

PlotLost

DigiHub Guide

Installation, configuration and everyday operation for your digital amateur radio station.

Debian · Raspberry Pi · WSL2

Updated 12 October 2026

[Read the latest guide at plotlost.com](https://plotlost.com)

Contents

Overview

Requirements

Installation

Getting Started

Identity, License & Location

Meshtastic GPS Forwarding

Digital Modes

Radio Services

GUI Applications

Raspberry Pi Notes

Windows Subsystem for Linux (WSL)

Updating & Removing

Command Reference

Credits

Overview

DigiHub is a curated collection of ham radio utilities and applications for digital ham radio operation. It's an alternative to the popular [DigiPi](#) - an excellent project and a great starting point for anyone new to digital modes - built around a different idea: instead of a fixed SD-card image, DigiHub installs onto an existing Debian system and stays reconfigurable afterward.

What DigiHub handles for you:

- **Callsign identity** - validates (US) callsigns, looks up license/name/address from a local FCC database (hamdb) kept current automatically, and generates the APRS passcode and a random AX.25 node password.
- **Location** - calculates the Maidenhead grid square from GPS or manually entered coordinates.
- **Configuration** - one set of settings, editable anytime from the terminal (`dhedit`) or a web browser (`dhweb`); works for an individual or a club callsign.
- **Cost** - entirely free and open-source, both DigiHub itself and everything it installs.

Requirements

- A Debian Trixie (64-bit) system, sudo access, and a network connection. This covers plain Debian, Raspberry Pi OS (tested on a Pi Zero 2W, 3, 4, and 5), and Windows Subsystem for Linux (WSL2) - see [WSL](#) below for the one caveat. The installation has also been tested on a ClockworkPi uconsole (<https://www.clockworkpi.com/home-uconsole>) with the HackerGadgets uConsole AIO V2 (<https://hackergadgets.com/products/uconsole-aio-v2>).
- hamdb.org's callsign lookup is reliable for the US, Canada, and Australia; the database updates daily for the Czech Republic and Germany, and monthly elsewhere. For countries without a reliable lookup, install with `./install.sh nodb` to enter your callsign and details manually instead.

Installation

Download DigiHub from the [PlotLost DigiHub page](#). Source downloads and the update repository are hosted here on PlotLost; a GitHub account is not required.

```
sudo apt update && sudo apt install git
git clone https://www.plotlost.com/static/downloads/DigiHubHam.git
cd DigiHubHam
chmod +x install.sh
./install.sh
```

Pass a callsign as an argument (`./install.sh W0FFS`) to skip the prompt, or `./install.sh nodb` for manual entry (see [Requirements](#)). If installation fails partway through, DigiHub rolls back everything it changed - packages, scripts, environment changes - except an OS update, which is never reverted.

Alternatively, download the [DigiHub source ZIP](#), extract it, open the DigiHubHam folder in a terminal, and run `chmod +x install.sh` followed by `./install.sh`.

Getting Started

Every DigiHub script lands in `/usr/local/bin` and has a full man page (`man dhedit`, `man dhmode`, etc.) - this guide covers what things are and how they fit together; the man pages cover exact usage, flags, files, and exit codes. `dhweb`'s Manual page (`/docs`) renders all of them in the browser too, via `man(1)` itself, for whenever a terminal isn't handy.

`dhweb`'s Home page (`http://<dighub-host>:8080/`) is a dashboard: a live GPS fix - read fresh on every load, not just whatever's cached in `.dhinfo` - shown first if a GPS device is configured, a summary of the current configuration, and a tile linking to every other page with a one-line description. Every page ends in a footer identifying DigiHub and its author.

Configuration lives in `$HOME/.dhinfo` and is edited with `dhedit` (terminal) or `dhweb`'s Configuration page (browser, installed and running by default at `http://<dighub-host>:8080/config`). Both validate and save fields the same way and back up the previous `.dhinfo` before every save, so it's safe to use whichever is convenient at the time. `dhweb` **has no authentication** and is meant only for a trusted local network - don't expose it to the open internet.

Most bash commands are thin wrappers around Python workers in DigiHub's own virtual environment; run one directly with `source "$DigiHubenv/bin/activate"` (and `deactivate` when done) if you need to.

For full shell access without SSH, `dhconsole` on starts a browser-based terminal (ttyd at `http://<dighub-host>:7681`) running directly as the DigiHub operator account. There's no login prompt - it follows the same trusted-network, no-auth model as the rest of `dhweb` rather than being the one page that asks. Off by default; anyone who can reach port 7681 while it's on has an unauthenticated shell, so don't expose DigiHub to the open internet.

`dhweb`'s System page (`dhpower`) reboots or shuts down the host, and can also reset DigiHub's configuration back to nothing (`dhreset` - deletes `.dhinfo` and its backup, leaving scripts/packages/systemd units installed but unconfigured). Each button needs an explicit browser confirmation first, since none of the three is reversible from `dhweb` itself. Rebooting brings DigiHub back on its own; shutting down needs a physical power-cycle to undo; a configuration reset has no undo at all. `dhreset` (from a terminal or the web console) asks for confirmation too unless run with `--yes`.

The Update page (`http://<dighub-host>:8080/update`) has no entry of its own in `dhweb`'s navigation - the System page shows a quick summary (is `$HOME/.dighub-update-available` present, the same flag file `sysinfo` checks) and links to it, rather than duplicating the full check-and-apply UI there.

Identity, License & Location

DigiHub keeps a local copy of the FCC amateur callsign database (`hamdb`, backed by MariaDB) so `dhedit/dhweb` can pull license, name, and address details automatically when you set a callsign. It's populated at install time and kept current by a daily timer (`dhhamdbupdate`); `hamdb full` re-imports everything by hand if needed.

For coordinates, DigiHub prefers a connected GPS device - detected automatically at install time (preferring a stable `/dev/serial/by-id` path, falling back to USB or native serial ports) and monitored afterward by `dhgpsmonitor`, which updates `.dhinfo`'s latitude/longitude/grid whenever the device moves far enough to matter. A [Waveshare LC29H](#) is a solid, tested choice. Without GPS, coordinates can be entered manually,

and DigiHub can optionally use [Mapbox](#) to fill them in from an address when hamdb doesn't have them - entirely optional, asked about at install time, editable anytime with `dhedit`.

Meshtastic GPS Forwarding

DigiHub can optionally forward fresh GPS coordinates to a local `meshtasticd` instance. `dhgpsmonitor` obtains the fix through `position.py`, then `gpsupdate.py` sends latitude and longitude through the Meshtastic TCP API at `127.0.0.1:4403`. This avoids having DigiHub and Meshtastic compete for the GPS serial port.

Requirements

- GPS configured in DigiHub and producing a valid fix.
- Meshtastic installed, configured for your radio and region, and running on the same machine.
- The Meshtastic Python CLI installed with `pipx` for the DigiHub user, available at `$HOME/.local/bin/meshtastic`.
- Meshtastic's direct GPS serial access disabled: remove or comment out its `GPS SerialPath` setting and restart `meshtasticd`. Keep any hardware-specific GPS power configuration needed by your board.

DigiHub uses an existing Meshtastic installation; it does not install or configure the radio hardware.

Enable during installation

When a GPS device is configured, `install.sh` offers:

```
Enable Meshtastic position forwarding? [y/N]:
```

Choose **y** to validate the Meshtastic Python environment, connect to the local daemon, and set `position.fixed_position=false` so incoming positions can update the node. The installer writes `/etc/systemd/system/dhgpsmonitor.service.d/meshtastic.conf`, setting `DigiHubMeshPython` to the Python executable in the existing Meshtastic environment, then reloads `systemd` and restarts `dhgpsmonitor`.

Forwarding defaults to disabled. An existing drop-in is preserved on reinstall. If Meshtastic is missing or cannot be configured, installation continues without enabling forwarding. To enable it later, rerun the current `install.sh` from your repository checkout. `dhupdate` updates scripts but does not run this installer prompt. There is currently no forwarding toggle in `dhedit`.

Behavior and verification

Each successful monitor fix is forwarded, including when movement is below DigiHub's threshold for updating `.dhinfo`. No fix means no new position is forwarded; Meshtastic may still display its previous position. Only latitude and longitude are forwarded, not altitude or satellite information.

The update targets the local node. Meshtastic's position-sharing settings govern subsequent LoRa broadcasts. Forwarding failures are logged and do not stop normal DigiHub GPS updates.

After enabling, allow a monitor cycle (normally about 30 seconds) with a valid GPS fix, then check:

```
systemctl is-active dhgpsmonitor meshtasticd
~/local/bin/meshtastic --host 127.0.0.1 --nodes
sudo journalctl -u dhgpsmonitor -b -n 50 --no-pager
```

The local Meshtastic node should show coordinates. This verifies the local position update; it does not verify reception by another LoRa node.

Disable forwarding

```
sudo rm -f /etc/systemd/system/dhgpsmonitor.service.d/meshtastic.conf
sudo systemctl daemon-reload
sudo systemctl restart dhgpsmonitor
```

This stops forwarding and leaves DigiHub GPS monitoring running. It does not clear Meshtastic's last position or restore its previous fixed-position setting.

Digital Modes

DigiHub can only run one Direwolf-backed digital mode at a time, since they share the same sound card or serial port. `dhmode <name>` switches between them, stopping whatever's active first - from the terminal or from `dhweb`'s Mode page.

Mode	What it does
standby	Nothing running (the default)
tnc	Plain KISS/AGW TNC (1200 baud) for other apps to use
tnc300b	Same, at 300 baud, for HF SSB packet
tracker	TNC plus an APRS position beacon
digipeater	TNC plus beaconing plus wide-area APRS digipeating
node	AX.25 packet node/BBS (uronode) - connect over the air and get a node prompt for mail and onward connections. Needs a kernel with AX.25 support (not WSL2's default kernel) and is logged into with the AX Node password from <code>.dhinfo</code> .
wsjtx / js8call / qsstv	GUI apps (WSJT-X , JS8Call , qSSTV) on a browser-based VNC desktop - opt-in at install time , see GUI Applications below.

`winLinkrms` (a future Winlink RMS Gateway mode) is the one remaining unimplemented mode. APRS WebChat and FLDigi are deliberately *not* `dhmode` modes - see below for why.

Radio Services

These run independently of the digital mode above (each can run alongside whichever mode, or none), off by default, toggled with `dh<service> on|off|status` or the matching `dhweb` page:

Service	Toggle	What it is
CAT control	dh <code>rig</code>	Hamlib <code>rigctl</code> d, giving other apps network CAT control instead of each needing its own serial connection. Needs a rig model number and device set first.
Winlink	dh <code>pat</code>	Pat, DigiHub's Winlink client. Has its own web UI, linked from <code>dhweb</code> rather than reimplemented.
ARDOP	dh <code>ardop</code>	<code>ardopcf</code> , most often used for Winlink-over-ARDOP with Pat. Also links to its own web GUI for audio levels.
APRS WebChat	dh <code>webchat</code>	<code>aprsd</code> browser-based APRS messaging, riding on whichever Direwolf-backed mode's KISS interface is active.

Radio Services all get their configuration from `.dhinfo` automatically (rig model/device, Winlink password, etc. - set them with `dhedit`/`dhweb` first) and cross-wire sensibly where it matters: turning on ARDOP wires Pat to control its PTT via CAT if a rig is configured, for example.

GUI Applications

FLDigi, WSJT-X, JS8Call, and qSSTV are all GUI-only ham radio apps with no headless mode, handled two different ways:

FLDigi is installed by default. Start it yourself at the console; `dhweb`'s FLDigi page then talks to its XML-RPC interface (on by default) for status and remote control - modem, frequency, TX/RX - without reimplementing its waterfall.

WSJT-X, JS8Call, and qSSTV need exclusive access to the same sound card Direwolf/ARDOP use, so each is a real `dhmode` mode (`dhmode wsjtx`, etc.), backed by a browser-accessible remote desktop ([TigerVNC](#) + [noVNC](#), no existing desktop environment required). Because this is the heaviest, most niche-interest part of DigiHub, it's **opt-in at install time** - declined by default, asked about right after the Mapbox prompt, and installable later by re-running `install.sh`. `dhweb`'s GUI Apps page reports install status for all three; its Mode page links directly to the remote desktop when one of these modes is active.

Raspberry Pi Notes

Audio HATs (Fe-Pi Audio, Audio Injector Zero, DRAWS-style boards - set via `.dhinfo`'s radio interface field) need a kernel device-tree overlay to work, applied with `sudo dhaudiohat` and a reboot; it's a no-op on non-Raspberry-Pi systems. GPIO status LEDs, physical pushbuttons, and TFT displays from DigiPi's own hardware add-ons are tied to one specific commercial HAT's exact wiring and are out of scope for DigiHub.

Windows Subsystem for Linux (WSL)

All scripts are tested and working in WSL2. A USB-attached GPS device needs `usbipd` installed and configured. The one exception is node mode (the AX.25 packet node/BBS), which needs a kernel built with AX.25 support that WSL2's default kernel doesn't have - it fails with a clear error rather than silently not working; everything else is unaffected.

Updating & Removing

New installations use the PlotLost update repository. To migrate an older GitHub installation, download and extract the current [source ZIP](#), then run `bash scripts/dhupdate` from its DigiHubHam folder once.

`dhupdate` syncs installed scripts against the latest PlotLost repository (new/changed/removed files, confirmed before applying; `dhupdate --yes` skips the prompt), restarting any affected running service automatically. A daily timer checks for available updates and surfaces a login-banner reminder; it never applies anything by itself. `.dhinfo` is restored automatically from its rolling backup if it ever goes missing. `dhweb`'s Update page offers the same check-and-apply from the browser, via the same NOPASSWD sudoers rule as the Mode/Rig Control/Winlink/ARDOP/WebChat/Console pages.

`dhremove` uninstalls DigiHub - packages, scripts, services, and (optionally) your `.dhinfo`/license data and Python environment. Run it interactively for a menu of what to keep; see `man dhremove` for the non-interactive flags.

Command Reference

Every command has a full man page (`man <command>`) with usage, arguments, files, and exit codes.

Command	Purpose
<code>aprspass</code>	Generate an APRS password
<code>axnodepass</code>	Generate a random alphanumeric AX Node password
<code>dhardop</code>	Turn DigiHub's ARDOP TNC (<code>ardopcf</code>) on or off
<code>dhardopd</code>	Runs <code>ardopcf</code> for DigiHub's ARDOP TNC
<code>dhaudiohat</code>	Activate the Raspberry Pi device-tree overlay for DigiHub's audio HAT
<code>dhconsole</code>	Turn DigiHub's web console (<code>ttyd</code>) on or off
<code>dhconsoled</code>	Runs <code>ttyd</code> for DigiHub's browser-based web console
<code>dhdirewolf</code>	Runs Direwolf for DigiHub's TNC/digipeater/tracker/node modes
<code>dhedit</code>	DigiHub configuration editor
<code>dhgpsmonitor</code>	Background service that updates <code>.dhinfo</code> when GPS position moves
<code>dhjs8call</code>	Runs JS8Call on a VNC + noVNC remote desktop for DigiHub's js8call mode
<code>dhmode</code>	Switch DigiHub's active digital mode
<code>dhnoded</code>	Runs <code>kissattach</code> + <code>ax25d</code> + <code>uronode</code> for DigiHub's AX.25 node/BBS mode
<code>dhqsstvd</code>	Runs qSSTV on a VNC + noVNC remote desktop for DigiHub's qsstv mode
<code>dhremove</code>	DigiHub uninstaller

Command	Purpose
dhpat	Turn DigiHub's Winlink client (Pat) on or off
dhpatd	Runs Pat's web interface for DigiHub's Winlink client
dhpower	Reboot or shut down the DigiHub host
dhreset	Reset DigiHub's configuration back to nothing
dhrig	Turn DigiHub's CAT control (rigctld) on or off
dhrigctld	Runs hamlib's rigctld for DigiHub CAT control
dhupdate	Sync installed DigiHub scripts against the PlotLost repository
dhweb	DigiHub web configuration interface
dhwebchat	Turn DigiHub's APRS WebChat (aprsd) on or off
dhwebchatd	Runs aprsd's webchat command for DigiHub's APRS WebChat
dhwsjtxd	Runs WSJT-X on a VNC + noVNC remote desktop for DigiHub's wsjtx mode
hamdb	FCC Amateur Radio license database
maidenhead	Calculate a Maidenhead ham grid from latitude and longitude
position	Get current GPS position from GPS device
qrz	Check a US callsign using hamdb, hamdb.org & Mapbox
sysinfo	System information
whohami	Show user information held for current configuration

Note: hamdb is installed using MariaDB rather than MySQL; starting with MariaDB 10.5, `mysqlimport` was replaced with `mariadb-import`, and the version distributed by PlotLost reflects that but is otherwise identical.

Credits

	Link	Purpose
hamdb.org	https://hamdb.org	API Calls (FCC)
mapbox	https://www.mapbox.com	Geolocation
Direwolf	https://github.com/wb2os/direwolf	Direwolf
scripts	https://github.com/dslotter/ham_radio_scripts	FLdigi Installation
HamPi	https://github.com/dslotter/HamPi	Content
hamdb	https://github.com/k3ng/hamdb	Hamdb (MariaDB)
DigiPi	https://digipi.org	Concept & content
usbipd	https://github.com/dorssel/usbipd-win	USB Passthrough (WSL)
Flask	https://flask.palletsprojects.com	Web Interface (dhweb)
waitress	https://github.com/Pylons/waitress	Web Interface (dhweb)
Hamlib	https://github.com/Hamlib/Hamlib	CAT Control (rigctld)
Pat	https://github.com/la5nta/pat	Winlink Client
ardopcf	https://github.com/pflarue/ardop	ARDOP TNC
uronode	https://sourceforge.net/projects/uronode/	AX.25 Node/BBS
aprsd	https://github.com/craigerl/aprsd	APRS WebChat
aprsd-webchat-extension	https://github.com/hemna/aprsd-webchat-extension	APRS WebChat
WSJT-X	https://wsjt.sourceforge.io/	Weak-signal modes
JS8Call	http://js8call.com/	JS8 messaging
qSSTV	https://github.com/ON4QZ/QSSTV	Slow-scan TV
TigerVNC	https://tigervnc.org/	Remote Desktop
noVNC	https://novnc.com/	Remote Desktop (browser bridge)
Fluxbox	http://fluxbox.org/	Remote Desktop (window manager)
tttyd	https://github.com/tsl0922/ttyd	Web Console
FLDigi	https://w1hkj.org/	Digital Modes (XML-RPC control)