

MeshCore Firmware Variants Explained

Accurate as of 13 July 2026. Firmware version numbers below should be checked against the current releases before you rely on them.

MeshCore is built in several distinct firmware types, each designed for a specific role in the mesh. Choosing the right one matters: the roles are not interchangeable, and a node flashed as a Repeater cannot be used as a personal messenger.

This page covers node roles. On this page, "variant" always means *role*. There is a second, separate choice: which *distribution* you flash. Stock MeshCore is one option, and several independent projects (EasySkyMesh, Keymind Cascade, MCLite, WADAMESH, ZephCore) build their own MeshCore firmware with different priorities. See [MeshCore Firmware Distributions](#). Pick a distribution first, then a role within it.

The Firmware Variants

Companion

The Companion firmware is for user-facing nodes. It is what you run on your personal device to send and receive messages through 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.
- **Three builds, not two.** BLE Companion and USB Serial Companion are both prebuilt by the official flasher. A **Wi-Fi Companion** build also exists, but the official flasher does not ship it, because the Wi-Fi SSID and password are compiled into the binary. You either build it yourself or take a prebuilt one from a fork (Keymind Cascade prebuilds Wi-Fi companion binaries).
- **When to use:** Personal handheld nodes, [base station nodes](#) used for human communication, any node a person interacts with via the app.
- **Firmware file example:** `Heltec_v3_companion_radio_ble-v1.16.0-07a3ca9.bin` (or the corresponding `...companion_radio_usb-...` build; RAK4631 nRF52840 boards ship a `.uf2` companion build)

Repeater

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

A MeshCore repeater is not a repeat-everything flood relay. This is the single most common misunderstanding, and it is MeshCore's main architectural difference from other LoRa mesh systems. Upstream's FAQ puts it in bold: a MeshCore repeater "does **not** forward or retransmit every packet it receives, unlike other LoRa mesh systems."

- **How it actually works:** MeshCore floods a packet only while it is discovering a route. Once a path is known, packets carry that path, and *only the repeaters named in the path* retransmit them. See [MeshCore Routing: Flood-First, Direct-Route-After](#).
- **No Bluetooth.** You cannot connect to a node running Repeater firmware over Bluetooth. It is administered over USB serial, or remotely over the air.
- **When to use:** Any node whose sole purpose is extending mesh coverage. Hilltop repeaters, building relays, infrastructure backbone nodes.
- **When NOT to use:** Do not flash Repeater on a node you intend to use as a personal communicator. It has no user-facing messaging.
- **Firmware file example:** `Heltec_v3_repeater-v1.16.0-07a3ca9.bin` (RAK4631 nRF52840 boards ship a `.uf2` repeater build)

Room Server

The Room Server firmware creates a store-and-forward message room. It works like a small BBS or persistent group chat reachable over LoRa.

- **Primary function:** Clients log in to the room and post messages. The server stores the room's posts and pushes recent unseen ones to a client when it next connects, so you can catch up on what you missed while offline.
- **Scope, precisely:** it stores *that room's* posts, and it pushes a limited backlog (upstream currently pushes the last 32 unseen posts on login). It is not a network-wide message archive and it does not store other people's direct messages.
- **When to use:** Fixed infrastructure serving as a community message hub, in a building, on a hilltop, or at an event site where a persistent group thread is useful.
- **Firmware file example:** `Heltec_v3_room_server-v1.16.0-07a3ca9.bin` (RAK4631 nRF52840 boards ship a `.uf2` room_server build)

GUI

For boards with a screen, the official flasher builds GUI firmware (and a GUI-with-SD-card variant). A GUI node is a Companion with an on-device interface, not a separate network role: it messages

and it does not relay.

KISS Radio

A KISS TNC build. It turns the node into a plain modem driven by a host computer, rather than a participant in the mesh in its own right. See [MeshCore KISS Modem Protocol](#).

Sensor: read this carefully

Sensor firmware is the most misunderstood item in this list, so be precise about what exists.

- The upstream MeshCore repository contains a `simple_sensor` example application. **The source exists.**
- **The official flasher does not build it.** The flasher's role list is Companion (BLE), Companion (USB), Repeater, Room Server, GUI, GUI-with-SD, and KISS. There is no Sensor entry, and the upstream releases contain no sensor binaries. You cannot flash a stock Sensor build from flasher.meshcore.io.
- To get a prebuilt Sensor binary you either compile it yourself or take one from a fork. [Keymind Cascade](#) ships Sensor builds for roughly 17 boards, and they are flashable from the Mesh America Device Configurator.
- Sensor readings are **not** broadcast in adverts. A MeshCore advert announces a node's presence, not its telemetry. Sensor data moves over MeshCore's request/response protocol. See [MeshCore Sensor Nodes](#).

Summary Table

Role	Messages	Relays for others	Stores messages	Prebuilt by official flasher
Companion (BLE / USB)	Yes	No	Its own only	Yes
Companion (Wi-Fi)	Yes	No	Its own only	No Self-compiled, or from a fork
Repeater	No	Yes, along known paths. Not everything	No	Yes
Room Server	No	No	Yes, that room's posts	Yes
GUI	Yes, on device	No	Its own only	Yes, on screen-equipped boards

KISS Radio	No, host-driven	No	No	Yes
Sensor	No	No	No	No Source is upstream; binaries from forks or your own build

Sources: the official MeshCore flasher catalog (flasher.meshcore.io), the MeshCore FAQ, and the firmware repository at github.com/meshcore-dev/MeshCore.

Revision #11
Created 2026-05-03 06:02:49 UTC by Mesh America Admin
Updated 2026-07-13 22:25:13 UTC by Mesh America Admin