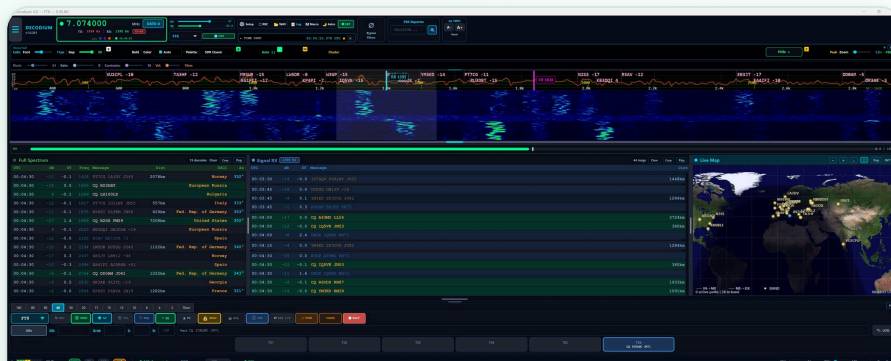


● PUBLIC BETA · V1.0.262 · COMPLETE USER MANUAL

# DECODIUM 4.0

“SHANNON” – COMPLETE USER MANUAL



▸ Chapters 1-10 · 10 chapters · 20+ features · 3 platforms

**10**

CHAPTERS  
The whole software

**-23 dB**

RAPTOR SNR  
5 decode passes

**ASYMX**

ASYNC TX  
No Even/Odd · No NTP

**3 OS**

NATIVE BUILDS  
Win / macOS / Linux

## Chapter 1 – Getting Started

### 1.1 What DECODIUM 4.0 is

DECODIUM 4.0 “Shannon” is an open-source platform for amateur radio digital modes, rewritten from scratch in C++ over six months of intensive development. It is heir and successor to DECODIUM 3.0 (“Raptor”) and introduces three structural innovations:

1. **C++ multi-decode engine** – the old Fortran core has been replaced by an OpenMP-parallel decoder that uses every available CPU core. Fortran code remains isolated only for LDPC, where the original mathematics by K1JT is unmatched.
2. **Native FT2 mode** – a new digital protocol with a 3.8-second T/R cycle, now certified ADIF 3.1.7 as `SUBMODE=FT2` (unanimous 22:0 vote).
3. **ASYMX – asynchronous transmission** – the first HF digital mode in history that requires no clock sync (no NTP, no GPS).

The stated goal of v4.0 is twofold: deliver to the amateur radio world **a decoder beyond the state of the art** (–23 dB SNR sensitivity on FT2 Raptor, +1.5 dB on fading channels versus WSJT-X baseline), and provide **a modern operating experience** (integrated Live Map, automatic Call Roster, robust CAT for Yaesu/Kenwood/Icom, native cross-platform support).

## 1.2 Differences vs DECODIUM 3.0 and WSJT-X

For those coming from DECODIUM 3.0 or WSJT-X / JTDX / MSHV, here are the substantive differences:

AREA	DECODIUM 3.0	WSJT-X	DECODIUM 4.0
Language	Fortran + Qt 5	Fortran + Qt 5	<b>C++ + Qt 6.11 + QML</b>
Multi-decode	Yes, Raptor	No	<b>Yes, Raptor + best-of</b>
FT8 sensitivity	-23 dB	-21 dB	<b>-23 dB</b>
FT2 mode	No	No	<b>Yes, native</b>
ASYMX async	No	No	<b>Yes (FT2 only)</b>
Integrated Live Map	No	No	<b>Yes</b>
HRD CAT bridge	Partial	Limited	<b>Robust, auto-reconnect</b>
Yaesu TCI bridge	No	No	<b>In development (Phase 1)</b>
Multi-instance	No	Yes	<b>Yes, MultiRig CLI</b>
macOS Apple Silicon	No	x86 (Rosetta)	<b>Native M1/M2/M3</b>
Linux	Compile-only	Source	<b>Portable Appliance</b>
License	GPLv3	GPLv3	<b>GPLv3</b>

**Important:** DECODIUM 4.0 is backward-compatible with DECODIUM 3.0 logs and settings. On first launch, the application detects any previous installation and offers to import the preferences.

# 1.3 System requirements

## Minimum requirements

COMPONENT	MINIMUM	RECOMMENDED
CPU	Dual-core 2.0 GHz	Quad-core 2.5+ GHz
RAM	4 GB	8 GB
Storage	200 MB free	1 GB free (for logs/cache)
Audio	Sound card compatible with radio	Dedicated USB CODEC interface
CAT connectivity	USB-Serial or CI-V cable	Manufacturer-certified cable
Internet	Not required	Recommended for Live Map / PSK Reporter / DX Cluster

## Supported operating systems

- **Windows 10/11 x64** – Inno Setup installer, runtime included
- **macOS 12.x+ Apple Silicon** – native M1/M2/M3 DMG (build by 9H1SR)
- **macOS 12.x+ Intel** – support via Rosetta, reduced performance
- **Linux x86\_64** – portable ApplImage (Qt 6.11), tested on Debian 12, Ubuntu 22.04+, Arch
- **Raspberry Pi (ARM Linux)** – LU7DID community port, ideal for portable stations and low-power use

**Note on Windows 10:** requires at least version 21H2. Earlier versions may have issues with Qt 6.11 and QML drag handles.

## 1.4 Project philosophy

DECODIUM 4.0 follows three guiding principles:

1. **Technical transparency** – every innovation is documented and credited. The LDPC decoder inherits K1JT's mathematics (WSJT-X), the Qt UI framework is by The Qt Company, and this heritage is explicit in the code and credits.
2. **Operational robustness** – we prefer a slow and reliable decoder over a fast and unstable one. The HRD CAT bridge mode has been rewritten three times to guarantee data-mode persistence.
3. **Community first, product second** – the Telegram tester community is the primary development organ. Bug reports, suggestions, and feedback have absolute priority in the release cycle.

---

# Chapter 2 – Installation

---

This chapter guides installation on every supported platform. Jump directly to the section that interests you.

## 2.1 Windows x64

### Standard procedure

1. Open your browser at <https://github.com/iu8lmc/Decodium-4.0-Core-Shannon/releases/latest>

2. In the **Assets** section, download `Decodium_1.0.262_Setup_x64.exe` (~90 MB)
3. Double-click the downloaded file
4. If Windows shows the SmartScreen warning, click **More info** → **Run anyway** (the binary isn't Microsoft EV-signed; it's community-certified)
5. Follow the wizard. The installer will ask for:
  - **Install folder** – default: `C:\Users\  
<your_user>\AppData\Local\Decodium`
  - **Create Desktop shortcut** – recommended
  - **Launch on finish** – optional

## User mode vs administrator

The DECODIUM installer requires `PrivilegesRequired=lowest`. This means:

- No admin password prompt
- Installs in user profile (not `Program Files`)
- Automatic cleanup of `*.qmlc` caches from previous installations
- Upgrade over 1.0.257/258/259/260 without issues

## Upgrading from v1.0.257 or earlier

The installer detects a previous version and: 1. Automatically backs up `decodium.ini` 2. Uninstalls compiled QML files (`.qmlc`) from the old version 3. Overwrites the executable and resources 4. Preserves ADIF logs, preferences, and callsign dictionary

**Important:** If you've manually customized `decodium.ini`, make a separate backup before upgrading. The installer respects the file but in case of a crash you might lose it.

## Uninstallation

Control Panel → Apps & Features → Decodium → Uninstall.

Uninstallation removes the executable but **preserves logs and user preferences**. For a complete removal, manually delete:

- `C:\Users\`
- `C:\Users\`

## 2.2 macOS Apple Silicon

### Native M1/M2/M3 build

The macOS build is personally maintained by Salvatore 9H1SR and is **native ARM64** – no Rosetta emulation, full performance.

1. Download `Decodium-1.0.262.dmg` (or the zip) from the release
2. Double-click the `.dmg`
3. Drag the DECODIUM icon to the **Applications** folder
4. Eject the DMG from Finder

### First launch and Gatekeeper

On first launch, macOS will block the app because it's not Apple Developer-signed (the signature costs \$99/year, unsustainable for community-driven GPL projects):

1. Go to **Applications**, find DECODIUM
2. **Right-click** (or Ctrl+click) → **Open**
3. The security dialog appears with a warning – click **Open** (this only happens once)
4. macOS will remember the authorization for subsequent launches

If the app still refuses to open, from terminal:

```
xattr -cr /Applications/DECODIUM.app
```

Removes quarantine attributes and allows execution.

## Required permissions

On first launch macOS will ask for:

- **Microphone** – required to read audio from the radio
- **Local network** – for UDP broadcast (Log4OM, GridTracker, etc.)
- **Documents / Downloads folders** – for ADIF file access

Grant all requested permissions, otherwise decoding will silently fail.

## 2.3 Linux x86\_64

### AppImage (recommended)

The AppImage is a single executable file containing all dependencies (Qt 6.11, libsndfile, libfftw3, hamlib 4.7):

```
# Download
wget https://github.com/iu8lmc/Decodium-4.0-Core-Shannon/releases/download/v1.0.262-call/Decodium-1.0.262-x86_64.AppImage

# Make executable
chmod +x Decodium-1.0.262-x86_64.AppImage

# Launch
./Decodium-1.0.262-x86_64.AppImage
```

The AppImage requires no installation. You can move it anywhere (`/opt/`, `~/Applications/`, desktop) and launch it from there.

### Desktop integration

To have it in the application menu with an icon:

```

mkdir -p ~/.local/share/applications/
cat > ~/.local/share/applications/decodium.desktop << EOF
[Desktop Entry]
Name=DECODIUM
Comment=FT8/FT2 Digital Mode Decoder
Exec=/full/path/to/Decodium-1.0.262-x86_64.AppImage
Icon=decodium
Terminal=false
Type=Application
Categories=HamRadio;AudioVideo;
EOF
update-desktop-database ~/.local/share/applications/

```

## Runtime dependencies

The AppImage includes almost everything, but the host system needs:

- **FUSE** (to mount the AppImage) – usually already present, otherwise  

```
apt install fuse libfuse2
```
- **PulseAudio** or **PipeWire** for audio
- **ALSA** for direct sound card access (optional)

On minimal distros (e.g. Debian server with graphical environment installed afterwards) you might need:

```

sudo apt install libxcb-cursor0 libxcb-icccm4 libxcb-image0
\
libxcb-keysyms1 libxcb-randr0 libxcb-
render-util0 \
libxcb-shape0 libxcb-sync1 libxcb-xfixed0
\
libxcb-xinerama0 libxcb-xkb1 libxkbcommon-
x11-0

```

## Serial permissions (CAT)

To access the CAT serial port without sudo, add your user to the

```
dialout
```

group:

```
sudo usermod -aG dialout $USER
# Log out and back in
```

Verify with:

```
groups # should show "dialout" among the groups
ls -l /dev/ttyUSB0 # should show "dialout" group
```

## 2.4 Raspberry Pi (ARM Linux)

The Raspberry port is maintained by **LU7DID** and is available as a separate repository. It's particularly suited for:

- Portable FT2 stations with low power consumption
- Embedded installations (e.g. remote-rig on the rooftop)
- Educational and laboratory setups

Visit <https://github.com/lu7did> for the specific instructions.

**Compatibility:** - Raspberry Pi 4 (8 GB RAM) – full functionality, multi-pass decode - Raspberry Pi 5 – recommended, performance comparable to x86 - Raspberry Pi 3 – limited, not recommended for FT2 multi-decode

## 2.5 Installation verification

After installation, launch DECODIUM. The top bar must show:

```
Decodium 4.0 – FT8 – IU8LMC
```

where **IU8LMC** will be replaced by your callsign (empty on first launch).

At the bottom left there should appear:

- **S:** signal level indicator
- **MON | TX | DEC | FT8** toggles

- **CAT: Disconnected** (yellow) – normal on first launch
- **PWR: --** – will populate after CAT configuration
- **Decodium 4.0** bottom right

If you see all of this, **the installation succeeded**. Proceed to Chapter 3 for configuration.

---

## Chapter 3 – Initial Setup

---

This chapter guides step-by-step station configuration. It's meant to be followed in order, because some choices (e.g. CAT) influence others (e.g. PTT).

### 3.1 Callsign and locator configuration

Open **Setup** from the hamburger menu (☰) at top-left, **My Station** section:

FIELD	EXAMPLE	NOTES
<b>My Call</b>	IU8LMC	Your callsign, without prefixes/suffixes (add separately if needed)
<b>My Grid</b>	JN71DC	6-character locator recommended (4 for standard QSO, 6 for finer map precision)
<b>DXCC</b>	Auto-detect	Sets automatically based on prefix
<b>Antenna</b>	e.g. Dipole 40m, 12m AGL	Informational only, appears in ADIF log
<b>Operator</b>	Empty or IU8LMC	For stations operated by multiple people (contest, special event)

**Tip:** For the 6-character locator, use <https://aprs.fi> or [http://www.levinecentral.com/ham/grid\\_square.php](http://www.levinecentral.com/ham/grid_square.php) to calculate from your station's GPS coordinates.

Press **Save**. The callsign appears in the title bar ( `Decodium 4.0 – FT8 – IU8LMC` ).

## 3.2 CAT configuration

CAT (Computer Aided Transceiver) is the communication protocol between software and radio. DECODIUM 4.0 supports three backends:

1. **Hamlib** – universal, recommended for most modern radios
2. **OmniRig** (Windows only) – alternative when other programs share the radio
3. **HRD CAT bridge** – for those using Ham Radio Deluxe as a central hub

### 3.2.1 CAT setup with Hamlib

Open **Setup** → **Radio**:

1. **Rig** – autocomplete field. Start typing the model:

- `FT-991A` → Yaesu FT-991A
- `TS-590S` → Kenwood TS-590S / 590SG
- `IC-7300` → Icom IC-7300
- `K3` → Elecraft K3 / K3S

2. **Port** – auto-selection of available ports:

- Windows: `COM3`, `COM4`, etc.
- Linux: `/dev/ttyUSB0`, `/dev/ttyACM0`
- macOS: `/dev/cu.usbserial-*`, `/dev/cu.SLAB_USBtoUART`

3. **Baud rate** – fundamental, must match the radio menu:

- Kenwood TS-590S: **57600** (recommended, max throughput) or 9600
- Yaesu FT-991A: **38400** or 9600
- Icom IC-7300: **19200** or 9600

4. **Stop bits** – usually 1, but for Yaesu FT-991A you might need 2

5. **Handshake** – None (default), Hardware (rare)

6. **CI-V Address** (Icom only) – `0x94` for IC-7300, `0xA4` for IC-7610

## 3.2.2 RTS / DTR

The RTS and DTR serial control lines are often used for hardware PTT. The DECODIUM 4.0 semantics are:

VALUE	BEHAVIOR
<b>Empty</b> (none)	DECODIUM doesn't touch the line (safe default)
<b>ON</b>	Line forced high (3.3V or 5V on USB-serial)
<b>OFF</b>	Line forced low (0V)

**Important:** If the radio is set for RTS-based PTT, leave RTS = ON and manage PTT via Setup → PTT (see section 3.4). Don't confuse the two.

## 3.2.3 CAT Test

After configuring all parameters, press **Test CAT**. A successful connection returns:

```
✓ Rig: Kenwood TS-590S
✓ Frequency: 7.074000 MHz
✓ Mode: USB
✓ Power level: read OK
```

At the bottom, the **CAT: Disconnected** (yellow) indicator turns to **CAT: Connected** (green).

## 3.2.4 CAT troubleshooting

SYMPTOM	CAUSE	SOLUTION
Cannot open port	Port in use or missing permissions	Close other software (WSJT-X, JS8Call); on Linux check <code>dialog</code> group
Timeout reading	Wrong baud rate	Compare radio menu with the configured value
Mode toggles to USB on TX	Old driver	Update USB-serial driver (FTDI, Silicon Labs CP210x, Prolific)
Connected but freq 0.000	Wrong CI-V address (Icom)	Verify the radio's Set menu

## 3.2.5 HRD bridge (advanced)

To use DECODIUM with HRD as central hub, enable **Setup** → **Radio** → **HRD bridge**:

1. Start HRD normally, connect to the radio
2. In HRD, enable the TCP server (Tools → TCP Server)
3. In DECODIUM, enter `127.0.0.1:7809` (or the address configured in HRD)
4. **Test HRD** – when you see `✓ HRD v6.x bridged`, you're set

**HRD/Icom data-mode stability (v1.0.261):** The new v1.0.261 has definitively fixed the issue of USB fallback during PTT settling. DATA mode is preserved across all TX/RX transitions and band changes. If you're coming from v1.0.260 or earlier, the upgrade is strongly recommended.

## 3.3 Audio configuration

Audio I/O is the backbone of any digital mode. Wrong configuration = no decodes.

### 3.3.1 Audio hardware

DECODIUM supports three topologies:

- 1. USB Audio CODEC integrated in the radio** (recommended)
  - Kenwood TS-590S/SG, Yaesu FT-991A, Icom IC-7300/7610 – single USB interface for audio + CAT
- 2. External interface** (e.g. SignalLink USB, Tigertronics, RigBlaster)
  - Audio separate from CAT, more cables but greater flexibility
- 3. PC sound card + direct cables** (legacy, not recommended)
  - Audio line from radio to PC sound card, manual level management

### 3.3.2 Device selection

Open **Setup** → **Audio**:

FIELD	TYPICAL VALUE
<b>Audio IN</b>	<code>USB Audio CODEC</code> (input from radio to PC)
<b>Audio OUT</b>	<code>USB Audio CODEC</code> (output from PC to radio)
<b>Sample rate</b>	48000 Hz (default) or 12000 Hz (legacy WSJT-X)
<b>Buffer size</b>	1024 frames (default) – reduce to 512 for lower latency

On Windows, **disable driver audio effects**:

- Control Panel → Sound → USB Audio CODEC → Properties → Advanced
- Disable “Audio Enhancements”, “Dolby Effects”, etc.

### 3.3.3 Level calibration

Audio levels are **critical**. Once CAT is connected and the radio is receiving on an active band:

1. Open the **VU meter** at bottom-left ( **S:** indicator)
2. The bar should swing in the green zone (-30 dB to -10 dB)
3. **Yellow zone (-10 to -5 dB)** = too high, possible clipping
4. **Red zone (>-5 dB)** = overload, decoder will fail

To adjust the IN level:

- **On the radio:** USB IN level menu (Kenwood Menu 64, Yaesu Menu 113, Icom USB AF Output Level)
- **On the PC:** audio control panel → USB CODEC microphone level

To adjust the OUT level (for transmission):

- **On the radio:** USB OUT level menu (Kenwood Menu 65, Yaesu Menu 114, Icom USB MOD Level)
- **On the PC:** audio preferences → USB CODEC speaker volume

**Tip Kenwood TS-590S:** The optimal setup requires three menus in coordination: - Menu **63** = DATA IN – set to **USB** - Menu **64** = USB IN level – start at **5**, increase until green zone - Menu **65** = USB OUT level – start at **5**, adjust to obtain desired PWR

## 3.4 PTT (Push To Talk)

PTT is the signal telling the radio “transmit now”. DECODIUM 4.0 supports five PTT modes:

### 3.4.1 Available PTT modes

MODE	WHEN TO USE	RELIABILITY
<b>CAT</b>	Modern radios with USB Audio CODEC	★★★★★ (recommended)
<b>VOX</b>	Radios without CAT or minimal setup	★★★ (may cut off TX start)
<b>RTS</b>	Serial cable with RTS line driving a PTT transistor	★★★★ (legacy but reliable)
<b>DTR</b>	Like RTS but on DTR line	★★★★
<b>External</b>	External hardware (pedal, USB footswitch)	★★★

### 3.4.2 PTT CAT setup (recommended)

In **Setup** → **PTT**: 1. **PTT method** – **CAT** 2. **PTT port** – same value as CAT (e.g. **COM3**) 3. Save

PTT travels on the same CAT channel. No additional cable or configuration needed.

### 3.4.3 PTT RTS or DTR setup

For legacy radios or external setups: 1. **PTT method** – **RTS** or **DTR** depending on cable 2. **PTT port** – can be same as CAT or a different serial port (if you have a dedicated PTT cable) 3. **Test PTT** – press the button; the radio must switch to TX (red LED lights)

### 3.4.4 PTT test with TUNE button

At any time, the **TUNE** button in the toolbar generates a 1500 Hz test tone and activates PTT. Use it to:

- Verify PTT actually fires

- Tune the antenna (SWR)
- Adjust output PWR level

**Important:** TUNE generates a continuous signal for 5-10 seconds. Don't leave it on for long to avoid stressing the radio's final stage. Use  $SWR \leq 1.5$  for prolonged TUNE.

## 3.5 Time synchronization

FT8, FT4, synchronous FT2 and other modes require the PC to be **UTC-synchronized** within **±1 second**.

### 3.5.1 TIME SYNC indicator

At top, next to the frequency display, there's the **TIME SYNC** line with current UTC time:

- **White** → synchronized, all OK
- **Yellow** → drift within 500 ms, tolerated
- **Red** → drift > 1 second, decoding compromised

### 3.5.2 Configuring NTP

#### Windows 10/11:

```
Control Panel → Date and Time → Internet Time → Change settings
✓ Synchronize with Internet time server
Server: time.windows.com (default) or pool.ntp.org
```

#### macOS:

```
System Preferences → Date & Time → Set automatically
Server: time.apple.com
```

#### Linux:

```
sudo systemctl enable --now systemd-timesyncd
timedatectl status # verify
```

### 3.5.3 ASYMX – no NTP needed

If you operate only in **FT2 ASYMX**, time sync is not required. ASYMX is designed precisely for scenarios where NTP/GPS are not available (emergency, maritime, portable). See Chapter 5.

---

## Chapter 4 – Operating in FT8

---

FT8 is the most widespread digital mode in the world. DECODIUM 4.0 offers full WSJT-X-compatible support, with the Raptor decoding engine improving sensitivity.

### 4.1 Typical workflow

An FT8 QSO in DECODIUM follows the classic flow:

1. **Select the band** (bottom toolbar: 160, 80, 60, 40, 30, 20, 17, 15, 12, 10, 6, 4, 2, 70cm)
2. **DECODIUM auto-tunes** the radio to the standard FT8 frequency (e.g. 7.074 MHz for 40m)
3. **Passive decode** – you see signals in the Full Spectrum
4. **Click a station calling CQ** – DECODIUM prepares TX messages
5. **Start the QSO** by pressing TX (or let auto-sequencing run)
6. **Auto-log** at end of QSO

## 4.2 Decode windows

### 4.2.1 Full Spectrum (left)

Shows **all decodes** across the entire audio band (50 Hz - 3000 Hz).

Columns:

COLUMN	MEANING
<b>UTC</b>	Slot start timestamp (e.g. <code>00:04:30</code> )
<b>dB</b>	Received SNR (weaker signals = more negative number)
<b>DT</b>	Time delta vs ideal slot ( $\pm 0.5$ s typical)
<b>Freq</b>	Audio frequency in Hz (1395 Hz, etc.)
<b>Message</b>	Decoded content ( <code>CQ N2SS FM29</code> )
<b>Dist</b>	Distance from your locator
<b>DXCC</b>	Decoded geographical entity
<b>Az</b>	Azimuth in degrees

### 4.2.2 Signal RX (right)

Shows **only decodes on the selected frequency** (shown at top: `1395 Hz`). Useful for following an ongoing QSO without distractions.

### 4.2.3 Live Map

World map with yellow dots for each recent decode. Green lines indicate active paths:

- `IN→ME` = stations you receive
- `ME→DX` = if you're spotted on PSK Reporter
- `BAND` = yellow dots without lines = raw decodes

Click a dot = filter Full Spectrum on that station.

## 4.3 Transmitting

### 4.3.1 Replying to a CQ

1. In Full Spectrum, **double-click** a line starting with `CQ`
2. DECODIUM populates the messages automatically:
  - TX1: `<DX> IU8LMC JN71`
  - TX2: `<DX> IU8LMC -10`
  - TX3: `<DX> IU8LMC R-10`
  - TX4: `<DX> IU8LMC RR73`
  - TX5: `73 to <DX>`
3. Press **TX** at bottom (or let auto-sequencing run)
4. Radio switches to TX, transmits for 13 seconds, returns to RX
5. If DX replies, autoSeq flow proceeds automatically through to log

### 4.3.2 Calling CQ

1. Press **TX6** at bottom (preset to `CQ IU8LMC JN71` )
2. DECODIUM alternates RX and TX automatically
3. When another station replies, you'll see their message in Signal RX
4. The software proceeds with autoSeq flow: receive report → send report → receive RR73 → send 73 → log

### 4.3.3 Auto-sequencing (autoSeq)

The **SEQ** button in the toolbar enables full QSO automation. When SEQ is green:

- DECODIUM auto-advances from TX1 → TX2 → TX3 → TX4 when it detects coherent replies
- Logs at end of QSO without intervention

- Stops transmission if no reply in 3 consecutive slots

### 4.3.4 Hold mode

The **HOLD** button (orange when active) blocks automatic changes to TX messages. Useful when:

- You're working a difficult station and don't want the software to change callsigns based on spurious decodes
- You want to maintain QSO with the specific DX even if another station calls

## 4.4 Logging

### 4.4.1 ADIF log

Every completed QSO is automatically logged to `~/.decodium/log.adi` (or `%APPDATA%\Decodium\log.adi` on Windows). The format is ADIF 3.1.7 standard, compatible with any logbook software.

### 4.4.2 Automatic uploads

In **Setup** → **Logbook** you can enable automatic upload to:

- **LoTW (Logbook of The World)** – requires TQSL configured
- **eQSL** – username/password
- **Club Log** – username/password
- **QRZ.com** – API key
- **Cloudlog / Wavelog** – endpoint URL + API key

**FT2 submode and LoTW:** ARRL confirmed that LoTW support for SUBMODE=FT2 requires updating the TQSL configuration file. In the meantime, upload uses SUBMODE=DATA as a workaround. The definitive fix is in implementation on the ARRL side.

## 4.4.3 Export and backup

From **Logbook** → **Export**:

- **Complete ADIF** – entire logbook
- **ADIF by band** – split per band
- **ADIF by mode** – split per mode
- **CSV** – for spreadsheet import

# 4.5 Macros and MAM (Memory Auto Macro)

## 4.5.1 Classic macros

The TX1-TX6 buttons can be customized. Right-click on a TX → **Edit macro**:

```
%CALL% – current DX callsign  
%MYCALL% – your callsign  
%MYGRID% – your locator  
%RPT% – received SNR report  
%SENT% – report you sent
```

Custom macro example: `%CALL% %MYCALL% TU 73 DE %MYGRID%`

## 4.5.2 MAM – Memory Auto Macro

MAM is an advanced feature that stores recurring QSO sequences. Typical application: contest or activations where you repeat the same flow dozens of times.

Open **MAM Window** from the hamburger menu. Create a compound macro: 1. **Trigger** – activation condition (e.g. “after QSO log”) 2. **Sequence** – series of actions (e.g. “change band, call CQ”) 3. **Stop** – stop condition (e.g. “after 10 QSOs” or “manual”)

The feature is detailed in Chapter 7 (Advanced Features).

---

# Chapter 5 – Operating in FT2

---

FT2 (Fast Transmission 2) is the original digital protocol developed by the DECODIUM team and certified ADIF 3.1.7 with a unanimous 22:0 vote on March 22, 2026.

## 5.1 What FT2 is

FT2 is an HF digital protocol designed to be **3-4 times faster than FT8** while maintaining comparable sensitivity. Technical specs:

PARAMETER	VALUE
Modulation	4-GFSK (4-tone Gaussian Frequency Shift Keying)
Bandwidth	167 Hz
Tone spacing	41.667 Hz
Symbol rate	41.667 baud
T/R cycle	3.75 – 3.8 seconds (vs FT8's 15s)
FEC	LDPC (174, 91) + CRC-14
Sync sequence	16-tone Costas array
Payload	77 useful bits per message
Complete QSO	~6 seconds (vs FT8's 60+)

### 5.1.1 Standard FT2 frequencies

Dedicated sub-band within each band's digital segment:

BAND	FT2 FREQUENCY
80m	3.580 MHz
40m	7.080 MHz
30m	10.137 MHz
20m	14.080 MHz
17m	18.105 MHz
15m	21.080 MHz
12m	24.920 MHz
10m	28.180 MHz
6m	50.318 MHz

DECODIUM auto-tunes these frequencies when you select the band in FT2 mode.

## 5.2 Enabling FT2

### 5.2.1 Switching to FT2 from FT8

1. On the mode toolbar (bottom-left), click the **FT8** button which opens the menu
2. Select **FT2**
3. DECODIUM automatically changes:
  - Radio frequency to FT2 sub-band
  - Decoder to dedicated FT2 engine
  - T/R cycle to 3.8 seconds
  - Interface with FT2-specific elements (AP, QQ buttons visible)

## 5.2.2 FT2 mode buttons

In the toolbar new controls will appear, specific to FT2:

BUTTON	FUNCTION
<b>DEEP</b>	Activates Raptor Engine (5 passes) – –23 dB SNR sensitivity
<b>AP</b>	Async Period – ASYMX mode (asynchronous transmission)
<b>QO</b>	Quick QSO – 4-message flow
<b>SWL</b>	Listen-only mode, no TX
<b>ACQ</b>	Acquisition – force sync on next slot

## 5.3 Raptor Engine (DEEP)

The **DEEP** button activates the multi-pass decoding engine. When DEEP is green:

1. The decoder runs **5 independent passes** on the same RX slot
2. Each pass tries an optimal sync with different methods (classical correlation, MMSE, EMA)
3. Results are combined with a **best-of** strategy
4. The final decode is the one with valid CRC and best SNR

The gain is significant:

- Standard decoder: –16 dB SNR floor
- DEEP(Raptor 5-pass): **–23 dB SNR floor** (+7 dB of sensitivity)

**Computational trade-off:** DEEP uses more CPU. On modern PCs (quad-core 2.5+ GHz) it's negligible. On Raspberry Pi 4 it might slow the interface – disable DEEP if you see lag.

## 5.4 ASYMX – Asynchronous mode

ASYMX is FT2's most radical innovation. It **eliminates Even/Odd sync** of classic FT8. Result: no NTP, no GPS, no "slot loss" if you change band at the wrong moment.

### 5.4.1 How to enable ASYMX

1. Band 40m, FT2 mode active
2. Press **AP** (Async Period) in toolbar – the button turns yellow
3. ASYMX is active. Transmission starts at first available TX without waiting for "even/odd slot"

### 5.4.2 Complete ASYMX sequence

T=0	You transmit CQ (2.47s waveform)
T=2.5	Waveform ends, radio returns to RX
T=2.8	You decode DX's reply (~300ms guard time)
T=3.1	You decode the reply (decoder <400ms)
T=3.5	Auto-fire TX2 (report)
T=6.0	You decode DX's TU
T=6.0	QSO complete, auto-log

**Total: ~6 seconds for complete QSO.**

### 5.4.3 When to use ASYMX

ASYMX is ideal for:

- **Maritime operations** – boats and sailing vessels without always-available GPS

- **Emergency/EmComm** – scenarios where time infrastructure is compromised
- **Portable/Field Day** – setup simplification, no PC sync required
- **Propagation experiments** – fast test sequences
- **High-rate contexts** – DXpedition with tight pile-ups

## 5.4.4 Compatibility

**Important:** The ASYMX signal is **identical to the standard FT2 signal** from the RF point of view. Anyone using WSJT-X or JTDX will receive your ASYMX signals perfectly. The difference is only in **TX timing**: you transmit freely, others receive normally.

## 5.5 Quick QSO (QQ)

QQ further optimizes the QSO flow, reducing from 5 to **4 messages**.

### 5.5.1 QQ vs standard flow

#### Standard FT2:

```
TX1: CQ MYCALL MYGRID
TX2: DXCALL MYCALL +05
TX3: DXCALL MYCALL R+03
TX4: DXCALL MYCALL RR73
TX5: 73 TO DXCALL
```

#### QQ FT2:

```
TX1: CQ MYCALL MYGRID
TX2: DXCALL MYCALL R+05    ← report sent already confirmed
TX3: DXCALL MYCALL RR73
TX4: TU MYCALL             ← closes the QSO
```

Savings: **1 full cycle** (~3.8 seconds).

## 5.5.2 QQ activation

1. Press **QQ** in toolbar – button turns green
2. Next QSO will use the optimized flow
3. If the partner station doesn't support QQ, **automatic fallback** to standard cycle




## 5.5.3 TU Detection

DECODIUM automatically recognizes the **TU MYCALL** message as valid QSO end. No need to press anything – log is automatic on TU reception.

# 5.6 FT2 best practices

## 5.6.1 Tone spacing and SSB

FT2 has **167 Hz RF bandwidth**, against FT8's 50 Hz. This means:

-  **More robust to fading and Doppler** (wider band = fewer catastrophic fluctuations)
-  **Easier multi-decode** (tones more spaced, less overlap)
-  **Faster band saturation** – with 10 FT2 stations in 3 kHz SSB you already have a crowded market

## 5.6.2 Power and SWR

FT2 is medium-duty cycle (~50% TX). Tips:

- Typical power: **20-50 W** sufficient for most QSOs
- SWR < 1.5:1 recommended
- For QRP activations, 5 W enough to work DX on 20m with decent propagation

## 5.6.3 Synchronous vs ASYMX

SCENARIO	RECOMMENDED MODE
Home, NTP working, crowded band	Synchronous FT2 (compatibility with all)
Home, NTP broken/firewall blocking NTP	ASYMX (no dependency)
Field Day, portable	ASYMX (simplifies setup)
Maritime mobile	ASYMX (no GPS dependency)
Tight DXpedition pile-up	ASYMX + QQ (max throughput)

# Chapter 6 – Decoder and Sensitivity

This chapter is for those who want to understand **how** DECODIUM 4.0 achieves superior decoding performance. It's not required to use the software, but it's essential for those who want to optimize setup or contribute to development.

## 6.1 Decoder architecture

The DECODIUM 4.0 decoder is structured as a **4-stage pipeline**:

1. Audio pre-processing (FFT, AGC, RMS norm)  
↓
2. Sync acquisition (Costas correlation, MMSE channel estimation)  
↓
3. LDPC decoding (Min-Sum normalized, 5-pass best-of)  
↓
4. Post-processing (CRC validation, EMA averaging, message extraction)

Each stage is **isolated** in separate C++ modules. The principle: if a new algorithm proves superior in one stage, it can be replaced without touching the others.

## 6.1.1 Audio pre-processing

Incoming audio is transformed into three parallel representations:

REPRESENTATION	USE
<b>Time-domain raw</b> (48 kHz, float32)	Diagnostics and logging
<b>FFT (2048-point Hann)</b>	Sync acquisition and waterfall display
<b>Symbol-rate filtered (41.667 Hz for FT2)</b>	Decoder input

**RMS normalization** operates on three simulated audio channels to make the decoder robust to level variations: if the RX signal drops suddenly, the decoder doesn't lose sync.

**Important:** RMS norm **is not radio-side AGC**. It's an internal decoder normalization. On the radio you can (and should) leave AGC as you prefer for listening.

## 6.1.2 Sync acquisition

DECODIUM searches for the **16-tone Costas** pattern at the start of every slot. Three parallel strategies:

1. **Classical correlation** – WSJT-X standard method. Fast but fading-sensitive.
2. **MMSE adaptive estimation** – uses all 16 tones as pilots to estimate the channel and correct the signal. **+1.5 dB** on fading channels.
3. **EMA-weighted multi-period** – for stations repeating the call, accumulates information across consecutive slots.

The **Best-of-N** selector picks the sync with the best **correlation score**. This avoids locking onto interfering signals on a false-positive sync.

## 6.1.3 LDPC decoding

LDPC (Low-Density Parity-Check) is the error correction code used in FT8, FT4, FT2. DECODIUM uses original **K1JT** (Joe Taylor) mathematics but with two optimizations:

1. **Normalized Min-Sum** – replaces classical Sum-Product. Reduced complexity per iteration (~30% faster) with equivalent performance on AWGN channels.
2. **Multi-pass best-of** – 5 independent attempts with different syncs. The final CRC-14 confirms validity. Only decodes passing CRC reach the log.

**Where the LDPC code lives:** isolated in

`lib/ft2/decode174_91_ft2.f90` – the only Fortran file remaining in DECODIUM 4.0. The choice is deliberate: K1JT's math is perfect, rewriting it in C++ would add risk with no measurable benefit.

## 6.1.4 Post-processing

After decoding, every message passes through:

1. **CRC-14 validation** – if CRC fails, decode discarded (no log, no display)

2. **EMA averaging** – for stations seen in previous slots, applies weighted average to stabilize SNR/DT
3. **Callsign hash extraction** – the 77-bit payload is decompressed into callsign + grid + text message
4. **Best-of-N consolidation** – if multiple passes found the same message, the decode with the best SNR is kept

## 6.2 Sensitivity table

DECODIUM Lab internal measurements, AWGN channel, 12 kHz sample rate:

MODE	STANDARD FLOOR	DEEP (RAPTOR) FLOOR	GAIN
FT8	-21 dB	-23 dB	+2 dB
FT4	-17 dB	-19 dB	+2 dB
FT2	-16 dB	<b>-23 dB</b>	<b>+7 dB</b>
FT2 + EMA 4-period	n/a	<b>-14.5 dB</b>	(on dynamic SNR)

**Fair comparison vs WSJT-X:** WSJT-X reports -21 dB floor on FT8 and ~-17 dB on FT4. DECODIUM 4.0 without DEEP is equivalent. With DEEP enabled the gain is real but costs CPU (~25-30% more per slot). On modern PCs it's negligible.

## 6.3 Optional filters

In **Setup** → **Filters** are available post-decode filters that reduce false positives at the cost of sensitivity.

## 6.3.1 FDR (Frequency Domain Resilience)

Filter rejecting decodes with **inconsistent SNR across passes**. If pass 1 says `-18 dB` and pass 5 says `-8 dB`, it's probably a QRM artifact.

STATE	BEHAVIOR
<b>FDR ON</b> (default)	Filters inconsistencies, loses ~5% of borderline decodes
<b>FDR OFF</b>	Shows everything, accepts some false positives

**When to disable FDR:** if you're convinced some valid decodes are being lost. Typical case: QRP stations at marginal propagation where every decode matters.

## 6.3.2 Spectral Mask

Filter rejecting decodes **outside the expected spectral mask**. Useful in crowded bands where adjacent signals can induce false syncs.

## 6.3.3 Sliding Window AGC

Internal AGC applied to the audio buffer before FFT. Not to be confused with radio AGC. Improves dynamic range on signals with highly variable levels within the slot.

# 6.4 Bandwidth and selectivity


DECODIUM works by default on **3 kHz of audio bandwidth** (50 Hz - 3000 Hz). You can modify in **Setup** → **Audio** → **Bandwidth**:

BAND	USE
<b>3 kHz</b> (default)	Standard, covers all digital sub-bands
<b>2.7 kHz</b>	For radios with narrow SSB filter
<b>2 kHz</b>	Only the selected FT8/FT2 signal, reduces QRM

**Caution:** reducing bandwidth **also reduces the number of stations simultaneously decodable**. Use only if you have specific out-of-band QRM to exclude.

## Chapter 7 – Advanced Features

### 7.1 CALL – Direct call (v1.0.262)

The  **CALL** button is the v1.0.262 novelty. It's located in the TX toolbar **next to ACQ**, and allows calling a specific callsign **without waiting for their CQ**.

#### 7.1.1 When CALL is useful

CALL solves three concrete scenarios:

- 1. Skeds with friends** – agree on a time, they don't CQ, you call them directly
- 2. Fast replies to missed CQs** – DECODIUM decoded a CQ but you clicked 100ms too late
- 3. Targeted DXpedition activations** – call repeatedly to a rare station without waiting for their CQ

## 7.1.2 CALL workflow step-by-step


1. Click  **CALL** – the `CallDialog` opens
2. Enter the target callsign – e.g. `F4CQS`, `JA1XYZ`, `VK7AB`
3. Configure parameters:


PARAMETER	MEANING	DEFAULT
<b>Max retries</b>	Maximum attempts before automatic stop	10
<b>Timeout</b>	Total seconds before abort	60s
<b>Period</b>	Call frequency (1 = every slot, 2 = every 2 slots, etc.)	1
<b>Message format</b>	TX1 message template	<code>TARGET MYCALL</code> <code>MYGRID</code>

4. Press **▶ Start**
5. Decodium starts calling at the next available slot: `F4CQS IU8LMC`  
`JN71`
6. The CALL button in the toolbar turns **green** with tooltip `retry N/M`
7. **Edge-detector on** `transmittingChanged` counts actual retries (not attempts failed by CAT errors)
8. **Automatic stop** if:
  - Max retries reached
  - Timeout expired
  - The target replies (automatic handoff)
9. If the target replies, **natural handoff** to existing autoSeq flow: R-XX  
→ RR73 → log

## 7.1.3 Operating conventions and etiquette

CALL is a powerful tool. Misusing it can create QRM. Guidelines:

 **Don't use CALL:** - To **call stations clearly in QSO**. Wait for their 73. - To **call DXpedition in tight pile-up without coordination**. You just create QRM to the existing pile-up. - **Continuously in series** toward 10 different stations. Useless split is chaos.

 **Use CALL:** - When you have a **confirmed sked** and the time has come - For **fast replies to CQs** decoded but missed by a few ms - When you want to **complete an interrupted QSO** with a previous station (calls + grid already exchanged) - In **EmComm** where calling directly is standard practice

## 7.1.4 CALL ↔ AutoSeq ↔ Hold interaction

CURRENT STATE	CALL BEHAVIOR
AutoSeq OFF, Hold OFF	CALL handles everything, starts and stops on its own
AutoSeq ON, Hold OFF	If target replies, AutoSeq takes control. CALL ends.
AutoSeq OFF, Hold ON	CALL runs but doesn't advance messages. You control the flow.
AutoSeq ON, Hold ON	CALL stops immediately – Hold blocks autoSeq.

## 7.2 Call Roster

The Call Roster is a **dedicated window** that automatically tracks callsigns heard during the session.

### 7.2.1 Activate the Call Roster

Hamburger menu (☰) → **Tools** → **Call Roster**. Or shortcut **F7**.

The window shows three tabs:

- **Heard** – all stations decoded in the current session
- **Calling** – stations currently calling CQ
- **Roster** – your personal “watch” callsign list (see 7.2.3)

## 7.2.2 Roster columns

COLUMN	MEANING
<b>Call</b>	Decoded callsign
<b>Grid</b>	Locator extracted from payload
<b>DXCC</b>	Automatically decoded entity
<b>Band</b>	Band where heard
<b>Mode</b>	FT8, FT2, FT4
<b>SNR</b>	Best SNR in the session
<b>First</b>	UTC of first decode
<b>Last</b>	UTC of last decode (useful for "still active?")
<b>Count</b>	How many times heard (frequency)
<b>Status</b>	★ if on your roster · ✓ if already worked (logbook check)

## 7.2.3 Personal roster

Right-click on a station → **Add to roster**. The personal roster is a list of "interesting stations" you want **not to miss**.

DECODIUM emits **audio/visual alert** when a station from your roster decodes. Configurable in **Setup** → **Roster** → **Alerts**.

**Tip:** Use the roster for Most Wanted DXCC, contest target list, recurring sked partners.

## 7.3 DX Cluster

### 7.3.1 Cluster connection

In **Setup** → **DX Cluster**:

FIELD	EXAMPLE
<b>Server</b>	<code>dxc.k0xm.net:7300</code>
<b>Login</b>	your callsign
<b>Password</b>	(usually empty, some clusters require registration)
<b>Auto-connect on startup</b>	✓ recommended

### 7.3.2 Cluster filters

DECODIUM applies filters to avoid being overwhelmed:

- **By band** – only active band or multi-band
- **By mode** – FT8, FT2, FT4, CW, SSB, etc.
- **By DXCC** – only entities of interest
- **By distance** – exclude local clusters (e.g. > 1000 km)
- **By age** – only recent spots (e.g. last 5 min)

### 7.3.3 Spotting

You can **spot stations** you decode: right-click on a decode → **Spot to cluster**. DECODIUM sends a spot to the connected cluster.

## 7.4 PSK Reporter

PSK Reporter is the worldwide “who hears whom” database. DECODIUM automatically uploads all valid decodes.

### 7.4.1 Enabling

In **Setup** → **PSK Reporter**:

- **Enable upload** – ✓ recommended
- **Upload interval** – 5 minutes (default)
- **Include FT2 decodes** – ✓ (requires updated PSK Reporter plugin)

### 7.4.2 Upload verification

In [Help → PSK Reporter status](#) you can see:

- Last upload performed
- Number of decodes uploaded in the session
- Any HTTP errors

**Privacy:** PSK Reporter publicly exposes your callsign + grid + station heard. If you don't want this, disable upload. It's entirely optional.

## 7.5 Astro Tools

Astro is the propagation tools suite integrated in DECODIUM.

### 7.5.1 Solar dashboard

In toolbar → **Astro**. The dashboard opens with:

- **Solar Flux Index (SFI)** – HF propagation indicator
- **K-index and A-index** – geomagnetic disturbance

- **Sunspot Number** – solar spots (SFI proxy)
- **MUF/LUF map** – Maximum/Lowest Usable Frequency in real time

## 7.5.2 Greyline tracker

DECODIUM displays the **terminator line (greyline)** on Live Map. Stations near the line are **propagatively favored** for DX on HF.

## 7.5.3 Auroral oval

When K-index > 4, DECODIUM highlights the auroral oval area on Live Map (high polar absorption). Useful for those seeking EME / VHF contacts.

# 7.6 In-depth Live Map

The Live Map isn't just decorative – it has operational features:

## 7.6.1 Map filters

Buttons above the map:

- **IN → ME** – only stations receiving you (requires PSK Reporter)
- **ME → DX** – only stations you decode
- **BAND** – only active band
- **All** – everything

## 7.6.2 Click on a dot

Single click on a station → filters Full Spectrum on that callsign. Double click → opens QRZ.com (configurable in Setup → Lookup) for details.

## 7.6.3 Counters

Bottom-right of the map:

- **Active paths** – number of active paths this slot

- **In band** – stations decoded on current band
- **UTC clock** – UTC time synced with system

## Chapter 8 – UI Customization

DECODIUM 4.0 is highly customizable. This flexibility is designed to accommodate different operating styles: from the contester wanting everything at a glance, to the home operator preferring a relaxing interface.

### 8.1 Themes

In **Setup** → **Interface** → **Theme**:

THEME	STYLE	WHEN TO USE
<b>Shannon Dark</b> (default)	Background #050816 , green accents #00e588	Night operations, dimly lit environments
<b>Shannon Light</b>	White background, dark green accents	Day operations, high ambient brightness
<b>Midnight</b>	Pure black background, cobalt blue accents	Maximum contrast, OLED-friendly
<b>Classic</b>	WSJT-X theme reproduction	For those not wanting to be disoriented

The theme changes immediately without restart.

## 8.2 Waterfall palette

In top toolbar → **Palette**. Four options:

PALETTE	CHARACTERISTICS
<b>SDR Classic</b> (default)	Blue → green → yellow, identical to WSJT-X
<b>Shannon Light</b>	White → amber → red, high brightness
<b>Shannon Dark</b>	Black → green → white, maximum contrast on weak signals
<b>Heat</b>	Red → yellow → white heatmap, vintage taste

**Tip:** To search for weak signals, use Shannon Dark. To quickly identify strong stations in a pile-up, use Heat.

## 8.3 Font scaling

In top toolbar → **Aa 100%**. Three sliders:

- **A-** reduces global font size
- **100%** restores default
- **A+** increases global font size

Range: 80% → 150%. Useful for operators with visual issues or for public demonstrations.

## 8.4 Compact mode

For multi-monitor setups or small screens, **Setup** → **Interface** → **Compact mode** enables:

- Reduced internal padding
- Toolbar collapse (icons only)
- Hides redundant labels
- Maintains accessibility via tooltip

## 8.5 Pop-out windows

All main windows can be **detached from the main window**:

- **Live Map** → **Pop** button top-right
- **Full Spectrum** → **Pop** top-right
- **Signal RX** → **Pop** top-right
- **Call Roster** → Tools menu → Pop-out

Pop-out windows **maintain live decoder connection**. Useful for:

- Multi-monitor setup where you want the map on monitor 2
- Streaming/video where the map is a separate scene
- “Only watch Signal RX and hide everything else” operations

## 8.6 Multi-instance (MultiRig CLI)

DECODIUM 4.0 supports **parallel multiple instances** for stations with multiple radios. Launch via CLI:

```
# Instance 1 - main radio
decodium --instance 1 --rig-port COM3 --config=primary.ini

# Instance 2 - secondary radio
decodium --instance 2 --rig-port COM4 --
config secondary.ini
```

Each instance has:

- Separate configuration (distinct `.ini`)
- Separate ADIF log
- Independent windows
- Separate audio device

**Important:** The two instances **must not share** the same sound card. Use two separate USB CODEC interfaces.

## 8.7 Interface languages

In **Setup** → **Interface** → **Language**:

-  **English** (default)
-  **Italiano**
-  **Español**
-  **Deutsch**
-  **Türkçe**

Language change is **immediate** without restart. Translated: all menus, dialogs, tooltips. **Not translated** (because technical): parameter names (SNR, DT, Freq), protocol terms (CQ, RR73, 73), and standard ham mode abbreviations.

## 8.7.1 Contributing translations

Translations are Qt Linguist `.ts` files. To contribute: 1. Fork the GitHub repository 2. Modify `translations/decodium_xx.ts` with Qt Linguist 3. Pull request

Translators are **credited** in the program's About page.

---

# Chapter 9 – Troubleshooting

---

This chapter collects real issues reported by the Telegram community during Public Beta. For each: symptom, probable cause, tested solution.

## 9.1 CAT problems

### 9.1.1 “Cannot open port”

**Symptom:** at startup or pressing Test CAT, error `Cannot open port COM3` (or Linux/macOS equivalent).

**Possible causes:** 1. Another program is using the same port (WSJT-X, JS8Call, FIDigi) 2. Missing permissions (Linux: not in `dialout` group) 3. USB-serial driver not installed or wrong 4. Defective or unplugged cable

**Solutions:**

```
Step 1: Close all other ham software and retry
Step 2: Verify the port exists
- Windows: Device Manager → Ports (COM and LPT)
- Linux: ls -l /dev/tty* | grep USB
- macOS: ls /dev/cu.*
Step 3: On Linux, add user to dialout group:
sudo usermod -aG dialout $USER
(then logout/login)
Step 4: Unplug and replug the USB cable
Step 5: Update USB-serial driver (FTDI, Silicon Labs CP210x,
Prolific)
```

## 9.1.2 “Timeout reading from rig”

**Symptom:** CAT connects but then errors with `Timeout`.

**Probable cause:** wrong baud rate (radio says X, software says Y).

**Solution:** 1. Go to the radio menus and read the configured baud rate 2. In DECODIUM, Setup → Radio → Baud rate, set the **same value** 3. Press Test CAT

**Common values:** - Kenwood TS-590: 57600 (recommended) - Yaesu FT-991A: 38400 - Icom IC-7300: 19200

## 9.1.3 “Mode unexpectedly toggles to USB”

**Symptom:** you’re operating in DATA-U but on every TX the radio switches to USB.

**Cause:** known bug fixed in v1.0.261. If on previous version, upgrade.

**Solution:** download v1.0.262 (or at least v1.0.261). The fix ensures DATA-U/D1 preserved across TX/RX transitions.

## 9.1.4 HRD doesn’t reconnect after restart

**Symptom:** close DECODIUM and HRD, reopen DECODIUM, HRD not detected.

**Cause:** HRD must be started and connected BEFORE DECODIUM retries the bridge.

**Solution (v1.0.261+):** the new “last-successful CAT retry path” logic automatically attempts HRD at startup. If still failing: 1. Start HRD manually 2. In HRD, verify TCP server active (Tools → TCP Server → Status) 3. In DECODIUM, click Test HRD to force reconnection

## 9.2 Audio problems

### 9.2.1 No decodes visible

**Symptom:** everything seems OK, but no decodes appear for minutes.

**Sequential diagnosis:**

Check 1: Correct band?

Standard FT8 bands:

40m = 7.074 MHz

20m = 14.074 MHz

15m = 21.074 MHz

10m = 28.074 MHz

If radio is on 7.080 (FT2 band), no FT8 there.

Check 2: Audio IN VU meter moving?

Open `S:` indicator at bottom-left.

Bar still → audio IN problem

Bar in red zone → audio too high (clipping)

Bar in green zone → audio OK, problem elsewhere

Check 3: TIME SYNC white or red?

If red, sync NTP. Without sync, no FT8 decoding.

Check 4: Coherent mode?

Radio display says USB-DATA? Top-left DECODIUM says FT8?

If radio in CW or LSB, no digital audio decoding.

Check 5: Filters active?

Setup → Filters: aggressive FDR? Disable and retry.

## 9.2.2 Audio too loud (clipping)

**Symptom:** VU meter in red zone, sporadic decodes.

**Solution:** 1. **On the radio:** lower USB IN level (Kenwood Menu 64, Yaesu Menu 113, Icom USB AF Output Level) 2. **On the PC:** Audio control panel → USB CODEC microphone → reduce level 3. Target: VU in green zone, peaks not above -10 dB

## 9.2.3 PTT doesn't fire

**Symptom:** you press TX or TUNE but the radio doesn't go into transmit.

**Diagnosis:**

Check 1: Coherent PTT method?

Setup → PTT → method. If CAT, must match CAT port.

If RTS/DTR, you must have a PTT cable connected to the radio side.

Check 2: Test PTT works?

Setup → PTT → Test PTT

If yes → problem in TX message or SEQ

If no → wiring/PTT configuration problem

Check 3: Correct RTS/DTR semantics?

Empty = DECODIUM doesn't touch the line

ON = forces high (line always active)

OFF = forces low (line always inactive)

If you have a PTT cable wanting "high = TX", set ON.

# 9.3 Live Map problems

## 9.3.1 Empty map

**Symptom:** Live Map shows no stations.

**Causes:** 1. Internet not connected → DECODIUM downloads map tiles from OpenStreetMap 2. Firewall blocking DECODIUM 3. Decodes still 0 →

Live Map shows only after valid decodes

**Solutions:** 1. Check internet connection (ping `tile.openstreetmap.org`) 2. Add DECODIUM to firewall exceptions (Windows Defender, ufw on Linux) 3. Wait at least 30 seconds after the first decode

### 9.3.2 Slow or “jerky” map

**Symptom:** map lags, zoom is jerky.

**Cause:** GPU rendering not accelerated.

**Solution (Windows):** 1. NVIDIA / AMD Control Panel → 3D settings 2. Add `decodium.exe` with “High performance” profile 3. Restart DECODIUM

**Solution (Linux):**

```
export QSG_RENDER_LOOP=threaded
./Decodium-1.0.262-x86_64.AppImage
```

## 9.4 Application startup problems

### 9.4.1 “QML import failed”

**Symptom:** at startup, error with failed QML import.

**Cause:** Qt 6.11 version not available on system (Linux), or corrupted QML cache.

**Solution:** 1. Delete QML cache: `~/.cache/Decodium/qmlcache/` 2. Restart

On Linux, if persistent, verify the AppImage is not on `noexec` filesystem:

```
mount | grep noexec # if current folder is noexec, move
the AppImage
```

## 9.4.2 Crash on first launch (macOS)

**Symptom:** click DECODIUM, window appears and disappears.

**Cause:** Gatekeeper blocking execution.

**Solution:**

```
xattr -cr /Applications/DECODIUM.app  
open /Applications/DECODIUM.app
```

## 9.5 Log files and debug mode

### 9.5.1 Where are the logs

OS	PATH
Windows	%APPDATA%\Decodium\logs\
macOS	~/Library/Application Support/Decodium/logs/
Linux	~/.local/share/Decodium/logs/

Three main files:

- `decodium.log` – general application log
- `cat.log` – CAT communications (only with CAT debug active)
- `decoder.log` – decoder log (rarely useful)

### 9.5.2 Enable debug mode

In **Setup** → **Advanced** → **Debug logging**, enable:

- **CAT debug** – logs every CAT command sent/received
- **Audio debug** – logs audio levels per slot
- **Decoder debug** – logs multi-pass details

**Important:** debug mode generates huge files (hundreds of MB per hour). Enable only for active troubleshooting, then disable.

### 9.5.3 Bug report

To report reproducible bugs:

1. Reproduce the bug with debug mode ON
2. Open **Help** → **Report bug**
3. Logs are automatically attached
4. Description: what you were doing, what happened, what you expected

Bug reports with logs → high priority in community tracking.

---

## Chapter 10 – Appendices

---

## A. Keyboard shortcuts

KEY	ACTION
<b>F1</b>	Help / About
<b>F5</b>	Refresh decode list
<b>F7</b>	Open Call Roster
<b>F8</b>	Open Live Map (pop-out)
<b>F9</b>	Open Full Spectrum (pop-out)
<b>Ctrl+T</b>	Toggle TX (TX on/off)
<b>Ctrl+H</b>	Toggle HOLD
<b>Ctrl+S</b>	Toggle SEQ(auto-sequence)
<b>Ctrl+L</b>	Log current QSO
<b>Ctrl+M</b>	Open Macros editor
<b>Ctrl+P</b>	Open Setup (Preferences)
<b>Ctrl+1..6</b>	Trigger TX1..TX6 manually
<b>Ctrl+Tab</b>	Switch between Full Spectrum / Signal RX
<b>Esc</b>	Immediate TX stop

## B. `decodium.ini` configuration file

The main configuration file resides in:

OS	PATH
Windows	<code>%APPDATA%\Decodium\decodium.ini</code>
macOS	<code>~/Library/Application Support/Decodium/decodium.ini</code>
Linux	<code>~/.config/Decodium/decodium.ini</code>

## B.1 Main sections

```
[Station]
MyCall=IU8LMC
MyGrid=JN71DC
Operator=
DXCC=auto

[Radio]
RigType=KenwoodTS590
RigPort=COM3
RigBaud=57600
StopBits=1
HRDBridge=false

[Audio]
AudioIn=USB Audio CODEC
AudioOut=USB Audio CODEC
SampleRate=48000
BufferSize=1024

[PTT]
Method=CAT
PortShared=true

[FT2]
EnableASYMX=false
EnableQQ=true
EnableDEEP=true

[Filters]
FDR=true
SpectralMask=true
SlidingAGC=false

[UI]
Theme=ShannonDark
WaterfallPalette=SDRClassic
FontScale=100
Language=en
```

## B.2 Manual edits

Edit `decodium.ini` with a text editor (DECODIUM must be **closed**).

**Automatic backup:** at every startup, DECODIUM creates `decodium.ini.bak`. If you modify and something goes wrong, restore from `.bak`.

## C. FT2 technical specifications

For those wanting to understand or implement the protocol:

### C.1 Modulation

- **Type:** 4-GFSK (4-tone Gaussian Frequency Shift Keying)
- **Nominal tone:** audio center (e.g. 1500 Hz)
- **Spacing between tones:** 41.667 Hz
- **Total RF bandwidth:** 167 Hz
- **Gauss BT filter:** 1.0 (inter-symbol smoothing)

### C.2 Timing

- **Symbol duration:** 24 ms (1/41.667 baud)
- **Total frame:** 79 symbols = ~1.896 s
- **Costas sync:** 16 symbols at edges (7+9)
- **Payload:** 174 raw symbols (= 87 effective symbols with FEC, 77-bit payload)
- **T/R cycle:** 3.75-3.8 s (guard time included)

### C.3 FEC

- **Code:** LDPC (174, 91) – rate ~0.52
- **Decode algorithm:** Normalized Min-Sum, 20 iterations max
- **CRC:** CRC-14 (polynomial 0x6757)

- **Correction capacity:** ~10 errored bits / 174

## C.4 Payload

- **Useful length:** 77 bits
- **Supported formats:**
  - Standard QSO (callsign1, callsign2, grid/report)
  - Telemetry (numeric sequence)
  - Free text (max 13 characters)
  - ARRL field day, DXpedition mode

## C.5 ASYMX

- **Specification:** 100% compatible extension with the FT2 standard physical layer
- **Implementation:** free TX without even/odd slot wait
- **Synchronization:** lives in the signal's Costas sequence, not in the station

# D. Credits and license

## D.1 Development

- **Martino Merola IU8LMC** – protocol designer, project lead
- **Salvatore Raccampo 9H1SR** – lead C++ developer, macOS/Linux builds
- **LU7DID** – Raspberry Pi port

## D.2 Technical heritage

- **Joe Taylor K1JT** – FT8/FT4/JT65/WSPR protocols (WSJT-X), LDPC mathematics
- **The Qt Company** – UI framework (Qt 6.11)
- **OpenStreetMap** – Live Map tile data

- **Hamlib team** – CAT library

## D.3 License


DECODIUM 4.0 is released under **GPL v3**. Complete source code:

- Main repository: <https://github.com/iu8lmc/Decodium-4.0-Core-Shannon>
- Salvatore mirror: <https://github.com/elisir80/Decodium-4.0-Core-Shannon>

## D.4 Standards

- **ADIF 3.1.7** – unanimous 22:0 vote of March 22, 2026 for SUBMODE=FT2
- **Graham G3ZOD** – ADIF secretary, certification mediator

## E. Changelog v1.0.0 → v1.0.262

VERSION	DATE	HIGHLIGHTS
<b>v1.0.262-call</b>	May 20, 2026	 CALL button (direct call)
<b>v1.0.261</b>	May 19, 2026	HRD/Icom data mode stability, QSO state hygiene
<b>v1.0.260</b>	May 18, 2026	UI state persistence, FT2 async stabilization
<b>v1.0.259</b>	May 17, 2026	QSO flow state machine, HRD data mode
<b>v1.0.258</b>	May 16, 2026	CAT and decode panel fixes
<b>v1.0.257</b>	May 15, 2026	macOS Apple Silicon build
<b>v1.0.250</b>	May 10, 2026	Public Beta entry point
<b>v1.0.200</b>	April 25, 2026	C++ rewrite complete, FT2 native
<b>v1.0.100</b>	March 22, 2026	ADIF 3.1.7 certification
<b>v1.0.000</b>	February 16, 2026	First FT2 QSO ever

**End of the Complete User Manual.** For advanced technical documentation (full API reference, complete troubleshooting, decoder internals, complete FT2 protocol specs), see the *Reference Manual*.

---

**73 de Martino IU8LMC & Salvatore 9H1SR** *DECODIUM / FT2 Team – ARI*  
*Caserta · Italy · GPLv3*