Programming notes

Setup: updated 7.7.25

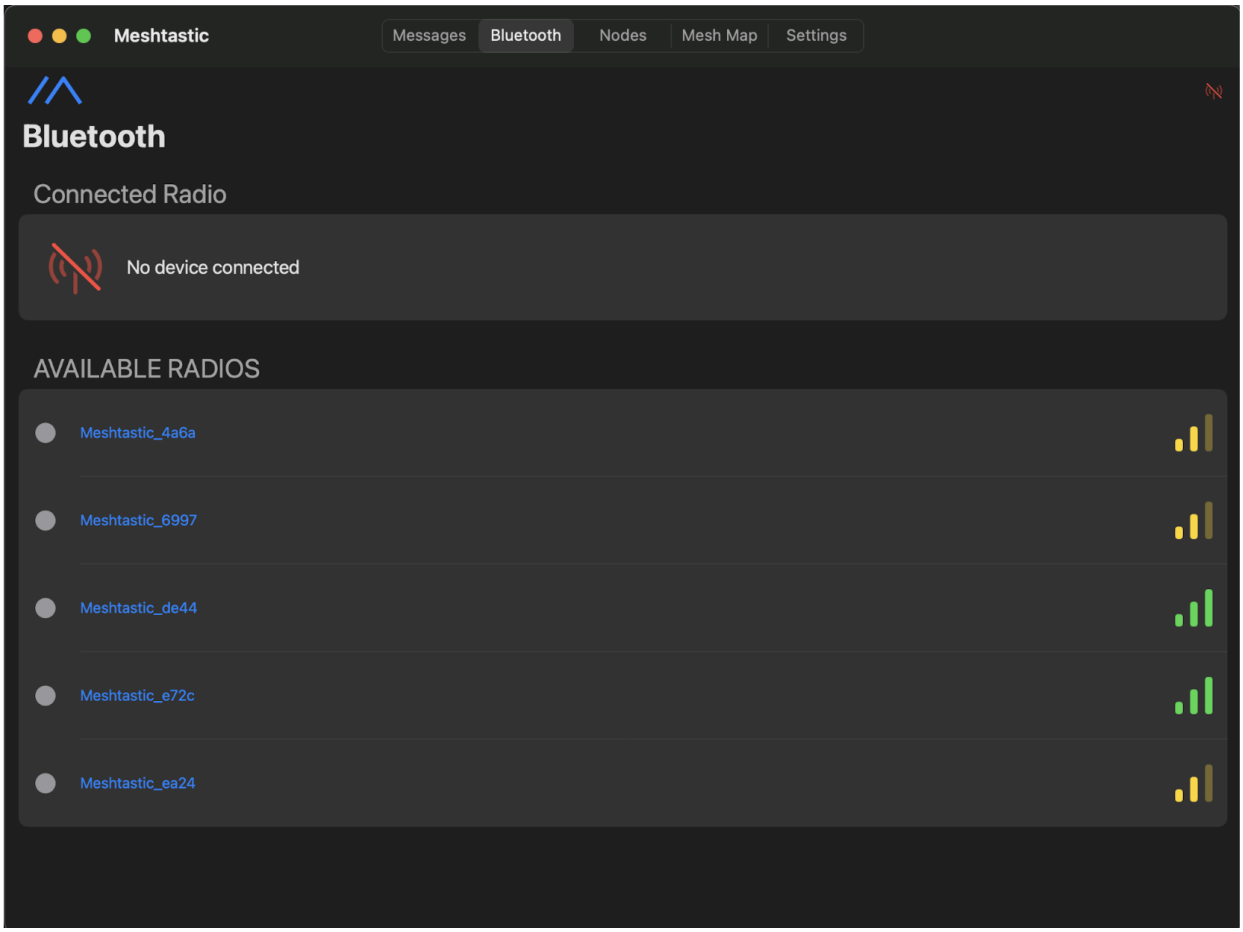
Here are a set of screenshots from a typical mesh setup I have.
Most devices fall into one of two categories: RAK 19007 boards and ESP32 boards. Both can be programmed initially using a chrome browser to flash and update the firmware.
After initial flashing, you can use either bluetooth or the client.meshtastic.org chrome client to configure the device.
n.b. the configuration options on the chrome client leave out some options found on the bluetooth meshtastic app.

My out of the box process:

1. connect antennas if present (critical: if you power up the device without, it can damage the RF chip)
2. connect via usb to computer running chrome (mac or pc, must be newer)-see below
3. update firmware using the app from meshtastic.org (include web server for ESP32 units)
4. remove from computer, power externally, join by bluetooth use app (iPhone, iPad, m1 mac, android)
5. set LoRa region (below)
6. if unit has a screen, go to bluetooth and change code to static (123456) or "no code, just works" which is much better.
7. enable MQTT messaging in LoRa and the primary channel
8. enable MQTT under modules if you like

We'll be using the meshtastic app on a mac (m1 "Apple siicon" or newer) computer, which looks the same on my iPhone. Don't have access to android, but it is similar.

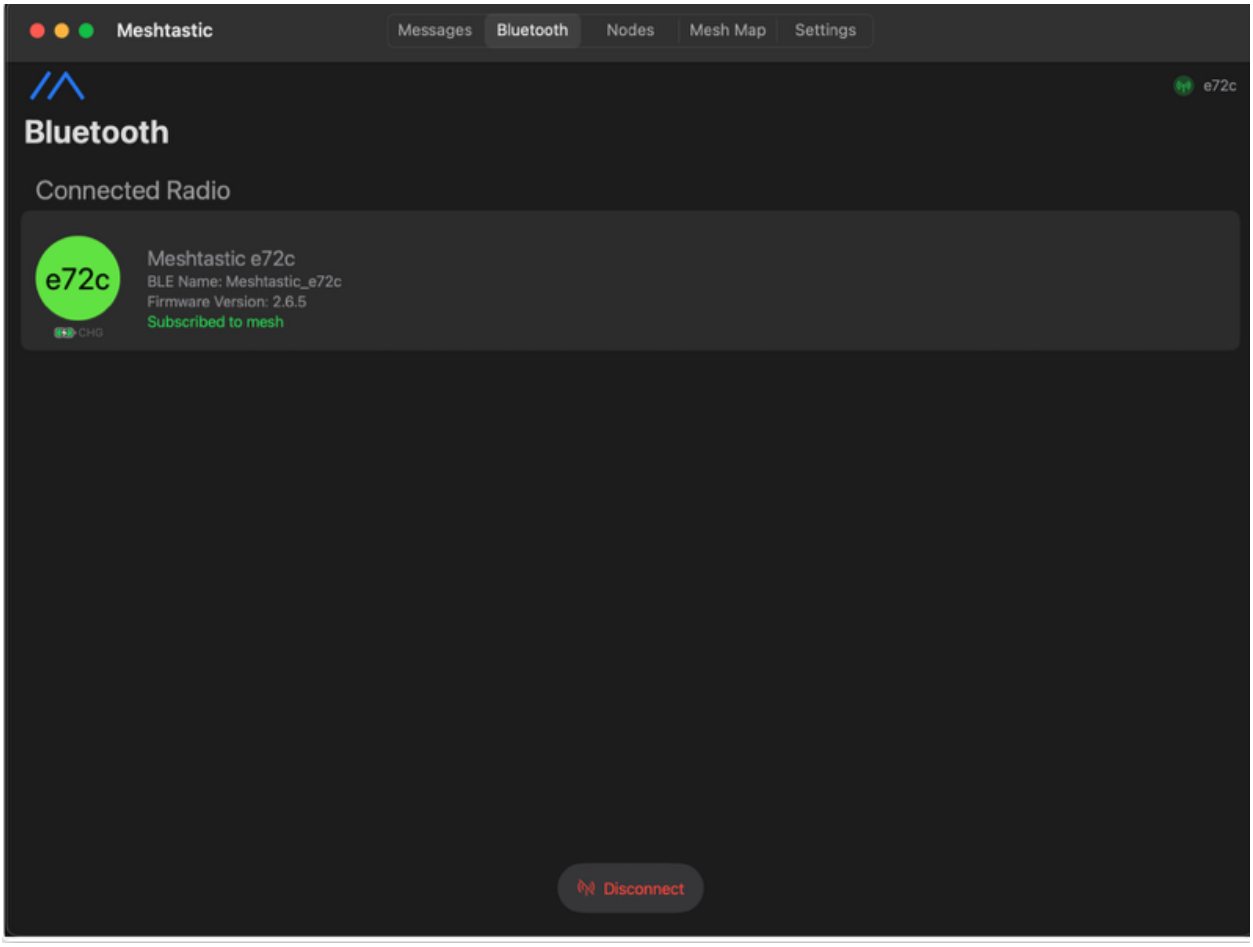
We will be using the 2.6.x version of firmware (see below)
You can also configure using chrome (web, bluetooth, serial) which I'll cover next.
Download the meshtasic app, should see this:



Screenshot 2025-07-07 at 4.39.24 AM.png

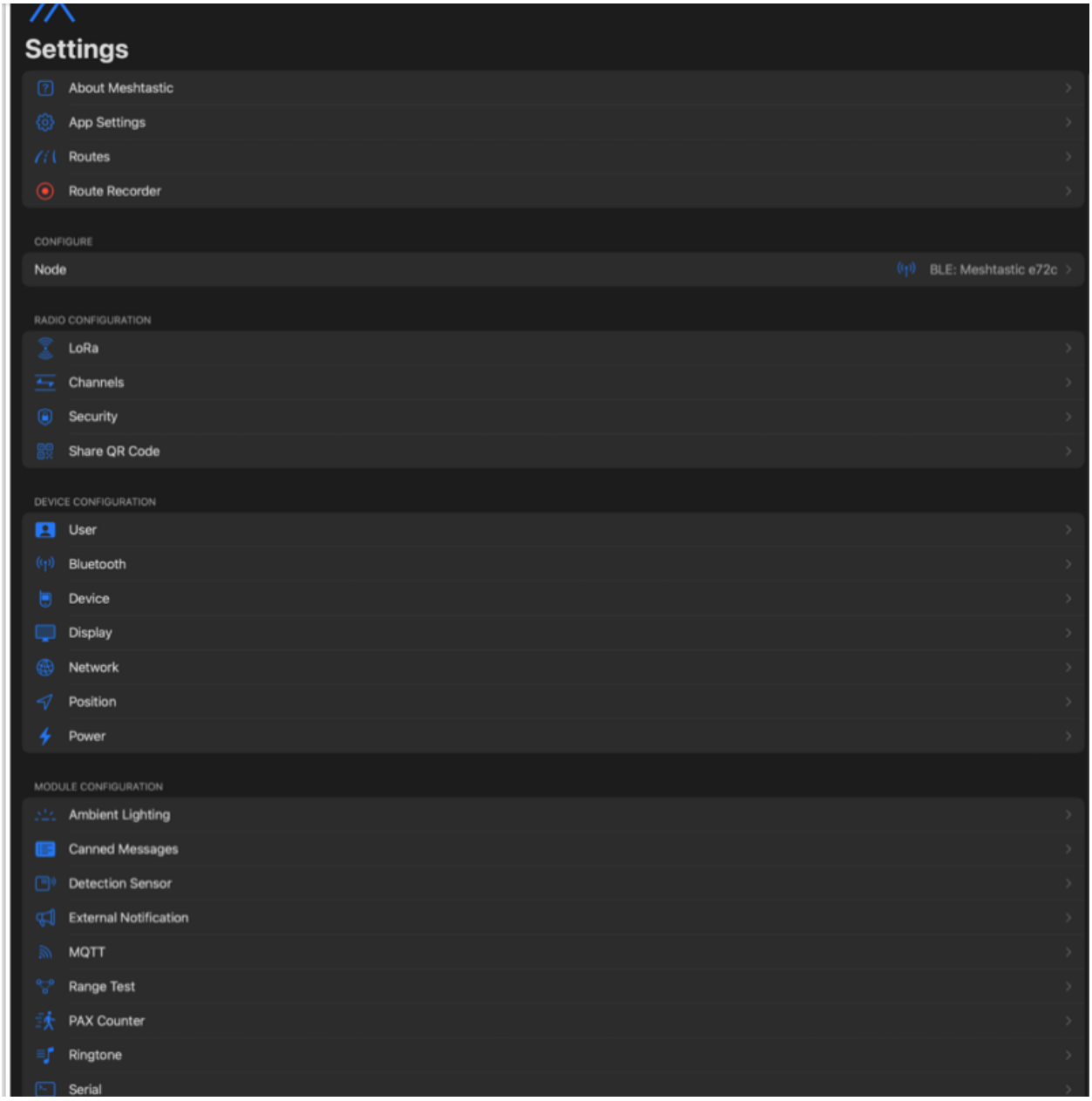
You may know the MAC (machine ID) for your device, if you have multiples, you can power them off, or wrap in foil (never disconnect antenna under power).
It is also a good idea to label your device immediately with the 4 character code.

Click the device, you should see this:



Screenshot 2025-07-07 at 10.03.52 AM.png

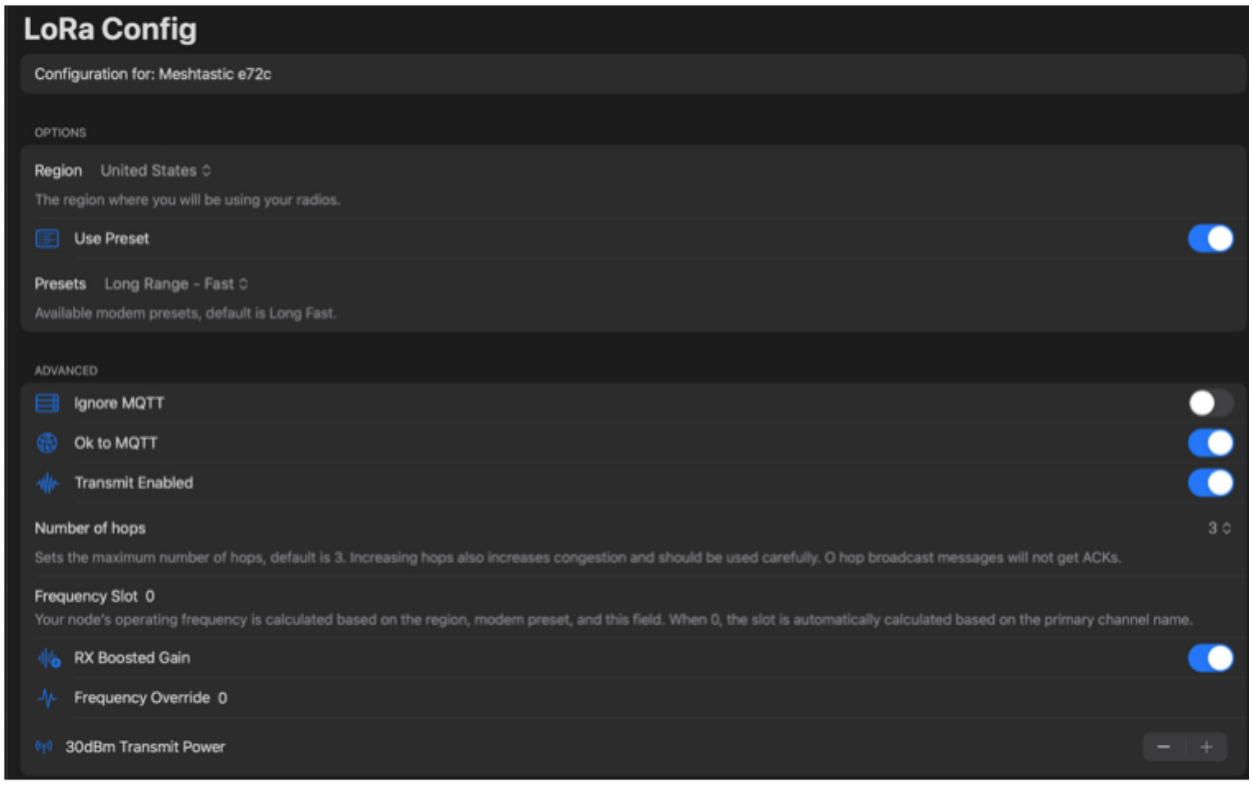
Upper left shows we are managing the device e72c
Big main page follows:



Screenshot 2025-07-07 at 10.06.53 AM.png

You may notice if you click on the "node" option, many devices may show up. This is for remote administration.

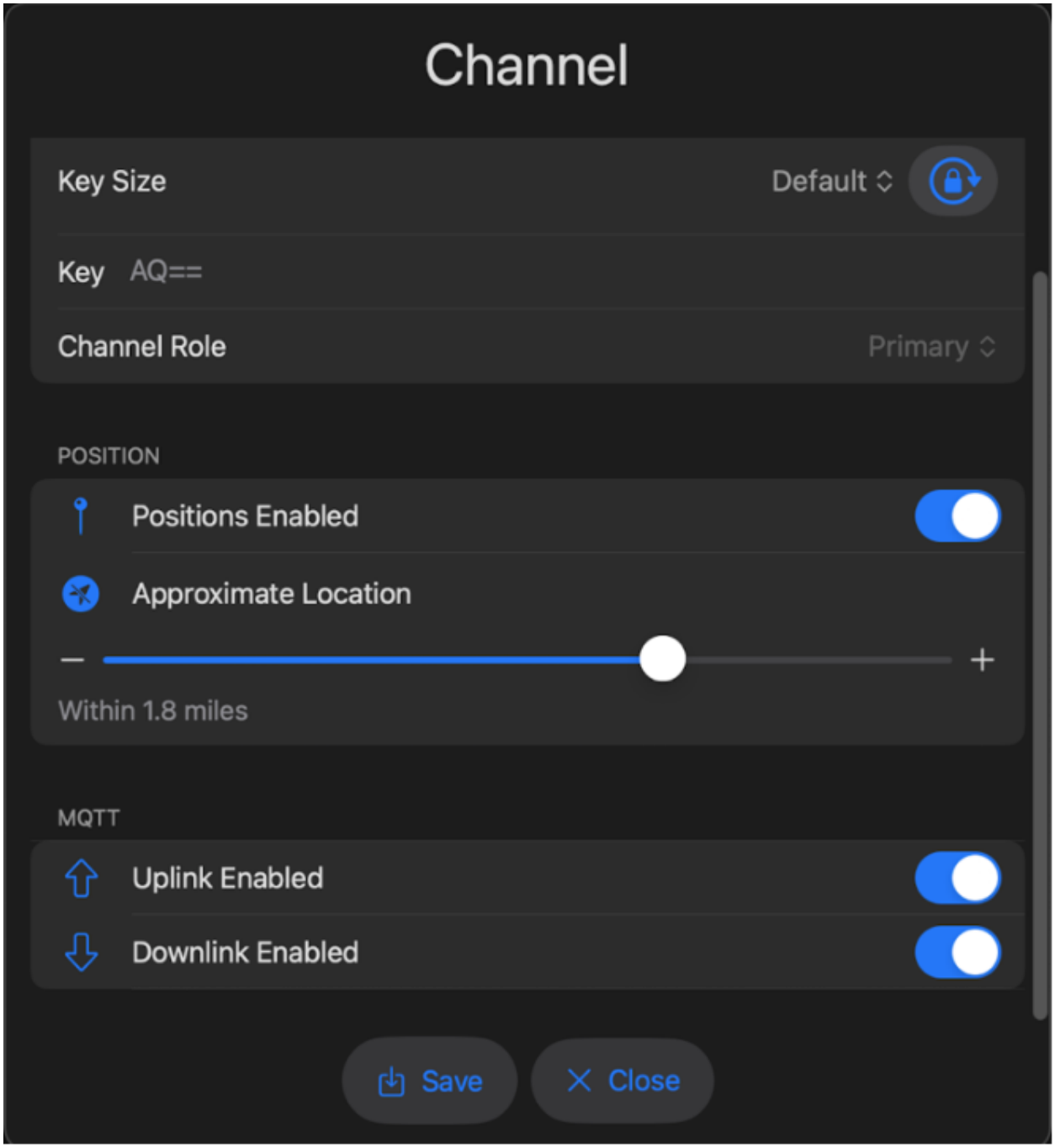
Start with LoRa: make sure the region is US, and OK to MQTT:



Screenshot 2025-07-07 at 10.07.38 AM.png

Even though the max power is 22 dBm, don't mess with the power setting unless you are using a station G2, in which case you may make this 19 dBm as feed for the PA (power amplifier).
I've found that the power decreases when you do this on a G2 unit, but it may reduce distortion. YMMV.

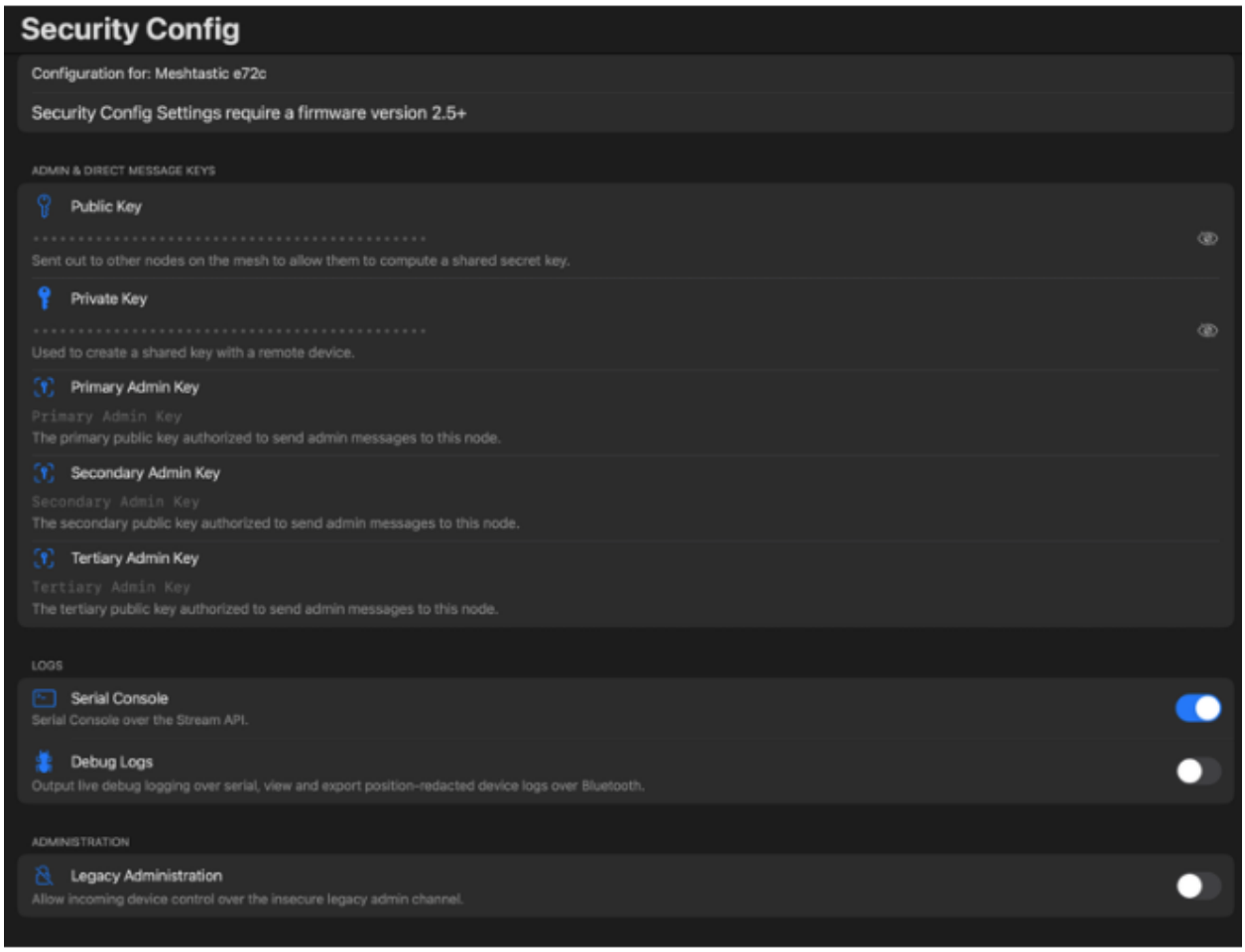
Channels is next, you should see at least the primary channel, used for all MQTT and general messages. You can see I've added another channel, more on this later.
Click on Primary, you should see this:



Screenshot 2025-07-07 at 10.08.24 AM.png

You can adjust the location accuracy if you like, but make sure uplink and downlink are enabled.

Security is not something you need to mess with unless you want to enable remote admin of your device. Also don't enable legacy admin unless you want to remote admin. Enabling serial and debugging are nice when you use either meshsense or chrome to look at the serial traffic. Legacy admin has been removed as of 2.6.11 so no worries.



Screenshot 2025-07-07 at 10.08.49 AM.png

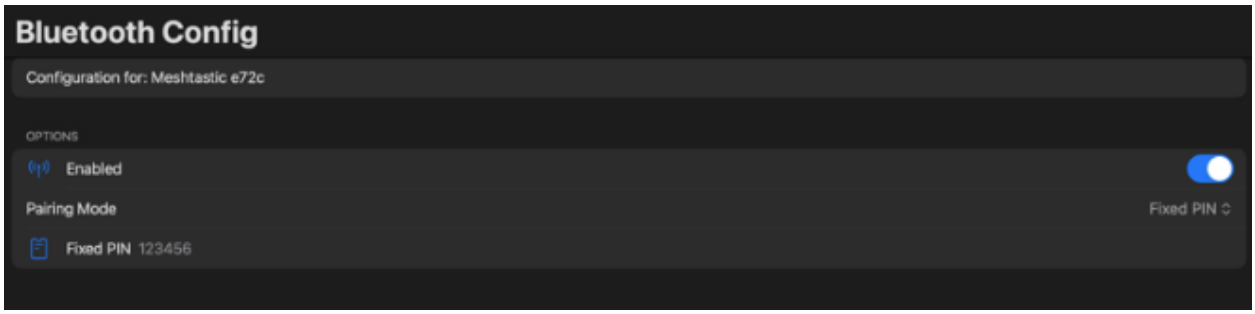
QR code is nice if you want to share the QR code for your channel, ignore otherwise.
You can also export as a URL, which is handy if you are programming many devices.
I've found using the command line interface in the meshtastic python app very useful for this, but be aware sometimes this importing does not fully work.

Be careful sharing this, it includes the key for your channel.

Next part may be of interest to hams: if you WANT to have your callsign, you don't have to click licensed operator, but if you do, it does it for you. It also disables encryption as per FCC regs, and gives you the "license" to push power beyond 1 Watt (30 dBm) which is the limit for most small amplifiers to 10 Watts max.
If you don't care about this, you can still put your callsign in there, but keep encryption as an option.
Good plan.

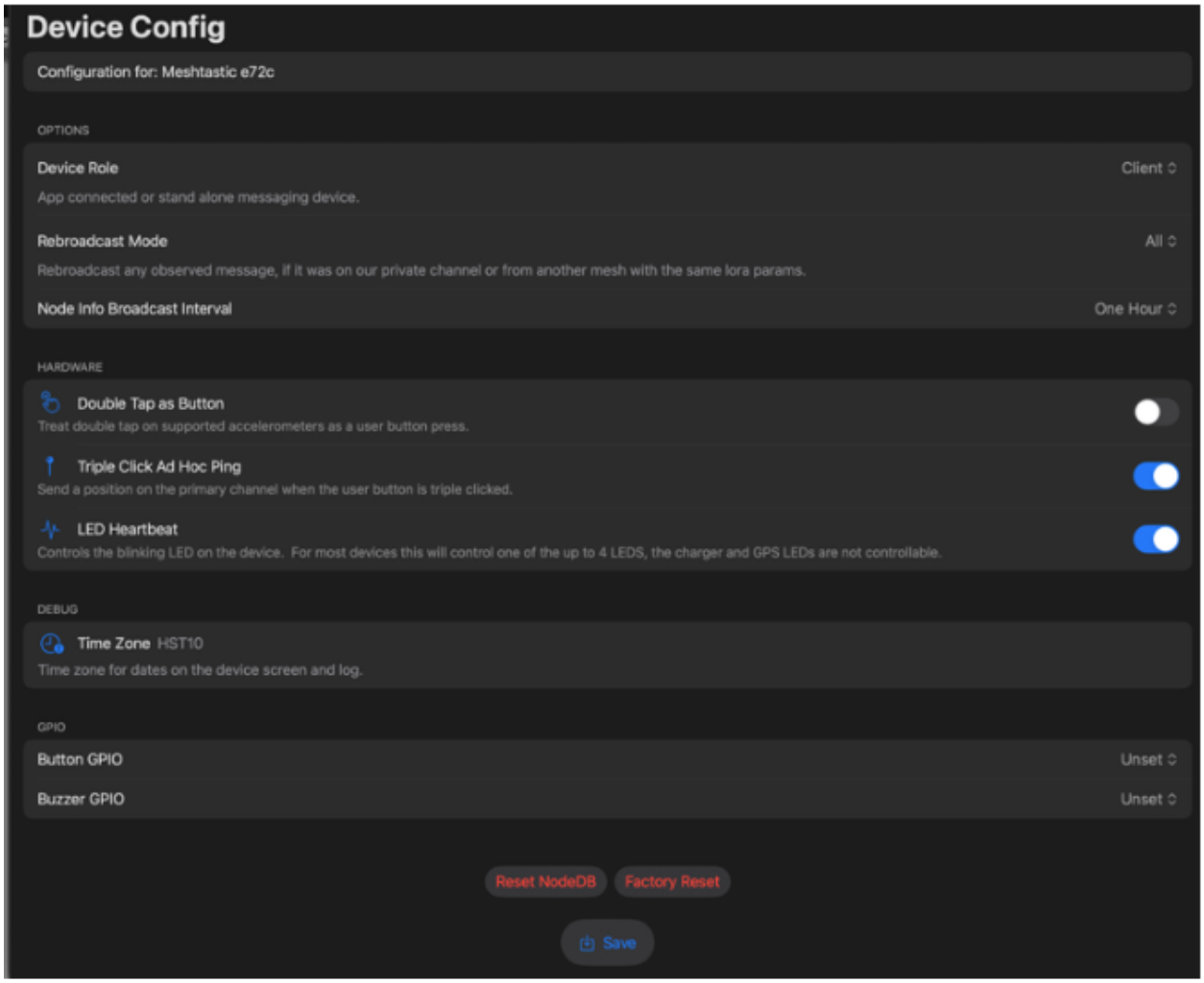
Make sure the next one is changed to fixed, which will make the bluetooth code 123456 or simply "no code, just works".. This way you won't have to look at the screen if it is away. For devices with no screen, the default code is 123456. If you change the code to random on a unit with no screen, the only way to rescue it is by the serial connection ("code suicide").

If you find your device cannot be seen, and it has a code (123456 or other) you may need to "forget" the device, which sets up a new set of BLE keys.
Good reason again, to have "no code, just works" as there is no BLE key.



Screenshot 2025-07-07 at 10.09.09 AM.png

Device is next:

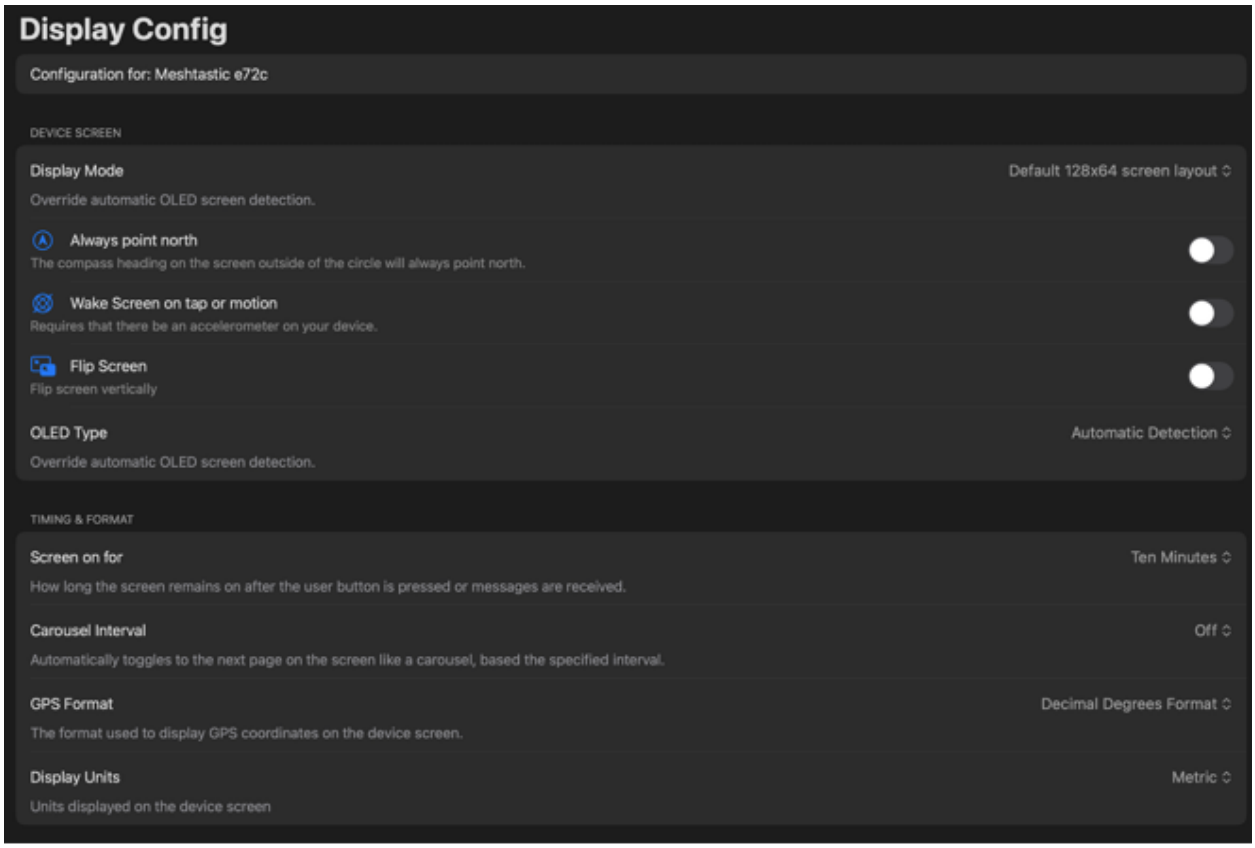


Screenshot 2025-07-07 at 10.09.33 AM.png

Unless you need to create a repeater, use client.
Simple.
You may also see nodes listed as router-which is a repeater by any other name, just shows up on the mapping servers, and smells as sweet.
Make sure rebroadcast all is enabled.

Note the lower buttons for reset, this erases the node info.
Steer clear of factory reset unless you really need to start from scratch.
Reset NodeDB is useful if you want to clear the database of your nodes, e.g. you relocate and want only fresh contacts
For more info on other roles, check out the meshtastic page, or the links in this homepage, which are excellent.

Display can be left alone, unless you want to invert the screen for some screen models:

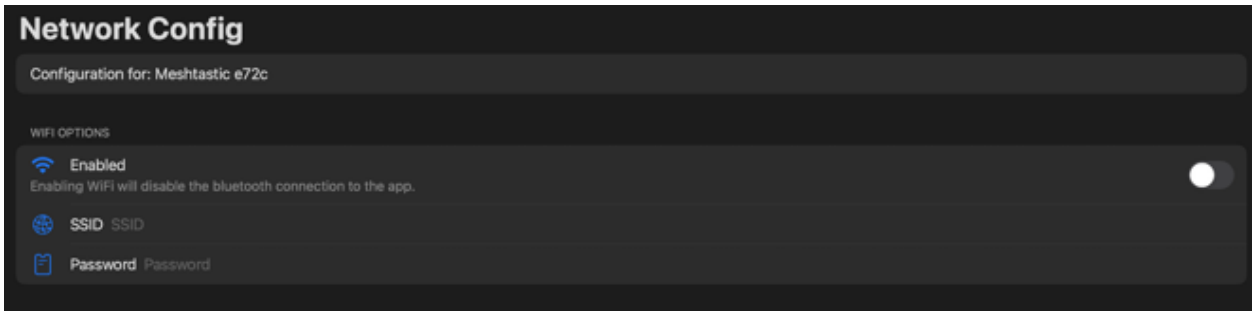


Screenshot 2025-07-07 at 10.10.00 AM.png

Network is only useful if you have an ESP32 or similar device with wifi (Lilygo T-beam):
n.b. if you enable this, you will lose bluetooth access, but you will be able to configure, monitor and other things with the device over your wifi network.
It also enables the device as an MQTT gateway if you have MQTT enabled.

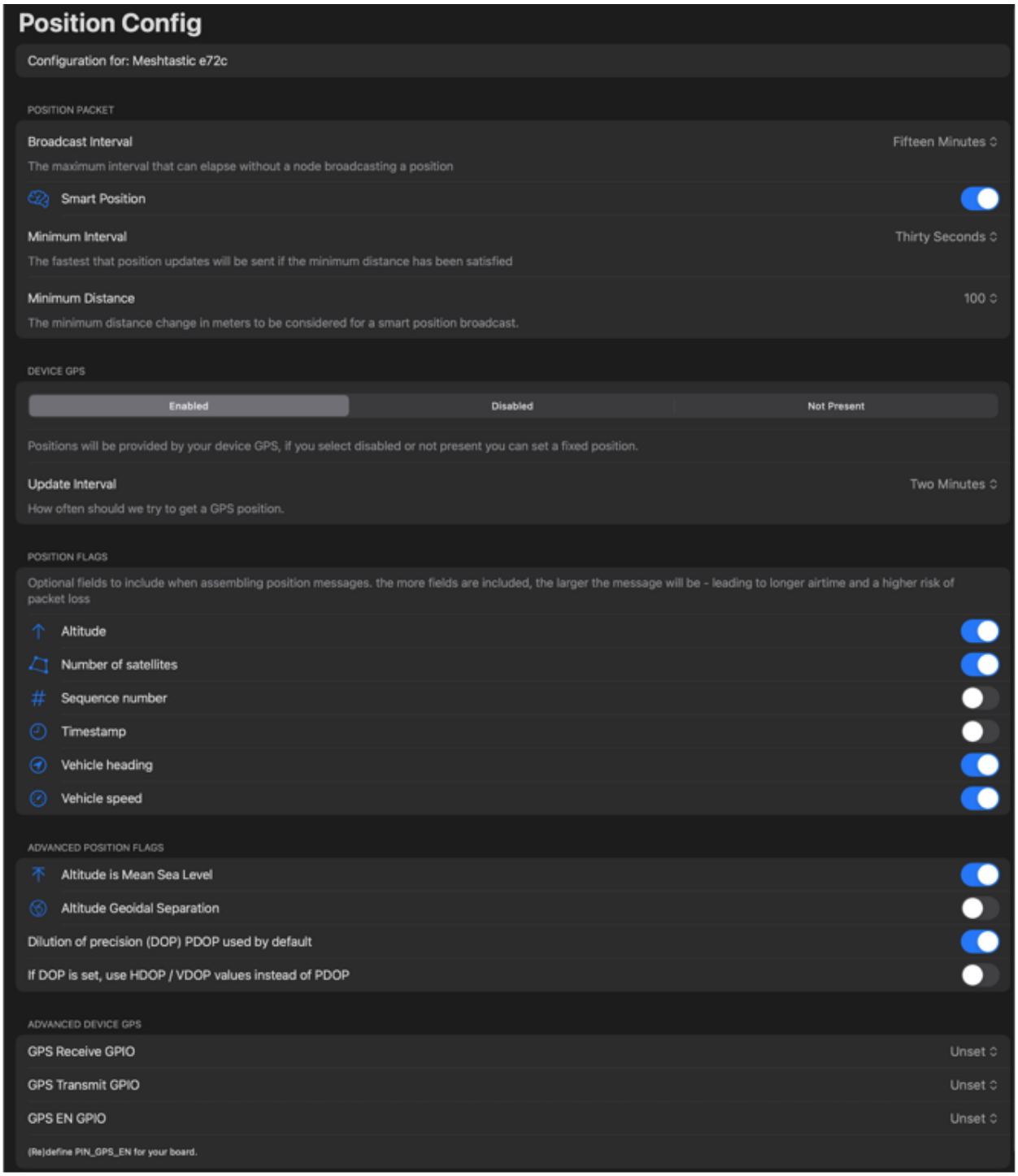
If you lose the wifi connection, you can regain access to the unit using the serial connection part of the Chrome process.
This can be useful if you have a device far away, but with internet access.
You can also use UDP to connect if the device has no IP address, which is fun to play with using the python CLI.

This is also the only way to get a csv file of range test information should you decide to do so on the esp-32 units, the only ones so far with this capacity.



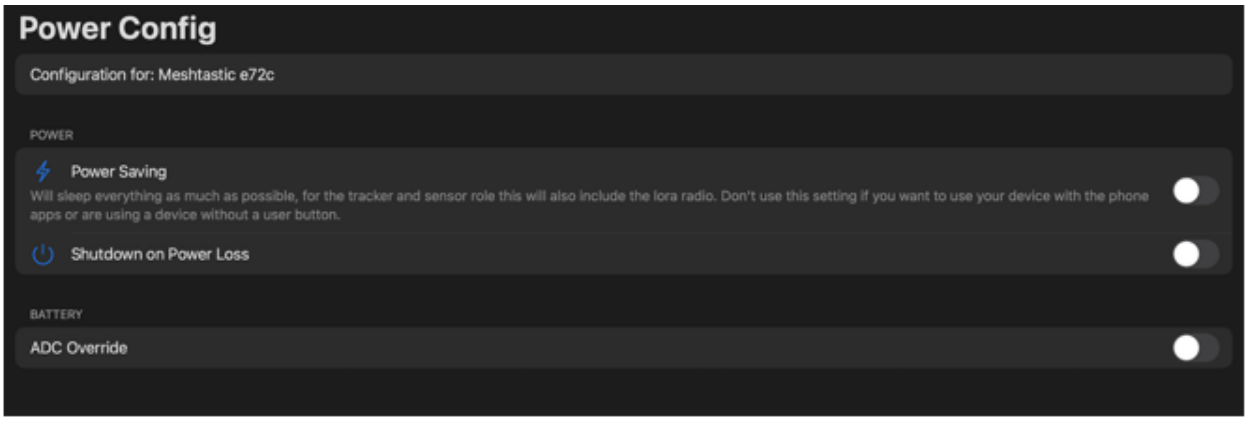
Screenshot 2025-07-07 at 10.10.19 AM.png

As you can see next, this device has GPS. If it is present, enable it. If not, you can choose "not present" and the device will take the GPS coordinates from your phone where you are at that moment. Works well, especially for locating your devices on a map. (see also meshsense)



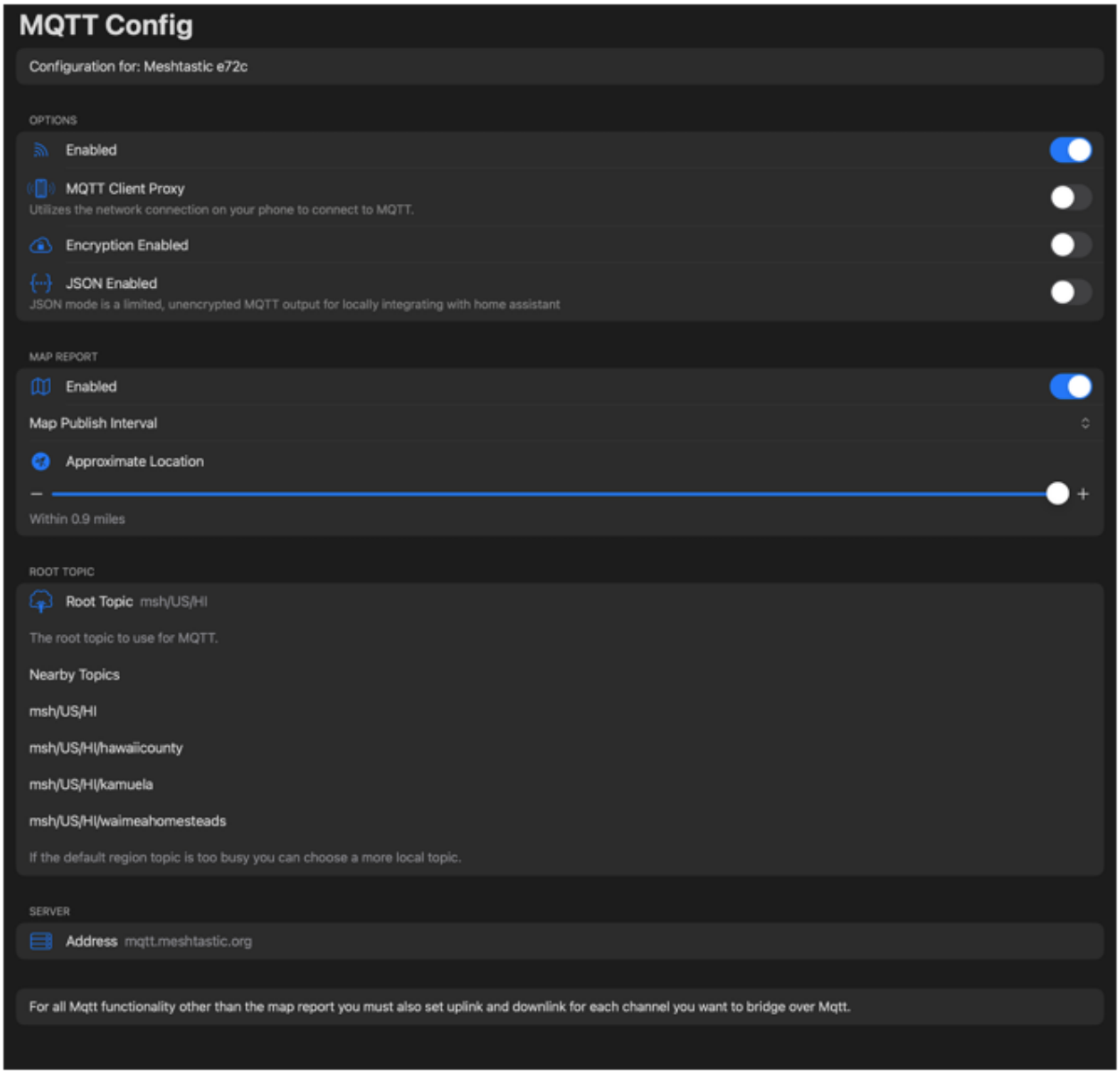
Screenshot 2025-07-07 at 10.10.56 AM.png

Power config is only useful if you are on solar power or want the device to shutdown on power loss:There are many modules, which I'll cover in detail later, but the two most important are MQTT and telemetry (only if you have a device with sensors included, like the T-beam):



Screenshot 2025-07-07 at 10.11.19 AM.png

MQTT:

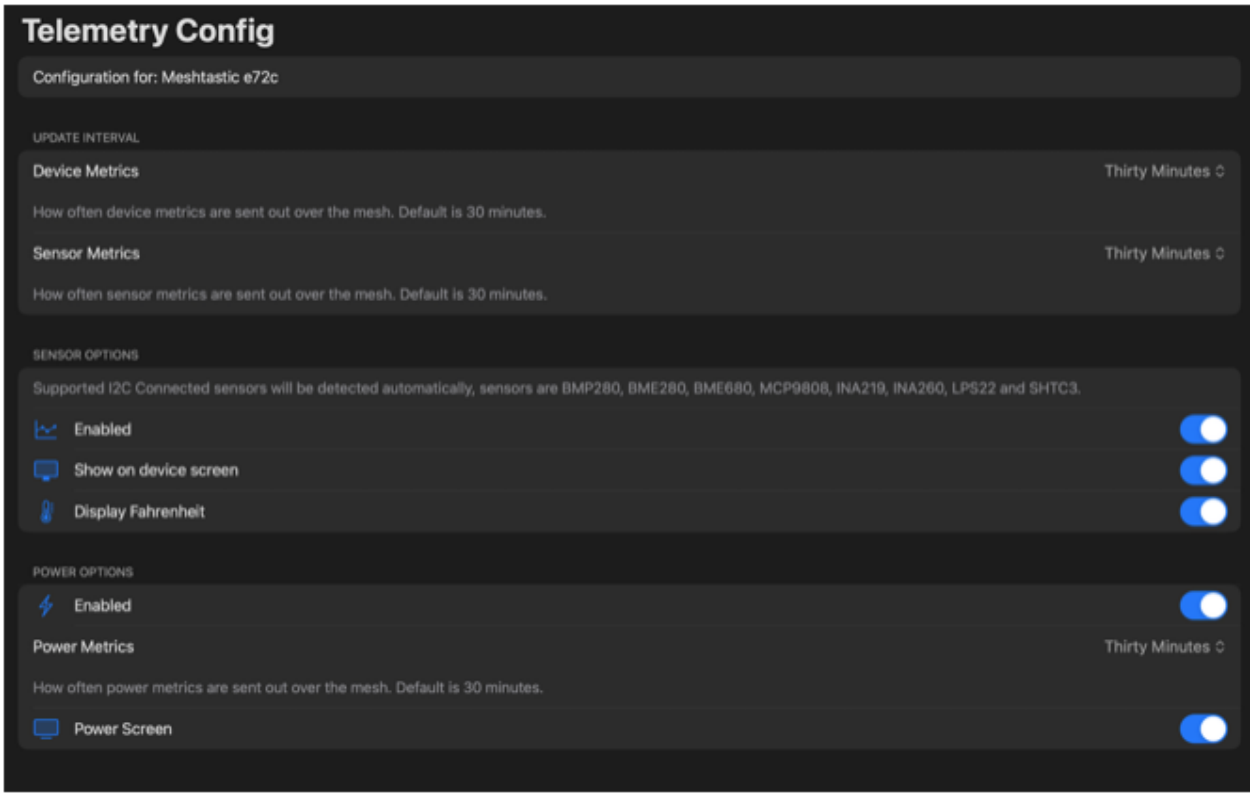


Screenshot 2025-07-07 at 10.11.40 AM.png

From the top: enable MQTT so your traffic can use MQTT servers to share messages.

Make sure uplink and downlink are also enabled on the channels page, and OK MQTT is enabled under LoRa.
Proxy uses your phone to connect to the MQTT server if you are not in range of any MQTT gateways-this works for solo users testing devices too.
One cool use for this is imagine you are in a remote location with many mesh users, only one needs to be using their phone as a MQTT proxy to get EVERYONE onto the MQTT server network.
Encryption off enables your traffic to be seen by others, but not a big deal, The impact is really to make your messages go through, change later if needed.
JSON is only for home automation, leave off unless you are using a meshsense server.
Enable the map and location.
Root topic is like a filter: you don't need to communicate with the US mainland, so I use the smaller "scope" of the MQTT server you see on the image, e.g. msh/US/HI.
You can also join other MQTT servers, like those sponsored on the web, or even your own if you want privacy.
If you want to see traffic, you can use many free MQTT explorer programs, just enter mqtt.meshtastic.org, user name meshdev, and password large4cats.
Cool stuff, you can see your own traffic.
Note the comment at the bottom-none of this works without enabling uplink and downlink in Channels above.

Telemetry is only for those with devices with sensor, but is really useful. I usually leave power monitor on.

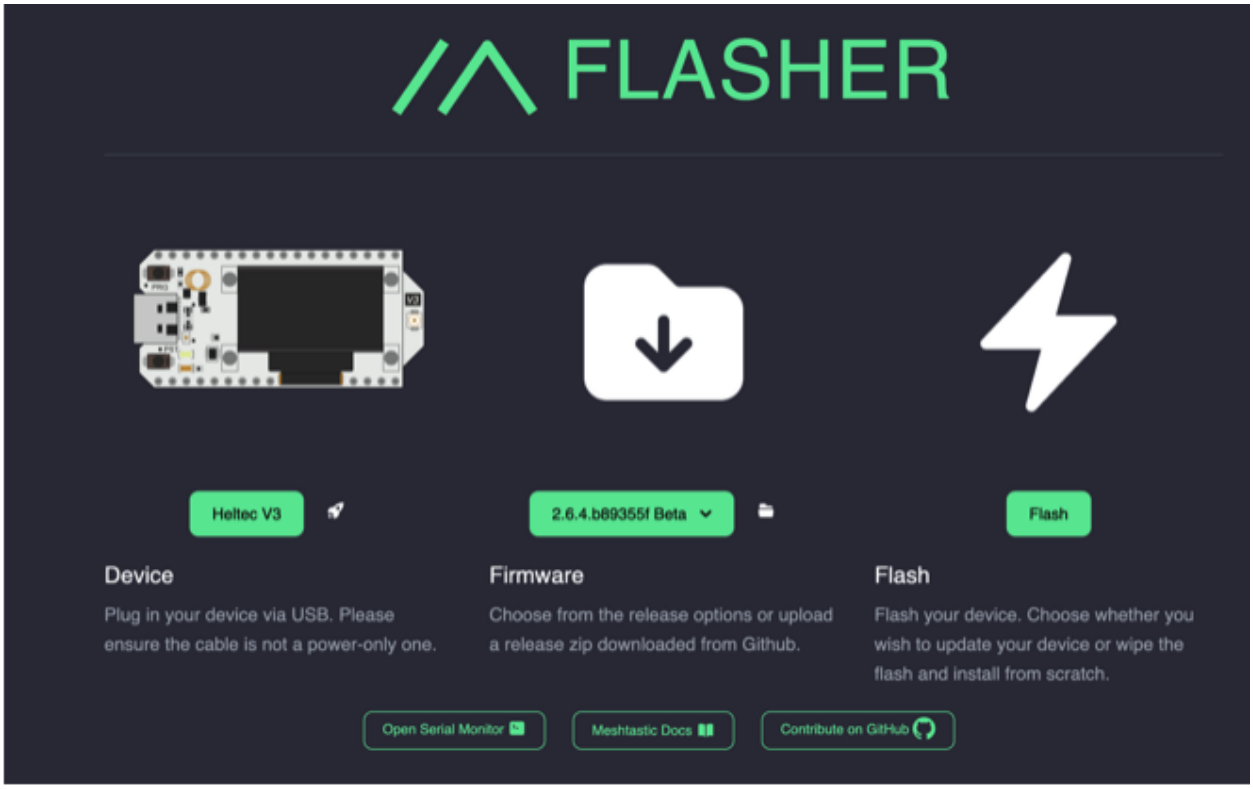


Screenshot 2025-07-07 at 10.12.06 AM.png

Chrome setup: Serial connection
We will be using a USB-C cable to connect a new ESP32-S3 unit to Chrome, on a mac. PC versions are similar, you may need help on the meshtastic site for drivers (ugh). You may also have a device with micro USB, cables vary on these.

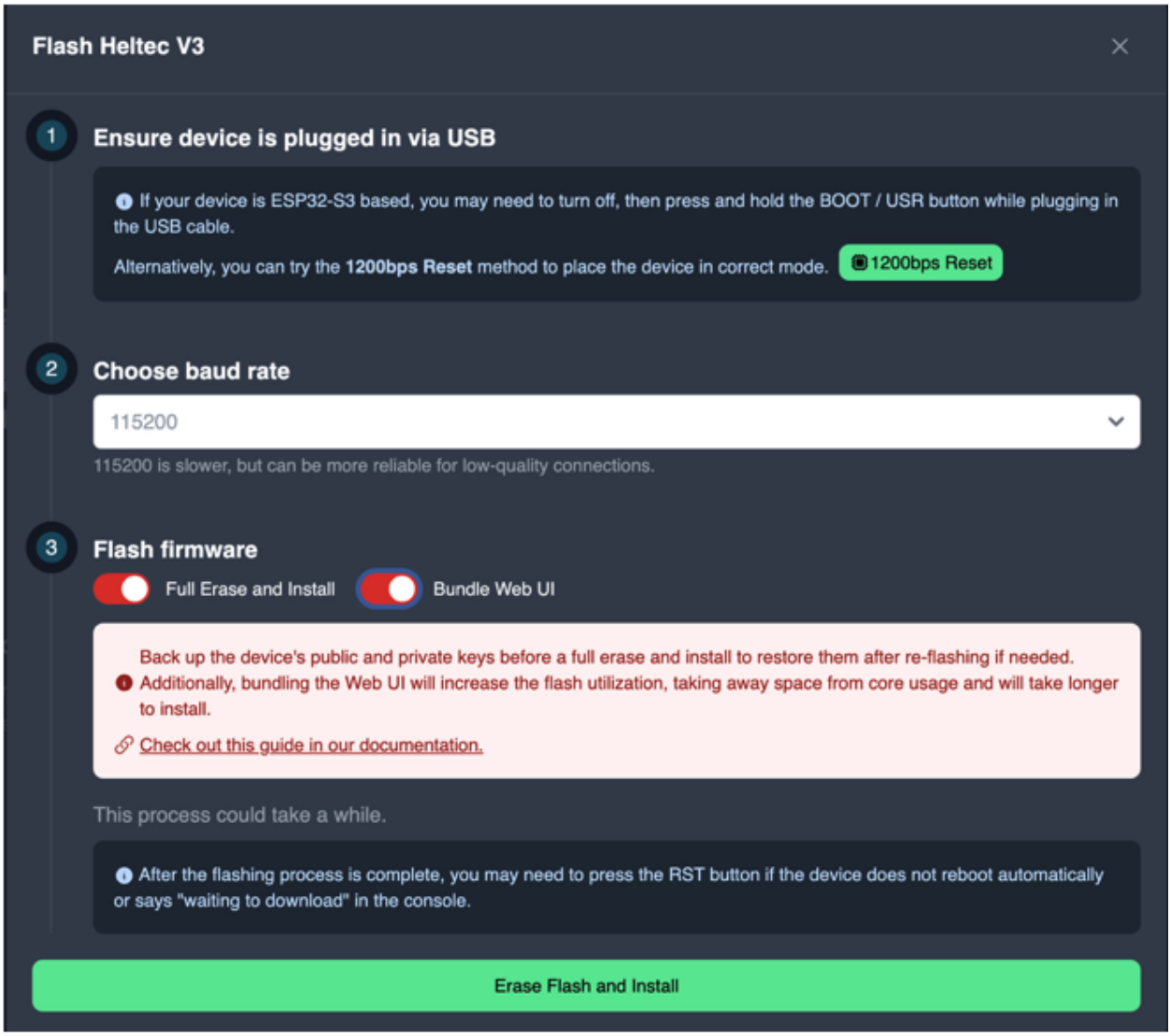
Step one: open Chrome:

Go to the meshtastic site, and download the flasher: https://flasher.meshtastic.org/
select target device (you can try auto, but it is not reliable):



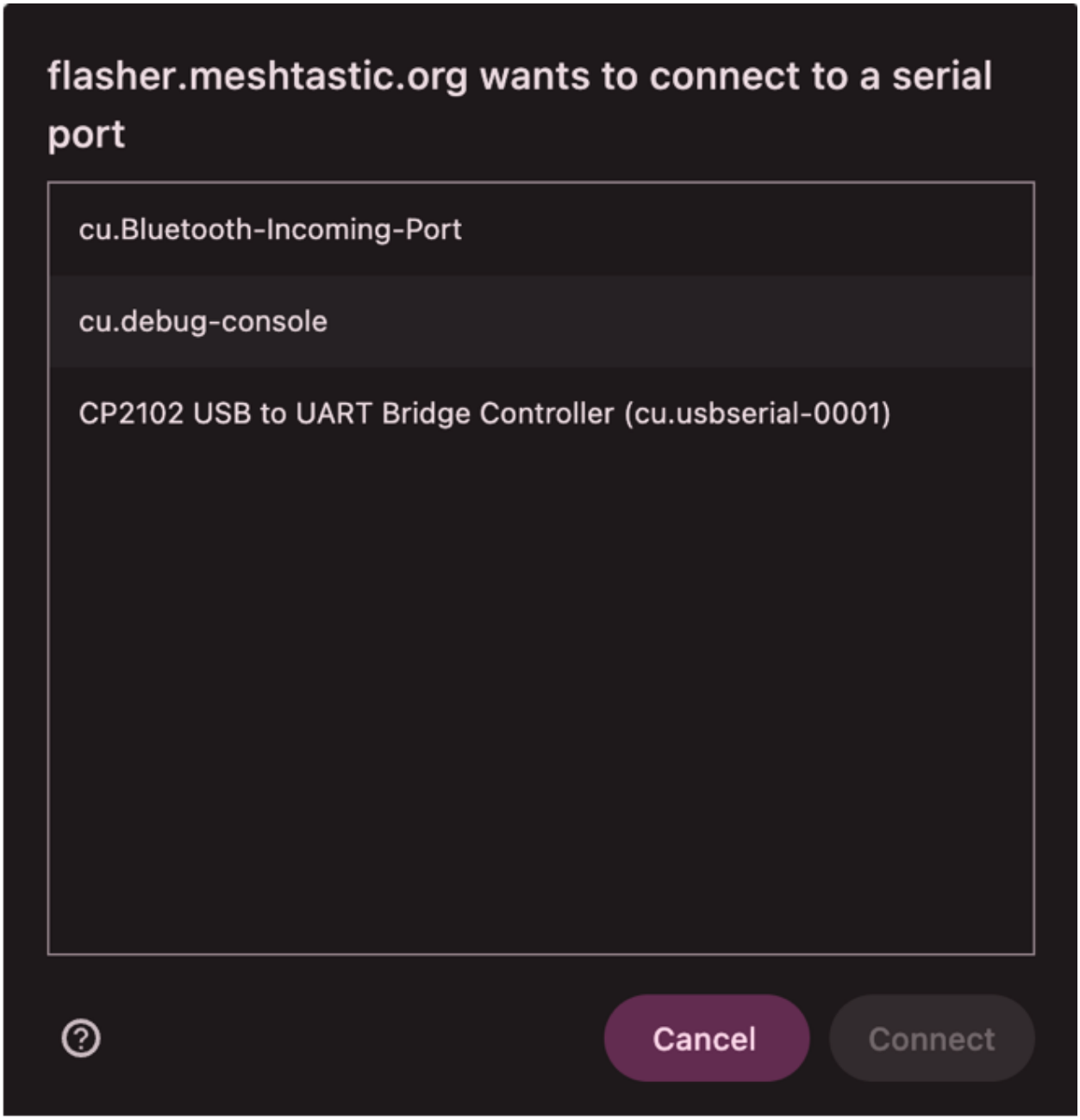
Screenshot 2025-07-07 at 10.12.30 AM.png

You can see I chose the Heltec V3, which matches this unit. You can choose beta or alpha software, up to you. If you want web access, you have to include the web package as well (see below):



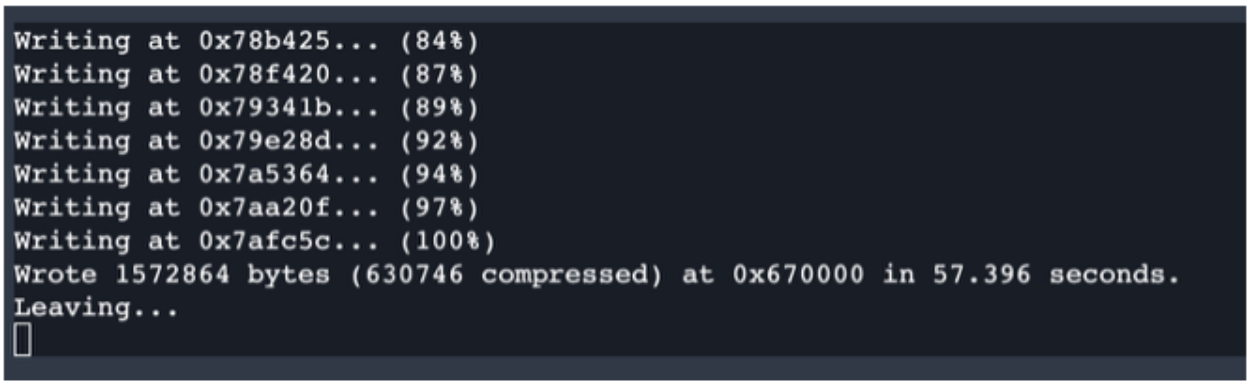
Screenshot 2025-07-07 at 10.12.51 AM.png

Some devices run just like it says, others you have to play a "plug and connect" dance.
In some cases (RAK) the device shows up as a drive on your computer, where you drop the firmware file.
Either way, it should show up in the devices menu of your computer:



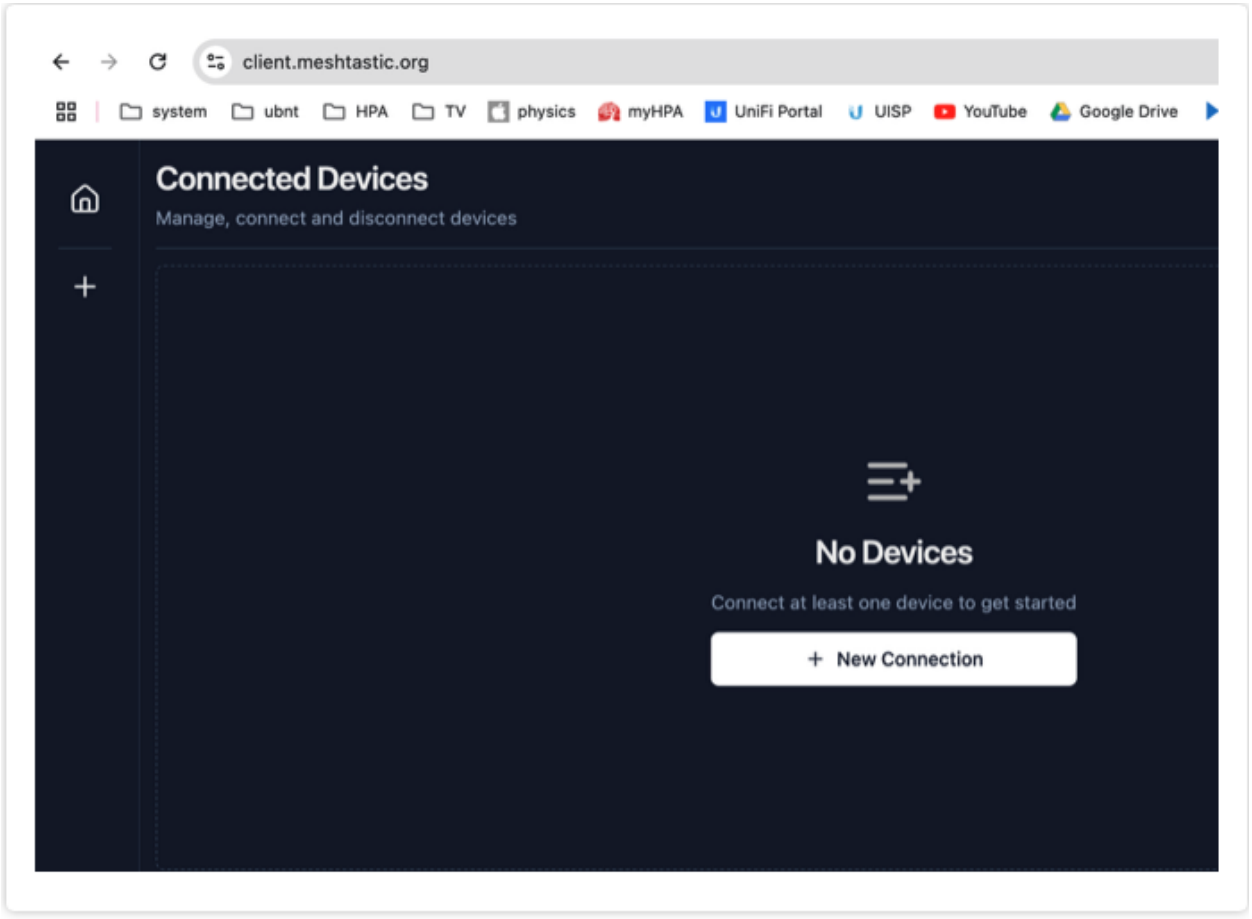
Screenshot 2025-07-07 at 10.13.13 AM.png

This may take a minute or so, when done it will say "leaving" in a petulant tone.
You may find the esp-32 units upgrade right away after 2.6 firmware, while the RAK units usually need to be set to DFU mode, in which case you copy the downloaded upgrade file to the drive image.



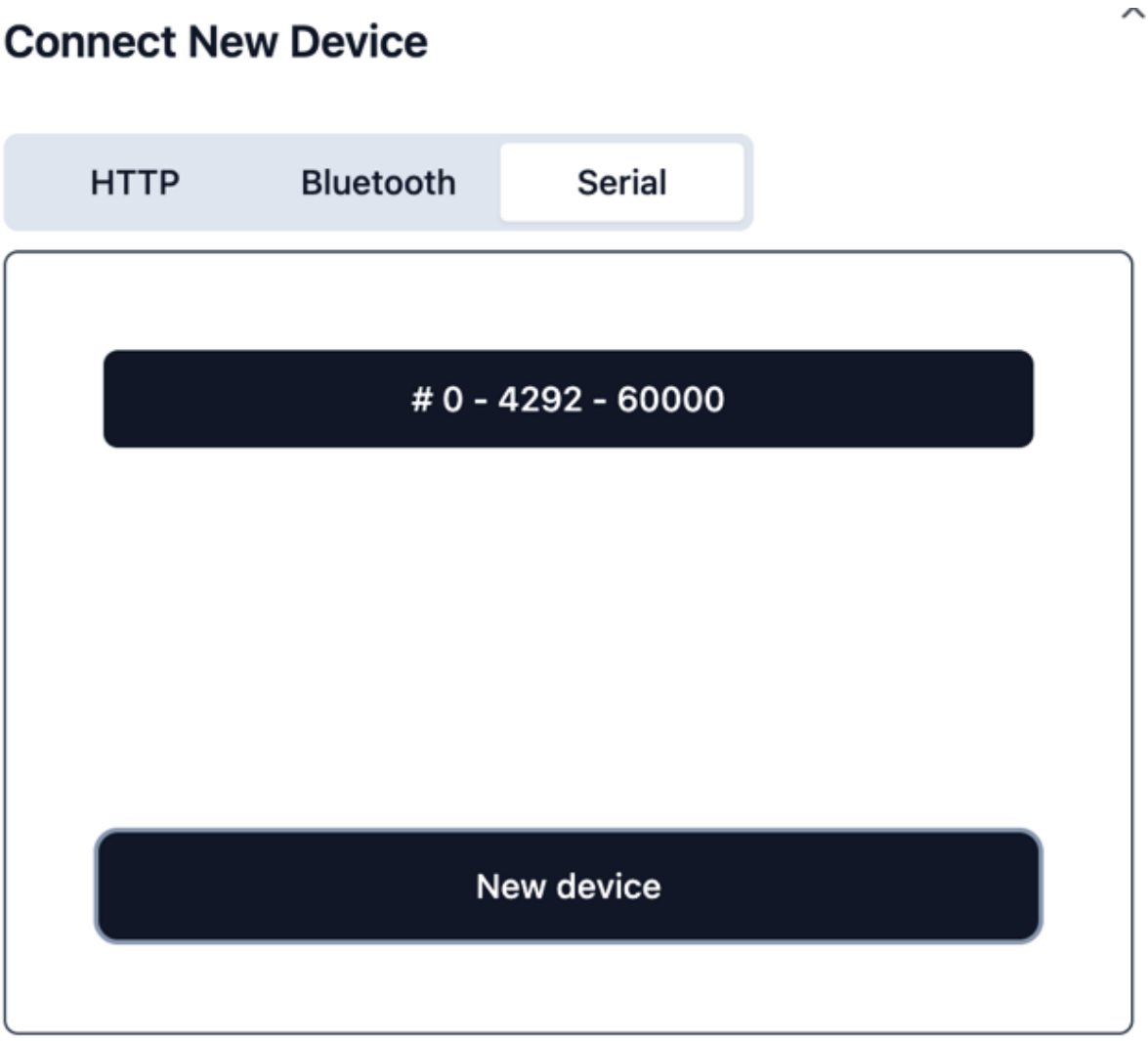
Screenshot 2025-07-07 at 10.13.32 AM.png

Once done, go to client.meshtastic.org on chrome...



Screenshot 2025-07-07 at 10.13.54 AM.png

choose serial, then a box should appear with devices, if not, search again:

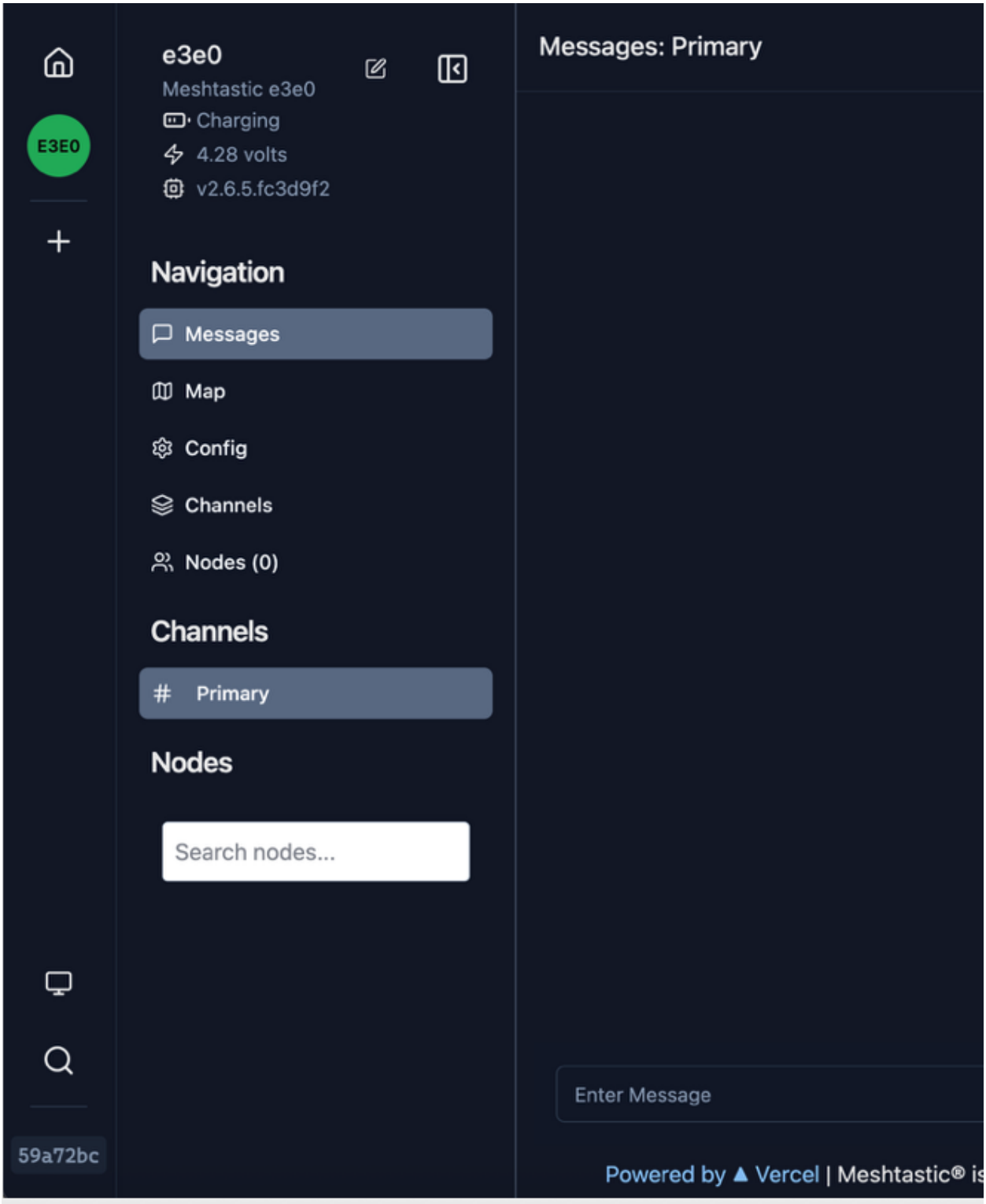


Screenshot 2025-07-07 at 10.14.17 AM.png

One reason to connect by usb is if you have multiple units, you can then label each one with the 4 character code.

Also make sure the bluetooth is set to either 123456 fixed or "no code, just works"

Nice stuff here:



Screenshot 2025-07-07 at 10.14.41 AM.png

- Same setups as above, main difference is this method ALWAYS works, even if you have disabled bluetooth.
- This is the only "rescue" way to connect to a unit if wifi is enabled and you lose wifi access.
- You will also notice the map is much more detailed and includes some fun graphics:

- When you make changes, click the drive icon in the upper right.
- You may also notice that on some units, bluetooth does not work if you are connected by serial.
- If you use your computer to charge your unit, make sure chrome is off, this solves the issue.
- You may also notice the long wait between changes and reboots when using the meshtastic app over bluetooth is gone-serial is instant.

Migrated from Apple Server Wiki. Original page ID: f363y2q2M

Revision #2
Created 2026-09-29 15:59:43 UTC by Admin
Updated 2026-09-29 16:22:33 UTC by Admin