

MeshCore Firmware

Firmware variants, flashing procedures, and update management for MeshCore nodes.

- [MeshCore Firmware Variants Explained](#)
- [Flashing MeshCore Firmware](#)
- [Keeping MeshCore Firmware Updated](#)
- [Flashing MeshCore Firmware OTA: The Definitive Guide](#)

MeshCore Firmware Variants Explained

MeshCore is distributed as several distinct firmware variants, each designed for a specific role in the mesh. Choosing the correct variant is essential - running the wrong firmware type wastes resources and can degrade network performance.

The Firmware Variants

Companion (COMPANION)

The Companion firmware is for user-facing nodes. It is the firmware that end-users run on their personal devices to send and receive messages via the MeshCore mobile app.

- **Primary function:** Sends and receives direct messages and channel (group) messages. Maintains a contact list. Connects to the MeshCore mobile app over BLE or USB serial (and, on Wi-Fi-capable boards, over a Wi-Fi/TCP connection).
- **Sub-variants:** There are two companion firmware builds - BLE Companion and USB Serial Companion. Wi-Fi is an additional connection mode available on capable boards (e.g. a Heltec V3), not a separate companion firmware build.
- **When to use:** Personal handheld nodes, [base station nodes](#) used for human communication, any node that a person interacts with via the MeshCore app.
- **Firmware file example:** `Heltec_v3_companion_radio_ble-v1.7.1-165fb33.bin` (or the corresponding `..._companion_radio_usb-...` build; RAK4631 nRF52840 boards ship a `.uf2` companion build)

Repeater (REPEATER)

The Repeater firmware turns a node into a dedicated packet relay. It has no user interface and no messaging capability of its own.

- **Primary function:** Receives MeshCore packets and re-transmits them. Forwards packets to extend network reach.

- **Resource profile:** Minimal. No display driver, no BLE advertising for user connections (only for CLI access), no message store. Designed for long-running operation on battery or solar when power is sized for worst-case conditions.
- **When to use:** Any node whose sole purpose is extending mesh coverage - hilltop repeaters, building relay nodes, infrastructure backbone nodes.
- **When NOT to use:** Do not flash Repeater on a node you plan to use as a personal communicator. It has no user-facing messaging.
- **Firmware file example:** `Heltec_v3_repeater-v1.7.1-165fb33.bin` (RAK4631 nRF52840 boards ship a `.uf2` repeater build)

Room Server (ROOM_SERVER)

The Room Server firmware creates a store-and-forward message hub. Room servers act as persistent group chat rooms accessible via LoRa.

- **Primary function:** Stores incoming messages and delivers them to nodes that connect later (offline delivery). Acts as a central hub for group communication.
- **When to use:** Fixed infrastructure nodes serving as community message hubs - in a building, on a hilltop tower, or at an event site where persistent message history is needed.
- **Firmware file example:** `Heltec_v3_room_server-v1.7.1-165fb33.bin` (RAK4631 nRF52840 boards ship a `.uf2` room_server build)

Sensor (SENSOR)

The Sensor firmware is for nodes that collect and transmit environmental or telemetry data.

- **Primary function:** Reads sensor data (temperature, humidity, GPS, etc.) and transmits it to the mesh network.
- **When to use:** Unattended monitoring nodes - weather stations, asset trackers, environmental sensors.
- **Hardware support:** Primarily nRF52840-based boards (RAK4631, T114) due to their low power consumption and hardware sensor interfaces.
- **Firmware file example:** RAK4631 nRF52840 boards ship a `.uf2` sensor build (other boards follow the `{board}_sensor-vX.Y.Z-{hash}.bin` pattern)

Summary Table

Firmware	User Messaging	Packet Relay	Message Store	Sensor Data
Companion	Yes	No	Local only	No

Repeater	No	Yes	No	No
Room Server	No	Optional	Yes (network-wide)	No
Sensor	No	No	No	Yes

Source: MeshCore official documentation and firmware repository at github.com/meshcore-dev/MeshCore

Flashing MeshCore Firmware

MeshCore firmware can be installed on supported hardware using two primary methods: the MeshCore Web Flasher (browser-based) and UF2 drag-and-drop (for nRF52840 boards only).

Method 1: MeshCore Web Flasher

The MeshCore Web Flasher is the recommended method for most users. It runs entirely in a browser and uses the WebSerial API to communicate with the board over USB.

URL: <https://flasher.meshcore.io> (the canonical flasher, run by the MeshCore core team. *flasher.meshcore.co.uk is a separate downstream flasher for the MeshOS variant - use the .io address for standard MeshCore.*)

Browser Requirements

The WebSerial API is only available in Chromium-based browsers (the WebSerial API shipped in Chrome/Edge 89 - see MDN/Can I Use):

- Google Chrome (version 89 or later) - recommended
- Microsoft Edge (version 89 or later) - supported
- Firefox, Safari - **NOT supported**. WebSerial is not implemented in these browsers.

Step-by-Step: Initial Flash

1. Open flasher.meshcore.io in Chrome or Edge.
2. Connect your board to your computer via USB.
3. Select your board type from the dropdown (e.g., RAK4631, T114, Heltec V3).
4. Select the firmware variant you want to flash:
 - **Companion** - for personal use nodes (connects to MeshCore app)
 - **Repeater** - for dedicated packet relay infrastructure nodes
 - **Room Server** - for store-and-forward message hub nodes
 - **Sensor** - for telemetry/environmental monitoring nodes
5. Select the firmware version (latest stable is selected by default).
6. Click **Connect**. A browser dialog will appear listing available serial ports - select your device.

7. Click **Flash**. The flasher will download the firmware and write it to the device. This typically takes 30-90 seconds.
8. The board will reboot automatically after flashing.
9. First-boot setup: connect via BLE using the MeshCore app to configure the node name and radio parameters (frequency, spreading factor, bandwidth, coding rate).

Method 2: UF2 Drag-and-Drop (nRF52840 boards only)

Boards based on the nRF52840 MCU (RAK4631, T114, Heltec HT-n62) support UF2 flashing without needing a browser or WebSerial.

1. Download the correct .uf2 file for your board and firmware variant from the MeshCore firmware releases page on GitHub.
2. Put the board into bootloader mode: double-tap the reset button rapidly. The board will appear as a USB mass storage drive whose name depends on the board's bootloader (for example, a RAK4631 mounts under its own board-specific label, while a nice!nano mounts as `NICENANO`) - the exact label varies by board, so look for any newly-appeared USB drive.
3. Copy the .uf2 file onto the USB drive. The board will automatically flash and reboot.

Platform-Specific Setup Notes

Windows

Many LoRa development boards use USB-to-serial bridge chips (CP2102, CH340, FTDI). If the board is not recognized, you may need to install the driver for your specific USB chip. Check Device Manager for unknown devices. Common driver sources:

- CP2102/CP2104: Silicon Labs VCP driver
- CH340/CH341: WCH driver
- FTDI: FTDI Virtual COM Port driver

Linux

Most USB-serial chips work out of the box on modern Linux. If you get permission errors with WebSerial or serial tools, add your user to the `dialout` group: `sudo usermod -a -G dialout $USER` and log out/in.

macOS

macOS 11+ includes a built-in CP210x (CP2102) driver. CH340/CH341 support varies by macOS version (it is absent or unreliable on several releases); if a CH340-based device is not recognized, install the WCH CH34x macOS driver. If the device doesn't appear, check System Information > USB.

Keeping MeshCore Firmware Updated

Keeping your MeshCore nodes on current firmware is important for stability, interoperability, and security. This page covers why updates matter, how to check your current version, update strategies for deployed infrastructure, and how to handle rollbacks.

Why Updates Matter

Bug Fixes

MeshCore is actively developed software. Each release typically resolves routing edge cases, BLE connectivity issues, memory leaks, and hardware-specific quirks. Running old firmware means running known bugs that may have already been fixed.

Performance Improvements

Routing algorithm refinements, radio parameter tuning, and message handling optimizations are regularly incorporated. A network of nodes all running the same recent firmware will generally route more efficiently than one running a mixture of old builds.

New Features

New capabilities - new sensor types, new room server features, new CLI commands, new position reporting formats - are only available in the firmware version that introduced them. Staying reasonably current ensures you can use new functionality as it becomes available.

Security Patches

While MeshCore is a mesh radio protocol rather than an internet-facing service, vulnerabilities can still exist. Malformed packet handling bugs, cryptographic implementation issues, and BLE pairing weaknesses are all possible attack surfaces. Security-relevant fixes are tagged in release notes;

apply them promptly.

Version Compatibility

MeshCore nodes on significantly different firmware versions may have interoperability limitations. Keeping your infrastructure nodes current minimizes the risk of incompatibility with nodes running newer client firmware.

Checking Your Current Firmware Version

There are two ways to check the firmware version on a node:

Via the MeshCore App

Connect to the node via the MeshCore app. Navigate to the node's detail or settings view. The firmware version is displayed in the device information section.

Via the MeshCore CLI

Connect to your node using a BLE serial terminal or the MeshCore CLI tool and run:

```
ver
```

This prints the node's firmware version. MeshCore firmware is currently in the 1.x series, so the output is an illustrative line such as:

```
v1.15.0
```

The `ver` command reports the firmware version. To see the hardware/board name, use the separate `board` command. For runtime health (battery, uptime, queue) use `stats-core`.

Update Strategy for Infrastructure Nodes

Repeaters and room servers are infrastructure - other users depend on them. Updating carelessly can cause network disruption. Follow this strategy:

1. Test on a Non-Critical Node First

If you operate multiple nodes, update one non-critical node (a spare, or the lowest-traffic repeater) to the new firmware first. Run it for 24 - 48 hours and verify:

- The node comes back online and connects to the mesh after the update.
- Routing works correctly through the node.
- No unexpected reboots or radio lockups occur.
- BLE connectivity from the app functions normally.

2. Preserve Configuration Before Updating

Before updating any node, record its current configuration:

- Node name
- Frequency preset and any custom radio parameters
- TX power setting
- Any custom channel configurations (for room servers: room name, password)

Use the CLI `get` queries (for example `get radio`, `get tx`) and `infos` to read back the current settings, and screenshot or copy the output. While configuration is generally preserved across firmware updates (stored in non-volatile flash separate from the firmware), a failed or interrupted flash can result in settings being wiped.

3. Update During Low-Traffic Periods

Infrastructure nodes go offline during flashing (typically 30 - 90 seconds). Schedule updates during periods when the network is least used to minimize impact on other users.

4. Update Infrastructure Before Clients

When a new major or minor version is released, update repeaters and room servers before client nodes. Infrastructure nodes carry traffic for all clients; having them on newer firmware ensures they can handle any new packet formats clients may start using.

How to Update

How you update depends on the board. ESP32 boards typically require USB flashing (the same process as initial flashing). nRF52 boards (RAK4631, T114, Seeed XIAO nRF52) additionally support over-the-air updates via the DFU app and the `start ota` CLI command, which avoids needing a USB connection. The USB web-flasher steps are:

1. Connect the node to a computer via USB.
2. Open the MeshCore Web Flasher at flasher.meshcore.io in Chrome or Edge.
3. Select your board type and firmware variant.
4. Select the new firmware version.
5. Click Connect, select the serial port, then click Flash.
6. Wait for the flash to complete and the board to reboot.
7. Verify the node is operational using `ver` (firmware version) and `stats-core` (battery/uptime/queue health).

For nRF52840 boards (RAK4631, T114, HT-n62): UF2 drag-and-drop is available as an alternative. Download the new `.uf2` file, enter bootloader mode (double-tap reset), and copy the file to the USB drive.

Rollback: Returning to a Previous Version

If a firmware update causes problems, you can return to any previous version:

1. Open the MeshCore Web Flasher.
2. Select your board and variant.
3. Use the **version selector** to choose the previous known-good version (older versions are retained in the flasher's version history).
4. Flash as normal.

For UF2 boards: download the previous version's `.uf2` file from the MeshCore GitHub releases page and flash it via drag-and-drop.

Note: Configuration is generally preserved across rollbacks. However, if a newer firmware version introduced a new configuration key that older firmware does not understand, the old firmware may ignore or reset that setting.

Coordinating Community Network Updates

If you operate nodes on a shared community network, coordinate updates with other network operators:

- Announce planned updates in your community's Discord, forum, or group chat before updating shared infrastructure.
- Share the release notes link so other operators can review what has changed.
- If a major version update is involved, agree on a migration window so all infrastructure nodes are updated together, minimizing the period of mixed-version operation.
- After updating, post a confirmation in the coordination channel so others know the node is back online and on the new version.

Same Version Compatibility Notes

Within the same major version, MeshCore nodes running different minor versions can generally communicate. However:

- As expected behavior for a forwarding mesh, a node running a minor version that introduced a new packet type may generate packets that older minor-version nodes do not fully process - they will typically still forward them but may not display them correctly. Check the release notes for any such changes.
- Patch releases within the same minor version are intended to be bug-fix-only and are generally interoperable, but MeshCore does not publish a formal compatibility guarantee - test before relying on mixed-version meshes.
- When in doubt, check the release notes for any compatibility warnings. The MeshCore team typically calls out cross-version compatibility issues explicitly.

Flashing MeshCore Firmware OTA: The Definitive Guide



Step-by-Step: OTA Update

Over-the-air (OTA) updating lets you reflash a deployed MeshCore node; a repeater, room server, or companion, without connecting it to a computer over USB. The method depends on the board's chip family: **nRF52** boards update over Bluetooth using Nordic's DFU app, while **ESP32** boards update over a temporary Wi-Fi access point in your browser. Both are covered below, followed by notes specific to companions.

OTA is convenient for nodes that are hard to reach physically (a repeater on a roof or tower). If a node is within easy reach, a USB flash from flasher.meshcore.io is faster and more reliable than OTA. Reserve OTA for when getting a cable to the device is impractical.

nRF52 Boards

nRF52 boards (RAK4631, Heltec Mesh Node T114, Seeed XIAO nRF52840, and similar) update over Bluetooth LE using Nordic's DFU app. The same process works for repeater, room server, and companion firmware, only the firmware image differs (see the Companions section for the companion firmware-version requirement).

Browser Requirements

The WebSerial API is only available in Chromium-based browsers (the WebSerial API shipped in Chrome/Edge 89 - see MDN/Can I Use):

- Google Chrome (version 89 or later) - recommended
- Microsoft Edge (version 89 or later) - supported
- Firefox, Safari - **NOT supported**. WebSerial is not implemented in these browsers.

Mobile App Requirements

Download the **nRF Device Firmware Update** app (you can find it by searching **nrf dfu** in your app store).

Note: After installation, this app is listed as "DFU" in the apps list, NOT nRF Device Firmware Update.

- **Android:** <https://play.google.com/store/apps/details?id=no.nordicsemi.android.dfu&hl=en-US>
- **iOS:** <https://apps.apple.com/us/app/nrf-device-firmware-update/id1624454660>

Get the OTAFIX Bootloader

The **OTAFIX bootloader** (by **oltaco**: Huw "Taco" Duddy, a MeshCore firmware developer) replaces the stock nRF52 bootloader and makes Bluetooth OTA DFU far more reliable: significantly faster OTA, automatic fallback to OTA DFU mode if an update fails, and the ability to enter OTA DFU mode by holding a button while resetting. It is strongly recommended before doing OTA on nRF52 boards. You install it once, over USB.

- Visit flasher.meshcore.io
- Find your device and select the repeater firmware you run
- Look for the following message:
 - We strongly recommend installing OTAFIX Bootloader by @oltaco for more reliable Bluetooth OTA DFU.

We strongly recommend installing [OTAFIX bootloader](#) by @oltaco for more reliable Bluetooth OTA DFU. [Read](#)

- Click the OTAFIX bootloader link to download the bootloader (the file is named something like `update-xxxx.uf2`)
- Place the device into DFU mode (on most nRF52 boards, double-press the RESET button, twice within about half a second). **On the Seeed XIAO nRF52840, press RESET once**

first; if no drive appears, double-press quickly. The T1000-E and ThinkNode M3 use a magnetic-cable button sequence: see the device's own instructions.

- Verify the device shows up as a drive on your computer
 - Windows: In Windows Explorer, look for a new device, such as HT-n5262 (G:)
 - macOS: a new removable volume mounts on the desktop, named after the board (e.g. T114 or RAK4631)
- Drag and drop the downloaded bootloader file onto the new device
- The device will reboot and you can now flash the device OTA

Download firmware images to your mobile device

On flasher.meshcore.io, download the firmware image for the device you want to flash. For OTA with the DFU app, choose the **DFU package (.zip)** variant of the firmware (not the `.uf2`, which is for USB drag-and-drop).

Flash the Device OTA!

- Open the DFU/nRF Device Firmware Update app on your mobile device
- Find the device you want to flash in the list, tap it
- Select the firmware (the `.zip` package) for the device you want to flash
- Ensure the correct device is selected
- Tap Start

Progress is slow. Ensure you have an unobstructed path to the device. **External Bluetooth antennas help tremendously.**

ESP32 Boards

ESP32 boards (Heltec V3, LilyGo T-Beam and T-Deck, Station G2, RAK11200, and similar) do **not** use the DFU app or Bluetooth for OTA. Instead, the device hosts a temporary Wi-Fi access point and you upload the firmware to it from a browser. You start this mode with a command, so you need admin access to the node in the MeshCore app.

Requirements

- The node must already be running a MeshCore firmware version that supports OTA. If it is on older firmware, do one USB flash from flasher.meshcore.io first, then future updates can be done OTA.
- A phone or laptop with Wi-Fi and a web browser.
- Admin access to the node in the MeshCore app (you issue the OTA command from its Command Line).

Get the firmware image

- Go to flasher.meshcore.io, select your device and the firmware you run.
- Download the **non-merged** `.bin` file. **Do not use the merged** `.bin`: the merged image is only for first-time USB flashing and will not work for OTA.

Start OTA mode on the device

- In the MeshCore app, log into the node and open its **Command Line**.
- Send the command `start ota`.
- The device replies with an address such as `Started: http://192.168.4.1/update` and creates a Wi-Fi access point named **MeshCore-OTA** (depending on firmware it may appear as "MeshCore OTA").

Upload the firmware

- On your phone or laptop, connect to the **MeshCore-OTA** Wi-Fi network.
- Open <http://192.168.4.1/update> in a browser.
- Choose the non-merged `.bin` you downloaded and start the upload. A progress bar tracks the flash.
- When it finishes, the device reboots onto the new firmware and the MeshCore-OTA network disappears. Reconnect your phone/laptop to your normal Wi-Fi.

While in OTA mode the device's only job is hosting this upload page, so it is briefly off the mesh. Keep your phone or laptop close to the node for a stable Wi-Fi link.

Companions

A companion is the node you pair with the MeshCore phone app. Companion firmware updates OTA using the **same mechanism as repeaters and room servers**. The difference is the firmware image you flash and, on nRF52, a minimum firmware version.

- **nRF52 companions** (RAK4631, Heltec T114, XIAO nRF52840): update over Bluetooth with the nRF Device Firmware Update (DFU) app, exactly as in the *nRF52 Boards* section above. **The node must already be running companion firmware v1.15 or later to update over OTA**. If it is older, flash it once over USB from flasher.meshcore.io to reach v1.15+, after which future updates can be done OTA.
- **ESP32 companions** (Heltec V3, T-Deck, etc.): update over Wi-Fi using the `start ota` method in the *ESP32 Boards* section above.

The actual firmware transfer happens in Nordic's DFU app (nRF52) or on the Wi-Fi upload page (ESP32). There is no separate "update firmware" button inside the MeshCore app itself. As always, if the companion is in your hand, a USB flash is the simplest path.