

Building the WYAR FM Probe

A One-Sitting Guide to Standing Up the Off-Site Witness — Built On-Site at Cumberland.

JEFF FRANZEN · JUNE 19, 2026 · DO IT TODAY

The website is already built, deployed, and live at `wyar.jbf.com`. The only job left is the physical probe: a Raspberry Pi with a USB software-defined radio that listens to 88.3 FM and reports to the website every minute.

You're building this **in place at the Cumberland Foreside location** — the off-site witness. That's the right spot on purpose: generator-backed power and internet separate from the station, so it can still report when the station goes dark. No bench-build, no relocation. Set it up here, on the UPS, and leave it running.

Until a real radio reports in, the dashboard reads `FM Carrier: WARNING (waiting)` and stays quiet. The finish line for today: that line flips to a **real reading**, and the probe heartbeat goes green.

The Model, in One Line

The Pi hears 88.3 at Cumberland → posts to `wyar.jbf.com` every 60 seconds → the dashboard shows the carrier and alerts the group the moment it drops.

It's already at the generator-backed off-site location, so once it's proven there's nothing left to move. A probe that only lives at the station can't tell anyone the station went dark — this one can.

Bring These to Cumberland

- ☐ Raspberry Pi kit — board, power supply, case, microSD card
- ☐ Nooelec NESDR SMARt v5 — the USB SDR dongle
- ☐ FM antenna — the one in the SDR bundle is fine to start
- ☐ Short USB extension cable · UPS / battery backup
- ☐ Your Mac — to flash the card and SSH into the Pi
- ☐ The Cumberland Wi-Fi name + password

Everything happens on the Cumberland network. The Mac you flash with must be on that same Wi-Fi when you SSH in later.

PHASE 1

Assemble the Pi and Flash the Card

- 1 Put the Pi in its case. Attach the heatsinks or fan if the kit included them. **Do not plug in power yet.**
- 2 On your Mac, install **Raspberry Pi Imager** from `raspberrypi.com/software`.
- 3 Insert the microSD card (use the kit's USB reader if your Mac has no slot).
- 4 In Imager, choose:
 - **Device:** your Pi model (Pi 5 or Pi 4).
 - **OS:** Raspberry Pi OS (other) → **Raspberry Pi OS Lite (64-bit)**. Lite means no desktop — you only ever reach it over SSH.
 - **Storage:** the microSD card. Double-check it's the card, not a backup drive.
- 5 Click **Next**, then **Edit Settings**. Set:
 - Hostname: `wyar-probe`
 - Enable SSH → password authentication
 - Username `wyar` and a password you'll remember
 - **Cumberland Wi-Fi** name + password — this is the final network, no re-flashing later
 - Your local time zone
- 6 Save, apply, and let it write and verify (a few minutes).

CHECKPOINT 1

Imager says "Write Successful." Eject the card.

PHASE 2

First Boot on the UPS, Then Connect Over SSH

- 1 Put the microSD card into the Pi.
- 2 Plug the **USB extension cable** into the Pi, the **SDR dongle** into the other end, and screw the **antenna** onto the SDR.
- 3 Plug the Pi's power into the **UPS**, and the UPS into the wall. On the UPS now means a power blip during setup won't cost you progress. It boots; give it ~90 seconds.
- 4 On your Mac (on the Cumberland Wi-Fi), open Terminal and connect:

```
ssh wyar@wyar-probe.local
```

Type `yes` if asked, then your password. If `wyar-probe.local` doesn't resolve, find the Pi's IP in the router and use `ssh wyar@THE.IP.ADDRESS`.

CHECKPOINT 2

Your prompt reads `wyar@wyar-probe:~ $`. You're on the Pi. Everything below runs on the Pi unless it says "on your Mac."

PHASE 3

Install the Software

Update the system and install the tools the probe needs:

```
sudo apt update
sudo apt install -y python3-venv ffmpeg rtl-sdr git
```

CHECKPOINT 3

No red errors; `apt` finishes.

PHASE 4

Confirm the SDR Hears 88.3 *From Here*

This matters more at Cumberland than at the station — you're across the water, so confirm the signal actually arrives at this spot.

- 1 Check the dongle is detected:

```
rtl_test -t
```

Good: prints Found 1 device(s): 0: Realtek, RTL2838UHIDIR . It may add "No E4000 tuner found" — that's fine. Press `Ctrl+C` to stop.

Bad: No supported devices found → see **Troubleshooting A**. Fix it, re-run.

- 2 Read the actual WYAR band for one sweep:

```
rtl_power -f 88.0M:88.6M:10k -g 30 -1 ~/wyar_test.csv && cat ~/wyar_test.csv
```

You want rows of numbers, not an error. If the levels look very weak (near the noise floor), reposition the antenna — near a window, up high, away from metal — and re-run. Good antenna placement here is the one thing only you can do on-site.

CHECKPOINT 4

`rtl_test` finds the device and `rtl_power` shows numbers. If weak, you've confirmed reception and can improve the antenna.

PHASE 5 · PART 1

Put the Probe on the Pi

Pull the project and lay it out where the service expects it:

```
sudo mkdir -p /opt/wyar-station-monitor
sudo chown -R wyar:wyar /opt/wyar-station-monitor
git clone https://github.com/franzenjb/wyar-station-monitor.git /tmp/wyar
cp -R /tmp/wyar/probe /opt/wyar-station-monitor/
```

Build the Python environment:

```
cd /opt/wyar-station-monitor/probe
python3 -m venv .venv
.venv/bin/pip install -r requirements.txt
```

PHASE 5 · PART 2

Configure the Probe

The probe needs one secret from Vercel so the website trusts its reports.

- 1 On your Mac, open Vercel → project **wyar-station-monitor** → **Settings** → **Environment Variables**.
- 2 Find `PROBE_SHARED_SECRET` and copy its value.
- 3 If it doesn't exist yet: make up a long random string, add it as `PROBE_SHARED_SECRET`, **redeploy**, and use that same string below.
- 4 Back on the Pi, create the config file:

```
sudo nano /etc/wyar-probe.env
```

Paste this in, replacing `PASTE_THE_SECRET_HERE`, then save with `Ctrl+O`, `Enter`, `Ctrl+X`:

```
WYAR_DASHBOARD_URL=https://wyar.jbf.com
WYAR_PROBE_SECRET=PASTE_THE_SECRET_HERE
WYAR_STREAM_URL=https://wyar.broadcasttool.stream/play
WYAR_FREQUENCY_MHZ=88.3
WYAR_RF_FLOOR_DBFS=-65
WYAR_SILENCE_FLOOR_DBFS=-50
WYAR_PROBE_LOCATION=Off-site, Cumberland Foreside
WYAR_PROBE_HARDWARE=Raspberry Pi + Nooelec NESDR SMarT v5 + FM antenna
WYAR_INTERVAL_SECONDS=60
WYAR_PORTABLE_RADIO_DEMO=false
```

CHECKPOINT 5

`cat /etc/wyar-probe.env` shows your real secret (not the placeholder) and `WYAR_PORTABLE_RADIO_DEMO=false`.

PHASE 6

Test One Report Before Going 24/7

Dry run first — measures the RF and prints the payload, posts nothing:

```
cd /opt/wyar-station-monitor/probe
set -a; . /etc/wyar-probe.env; set +a
.venv/bin/python wyar_probe.py --once --dry-run
```

You want a real RF level in the payload, **not** "rtl-sdr tools are not installed." If you see that, go back to Phase 4.

Now post one real report (drop `--dry-run`):

```
.venv/bin/python wyar_probe.py --once
```

CHECKPOINT 6

On your Mac, open `wyar.jbf.com`. The FM Carrier line changed from `WARNING (waiting)` to a real reading. If not, see **Troubleshooting B**.

PHASE 7

Run It Forever

Install the background service so the probe runs every 60 seconds and restarts itself after any power blip or reboot:

```
sudo useradd --system --create-home wyarprobe 2>/dev/null; true
sudo chown -R wyarprobe:wyarprobe /opt/wyar-station-monitor
sudo cp /opt/wyar-station-monitor/probe/systemd/wyar-probe.service /etc/systemd/system/
sudo chmod 600 /etc/wyar-probe.env
sudo chown wyarprobe:wyarprobe /etc/wyar-probe.env
sudo systemctl daemon-reload
sudo systemctl enable --now wyar-probe
```

Confirm it's running and watch a couple of cycles:

```
systemctl status wyar-probe --no-pager
journalctl -u wyar-probe -f
```

Ctrl+C stops watching the log; the service keeps running.

CHECKPOINT 7

`systemctl status` says **active (running)**. The dashboard updates on its own every minute.

PHASE 8

Prove the Loop, Then Leave It In Place

- 1 Pull the antenna off the SDR for about two minutes and watch `wyar.jbf.com`. The FM carrier reading should weaken or drop — that's the alarm path working. Screw the antenna back on and it recovers.
- 2 Settle the Pi into its permanent spot: antenna in the position that read best in Phase 4, Pi on the UPS, somewhere it won't get unplugged or overheat. You're already off-site — nothing left to move.

CHECKPOINT 8

You saw the reading react to losing the antenna, then recover, and the Pi is in its final spot on the UPS. **Done — the off-site witness is live.**

Troubleshooting

A · RTL_TEST SAYS "NO SUPPORTED DEVICES FOUND"

Linux grabbed the dongle as a TV tuner. Block that driver, then reboot:

```
echo 'blacklist dvb_usb_rtl28xxu' | sudo tee /etc/modprobe.d/blacklist-rtl.conf  
sudo reboot
```

Wait about 90 seconds, SSH back in, re-run `rtl_test -t`.

B · DASHBOARD DIDN'T CHANGE AFTER A REAL POST

- Secret mismatch is the #1 cause. `WYAR_PROBE_SECRET` on the Pi must exactly equal `PROBE_SHARED_SECRET` in Vercel. Re-check both; redeploy Vercel if you changed it.
- Confirm the Pi has internet: `ping -c 3 wyar.jbf.com`.
- Re-run with `--dry-run` off and read the error it prints.

C · SERVICE WON'T START (STATUS SHOWS FAILED)

```
journalctl -u wyar-probe -n 50 --no-pager
```

Read the last lines — usually a path or permissions issue on `/etc/wyar-probe.env`.

D · SIGNAL WEAK AT CUMBERLAND

Across the water, reception can be marginal. Move the antenna near a window or up high, away from metal and electronics. The bundled antenna is a starting point; a better FM/VHF antenna is the upgrade if `rtl_power` stays near the floor.

If You Only Remember One Thing

It's already at the off-site, generator-backed location. Build it here, prove it reacts to losing the antenna, and leave it on the UPS.

A probe that only lives at the station can't report that the station went dark. The Cumberland witness across the water is what makes this monitor honest.