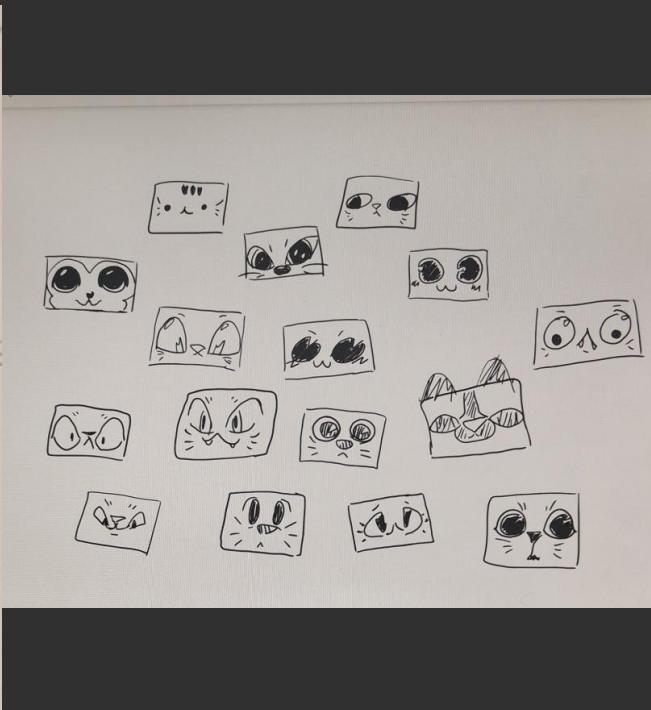
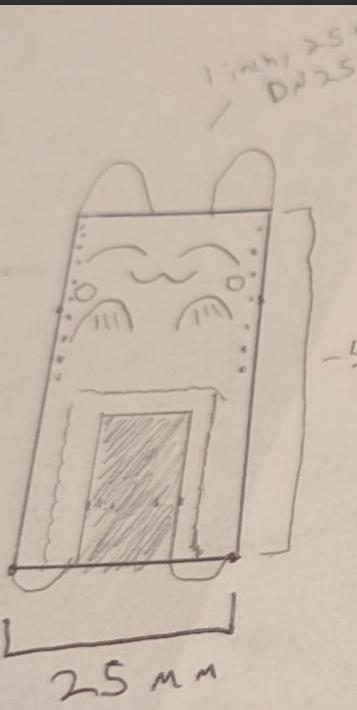
# LoRa Prototyping Powerhouse

- 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

Want one? Usually \$100 online, get a \$75 Bluetooth Nugget + LoRa backpack!

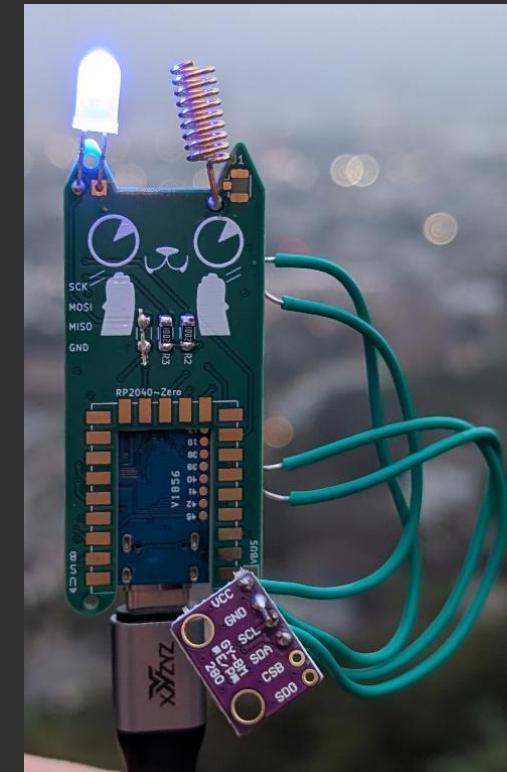


# Making 300 Custom Meshtastic Kits: The Nibble

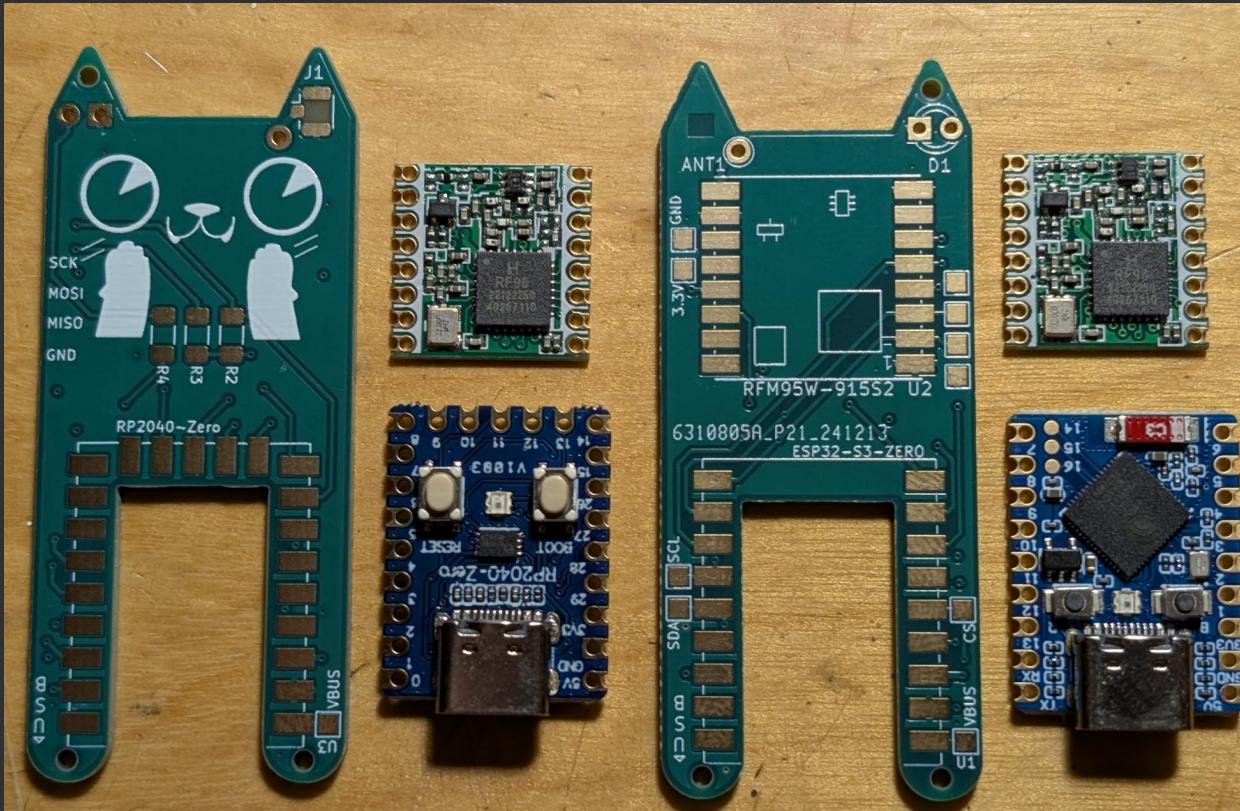


# The Nibble - The Original Cat Meshtastic Node Kit

- 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

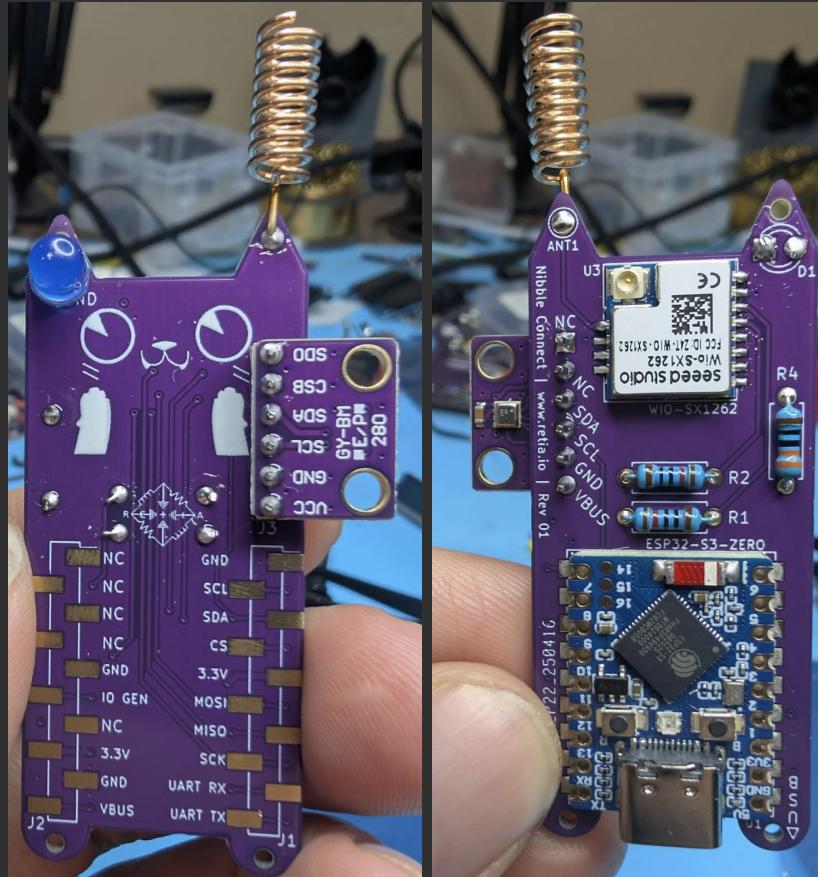


Make your own:  
[github.com/retiallc/nibble](https://github.com/retiallc/nibble)

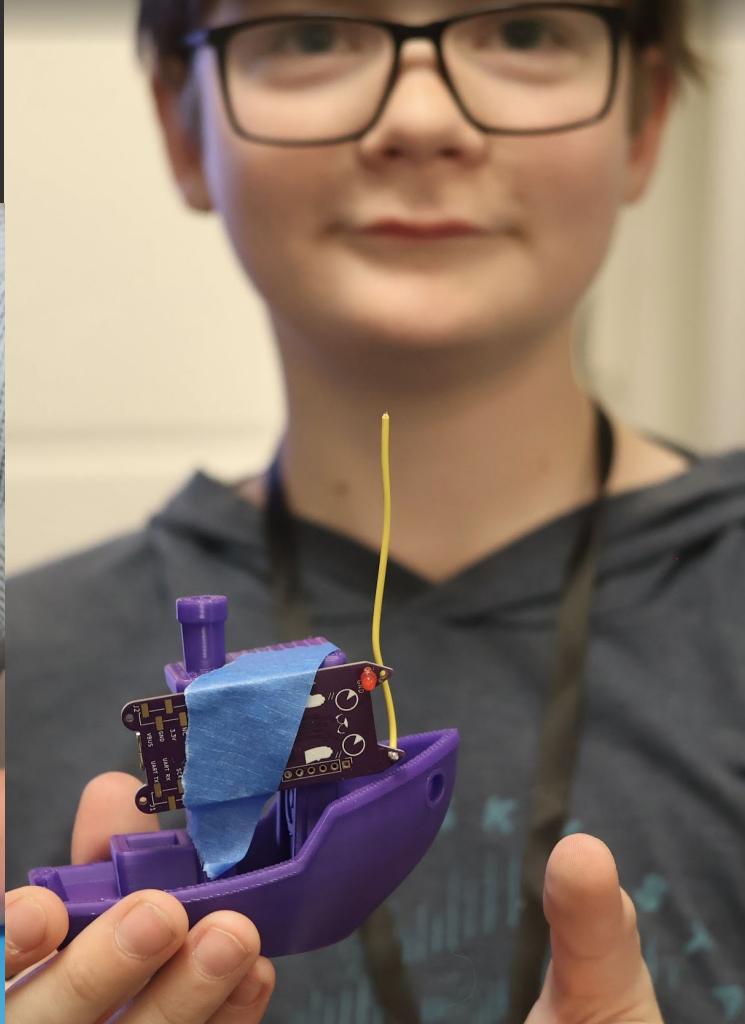
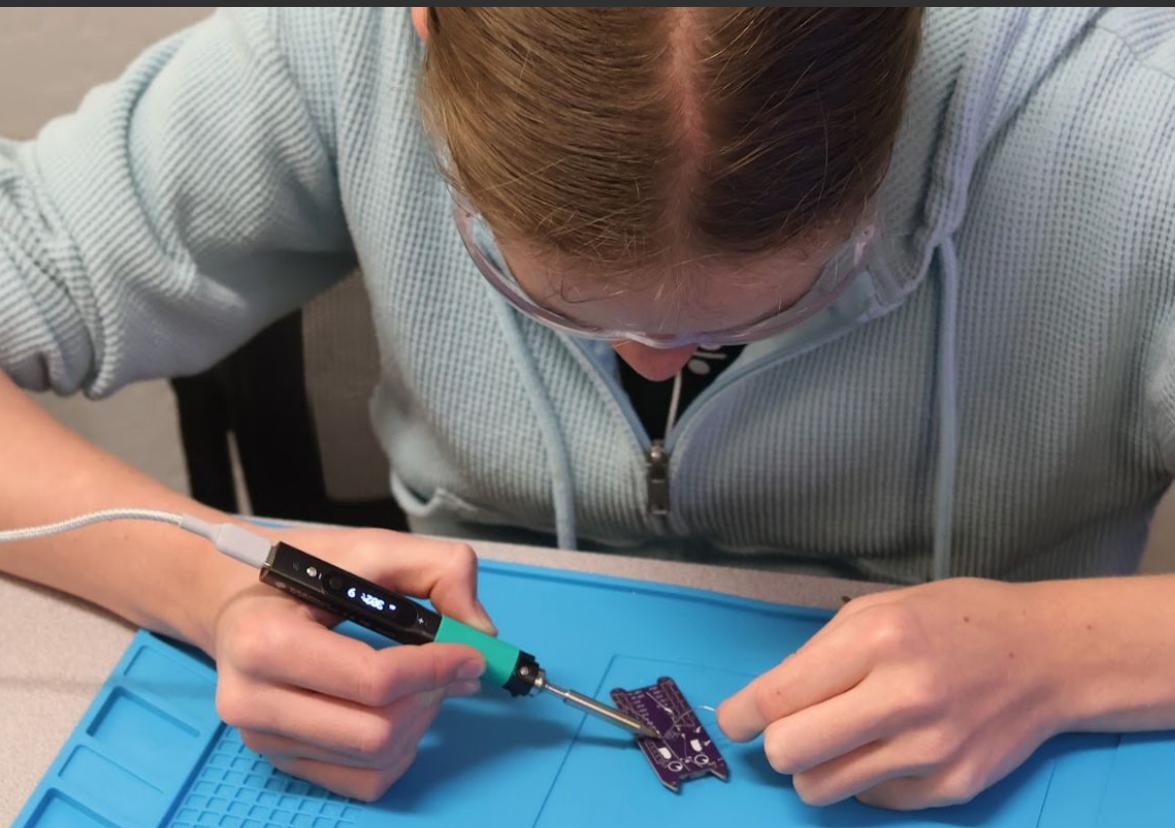
# Update: The Nibble Connect

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



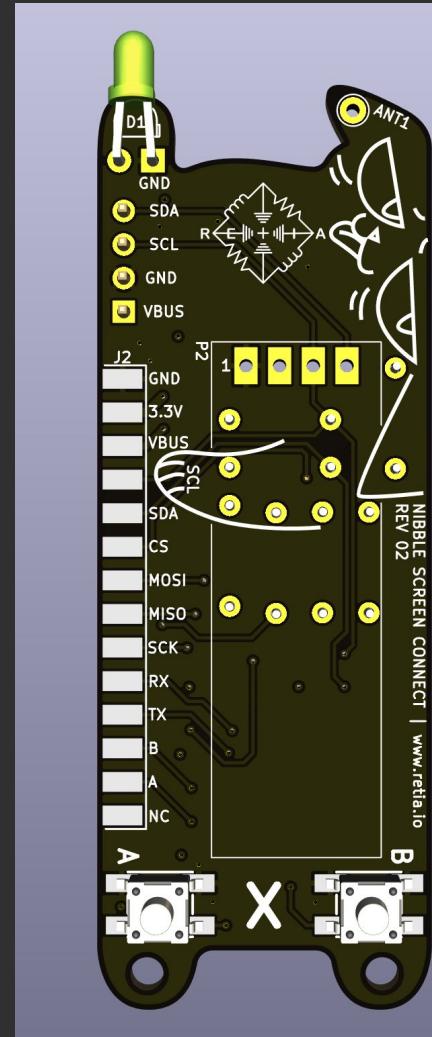
# Update: The Nibble Connect



# For Sec-T: New Nibble Screen Connect

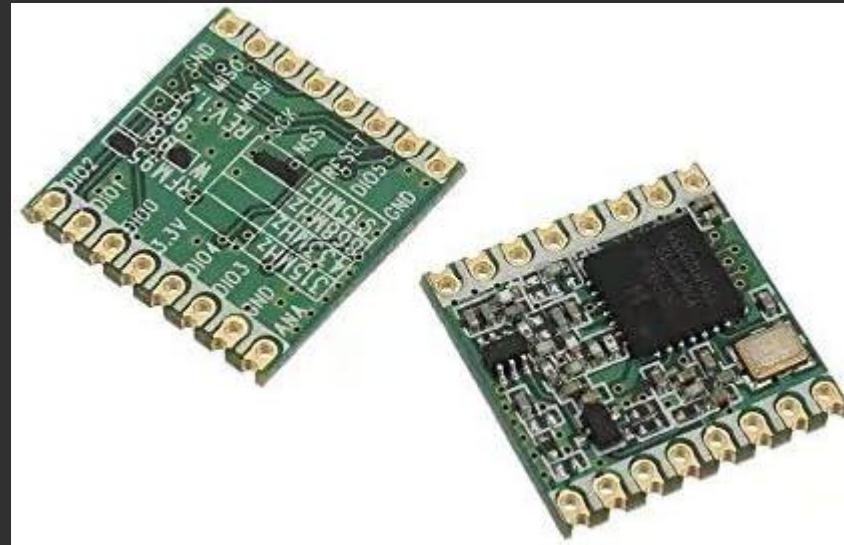
We asked - what about a modular DIY node that can be scaled up or down, & use all the features of Meshtastic?

- 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**



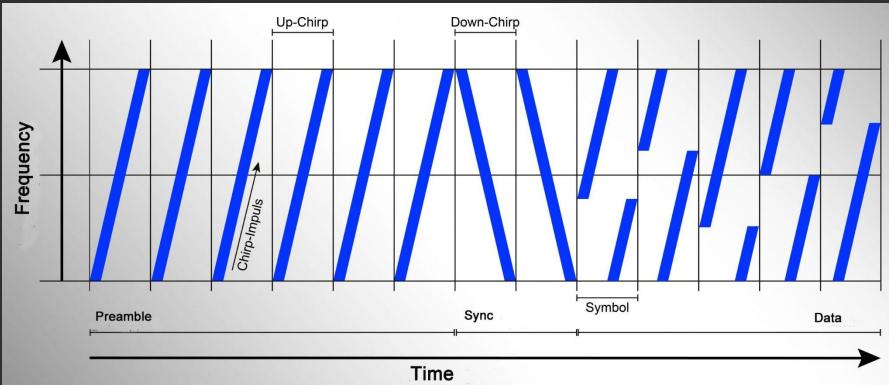
# Introduction to LoRa

- LoRa stands for Long Range, alternative radio standard
- Radios that operate in unlicensed sub-GHz frequency bands
- Slow data rates but very long range
- Not for webcam monitoring, but ideal for telemetry, chat, control signals



# What makes LoRa special?

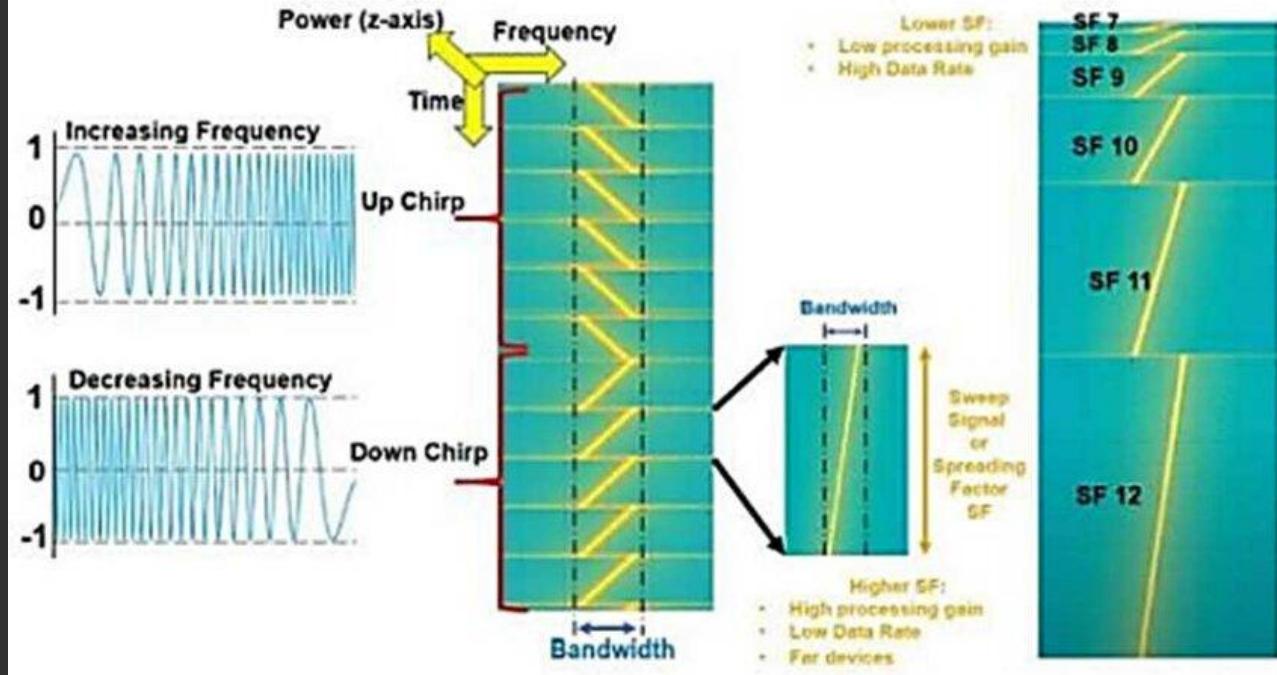
- Long range protocol uses up and down “chirps” to send data
- Chirp Spread Spectrum modulation helps packets survive noise & interference
- Successfully receive & decode packets 20 dB below the noise level
- Receiver sensitivity of a maximum of -149 dBm



Data rate	Sensitivity			
	CC1020	CC1101	SX1261	SX1261
FSK	FSK dBm	FSK	LoRa	
1.2 kBit/s	-117 dBm	-112 dBm	-123 dBm	-129 dBm
2.4 kBit/s	-117 dBm	-111 dBm	-121 dBm	-126 dBm
4.8 kBit/s	-114 dBm	-109 dBm	-118 dBm	-123 dBm
9.6 kBit/s	-112 dBm	-107 dBm	-116 dBm	-120 dBm

Source: interline.pl

# What are chirps?



- Chirp is a signal in which the frequency increases (up-chirp) or decreases (down-chirp) linearly over time
- Encodes information by varying the frequency over a set bandwidth
- Spreads the signal across time & frequency, making it easier to detect
- Require minimal power, ideal for battery-operated IoT devices

# Common types of LoRa Radios

SX1276 - Older LoRa Module used in RFM95

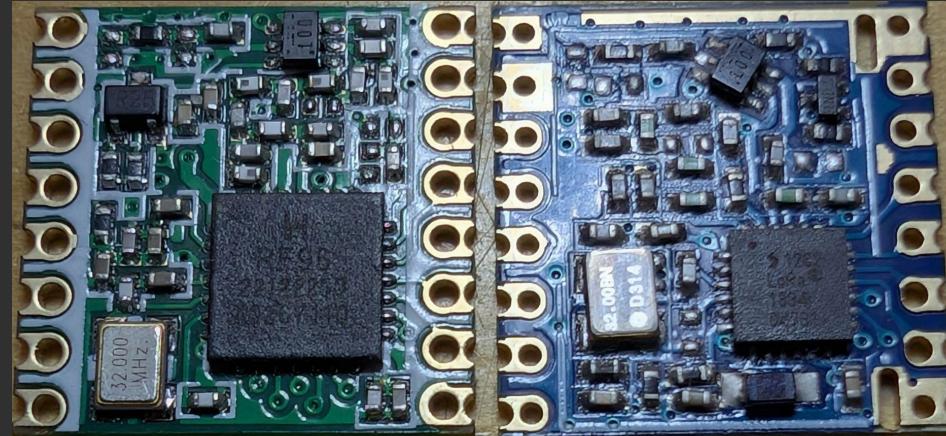
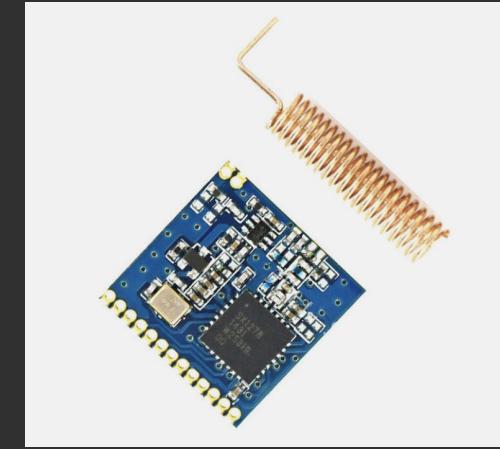
SX1278 - Also older generation

SX1261 - Newer version with improved sensitivity

SX1262- Newer generation, preferred by Meshtastic

Fun Fact: Meshtastic requires radios to work in SPI mode (supported by radiolib).

UART modules are 'too intelligent' & try to handle the radio themselves. The added layer prohibits using the chip for meshtastic

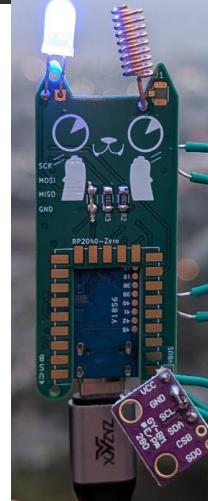
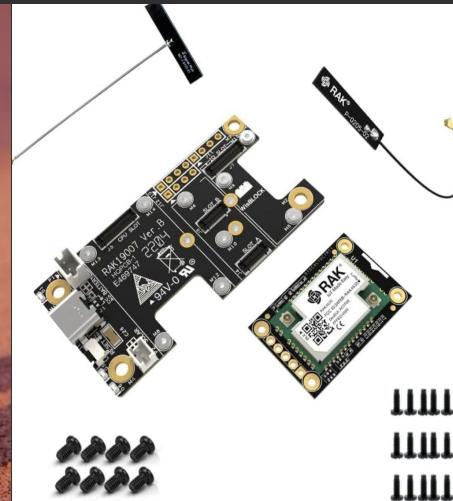


# What other devices can I use?

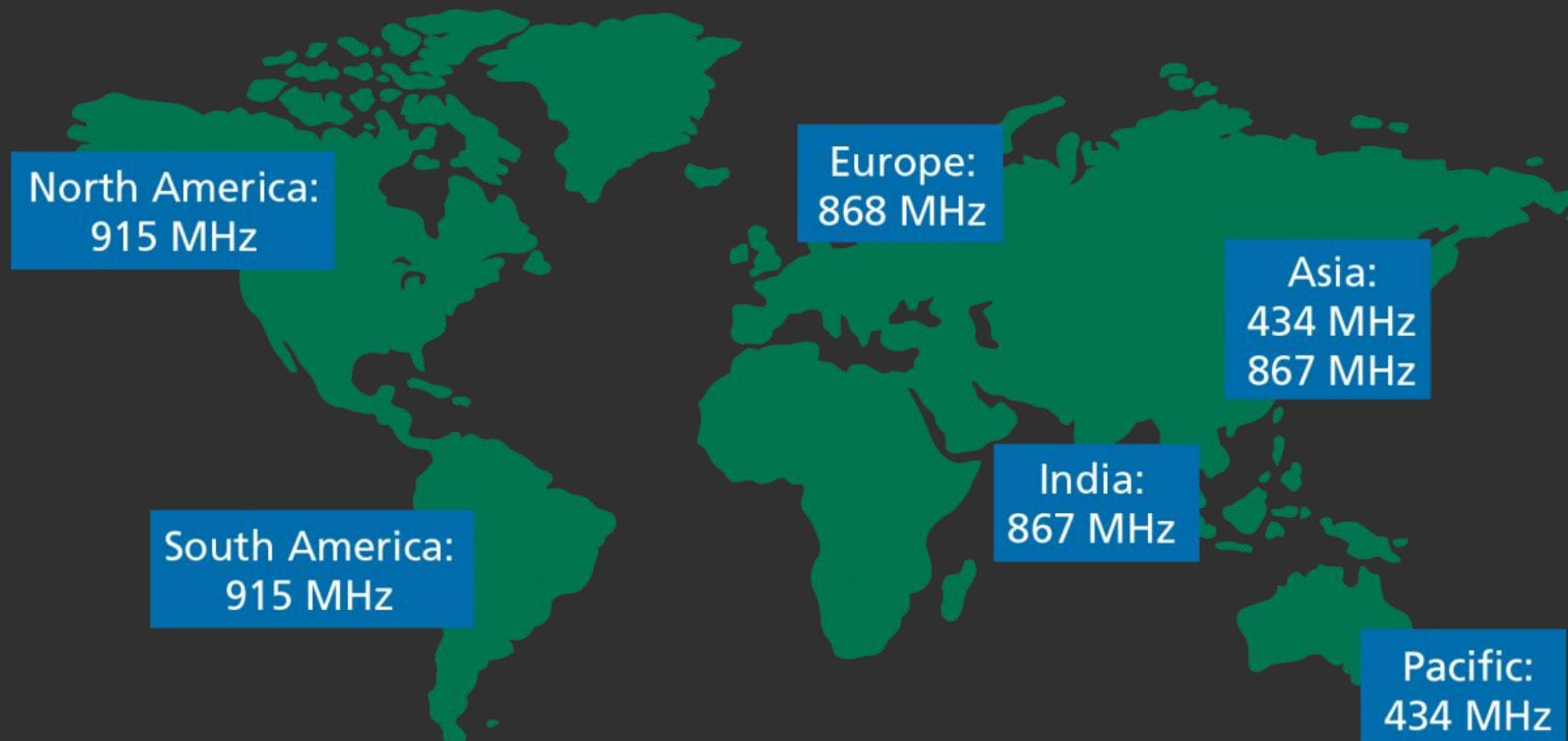
Here are some of the usual suspects:

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

Lilygo T-Deck, Nibble, Nugget



# Regional frequencies - Check your radios!



# What is Meshtastic?

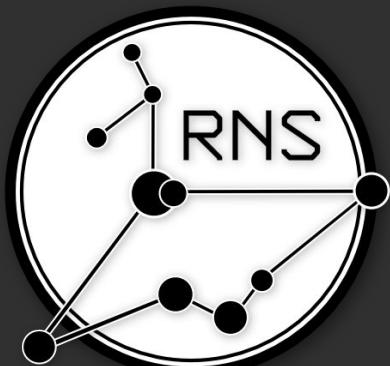
Meshtastic adds encryption, managed flood routing, & convenient apps to LoRa

- Users control nodes via Bluetooth, Wi-Fi, or Serial connection
- Managed with iOS, Android, MacOS or Windows apps
- Can also connect via any chrome browser with WebSerial webapp!
- Encrypts packets & removes headaches of matching radio settings



# 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

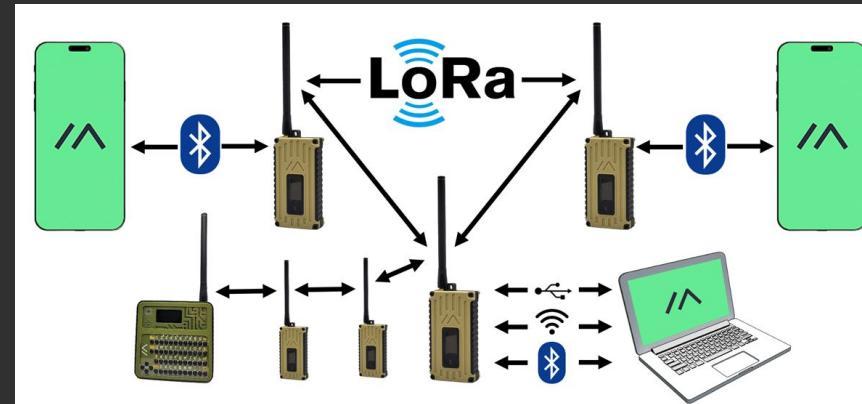
Use our web flasher to install MeshCore firmware on popular devices like the Lilygo T-Deck or Heltec V3 LoRa32, you can also use our web client to communicate securely over radio with other users just using a Chrome web browser - and yes of course it works without the internet!

# 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.

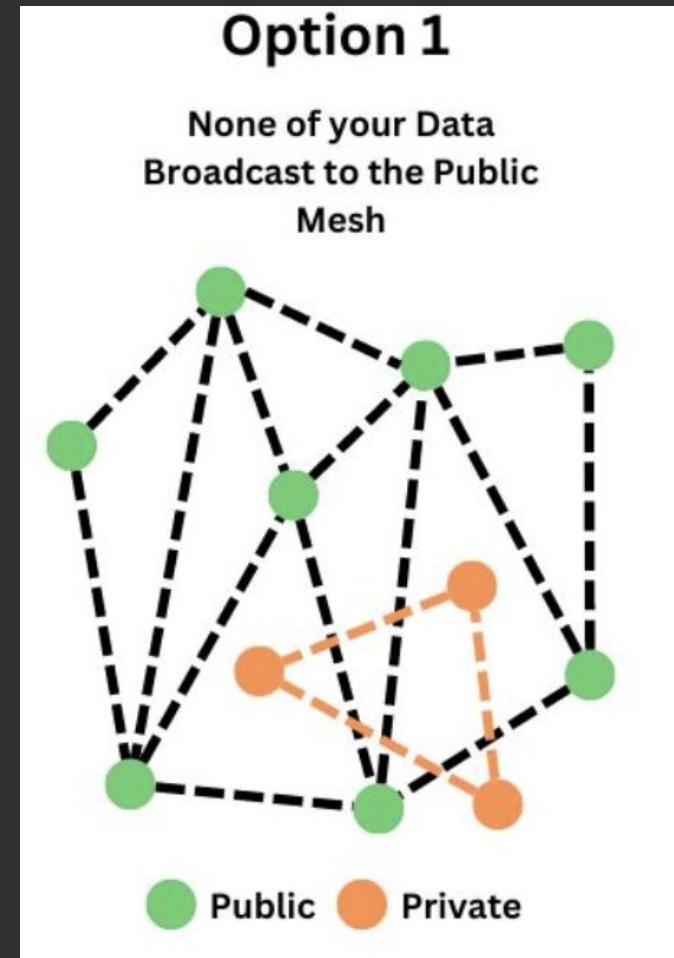


# Public, private, semi-private mesh

## 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

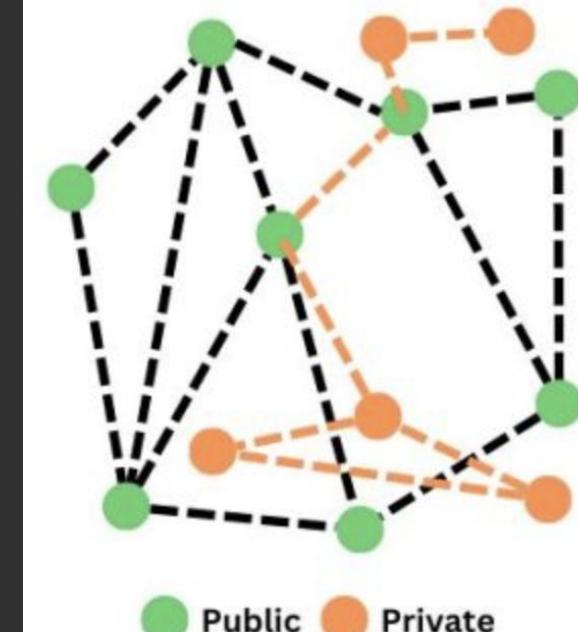
## Option 2: Semi-Private

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.

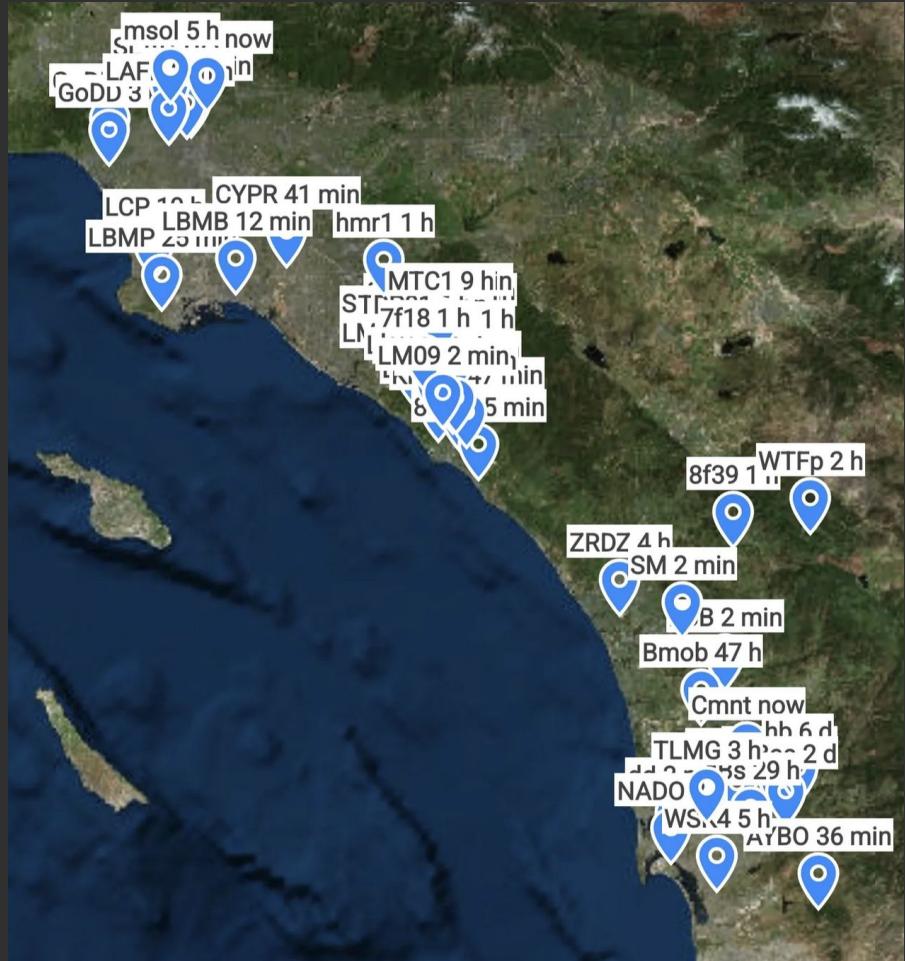
### Option 2

Utilizing the Public Mesh Whilst Remaining Hidden



# Regional Networks

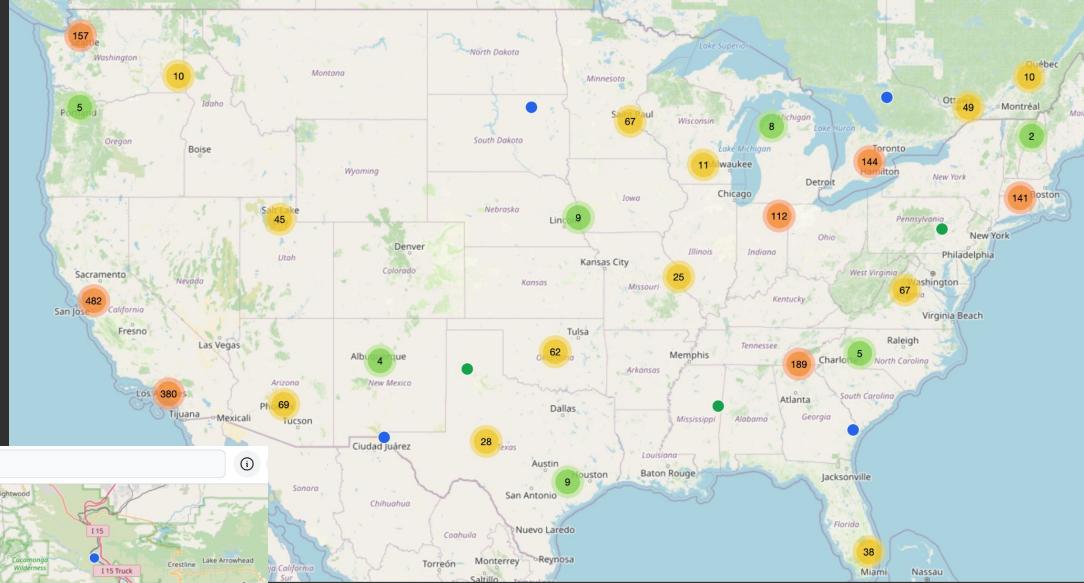
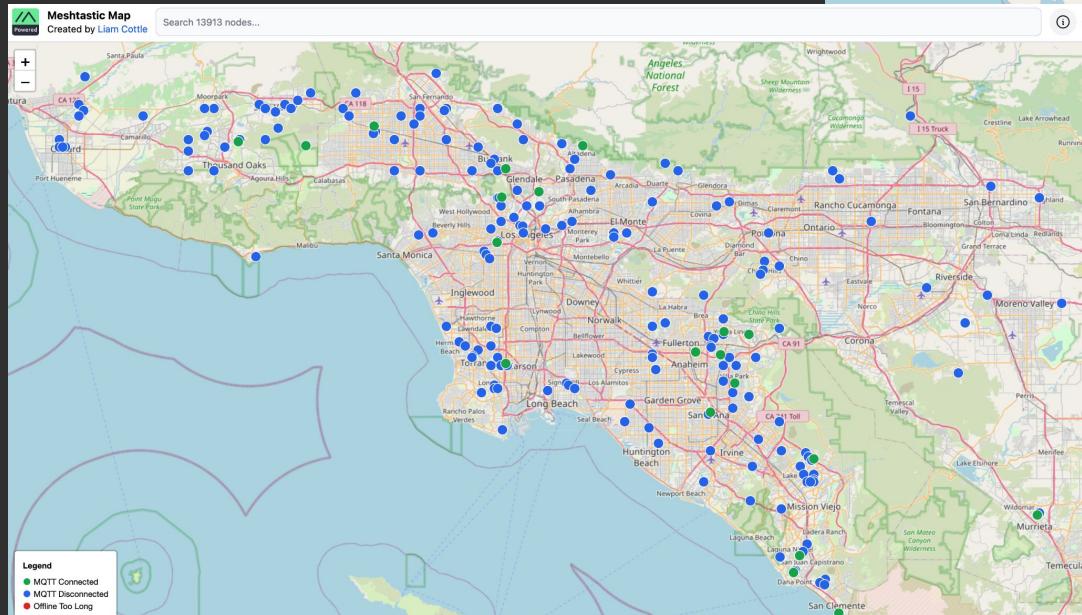
- **SoCal Mesh** - [socalmesh.org](http://socalmesh.org) - San Diego to Santa Clarita
- **Meshtastic Bay Area Group** - [bayme.sh](http://bayme.sh) - San Francisco Bay Area
- **Central Valley Mesh** - [centralvalleymesh.net](http://centralvalleymesh.net) - From Fresno to Santa Barbara



# Real World Mapping

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

Uses local MQTT collectors

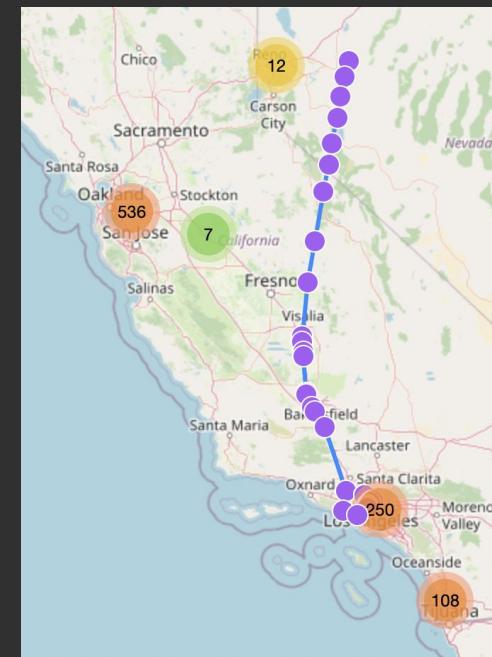
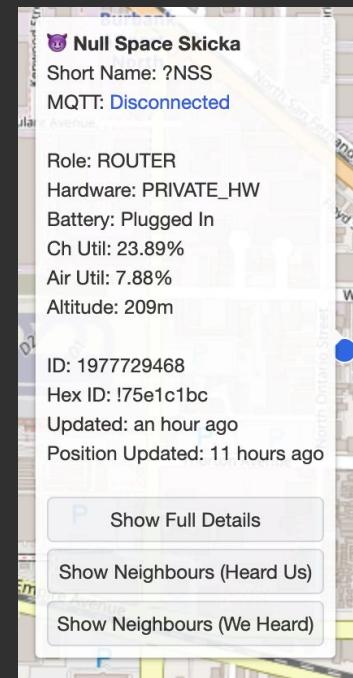
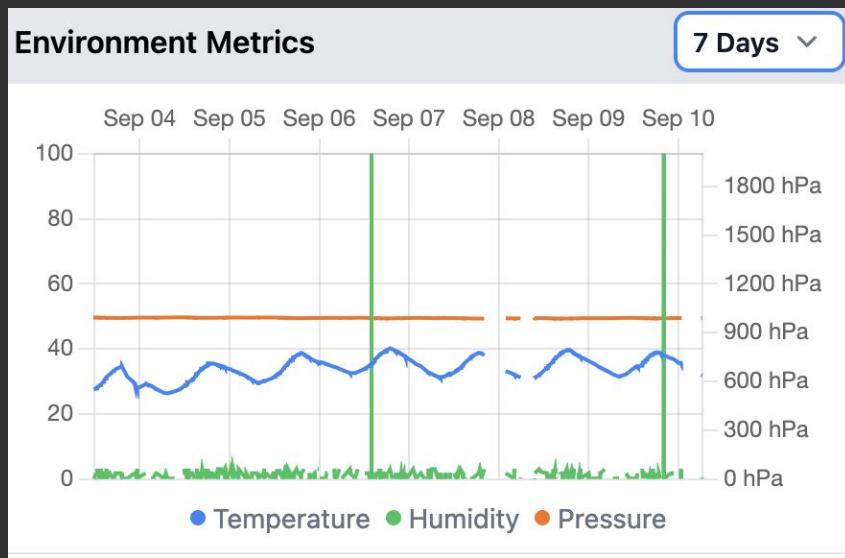


- Collector nodes forward observed nodes over internet to the map via MQTT
- Blind spots, but I can track location of nodes over time
- Tracked my flight!

# Nodes On Default Channel Are Not Private

For nodes in range of collectors:

7 days of sensor, telemetry, and location data



# Encryption Cheat Sheet

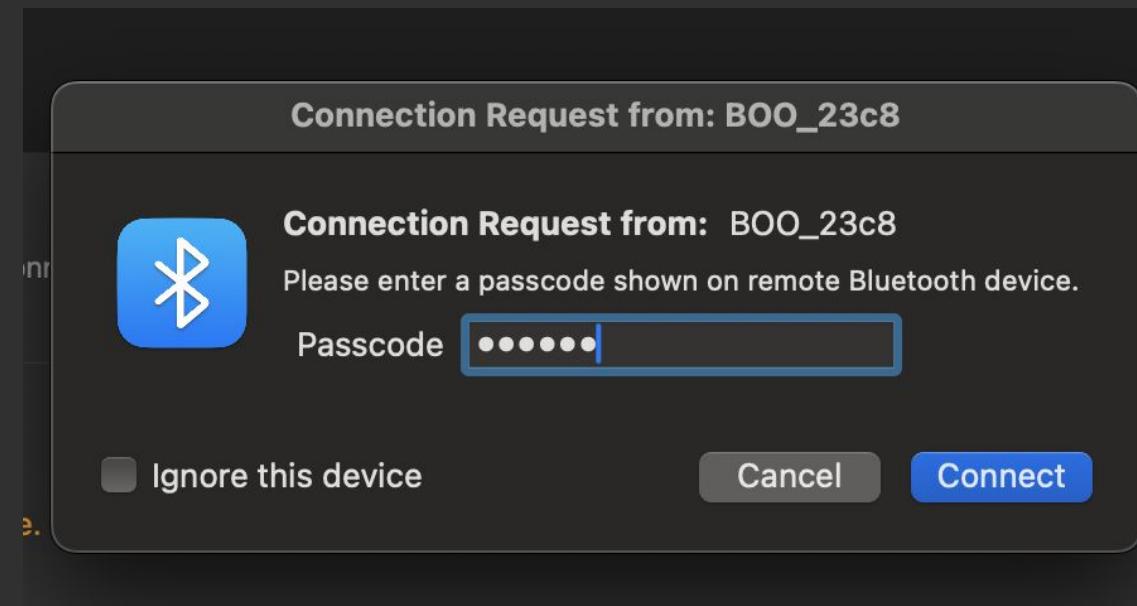
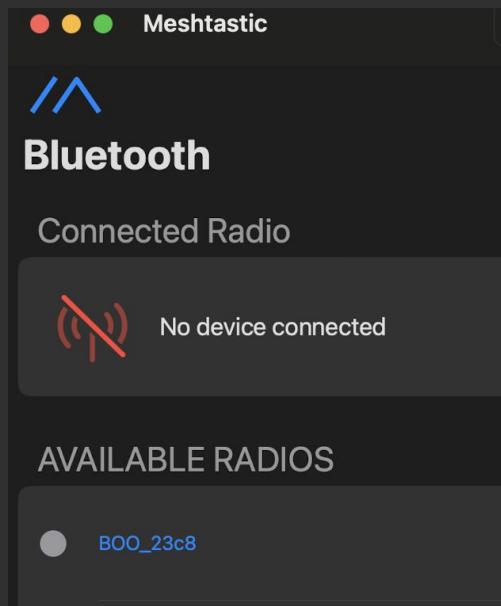
- Nodes communicating on LongFast channel with default encryption key:
  - All nodes can see all messages: unless they are DM's encrypted with keys exchanged between nodes
- Nodes communicating on LongFast channel with secret encryption key:
  - All nodes who know the channel key can read all messages, 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

# How do we compile it?



# Now Connect Via Bluetooth

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



# 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

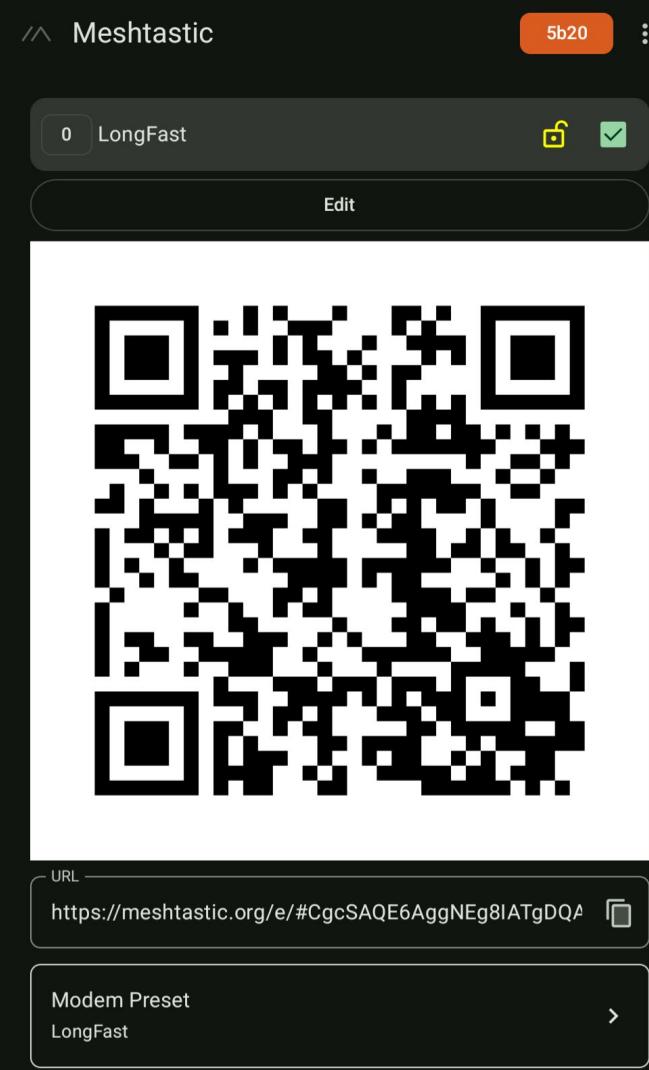
 Presets  
Available regions:

 Available regions:  
ADVANCE

# Let's Enable Our Radio!

Scan these settings to go to the default LongFast channel in the EU

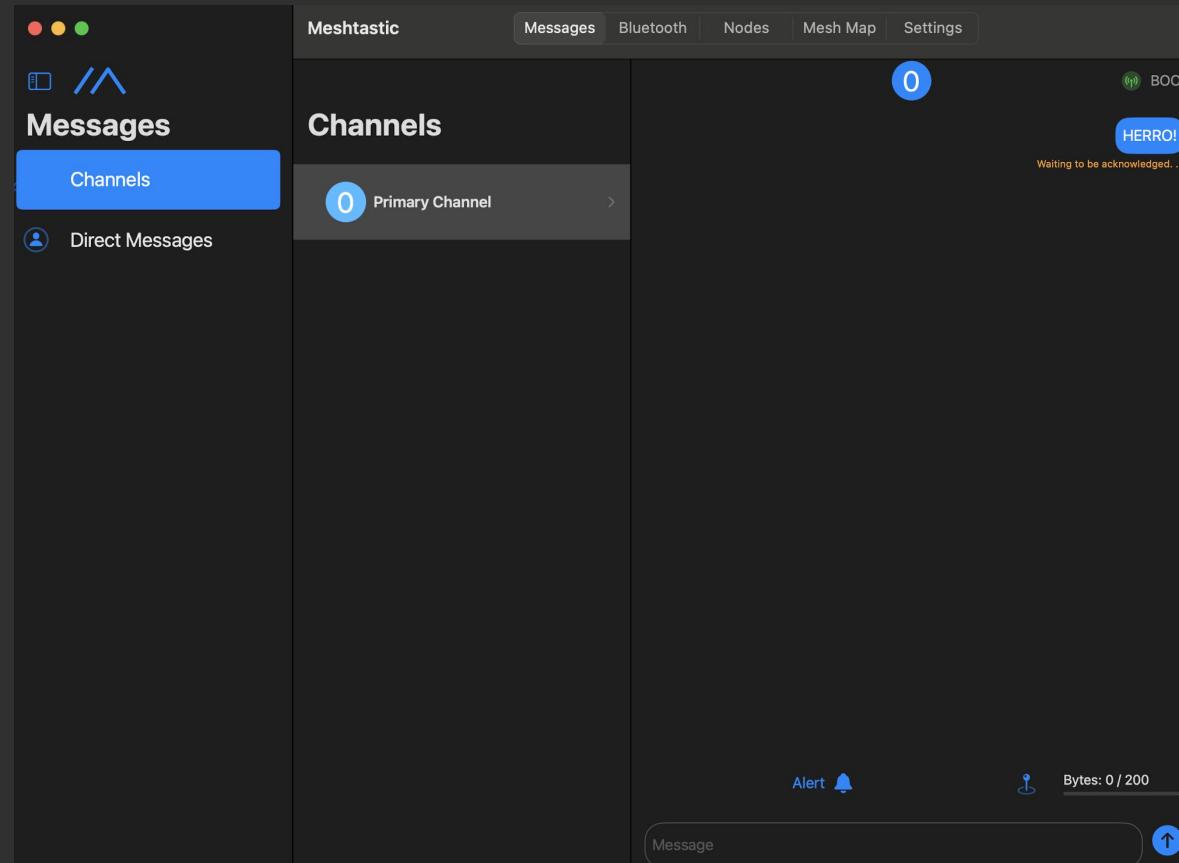
Make sure you see the option to “replace” all channels.



# 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, there are three colored dots (red, yellow, green) and the text "Meshtastic". In the top right corner, it says "TaMu1" with a small square icon.

The main area is titled "Nodes (11)" with a search bar below it. A button labeled "Find a node" with a magnifying glass icon is also present. Below the search bar, there is a list of nodes:

- YOUR\_-\_NAME\_HERE (Blue circle, Bluetooth Connected, 12/29/2024, 4:05 PM, Role: Client)
- drc 7489 (Green circle, 12/29/2024, 4:06 PM, Role: Client, Signal Good)
- Meshtastic 9068 (Blue circle, 12/29/2024, 4:06 PM, Role: Client, Signal Good)
- Nicolas (Green circle, 12/29/2024, 4:05 PM, Role: Client Mute, Signal Fair)
- AAAAA (Green circle, 12/29/2024, 4:05 PM, Role: Client, Signal Good)
- Meshtastic 329c (Blue circle, 12/29/2024, 4:05 PM, Role: Client, Signal Good)
- ricl (Purple circle, 12/29/2024, 4:04 PM, Role: Client, Signal Good)
- Clem (Maroon circle, 12/29/2024, 4:04 PM, Role: Client Mute, Signal Good)
- TaMu1 (Yellow circle, 12/29/2024, 4:04 PM, Role: Client, Signal Good)

Below the node list, there is a section titled "HARDWARE" which shows an "Unsupported" device (with a red X icon) and a "Model" device (LILYGO T-Echo).

The right side of the screen has a "NODE" section with detailed information for the "TaMu1" node:

Detail	Value
#	Node Number
User Id	!66c408f9
Role	Client
First heard	2 minutes ago
Last heard	2 minutes ago

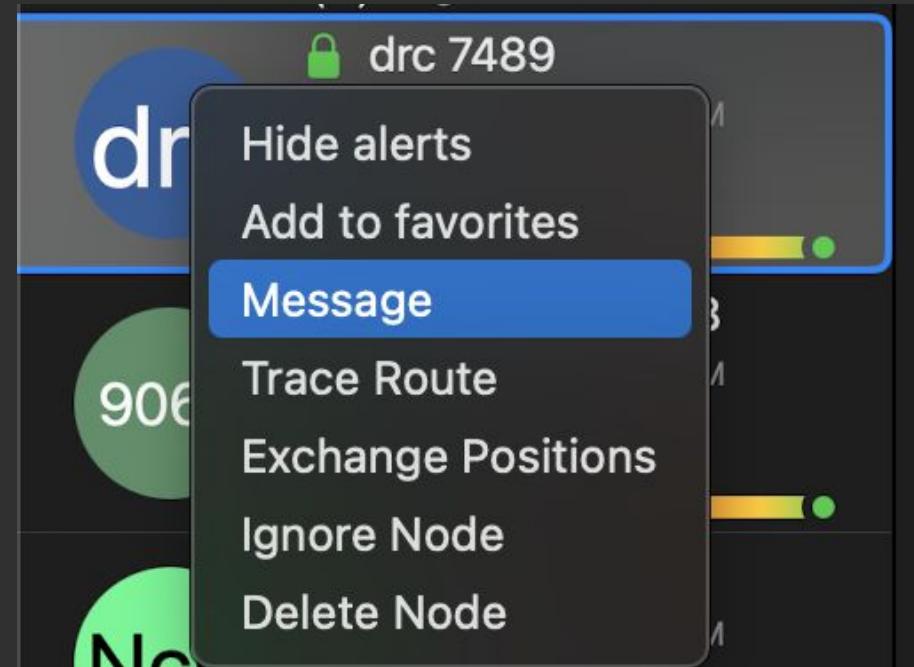
At the bottom, there is a "LOGS" section with links to various logs:

- Device Metrics Log
- Node Map
- Position Log
- Environment Metrics Log
- Trace Route Log
- Detection Sensor Log

# 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

---

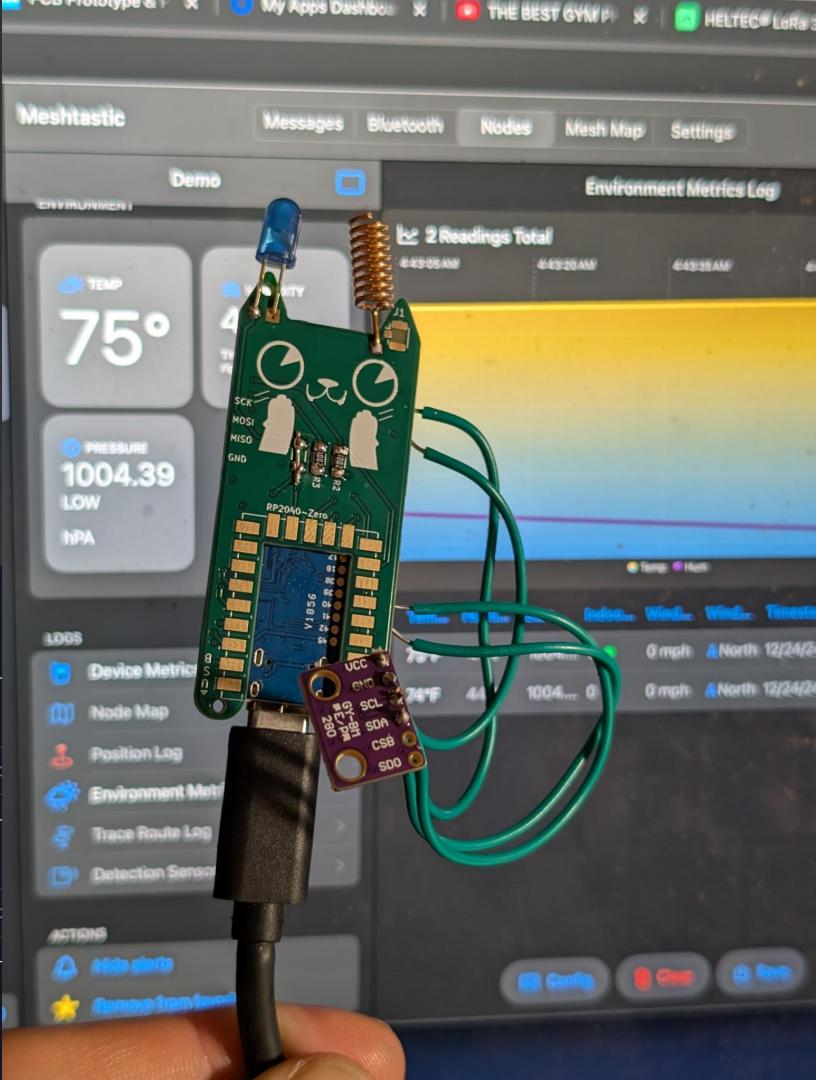
**Query Interval**      Interval to get telemetry data  
0      Seconds

---

**Update Interval**      How often to send Metrics over the mesh  
0      Seconds

---

**Module Enabled**      Enable the Environment Telemetry



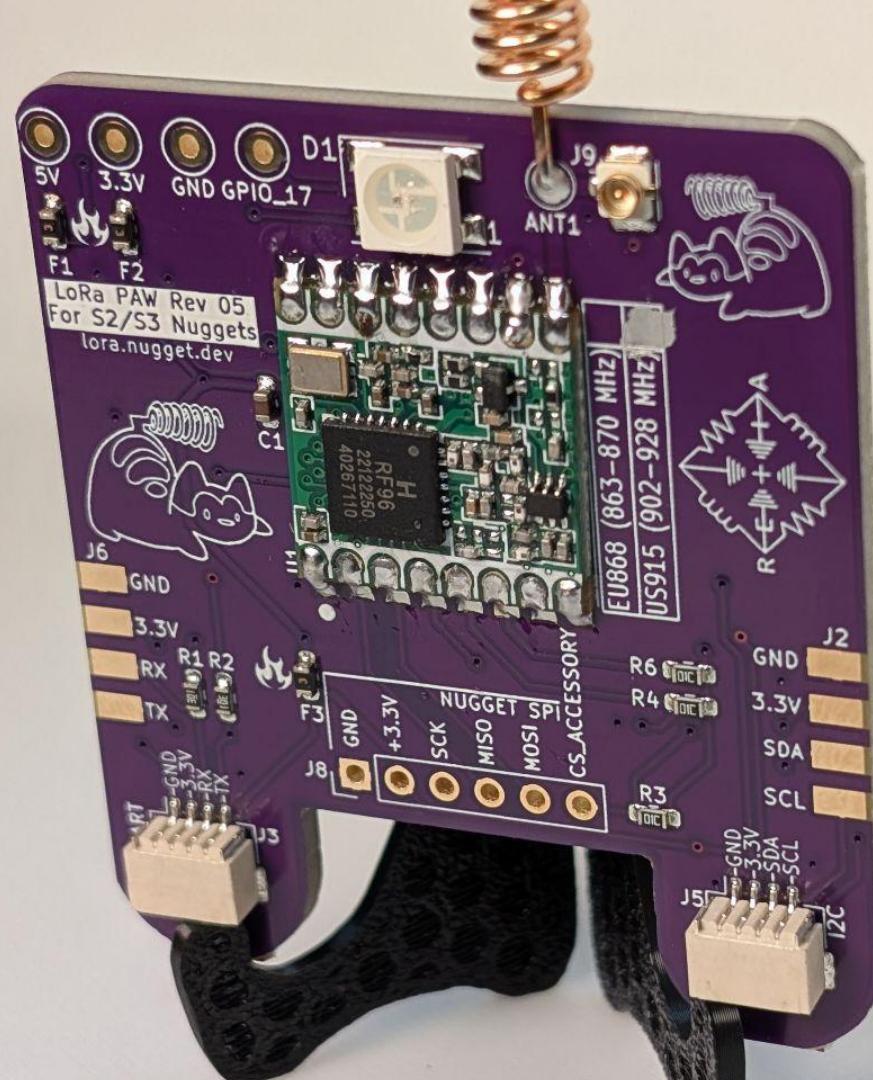
## Currently Supported Sensor Types

# 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

# Also Supported:

- UART devices like GPS for creating trackers or doing range tests
- SPI devices like a screen or other sensors
- Analog out devices like presence sensors
- Notification LED's
- Neopixel Strips!



# 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.

# 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
```

# Man In The Middle

Any node retransmitting a message on LongFast can see:

- Traceroutes
- Direct messages between nodes
- Currently broadcasted location
- Other nodes in the traceroute chain

## Trace Routes

Only 5 most recent are shown

FBI Open UP to Taylor Mountain W4  
2 days ago - 4 hops on LongFast

FBI Open UP to Meshtastic bc86  
2 days ago - 4 hops on LongFast

FBI Open UP to Meshtastic bc86  
2 days ago - 4 hops on LongFast

FBI Open UP to Meshtastic bc86  
2 days ago - 4 hops on LongFast

FBI Open UP to Meshtastic bc86  
2 days ago - 4 hops on LongFast

# Attacks Against Meshtastic

Several attacks have been demonstrated against Meshtastic:

- 1) Malicious router sinkhole
- 2) Adding lock emoji to unencrypted node names
- 3) Bee movie script spamming
- 4) Soft-ban users via sequence spoofing
- 5) Node reflection/replay attack - spam many fake nodes
- 6) Change the long and short name of a target radio remotely
- 7) Memory exhaustion of stored nodes/encryption keys
- 8) PSK-only channel spoofing - spoof messages from other users in group chat

...That one time at DEF CON!

August 12, 2025 · 7 min read



Meshtastic Team

# Attacks Against Meshtastic: Malicious Router Sinkhole

Setting up many routers in bad areas can “trap” or sinkhole packets in a local area

Exhausts hops by transmitting between routers before it can hop elsewhere

Router settings are often disabled at conferences to prevent this attack from happening accidentally.

# Attacks Against Meshtastic: Lock Emoji

A simple attack - just adding a lock emoji to a node's long name to suggest it's using encryption.

Relies on a simple trick to fool users.

# Attacks Against Meshtastic: Spammers

At LayerOne, someone transmitted the Bee Movie and Shrek scripts continuously on the conference network.

Nearby packets relayed the attack, causing it to spread and take up all available bandwidth.

Members of the Meshtastic security discord were dismissive and uninterested in fixing the issue.



spudgunman ✨ 5/30/25, 5:36 AM

Yea go ask a sovereign citizen about it, this isn't the spot



Lou DiMaggio 5/26/25, 1:35 PM

Someone there was spamming the Bee Movie script and made the mesh impossible to use

```
# Welcome  Meta_Bee_Spammer.py  Shrek_Spammer.py 
Users > skrika > Documents > Shrek_Spammer.py > send_chunks_on_node
#!/usr/bin/env python
#
# Import os
# Import time
# Import requests
# Import serial.tools.list_ports
# Import serial
# Import meshtastic.serial_interface
# From threading import Thread
#
# Configuration
SCRIPT_URL = "https://raw.githubusercontent.com/Liuhemtricks/5FB342659FF1341b9e4f32bca7a26d4/raw/3c1da7397573b53aaef424d9ff1080238785ef/shrek.txt"
CHUNK_SIZE = 100
REPEAT_DELAY = 10
TIME_SLEEP = 1 # Skip first 85 chunks
RECONNECT_DELAY = 5 # Seconds between reconnect attempts after failure
#
def download_script():
    print("Downloading script...")
    response = requests.get(SCRIPT_URL)
    if response.status_code == 200:
        with open(SCRIPT_PATH, 'w', encoding='utf-8') as f:
            f.write(response.text)
    else:
        raise RuntimeError(f"Failed to download script from GitHub!")
#
def load_script_chunks(path):
    if path == None or not os.path.exists(path):
        download_script()
    with open(path, 'r', encoding='utf-8') as f:
        text = f.read().replace("\r\n", "\n").replace("\r", "\n")
    return [text[i:i+CHUNK_SIZE] for i in range(0, len(text), CHUNK_SIZE)]
#
def find_meshtastic_ports():
    return [
        port.device for port in serial.tools.list_ports.comports()
        if "meshtastic" in port.device or "usbserial" in port.device or "ttyUSB" in port.device
    ]
#
def send_chunks_on_node(port, chunks, offset, stride):
    while True:
        try:
            if port == meshtastic.serial_interface.SerialInterface(devPath=port):
                print(f"({port}) Connected")
            while True:
                for i in range(offset + START_CHUNK_OFFSET, len(chunks), stride):
                    if chunk == None:
                        chunk = chunks[i]
                    if chunk == None:
                        print(f"({port}) Send error: ({e})")
                    time.sleep(0.1)
                    time.sleep(TIME_SLEEP)
                except Exception as e:
                    print(f"({port}) Connection lost or failed: ({e})")
                    print(f"({port}) Retrying in {RECONNECT_DELAY} seconds...")
                    time.sleep(RECONNECT_DELAY)
        except KeyboardInterrupt:
            print(f"({port}) Shutting down gracefully...")
            time.sleep(0.1)
            threads.append(t)
        except:
            print(f"({port}) Unknown error: {e}")
            time.sleep(0.1)
            threads.append(t)
    if __name__ == "__main__":
        main()
```

# Attacks Against Meshtastic: Softban Sequence Spoofing

Also at LayerOne, a softban attack was demonstrated.

Meshtastic packets use sequence numbers to prevent rebroadcasting the same message.

An attacker looking to silence a target node could spoof packets pretending to be from the target, but with sequence numbers that haven't happened yet.

When nearby nodes get a packet from the attacker pretending to be the victim, they will ignore messages from the victim thinking they've already seen those packets.

Effectively blocks nearby nodes from retransmitting a certain node.

# Attacks Against Meshtastic: Node Reflection/Replay

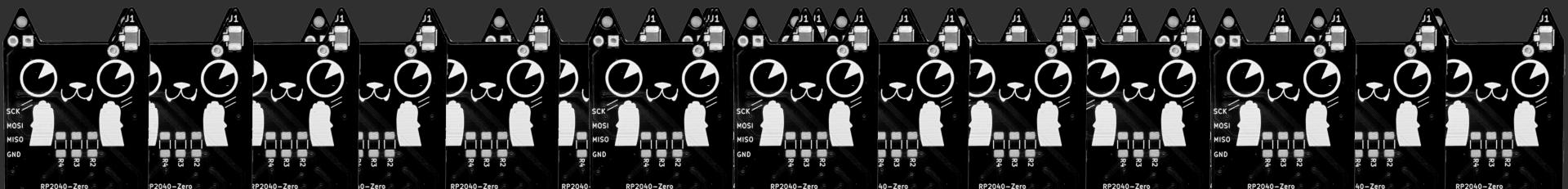
The node reflection attack listens for Meshtastic nodes announcing themselves, then spoofs more of those announcements.

The result is many fake nodes, all with the same name.

This can exhaust the number of node slots (100) by filling them with fake nodes

It becomes impossible to find real nodes, and you don't know who you are messaging.

Must know the channel key to do this attack, but highly disruptive.

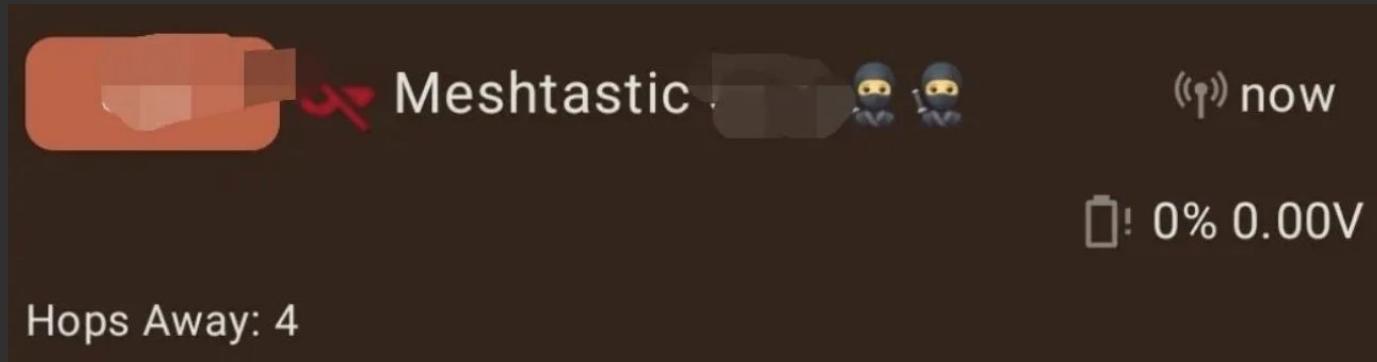


# Attacks Against Meshtastic: Change Node Name

At Defcon, an attack was demonstrated where an attacker can change a target's long and short node name.

This is done by spoofing a “nodeinfo” packet pretending to be from the target itself, which the node receives and uses to update it's long and short name.

This allowed hackers to append a ninja emoji to other's node names!



# Attacks Against Meshtastic: Memory Exhaustion

Nodes can only hold about 100 other devices in memory

If you spoof many fake nodes, the real nodes may be pushed out of the slots, preventing discovery of any real nodes on the mesh

You can favorite nodes, but otherwise may not be able to tell the difference between a real and fake node if an attacker can completely fill the memory slots with junk nodes.

# Attacks Against Meshtastic: PSK Channel Spoofing

In channels that use a PSK, messages are not signed

That means anyone who knows the PSK can send messages appearing to be from anyone else in the channel.

Attackers can spoof messages from other users. Combined with the soft ban attack, this could lead to impersonation.

# Attacks Against Encryption

Deep learning can take a public key and create enough samples to train a model on.

The model can potentially recover text after being trained on text generated from your public key

By design, your public key is exposed

## Deep-Learning-based Cryptanalysis through Topic Modeling

Kishore Kumar, Sarvesh Tanwar, Shishir Kumar • Published in Engineering, Technology & Computer Science • Engineering, Technology & Applied Science Research

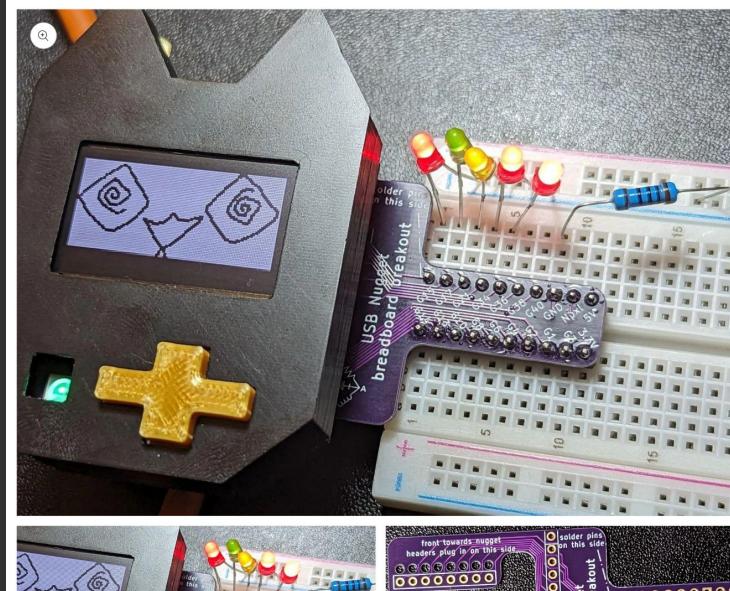
**TLDR** This study addresses chosen plaintext cryptanalysis within public key cryptography, to categorize topics of encrypted text and proposes a framework that identifies the topic associated with ciphertext, using diverse plaintexts and their corresponding cipher texts.

**Abstract** Neural cryptography is a technique that uses neural networks for secure data encryption. Cryptoanalysis, on the other hand, deals with analyzing and decrypting ciphers, codes, and encrypted text without using a real key. Chosen-plaintext cryptanalysis is a subfield of cryptanalysis where both plain text and ciphertext are available and the goal is either to find the encryption technique, the encryption key, or both. This study addresses chosen plaintext cryptanalysis within public key cryptography, to categorize topics of encrypted text. Using a fixed encryption technique and key, the focus was placed on creating a framework that identifies the topic associated with ciphertext, using diverse plaintexts and their corresponding cipher texts. To our knowledge, this is the first time that chosen-plaintext cryptanalysis has been discussed in the context of topic modeling. The paper used deep learning techniques such as CNNs, GRUs, and LSTMs to process sequential data. The proposed framework achieved up to 67% precision, 99% recall, 80% F1-score, and 71% AUPR on a dataset, showcasing promising results and opening avenues for further research in this cryptanalysis subarea.

# Teach a friend!

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

- Nuggets
- Add-ons
- Online classes



RETIA.IO

**USB Nugget  
Breadboard Tail  
Breakout**

\$15.00 USD  
Tax included.

Quantity

- 1 +

Add to cart

Buy with  

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)

