MICROWAVE UPDATE 2026

From Soldering Irons to Syntax

Building real station tools with generative AI, and what good looks like

Rus Healy, K2UA

Springwater, NY · FN12fr

10ghzdx.com/mud2026

2025 ARRL 10 GHZ AND UP · DAY 3

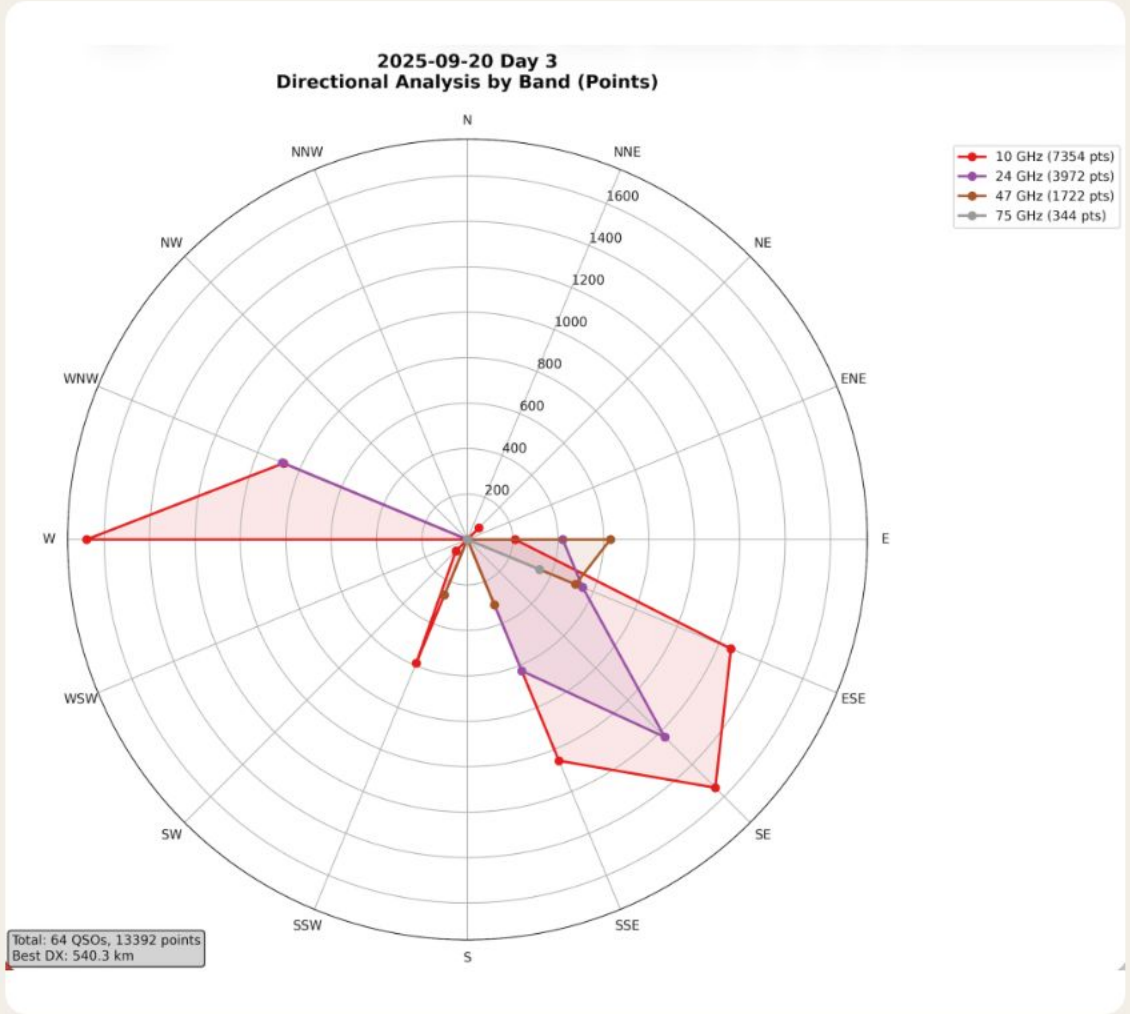
Built by a hardware guy, the week after the contest

64 QSOs

13,392 points

540 km best DX

Points by compass direction and band. Where is my site strong, and where is the hill?



I build systems. I don’t write syntax.

Day job

Retired aerospace and satellite solutions architect, Amazon Web Services. AWS Certified AI Practitioner.

Not a developer

No CS degree, no shipping code career. I architect; the AI is the high-speed technician.

On the air

HF through 10 GHz from FN12fr. Team lead, New York QSO Party.

Every project in this talk is public at github.com/rusk2ua

The software wall, and who does what now

You keep this

- RF intuition and systems thinking
- The requirements and the logic
- Test discipline: does it actually work?
- The source material: rules, logs, datasheets



AI takes this

- Syntax and boilerplate
- Library and framework choices
- Build tooling, CI, deployment glue
- First drafts of docs and tests

It’s a new building material, like copper wire or silicon. You still decide what to build.

SINCE THE MICROWAVE UPDATE PROCEEDINGS PAPER WENT TO PRESS

August to October: what moved

“Cowork” is just Claude now

The app reads and writes your project folders, runs code and keeps working while you step away.

House rules, written once

Skills and project instructions carry your conventions: versioning, README style, commit identity.

It remembers the station

Memory across chats: your rig, your repos, your decisions. Less re-explaining every session.

My toolbelt narrowed

Claude now covers nearly everything I used Kiro for. This deck and its website were built with it.

The grounded builder: three rules

1

Curate before you code

Hand it the primary source. The ARRL rules PDF, data sheets, etc--not a web search for them.

2

Isolate the context

One project, one folder, one conversation. Only the sources that matter.

3

Be the control operator

The AI writes syntax. Logic, limits and compliance remain yours.

Without these, you get **AI slop**: plausible, bloated, wrong. With them, you get tools you trust on contest weekend.

The workbench, October 2026

Tool	What I use it for	My verdict
Claude	Multi-file projects in local folders, code, tests, docs, decks, websites	Primary. Nearly everything
Gemini Notebook	Source-grounded research, image generation, podcast-style overviews	Very much worth a look
Kiro	Spec-first IDE for people who live in VS Code	Rarely, now
ChatGPT	Nothing.	Hard no (next slide)

My experience, on my projects. Test your own tools against your own source material.

A WAR STORY FROM LAST WEEK

One schematic. Many iterations. Still wrong.

What I asked for

A simple schematic diagram for a dual MOSFET driver for an IR LED transmitter

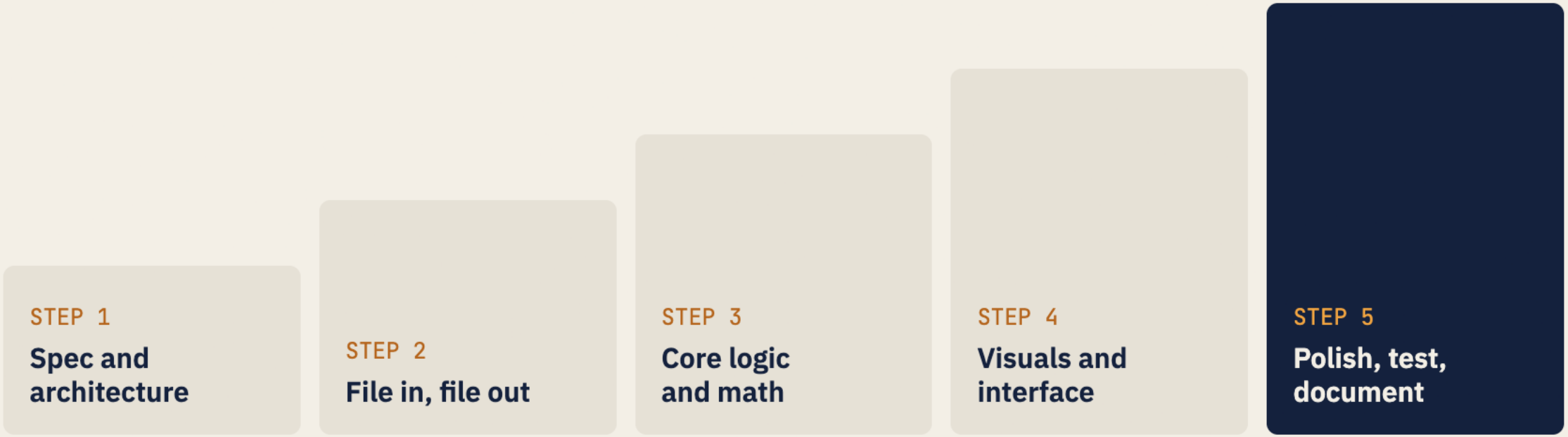
What came back

A shorted dropping resistor, wires connected incorrectly to the MOSFETs

The lesson

Check every output against the datasheet, regardless of what tool made it.

Climb the staircase. Never one giant prompt.



“Ask me clarifying questions before you write any code.” The questions you'll get in return surface design decisions you hadn’t considered yet.

Anatomy of a prompt that works

Role	You're a Python developer who values clean, single-file scripts.
Context	A grid-square map of a 10 GHz contest log, to share with other rovers.
Constraints	Python only, runs on Mac and Windows, no database. Score per the attached rules.
Input	A Cabrillo 3.0 log. Sample attached.
Output	One high-resolution PNG per band, darker grids for more contacts.

Not a developer? Leave out the libraries. Describe the picture you want and where it will run.

LIVE DEMOS

What good looks like

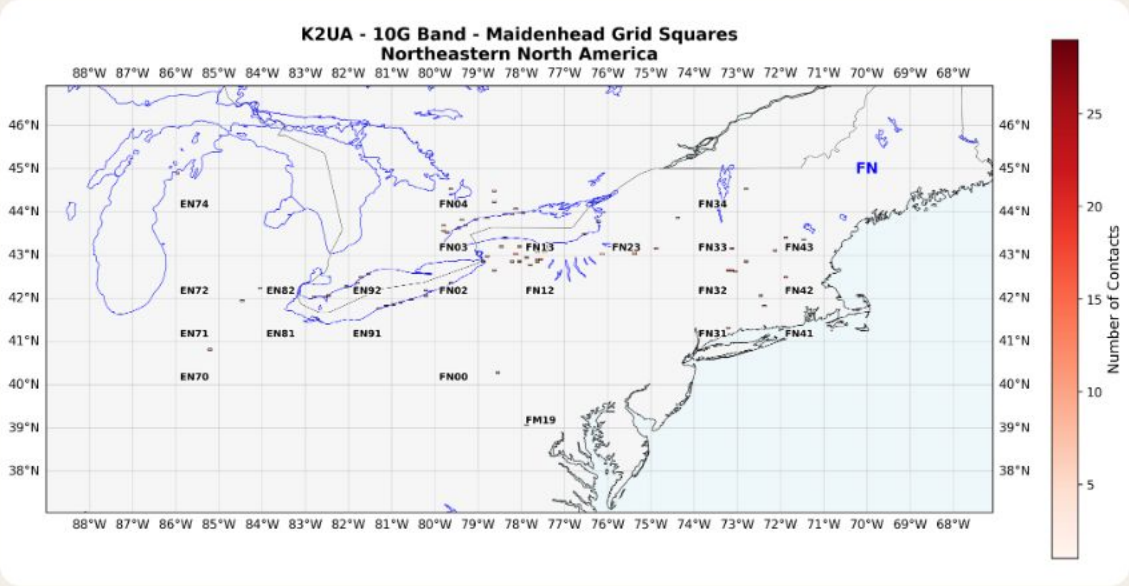
1	**10 GHz log analyzer, web edition**	Build the engine before the dashboard
2	**940 nm optical beacon and receiver**	Respect the source, show the math
3	**remkey: remote CW operation for my 10 GHz home station**	Step-by-step build

DEMO 1

Upload a log, get the whole weekend back

- Cabrillo, CSV or a Google Sheets link in
- Submittable Cabrillo with claimed score out
- Polar plots per day, or per rover location
- Path maps and grid-density maps per band
- Log comparison, dupe check, one .zip

10ghz.microwavedx.com



TECHNIQUE: BUILD THE ENGINE BEFORE THE DASHBOARD

Command line first. The web app came last.

CLI tools
10ghz-log-analyzer v1.6.1 and grid-mapper v1.5.0

→

Vendored
Copied in unmodified. A sync script, never hand edits

→

Lambda job
Container image, background jobs up to 15 min

→

Browser
Static page via CloudFront; results expire in a day

Parity test: the web app's Cabrillo log, summary and reports must match the CLI's, character for character.

TECHNIQUE: BE THE CONTROL OPERATOR

Two bugs that looked fine at a glance

v1.5.1 · the phantom QSO

A blank spreadsheet row became a duplicate contact

Forward-fill copied *every* column, call and grid included. Fix: fill only date, band, grid and time. Never invent a call sign.

v1.6.1 · the wrong multiplier

134+ GHz scored 6x; 241 and 300 GHz scored 10x

Rules section 5.2: every band from 122 GHz up is 5x. It changed the claimed score. Now a test pins the table to the rules.

Plausible code passes a glance. **Only the rules document and real logs catch it.**

DEMO 2

A 940 nm optical link, both ends

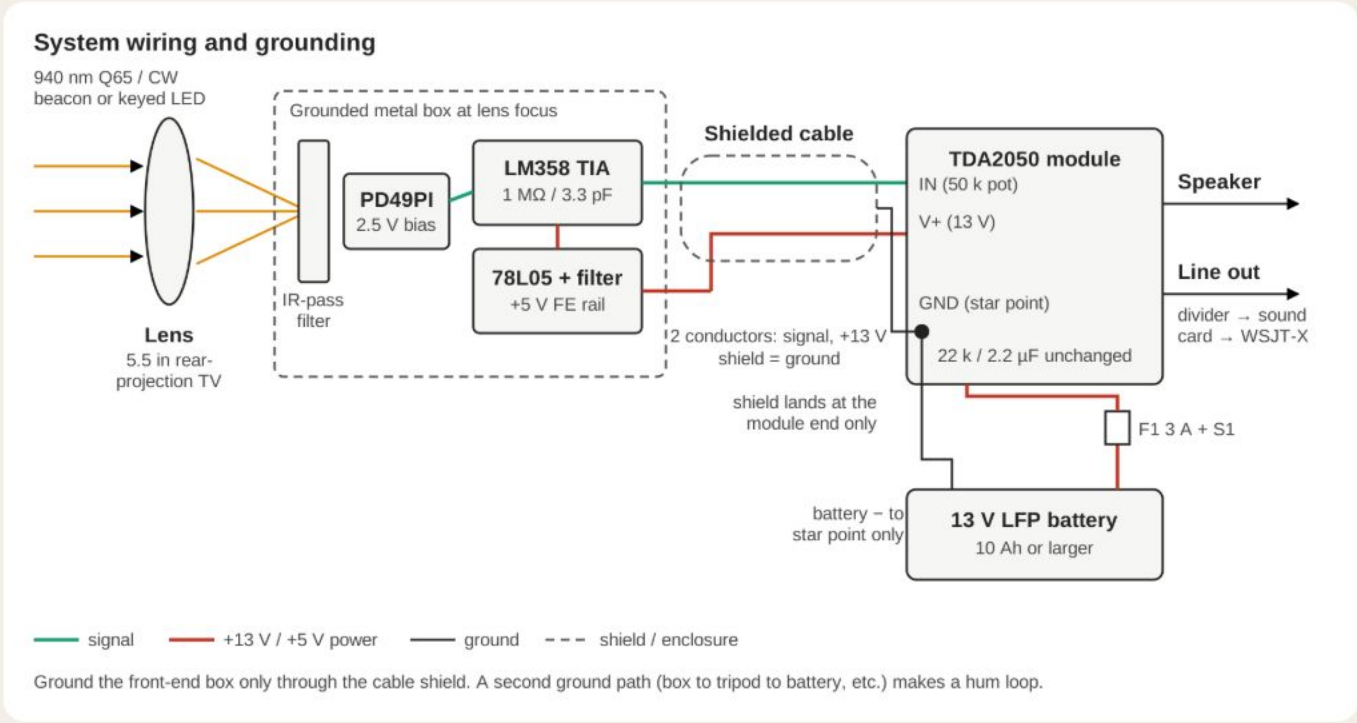
Transmit

VE2UG's ESP8266 beacon: Q65-30, then CW, then carrier, locked to GPS or NTP time.

Receive

Surplus 5.5 in TV lens, PD49PI photodiode, LM358 TIA, TDA2050 module, into WSJT-X.

`ve2ug-optical-beacon-project`
`optical-receive-chain`



TECHNIQUE: RESPECT THE SOURCE

Improving someone else's design, properly

Credit up front

VE2UG named as original author in the sketch, README and changelog.

Say what you kept

An "Unchanged" section: symbol timing, keyer, pins and jumper logic are his.

Honest test status

NTP mode verified on the bench. GPS mode: "not yet tested." In writing.

No secrets in git

Wi-Fi credentials ship as placeholders, with a warning beside them.

Respect licenses

The photodiode datasheet forbids copying, so the repo links to it instead.

Real bugs fixed

An NTP sync guard: no transmitting at a 1970 timestamp right after boot.

TECHNIQUE: MAKE IT SHOW ITS MATH

Design reasoning you can re-run

```
$ python3 tools/tia_calc.py
Op-amp LM358, Rf 1 Mohm, Cf 3.3 pF
Min Cf for ~45° margin : 2.22 pF (OK)
Modeled -3 dB point : 65.1 kHz
15 kHz response : -0.05 dB
Input noise, 300 Hz–20 kHz : 36.7 pA RMS
Max DC photocurrent : 2.45 µA
```

The Rf/Cf choice is a script, not a sentence in a chat. When the OPA350 arrives, one flag re-runs the whole analysis.

And it says so when it’s guessing: **“ballpark datasheet values; measure your own build.”**

DEMO 3 · REMKEY

Remote paddle CW: send characters, not timing

Paddle + Nano
Local keyer, zero-latency sidetone, decodes to characters

→

Mac client
Sends text over TCP with auth and heartbeat

→

Shack service
Keys off safely if the link drops

→

Station Nano
Re-sends with exact timing via opto to the 7610 KEY jack

remtlm: tower-top PA power, temperatures, reference lock, Kuhne transverter status and supply current, over LoRa.

TECHNIQUE: TESTS BEFORE THE TOWER

Software done and tested, hardware not yet built

- ✓ **CI on every push:** Python on Linux, Windows and macOS
- ✓ **Firmware simulated** on a PC: keyer timing, decoder, watchdog
- ✓ **End-to-end network tests** over localhost, client to station
- ✓ **CI caught a real break:** a firmware image that wouldn’t compile

The README’s first line says it plainly: **not yet run on real hardware.** A phased build plan gets it from parts to tower.

WHAT GOOD LOOKS LIKE

The checklist every repo in this talk follows

- ✓ README: what it is, how to run it, test status
- ✓ CHANGELOG with the reason for each fix
- ✓ One versioning scheme, used consistently
- ✓ A LICENSE, and credit for upstream authors
- ✓ Link restricted datasheets; don’t copy them
- ✓ Tests, and CI that runs them on every push
- ✓ Sample data, so a fresh clone runs at once
- ✓ No secrets: placeholders and a .gitignore
- ✓ Commits use your GitHub noreply email
- ✓ House rules written once, so the AI does all this

MCP: THE TNC BETWEEN AI AND YOUR DATA

Powerful, and a security minefield

22,775
MCP servers listed in one directory

36.7%
carry SSRF vulnerabilities

41%
have no authentication, mine included

8.5%
use OAuth

Start with vendor-maintained servers: Filesystem, GitHub's official server, cloud providers' documentation servers. Add community ones once you can read their code.

Figures as of May 2026 (Glama directory and published security research), from the paper · 10ghzdx.com/mud2026

Your first project, this week

1	**GitHub account and the CLI**	`gh auth login`
2	**Protect your email first**	`git config --global user.email ID+you@users.noreply.github.com`
3	**A folder for one project**	`git init -b main`
4	**Open Claude on that folder**	Attach the rules or datasheet. Climb the staircase.
5	**Share it**	`gh repo create my-tool --private --source=. --push`

A web page? Put it in /docs, add an empty .nojekyll, turn on GitHub Pages. Free hosting.

Full step-by-step guide: 10ghzdx.com/mud2026

The four questions I always get

“Isn’t it all AI slop?”

Without source material, often. With the rules, datasheets and sample logs in hand, it’s a different tool.

“Isn’t that cheating?”

We didn’t stop building when tubes became transistors. The design, the testing and the judgment are still yours.

“What does it cost?”

Free tiers are capable for a first project. Paid plans start around $20 a month when you outgrow them.

“What about my private logs?”

Work in a local folder and share only what the task needs. Check your tool’s privacy settings.

Questions?

Slides, prompt library, project gallery and the setup guide, all in one place.

10ghzdx.com/mud2026

github.com/rusk2ua

73, Rus Healy, K2UA
k2ua@pm.me



Scan for everything