Ham Signaling

Introduction

Ham Signaling is Unixeer's digital-modes companion for amateur radio: it decodes and transmits the open digital modes using the audio from your radio. Point your phone at the rig's speaker and it turns the tones into text; type a reply and it keys the mode back out. This release operates fourteen modes — FT8, FT4 and JS8 weak-signal work, a WSPR transmit beacon, the PSK31, PSK63, PSK125, RTTY, Olivia and Contestia keyboard modes, Hellschreiber's painted text, APRS and Packet, and a multilingual Messaging mode. Every one is open, with no proprietary codecs and no licensing walls, and the work is signal processing, not paperwork.

What this build is

This is the first public release: a complete operating app, with the Modes console, Settings, and the standard About, Support, Manual and Version-history pages, on the family theme with weekly tips and global error handling throughout. Eleven modes are live today: ten of them decode and transmit, and WSPR transmits as a propagation beacon.

Open modes only

Ham Signaling deliberately covers only open amateur modes. It is not an image-mode app: SSTV and other picture modes live in Ham Imaging, and Ham Satting handles ISS SSTV, so those are intentionally absent here rather than duplicated.

You must be licensed

Transmitting any of these modes puts real signals on the amateur bands. Ham Signaling is an operating tool, not a licence — you must hold a valid amateur radio licence and stay within your privileges whenever you transmit.

The Modes Screen

The Modes screen is where you operate — everything happens here, with no jumping between pages. At the top are a mode dropdown and a single button; the space below them becomes the operating console once you start.

Initiate and Terminate

Pick a mode from the dropdown — it lists the modes you have switched on in Settings, Digital modes — and press Initiate. The dropdown locks and that mode's full console opens below: the waterfall, the decode window, a message box and the controls. Press Terminate to close it and choose another mode. Your last-used mode is remembered, so it is already selected the next time you open the app.

The built-in tutorial

Before you press Initiate, the Modes screen shows the selected mode’s tutorial: an Input line naming what the mode can carry (English/ASCII only, auto-wrapped scripts, or any language), followed by How to use, Best Practice and Usage Examples panels. While operating, the same

tutorial stays one tap away behind the (i) at the top right of the app bar, and in Settings a small (i) sits beside every mode toggle — so the help is available everywhere, for every mode.

Logging & sharing QSOs

Ham Signaling keeps a small log of the contacts you make. On FT8 and FT4 a completed exchange is logged automatically; on RTTY, PSK and JS8 the bookmark button on the console opens a short form (their callsign, signal reports and grid) that you save by hand. In Settings, the QSO log panel shows how many contacts are stored and lets you share them all as an ADIF file — import that file into Ham Logging or any other logger. The reply icon on a received line fills the message box with the standard next transmission, and the re-use icon resends one of your earlier messages.

Listen and Send

Inside the console, press Listen to start decoding from your radio's audio, and hold the phone to the speaker. Type a message and press Send to transmit it. Your sent messages appear in the receive view in an accent colour so they stand out from received traffic, each with a small re-use icon that drops the message back into the box to send again; the box clears itself after each send.

When the list is empty

If you switch every mode off, the Modes screen shows a short hint pointing you to Settings, Digital modes. Turn any mode back on and it reappears in the dropdown immediately.

Settings

Settings is organised into two panels plus the resets. Everything auto-saves the moment you change it — there is no Save button — and every value is stored only on your device.

Your Details

The first panel holds your operator details and the app's look and feel. Enter your callsign and Maidenhead grid (both optional; they tag your decodes and pre-fill the Support form) and a contact email. Below them are the appearance controls: a font-size slider, a dark-mode switch, a weekly-tips switch, a Waterfall theme dropdown (Ocean, Classic, Green and Matrix), and a Keep decoding in background switch. Identifier fields auto-capitalise and the grid uses Maidenhead formatting as you type.

Digital modes

The second panel lists every mode with its own on/off switch. Turn on the modes you operate and they appear in the Modes dropdown. A Reset digital-mode settings button at the foot of the panel turns every mode back on and clears any per-mode settings, while leaving your personal details and appearance untouched. The small (i) beside each mode opens that mode’s tutorial, so you can read about a mode before enabling it.

Resets and storage

A separate Reset all settings to defaults button below the panels restores every app setting — callsign, grid, email, theme, font size and tips — to its default. Both resets ask for confirmation first and cannot be undone. Nothing you enter in Settings ever leaves the device: there is no account and no Unixeer server.

How the Modes Work

The fourteen modes share one operating pattern — audio in, waterfall, decode, transmit — with a few differences worth knowing. The one exception is WSPR, which only transmits and has no decode side.

Audio in from the rig

Feed your radio's receive audio to the app acoustically: hold the phone's microphone near the radio's speaker. Watch the input-level meter under the waterfall and set the radio's volume so the bar sits in the green — too loud clips and garbles the decode, too soft misses weak signals.

Transmit is half-duplex

Press Send and the app plays the mode's tones out of the phone's loudspeaker; hold it to the rig's microphone (or couple it in) and key up. Receiving pauses while you transmit and resumes automatically when the transmission finishes.

The waterfall

A live waterfall scrolls the received passband in the palette you chose in Settings, so you can see signals and tune the radio to place them cleanly. The waterfall is interactive: on the keyboard modes, tap a signal’s trace to tune the decoder — and your transmitter — to it, and on FT8, FT4 and JS8 tap to place your transmit offset, shown as a red line. A soft automatic gain control lifts a quiet signal so the trace stays readable (the level meter still shows the true input), and the zoom button in the corner switches between the full span and the HF audio window where the modes live.

Keyboard modes — RTTY and PSK31/63/125

Decoded text scrolls in a single window. Type in the multi-line box and press Send; the text goes out as tones and is echoed into the window in the accent colour. PSK comes in three speeds, each its own mode in the dropdown — PSK31 is the narrow, sensitive classic, PSK63 and PSK125 trade bandwidth for typing speed. RTTY adds two selectors on the console: baud (45.45 or 75) and shift (170, 425, 450 or 850 Hz), so you can match whatever flavour is on the air. Under the entry field a live counter shows the character count and an estimate of the time on air for the current mode, updating as you type. The on-air estimate under the entry reflects the exact transmission length, including multilingual-wrapped text; wrapped (non-English) messages are limited to roughly 1,280 bytes so the receiving station can always rebuild them. Off-air signals need not sit exactly on frequency: the decoders acquire, track and follow slow drift by

themselves (RTTY within about ±100 Hz of the tone pair, PSK within about ±45 Hz of the tapped carrier), the small Hz readout beside the level meter turns green when locked, and a squelch keeps noise from printing garbage. RTTY also follows the on-air unshift-on-space convention, so digit groups like 599 survive fades.

Hellschreiber — painted text

Hellschreiber (Feld Hell) is the oldest mode here and works unlike any other: it does not decode to text at all — it paints. Each received column of dots is drawn straight onto a scrolling canvas and you read the letters with your eye, the way a fax machine prints, so a slightly fuzzy character is still readable and a little noise or drift does no harm. Type a short, uppercase message and press Send; the app keys it out as simple on/off audio tones. It is slow, so keep to a few words at a time. Tune a Feld Hell frequency such as 14.063 MHz USB.

Olivia and Contestia — robust MFSK

Olivia and Contestia are robust keyboard modes that shift between several tones and add heavy forward-error correction, so they recover text well below the noise where even PSK31 fails — at the cost of speed. Each comes in variants named by tone count and bandwidth (for example Olivia 16/500 or Contestia 8/250); pick a variant from the console and make sure the other station is using exactly the same one, or nothing will decode. Contestia is the faster, slightly less robust cousin of Olivia. Both carry mixed-case text; keep overs short. A common frequency is around 14.076 MHz USB.

Weak-signal modes — FT8 and FT4

These are slot-timed: FT8 works in 15-second windows aligned to UTC, FT4 in 7.5-second windows. Each finished slot is decoded to a list of callsigns, grids and reports, with your own callsign and grid highlighted. Press Send to arm a message and it transmits on the next slot boundary — the button shows an Armed countdown until it fires. Keep your phone's clock on automatic (UTC-synced) time or nothing decodes. Messages are checked the moment you press Send — anything the protocol cannot pack is refused immediately rather than after the countdown. Tap the waterfall to place your transmit audio offset (the red line) in a clear spot before arming.

JS8 — keyboard messaging on slots

JS8 combines the two families: it uses the FT8 waveform on the same 15-second UTC slots (Normal speed) but carries free-text keyboard messages, and it is compatible with the JS8Call network. A message longer than one frame is split across consecutive slots and sent frame by frame — the counter under the entry box shows how many frames and roughly how long on air the text needs — and received multi-frame overs are reassembled into one message in the decode window. As with FT8 and FT4, keep the phone's clock on automatic (UTC-synced) time.

WSPR — a transmit-only beacon

WSPR is different from every other mode here: it only transmits. The beacon sends your callsign, grid and power (dBm) as a two-minute transmission starting on the even UTC minutes, and repeats it while it is running. Receiving stations around the world report what they hear to wsprnet.org — search your callsign there to see where your signal reached. Ham Signaling does not decode WSPR, so spots appear on wsprnet.org, never in the app; the beacon also stops while the app is in the background. Set your callsign and grid in Settings before starting it.

Data modes — APRS and Packet

These decode AX.25 frames at 1200 baud. Received frames scroll in a monitor in TNC-2 form; to transmit, type a frame and press Send. Frames are English/ASCII and capped at the AX.25 size limit; the composer refuses anything a receiving station could not decode — for other scripts, use the Messaging mode. Position reports are decoded to coordinates, and when your grid is set the app adds the distance and bearing to the station.

Messaging — any language

The Messaging mode carries short text over the same 1200-baud packet link. Type in any language: plain English goes out as an ordinary packet frame that any TNC can read, while Arabic, Chinese, Japanese and other scripts are carried so another Ham Signaling station shows them in the original writing. The same multilingual carriage rides the keyboard modes too — PSK31, PSK63, PSK125, RTTY and JS8 — with the identical rule: plain text stays plain for every station on the mode, other scripts arrive intact on another Ham Signaling station. A frame holds a limited number of bytes, so the composer counts the space left as you type — multi-byte scripts use it faster. Set your callsign in Settings before you send.

Themes & Accessibility

Ham Signaling ships the family's light and dark themes on the standard ocean-blue accent. Toggle dark mode from Settings, Your Details and the change applies instantly across every screen.

Font scaling

The font-size slider scales text across the whole app from 80% to 140%, so you can size the interface for a bench display or a bright field day without leaving the app.

Waterfall themes

The Waterfall theme dropdown — Ocean, Classic, Green and Matrix — sets the colour scheme of the decode waterfall, and the change takes effect immediately in any open console.

Support & Privacy

Support

The Support page sends a message to the developer through your own email app, pre-filled with your callsign, email and the app version so a report carries the context it needs. You can reach Unixeer at unixeer@gmail.com.

Privacy

Everything Ham Signaling stores — your callsign, grid, email and settings — stays on your device. There is no account, no Unixeer server, no analytics and no ads.

Microphone & background

Ham Signaling uses the microphone only to listen to your radio's audio so it can decode — never for anything else. The Keep decoding in background switch (on by default) lets decoding continue when the app is minimised or the screen is off, shown by an ongoing notification on Android; turn it off and decoding stops the moment you leave the app.

About & Credits

About

The About page shows the current version, a summary of the app, and links to this manual, the Support form and the version history.

Credits

Ham Signaling is part of the Unixeer family of amateur-radio apps by Yousuf AL Balushi, A41UX. Unixeer™ — RF for Life.

Rig Control (CAT)

What rig control does here

Ham Signaling can read your radio so that every QSO you log carries the true dial frequency and band, taken straight from the rig at the moment you work the station — no more typing the frequency by hand or trusting the band you think you were on. When a radio is connected, the frequency and mode shown in the app follow the knob on your rig.

This is READ-ONLY rig control, and that guarantee is deliberate. Ham Signaling asks the radio only for its current frequency and mode. It never sends a command that changes the radio, never tunes it, never switches VFOs or modes, and never keys the transmitter. Your rig control

link cannot put you on the air — transmit in this app is always your own audio through the microphone path, exactly as described in “How the Modes Work.” The rig link only listens, so pointing it at your station is safe even in the middle of a contest.

Because it only listens, rig control is optional. Everything in the app works without it; connecting a radio simply makes your log more accurate.

Turning it on

Rig control is off by default. Open Settings and find the Rig Control block. Switch it on, choose your radio, and enter the connection details for whichever link you are using. Once a radio is selected the app shows a small status chip: it turns green and displays the live frequency and mode when the rig is answering, and stays neutral when nothing is connected. You can leave rig control switched on permanently — the link suspends cleanly when the app is in the background and re-establishes itself when you return.

One platform note. Reaching a radio over your home network or over Bluetooth needs the operating system's permission for local-network or Bluetooth access; grant it when the phone asks on the first connection. USB radios are Android only — Apple does not allow USB serial on iPhone or iPad, so on iOS you reach a USB rig through a network or Bluetooth bridge instead.

Choosing your radio

The radio picker is a searchable list — start typing your model (for example “705”, “890” or “K4”) and the list narrows as you go. Picking a specific model matters: it sets the correct command dialect, the default baud rate, and, for Icom rigs, the CI-V address, so you rarely have to fill those in by hand.

Every entry carries an honest support label. VERIFIED means the model has been exercised on real hardware and confirmed working. IMPLEMENTED-PER-SPEC means the codec is written from the manufacturer's published CAT documentation and should work, but has not yet been proven on that exact radio on the bench — it is included so you are not blocked, with the caveat stated plainly rather than hidden. Both are usable; the label simply tells you how far the model has been tested.

A word on radios that are not listed: the FTM-family handhelds and mobiles (Yaesu's C4FM FTM series) have no CAT port at all, so no app can read their frequency. If you cannot find your radio, check first whether it actually offers a CAT / serial / network control interface — many purely operating-position rigs do not.

Connecting over the network

If your radio, or a computer attached to it, is on your Wi-Fi or LAN, the network path is usually the easiest. Ham Signaling speaks several network dialects directly:

rigctld (Hamlib). Point the app at a running Hamlib rigctld daemon by host and port; the default port is 4532. This is the universal fallback — anything Hamlib supports, rigctld exposes. wfview

publishes the same rigctld emulation on port 4533, so tick “Enable RigCtld” in wfview and connect to that port to reach a USB Icom through your PC or Mac.

Icom WLAN direct (RS-BA1 over Wi-Fi / LAN). Network-capable Icoms — IC-705, IC-9700, IC-7610, IC-905, the IC-7300 MK2 and others — can be reached over the air with no computer in between, using the same RS-BA1 server protocol as Icom's own software. On the radio, set up a user name and password and turn Network Control on (on the IC-705 use its 2.4 GHz network); then enter the radio's address and those credentials in the app. Note that the original IC-7300 (MK I) has no network port — reach it through wfview or rigctld instead.

Kenwood KNS. The TS-890S and TS-990S offer network CAT over their built-in KNS server on TCP port 60000. Enable KNS on the radio, create the login, and enter the host and your credentials. The TS-890S is verified; the TS-990S is implemented per spec. (The TS-590S / SG are not directly reachable — they need Kenwood's ARHP host software, so use rigctld for those.)

Elecraft K4. The K4 speaks CAT over Ethernet on TCP port 9200 — enter the radio's address and that port. Multiple clients are allowed, and the radio advertises itself on the network, so a K4 on your LAN is a one-line setup.

Connecting through a serial bridge

The Network Serial Bridge option turns any transparent serial-over-TCP pipe into a rig link, which means practically every serial-CAT radio ever made becomes reachable from your phone — including from iOS. You give the app a host and port; the bytes flow straight through to the radio's serial port at the other end. Pick your exact radio model in the picker as usual so the app knows which dialect to speak; the baud rate lives in the bridge's own configuration.

Free software bridges. ser2net (version 4) running on a Raspberry Pi, Linux, macOS or Windows publishes a serial port as a TCP port in three lines of YAML — an accepter line for the TCP port, a connector line naming the serial device and its baud string (for example 38400n81, or 4800n82 for a Yaesu FT-8x7's 8N2 framing), and kickolduser: true. Be sure to use the version-4 YAML syntax, not the old version-3 colon format. wfview also exposes a raw CI-V TCP port for a USB-attached Icom, which the app's CI-V codec drives directly.

WiFi-serial dongles. If you would rather not run a computer, a small WiFi-to-serial adapter does the same job. The USR-W610 (a solid RS-232 unit with a web setup page) is the first recommendation; the Elfin EW10 / EW10A (RS-232) and the EW12 (TTL) are cheaper options. Do not use the EW11 — it is RS-485, which is wrong for CAT. Configure the dongle as a transparent TCP server on your Wi-Fi, then give the app its address and port.

Two safety facts worth knowing before you wire anything. First, PTT keying over the RTS/DTR control lines does not cross a transparent bridge, so keying is never something this app does anyway — it stays read-only. Second, mind the electrical levels: an Icom CI-V jack is a single-wire ~5 V bus and supplies no power; a Yaesu FT-817/818/857/897 CAT port is 5 V TTL at 8N2 and is damaged by RS-232 levels (that is what a CT-62-style converter is for); DB9 rigs such as the

Kenwood TS-480/570/590/2000 and the Elecraft K3/K3S are plain RS-232 and need only a null-modem check. A transparent bridge has no password, so keep it on your WPA2 Wi-Fi and never forward its port to the internet.

Connecting over USB (Android)

On Android you can plug the radio straight into the phone with a USB-OTG cable — the same CI-V, Kenwood, Elecraft or Yaesu cable you would use with a computer. Android re-asks for USB permission each time you connect, so approve the prompt when it appears. A few per-rig points save a lot of head-scratching:

Icom. On the radio's USB CI-V menu, set CI-V USB Port to Unlink from [REMOTE], set the CI-V baud to 115200, and turn CI-V USB Echo Back on. The IC-705 and IC-905 present as CDC-ACM devices (not the usual CP210x), which the app handles automatically.

Yaesu. The modern CAT radios (FT-991A, FTDX10, FT-710, FTDX101, FTX-1) use the CP2105 Enhanced port for CAT — except the FT-891, which is inverted: its CAT is on the Standard port, so select port index 1. The legacy FT-817/818/857/897 use 8N2 framing; that path is Android-only.

Elecraft and others. The KX2 / KX3 connect at 38400; the K4 exposes two ports, with control on the first. The Xiegu X6200 runs its CAT on the SERIAL-B port at 19200. If a radio offers a USB-audio function alongside CAT, note that “disabling USB audio” only affects audio and never the CAT link.

Connecting over Bluetooth

The IC-705 has Bluetooth Low Energy built in and can be read wirelessly with no extra hardware. On the radio, open SET > Bluetooth > Pairing Reception for the first pairing, then connect from the app; Ham Signaling remembers the pairing so later reconnects are automatic. This works on both Android and iOS.

For any other TTL-serial radio you can add a small BLE-UART bridge module. There is still no off-the-shelf Bluetooth CAT dongle that works with iPhones, but a DIY module does, and it is App-Store-legal. Two modules are recommended: the DSD TECH HM-19 (genuine TI chipset, default 9600-8-N-1, powered at 3.6–6 V — never from the 13.8 V shack rail) and the Adafruit Bluefruit LE UART Friend. Strongly avoid unbranded HM-10 clones (the CC41-A copies), which suffer scanning and pairing faults. Wire the module's TX and RX to the radio's serial pins at matching levels, select your radio in the picker, and connect. BLE works on both platforms.

On Android only, radios or adapters that use classic Bluetooth Serial Port Profile (SPP) are also supported. Apple does not permit classic-SPP serial on iOS, so on iPhone and iPad the BLE paths above are the wireless route.

When no radio is connected

Rig control is a convenience, never a requirement. If no radio is connected — because you have not set one up, or the link has dropped — the app simply falls back to the frequency you enter yourself, and the status chip stays neutral so you are never misled into thinking a stale reading is live. Your logging, decoding and transmitting all continue exactly as normal; the only difference is that you fill in the frequency by hand. When the radio comes back, the live reading resumes and QSOs are stamped from the rig again.