AMATEUR RADIO · AUGUST 15, 2026

Weak-Signal Receiver Software for Amateur Radio: Advances Through August 2026

The WSJT project shipped its first major release in over a year (WSJT-X 3.0.x, with MAP65 3.6 and QMAP 0.7), moved to GitHub with an automated build pipeline, and has a new JTTY mode in the pipeline; community work added the 3.8-second FT2 mode, multithreaded FT8 decoders, SDR front-ends with built-in FT8/FT4 decoding, and AI-based denoising.

10 min read · 15 sources

FT2 FT8 JTTY SDR WSJT-X digital modes open source weak signal

CONTENTS

The WSJT-X 3.0 line: release cadence, new features, project move 1
Decoder engines: from single-threaded to multithreaded and merged 3
New modes: JTTY (official, in development) and FT2 (community) 3
SDR receiver software with built-in weak-signal decoding 5
AI-assisted reception: early but moving 6
Mobile and the fork ecosystem 6
Bottom line 7
Sources 7

For roughly eighteen months — from February 2025 through August 2026 — weak-signal software for amateur radio has been through its busiest period in years: the WSJT project (WSJT-X, MAP65, QMAP) shipped its first major release since 2025 and reorganized around GitHub and a cloud build pipeline, a brand-new official mode called JTTY is in the pipeline, a community-developed **FT2** mode compresses FT8-style QSOs to a few seconds, and decoders have moved from single-threaded to multithreaded. Around the core, SDR receiver programs now bundle their own FT8/FT4 decoders, and the first AI-based denoisers aimed at pulling weak signals out of modern RFI have appeared. This document covers what changed and what it means for the operator.

3.0.2	3.8 s	+10 dB	1–12
Current WSJT-X (19 Jun 2026)	FT2 transmission cycle	SuperFox system gain	Configurable FT8 decode threads
+1 major vs Feb 2025	4× faster than FT8	vs 5-slot Fox/Hound	WSJT-Z / JTDX merge

The WSJT-X 3.0 line: release cadence, new features, project move

WSJT-X 2.7.0, released **17 February 2025**, was the first GA release in two years and is the baseline for everything since[1][4]. It introduced three things still shaping the ecosystem: **QMAP**, a

wideband receive-only companion for EME that monitors a 90 kHz sub-band while WSJT-X works a QSO; **Q65 Pileup** mode for DX operators with many simultaneous callers and very weak signals; and **SuperFox**, a Special Operating Activity that lets a DXpedition address up to **9 Hounds simultaneously** with a constant-envelope waveform and no signal-strength penalty — about **+10 dB system gain** over the old 5-slot Fox-and-Hound scheme[1][4]. It also added a structured Message System for contest QSY requests and one-click Hamlib/cty.dat updates[1].

- 17 Feb 2025 — **WSJT-X 2.7.0 GA**
  QMAP, Q65 Pileup, SuperFox, Message System
- 15 Sep 2025 — **WSJT-X 3.0.0-rc1**
  Full duplex, TCI, parallel FT8 decoding, filters, alerts
- 8 Apr 2026 — **WSJT-X 3.0.0 GA**
  First GA since 2.7.0; ships QMAP 0.6, MAP65 3.2
- 4 May 2026 — **WSJT-X 3.0.1**
  Bug-fix; macOS Tahoe compatibility, CloudLog fix
- 19 Jun 2026 — **WSJT-X 3.0.2 GA**
  Bug-fix; MAP65 back in packages, automated pipeline
- 26 Jun 2026 — **JTTY announced**
  Fast RTTY-like mode in development for upcoming RC

WSJT-X 3.0.0 (GA announced 6–8 April 2026) is the first major release since 2.7.0, and its headline items are operator-facing: single-click band buttons, **parallel FT8 decoding in concurrent threads** for faster decode, message-display filters, “Wait and Reply / Wait and Call / Wait and Pounce” sequencing, audible alerts on matching decodes, transmitted power and SWR in the status bar with automatic transmit stop above **SWR 2.5**, band hopping on FT8/FT4/MSK144 across up to eight frequencies, HiDPI scaling, directional-call highlighting (DX/SOTA/POTA/YOTA), downloadable moon ephemeris charts, and an Echo mode that sends your callsign to the Moon and decodes the return[1][4]. Release candidates had already added full-duplex operation, native TCI (ExpertSDR) rig control, and bundled OpenSSL[1]. QMAP 0.6 and MAP65 3.2 shipped in the same packages, with CQLive and PSKReporter spotting[1][4].

3.0.1 (4 May 2026) and **3.0.2** (19 June 2026) are bug-fix releases, but 3.0.2 matters structurally: it is the first release built by the team’s **automated build-and-package pipeline**, and it restored MAP65 to the installers after a modernization pass (cross-platform work, improved decoder output, writable user data locations)[1][2]. Fixes of note: HRD and OmniRig CAT failures, TCI transmit-audio issues in non-15-second modes, a Windows TLS bug that had broken SuperFox verification and HTTPS uploads, and a macOS packaging fix that restored FT8 callsign filtering on fresh installs[1].

Behind the releases, the project reorganized itself. Development moved to **github.com/WSJTX/wsjtx**, with the full open-source history back to 2005; on 26 June 2026 the team announced completion of the move to “a modern cloud-based development environment with automated

package building and testing”[2]. Signed Windows/macOS installers are planned for the next release[2].

KEY FINDING

The cadence change is real: 2.6.1 → 2.7.0 took two years; 3.0.0 → 3.0.2 took ten weeks. The WSJT team explicitly says the new pipeline “allow[s] for faster development and more frequent releases”[2].

Decoder engines: from single-threaded to multithreaded and merged

The quietest but most consequential advance is in the decoders themselves. WSJT-X 3.0.0 added an option for **parallel FT8 decoding in concurrent threads**, “and other changes that yield improved decoding speed and performance”[1][4]. The community forks went further. WSJT-X Improved (DG2YCB) introduced a **multithreaded FT8 decoder (MTD)** alongside the single-threaded one, with “2-stage” and “3-stage” options that combine both “intelligently … resulting in the best FT8 decoding performance to date” [search: SourceForge WSJT-X Improved]. And **WSJT-Z** — the successor to WSJT-X Improved, which moved to GitHub in May 2026 — merged the JTDX decoder into the WSJT-X 3.0 baseline: it ported 18 modified and 11 new Fortran files plus the full `ft8var/` directory from JTDX, and exposes a **1–12 thread FT8 decode control** with an Auto setting[11]. JTDX itself remains independently maintained (last SourceForge update May 2026) and is prized on crowded bands for its iterative multi-pass decoding that re-runs and subtracts successful decodes[13] [search].

DECODER APPROACH	WHERE IT LIVES	WHAT IT DOES
Single-threaded FT8 (STD)	WSJT-X 2.7 and earlier	One decode pass per 15 s period
Parallel FT8 (threads)	WSJT-X 3.0.0+	Optional concurrent-thread decoding
MTD (multithreaded), 2/3-stage	WSJT-X Improved	Combines STD + MTD passes; best-in-class FT8
JTDX decoder merged	WSJT-Z 2.0.1	JTDX iterative deep-decode inside WSJT-X 3.0; 1–12 threads

The practical effect: on a crowded FT8 band, where decode rate is the difference between working the DX and watching it, the 2026 builds decode more stations per period on ordinary hardware than any prior release.

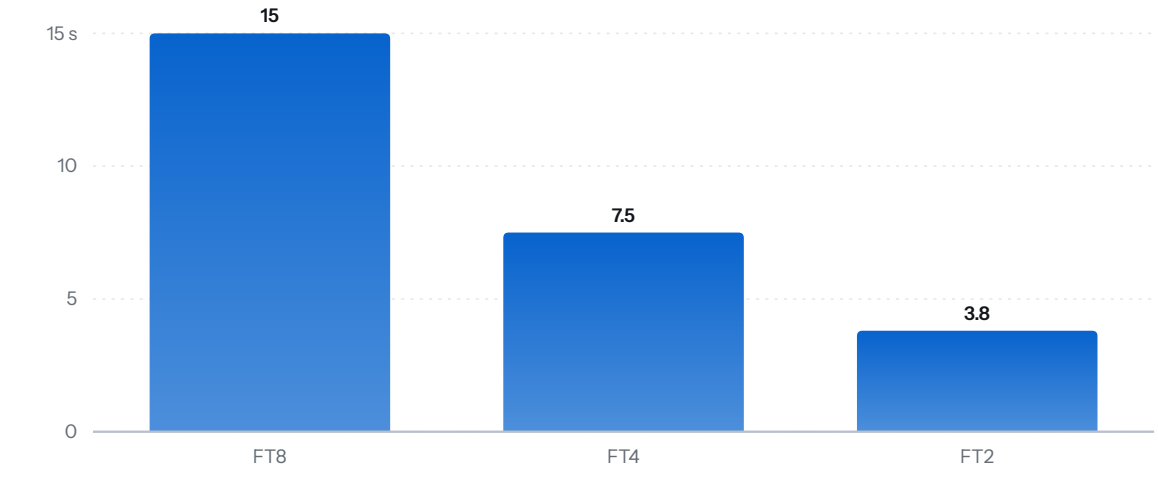
New modes: JTTY (official, in development) and FT2 (community)

JTTY is the WSJT Development Team’s next mode, announced 26 June 2026[2]. From the announcement:

> *"We are well along in the development of a new mode called JTTY, designed for fast RTTY-like contest exchanges and other keyboard-to-keyboard communication. Its general goal is an operational feel similar to standard RTTY, but with much better weak-signal performance and far lower error rate."*

It is expected "in an upcoming WSJT-X release candidate," with on-the-air testing requested[2]. This is the first mode built under the new faster-release regime — a marker of where the project is heading after the FT8/Fox-and-Hound era.

FT2, announced 18 February 2026, is a community mode by Martino IU8LMC with the ARI Caserta team, built as a modified WSJT-X[3]. It reuses the FT8/FT4 codec exactly — **77-bit payload, LDPC (174,91), 8-GFSK** — and only compresses the timing: symbols are shorter and the transmit cycle drops from 15 s (FT8) to **3.8 seconds**, four times faster than FT8 and twice as fast as FT4. A complete QSO takes **7–11 seconds** instead of the full FT8 minute[3]. Sensitivity is the trade-off: roughly **−12/−13 dB SNR** at 150 Hz bandwidth — weaker than FT8, but ample for contests, DXpeditions and pile-ups where signals are strong[3]. The developer reports dozens of real contacts on 40 m and 80 m on 16 February 2026, spanning Campania, Capri, Sardinia and Turin, with decodes down to −12 dB[3]. FT2 is not part of official WSJT-X; it circulates as beta software with a WhatsApp group for testers, and MSHV already supports it (see below)[3][9].



FT2 figure per IU8LMC (18 Feb 2026); FT4 is twice FT2's speed per the same source.

	FT8	vs	FT2
Transmission cycle	15 s	·	3.8 s
Typical QSO time	~60 s	·	7–11 s
Modulation	8-GFSK, LDPC (174,91)	·	identical codec
Sensitivity	≈ −20+ dB	·	≈ −12/−13 dB
Status	Official, WSJT-X	·	Community beta (IU8LMC)

FT2 announced 18 Feb 2026; parameters per the developer.

CAUTION

FT2 buys speed at roughly 7 dB of sensitivity versus FT8. It is positioned for contests, DXpeditions and pile-ups where signals are strong — not as an FT8 replacement on weak paths [3].

For context, **SuperFox** and **Q65 Pileup** (both GA since 2.7.0) remain the throughput story for DXpeditions: SuperFox verification now uses a real-time one-time-password system, with keys issued by the Northern California DX Foundation, and 3.0.2 fixed the Windows TLS issue that had been breaking that verification for some users [1].

SDR receiver software with built-in weak-signal decoding

A clear 2025–2026 trend: SDR front-ends and network receivers now decode FT8/FT4/WSPR themselves, without a separate audio pipe into WSJT-X.

SDR++ Brown — a fork of SDR++ by SannySanoff, covered by RTL-SDR.com in June 2025 — bundles a **built-in FT4/FT8 decoder extracted from MSHV**, stripped of Qt and made lightweight [6] [7]. It decodes the whole SDR passband at once, so you can tune elsewhere and keep seeing decodes, groups them by distance from your grid, reports to PSKreporter, and writes WSJT-X-compatible logs. The fork also adds DSD (DMR/P25/NXDN), direct KiwiSDR/WebSDR connection, Hermes Lite 2 support with transmit, and wideband noise reduction — the mainline SDR++ project has not absorbed these features [6] [7].

KiwiSDR 2 is the hardware/software flagship of network weak-signal reception: the store opened (kiwisdr.nz), an ARRL *QST* review ran in February 2026, and the site was updated 4 August 2026 [14]. Each unit serves its own web UI with built-in WSPR/FT8 decoding, and thousands of networked receivers feed propagation tools. **WSPRdaemon 3.3.2** is the software that turns KiwiSDRs (and RX888 Mk II via Phil Karn's ka9q-radio) into unattended weak-signal machines: it decodes **WSPR, FST4W, FT8/FT4 and the WWV/WWVH/CHU frequency standards** across at least 8 channels, reports to wsprnet.org, PSKreporter and HamSCI, estimates noise with RMS and FFT methods, and pushes spots plus noise floors into a TimescaleDB/Grafana stack [10]. This is the backbone of the HamSCI personal space-weather station program.

SOFTWARE	PLATFORM	BUILT-IN WEAK-SIGNAL DECODE	EXTRA
SDR++ Brown	Win/Linux/macOS/ Android	FT8 + FT4 (from MSHV)	PSKreporter, DSD, KiwiSDR remote, Hermes Lite 2 TX
KiwiSDR 2	Web/network receiver	WSPR, FT8 + others in web UI	Global receiver network
WSPRdaemon 3.3.2	Linux service	WSPR, FST4W, FT8/FT4, WWV/CHU	8+ channels, noise metrics, HamSCI reporting

AI-assisted reception: early but moving

The first AI-driven weak-signal reception tools appeared in 2025–2026, all still experimental. **RFWhisper** (GitHub, latest commit 20 July 2026) is a fully local, open-source, real-time ML noise-reduction pipeline for amateur radio: a ham-tuned **DeepFilterNet3** denoiser (with an RNNoise fallback) inside a GNU Radio + SoapySDR + ONNX Runtime chain, aimed at pulling weak SSB/CW/FT8 out of powerline buzz and inverter hash without cloud processing[8]. Honesty caveat from the project itself: it is an early alpha, the neural denoiser is not built yet, and the current placeholder spectral-subtraction stub measures **+1.1 dB SNR gain** against a v0.1 target of +3 dB average and +6 dB on powerline buzz[8].

> CAUTION
>
> The runtime-AI story is real but not production-grade. RFWhisper's own README warns "Do not expect this to clean up your audio today" — a classical spectral-subtraction placeholder sits in the denoiser slot, and measured gain is +1.1 dB against a +3 dB target[8].

Elsewhere, AI is used as a *development* tool rather than a runtime one: the FT2 mode "was conceived and developed by IU8LMC … Artificial intelligence was used as a development tool to assist with modifications to the WSJT-X source code"[3], and the 2026 Northeast HamXposition featured a talk on extending FT8 via AI-generated Android/iOS apps [search]. Expect the runtime-AI story (neural denoisers feeding WSJT-X) to be the next frontier, but today none of it is production-grade — treat RFWhisper-class tools as research previews.

Mobile and the fork ecosystem

Mobile weak-signal operation matured too. **FT8CN** (BG7YOZ) brought native FT8 to Android; its fork **FT8AF** modernizes it with a Jetpack Compose/Material 3 UI, localization in seven languages, USB CAT rig control, an active-QSO monitor, a caller queue, TX1/TX2 slot selection and a CQ/Stop toggle — with releases tagged as recently as August 2026[12]. On the desktop, the fork ecology around WSJT-X consolidated: **WSJT-X Improved became WSJT-Z** (releases on GitHub from May 2026), **JTDX** continues (May 2026 update), and **MSHV 2.76.6** (28 April 2026) added **FT2 and FT4 AP7 decoding** (the deep-decode pass previously FT8-only) plus compound-callsign support for all 77-bit modes — making it the first mainstream program besides the FT2 beta to

decode the new mode[9][11]. One caveat the official team repeats: only github.com/WSJTX/wsjtx is official, and they ask derivative builds not to publish WSJT-X features before GA[2][4].

Bottom line

For a weak-signal operator the practical takeaway of 2025–2026 is: **upgrade to WSJT-X 3.0.2** (or a maintained fork like WSJT-Z if you want the multithreaded/JTDX decoder on crowded bands), because 3.0.x brings parallel decoding, better EME tooling, and the first releases built by the project's new automated pipeline — and watch for **JTTY** in the next release candidate, plus **FT2** if contest-speed QSOs matter more to you than ultimate sensitivity. On the receiver side, SDR++ Brown, KiwiSDR 2 and WSPRdaemon now make FT8/WSPR spotting a built-in feature of the SDR itself rather than a separate audio-chain task. The genuinely new frontier — neural-network denoising in real time, on your own hardware — is real but still alpha; its targets (+3 dB average SNR gain) are modest, and claims should be checked against the project's own measurements before building a shack around them.

SOURCES

1 WSJT-X Release Notes (full text, 2.7.0 through 3.0.2)
https://wsjtx.github.io/wsjtx/Release_Notes.txt

2 New Digital Mode – JTTY (WSJT Development Team announcement)
https://www.dx-world.net/new-digital-mode-jtty/

3 New Digital Mode: FT2 (IU8LMC / ARI Caserta)
https://www.dx-world.net/new-digital-mode-ft2/

4 WSJT-X Version 3.0.0 GA has been released — Mac Ham Radio
https://machamradio.com/blog/2026/04/10/wsjt-x-version-3-0-0-ga-has-been-released/

5 WSJTX/wsjtx Releases — GitHub
https://github.com/WSJTX/wsjtx/releases

6 SDR++Brown is a fork (patch-set)
http://sdrpp-brown.san.systems/

7 TechMinds: Testing the SDR++ Brown Fork with Built-In DSD and Remote KiwiSDR Support — RTL-SDR.com
https://www.rtl-sdr.com/techminds-testing-the-sdr-brown-fork-with-built-in-dsd-and-remote-kiwisdr-support/

8 RFWhisper — real-time, fully local AI denoising for amateur radio (GitHub)
https://github.com/JakenHerman/RFWhisper

9 MSHV Amateur Radio Software — LZ2HV
http://lz2hv.org/mshv

10 WSPRDAEMON Documentation — wsprdaemon 3.3.2
https://wsprdaemon.readthedocs.io/en/master/

11 WSJT-Z — WSJT-X fork with extended automation (GitHub)
https://github.com/sq9fve/wsjt-z

12 FT8AF — FT8 on Android, modernized (GitHub)
https://github.com/patrickrb/FT8AF

13 Weak Signal Modes — ARRL
https://www.arrl.org/weak-signal-modes

14 KiwiSDR — home page
http://kiwisdr.com/

15 WSJT-X — Official Home
https://wsjtx.github.io/wsjtx/index.html

Researched and written with deepseek/deepseek-v4-flash-0731, from the sources listed above. Figures are as reported by those sources. Exported November 29, 2026 from https://intel.innerpulse.net/doc/weak-signal-receiver-software-for-amateur-radio-advances-through-august-2026, where the current version lives.