Firmware Flashing

Flashing MeshCore Firmware

Flashing MeshCore Firmware

MeshCore firmware can be installed via the web flasher (easiest), the CLI tool, or OTA (over-the-air) for updates on already-running devices. All methods are covered below.

Method 1: Web Flasher (Recommended)

The web flasher at **flasher.meshcore.io** requires a Chromium-based browser (Chrome or Edge). Firefox does not support the WebSerial API and will not work.

1. Open **flasher.meshcore.io** in Chrome or Edge.
2. Connect your device via USB. Use a **data-capable** USB cable. Charge-only cables (common with power banks) will not expose the serial port. If the device does not appear, try a different cable first.
3. Hold the BOOT button while plugging in (ESP32 devices) or double-tap the reset button (nRF52 devices) to enter bootloader mode.
4. Select your device type from the dropdown.
5. Select the firmware variant:
   - **Companion** - pairs with a phone app over BLE or USB
   - **Repeater** - autonomous relay node, no interaction needed
   - **Room Server** - store-and-forward message hub
6. Click **Flash**. The process takes 1 - 2 minutes. Do not disconnect during flashing.
7. After flashing completes, configure the device via the MeshCore app or CLI.

Method 2: CLI Flashing

The MeshCore CLI tool allows flashing and configuration from a terminal. Useful for bulk deployments or when the web flasher is unavailable.

```
pip install meshcore-cli
```

Connect via serial (USB):

```
meshcore-cli --serial COM3 # Windows
meshcore-cli --serial /dev/ttyUSB0 # Linux/macOS
```

Connect via BLE:

```
meshcore-cli --ble connect
```

Connect via TCP (remote node on same network):

```
meshcore-cli --tcp 192.168.1.100:4403
```

Method 3: OTA Update (ESP32 devices only)

For devices already running MeshCore firmware, OTA updates avoid needing a USB connection.

1. In the MeshCore app, open the Command Line for your device.
2. Type: `start ota`
3. The device will create a Wi-Fi hotspot named **MeshCore OTA**.
4. Connect your phone or computer to the MeshCore OTA Wi-Fi network.
5. Open a browser and navigate to **http://192.168.4.1/update**
6. Upload the new firmware file (.bin). Wait for the device to reboot.

Method 4: OTA Update (nRF52 devices)

nRF52-based devices use the Nordic DFU protocol for OTA updates.

1. In the MeshCore app, type `start ota` in the Command Line.
2. Use the **nRF Device Firmware Update** app (available for Android/iOS).
3. Set packet count:
   - RAK4631: use **10**
   - Heltec T114: use **8**
4. Select the firmware .zip DFU package and transfer.

Bootloader Entry by Device

Device	Method
Most ESP32 devices (V3, V4, T-Beam)	Hold BOOT button while connecting USB
nRF52 (T-Echo, RAK WisBlock, Wio series)	Double-tap reset button quickly
LilyGo T-Deck variants	Depress trackball while connecting USB
Heltec V4	May need CH340 USB-serial drivers installed first

Post-Flash Configuration

After flashing, the device needs basic configuration before it will function on the network:

- Set device name (used to identify you in the mesh)

- Set region/frequency (ensure this matches your local regulations - 915 MHz for North America)
- Set TX power (default is usually fine; reduce for indoor testing)
- For Repeater variant: set the repeater name and ensure auto-start is enabled

Flashing Meshtastic Firmware

Flashing Meshtastic Firmware

Meshtastic firmware is flashed via the web flasher at **flasher.meshtastic.org** or via the Meshtastic Python CLI. The process is similar to MeshCore but has some differences in device selection and channels.

Web Flasher

1. Open **flasher.meshtastic.org** in Chrome or Edge. Firefox will not work (no WebSerial support).
2. Connect the device via a USB data cable. Enter bootloader mode:
   - ESP32: hold BOOT button while plugging in
   - nRF52: double-tap reset button
   - T-Deck: depress trackball while connecting
3. Select your device from the dropdown. If your device is not listed, check the Meshtastic hardware support page.
4. Choose firmware channel:
   - **Stable** - recommended for most users; well-tested
   - **Alpha** - latest features, may have bugs
5. Click Flash. The process takes 1 - 3 minutes depending on device.

Driver Requirements

Some devices require USB-serial drivers before the OS will recognise them:

Chip	Driver	Common Devices
CH340/CH341	CH340 driver (Windows/macOS)	Heltec V3, V4, some LilyGo
CP2102	Silicon Labs CP210x driver	Some T-Beam variants
USB native	No driver needed	T-Echo, RAK WisBlock, most nRF52

First-Time Configuration

After flashing Meshtastic, use the Meshtastic app (Android/iOS) or web client to configure:

- **Region:** Set to US (915 MHz) for North America. Wrong region = cannot communicate with local nodes.
- **Role:** CLIENT for personal devices; ROUTER or REPEATER for infrastructure/repeater nodes (ROUTER_CLIENT is deprecated in recent firmware).
- **Long name / short name:** How you appear to other users on the mesh.
- **Channel:** Must match other nodes you want to communicate with. Default channel works for public networks.

Re-Flashing Between Firmware Versions

You can move between Meshtastic stable and alpha, or between Meshtastic and MeshCore, at any time. Re-flashing is non-destructive to the hardware. Configuration is reset when flashing a new firmware type, so note your settings before switching.

Flashing Troubleshooting

Flashing Troubleshooting

Most flashing failures fall into a small set of categories. Work through this table before assuming the device is damaged.

Troubleshooting Table

Symptom	Likely Cause	Fix
Device not detected by browser or OS	Charge-only USB cable; wrong USB port; missing drivers	Try a different cable (data-capable); try a different USB port; install CH340 or CP2102 drivers; try a different computer
Device detected but flash fails immediately	Not in bootloader mode	Hold BOOT while connecting (ESP32) or double-tap reset (nRF52); consult device-specific instructions
Device won't boot after flashing	Wrong firmware build selected	Verify you selected the correct device in the flasher; re-flash with correct build
Flashed wrong variant (e.g., Repeater instead of Companion)	User error	Re-flash with correct variant; no permanent damage
ESP32 completely unresponsive / "bricked"	Corrupted flash	Hold BOOT button → connect USB → run: `esptool.py erase_flash` → re-flash firmware
nRF52 unresponsive	Corrupted firmware	Double-tap reset button to enter DFU mode → reflash via DFU
Linux: "Permission denied" on /dev/ttyUSB0	User not in dialout group	`sudo usermod -a -G dialout $USER` then log out and back in
Linux: Web flasher cannot connect	udev rules / ACL issue	`setfacl -m u:$USER:rw /dev/ttyUSB0`
macOS: Device not appearing in /dev/	Missing CH340 driver on macOS	Install CH34xVCPDriver from wch.cn or use the Homebrew formula
Flash completes but device shows wrong region or settings	Old config preserved in flash	Perform a factory reset via the app or by flashing with "erase before flash" option checked

esptool.py Emergency Erase

If an ESP32 device is completely unresponsive and normal bootloader entry fails:

```
pip install esptool
esptool.py --port COM3 erase_flash # Windows
esptool.py --port /dev/ttyUSB0 erase_flash # Linux
```

After erasing, the chip will be blank. Re-flash the firmware normally via the web flasher.

Identifying Your Serial Port

Windows: Device Manager → Ports (COM & LPT) → look for CH340 or CP210x device. Note the COM number (e.g., COM5).

Linux: Run `ls /dev/tty*` before and after plugging in the device. The new entry is your device (typically /dev/ttyUSB0 or /dev/ttyACM0).

macOS: Run `ls /dev/cu.*`. Look for cu.usbserial-* or cu.wchusbserial*.