

Project Drogheda — GNSS & Log Analysis

HBOX RTK fleet: bench triage, healthy baseline, and observability

Helgen Technologies – CTO/CSO

20 June 2026

Agenda

Context

Backend (retranslator)

GNSS log findings

Changes & tooling

Next steps

Context

HBOX RTK trackers on Hutchison Ports heavy equipment (ICAVE, Veracruz) – worked, then failed.

Three independent failure fronts, fixed in parallel:

- Backend** (retranslator) – could drop the fleet / black-hole a unit

- Thermal** – PLA box in a sealed cabinet (heat trap)

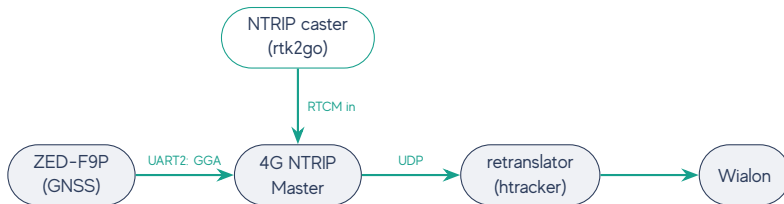
- Prototype HW + observability** – hand-wired; logs untrustworthy

Field units pulled → bench triage in Querétaro (firmware v1, no thermal log).

Invariant

Corrections ride the **4G modem**, not the GNSS – recover the link by power-cycling the stack.

The signal chain



Corrections come down from the caster over 4G; position goes out by UDP to the retranslator. The GNSS receiver never talks to the network directly.

Backend (retranslator)

Backend — hardened & deployed to prod

Head-of-line blocking removed: a WialonForwarder thread pool does all Wialon TCP I/O off the UDP receive loop – one slow upstream no longer drops the fleet.

Anti-jump now time-aware + self-healing: re-anchors after rejects, clears on disconnect – a unit is never permanently black-holed.

Identity keyed by `cell_id` (CGNAT-safe), not `(ip,port)`.

Monitor gained a **backend-victim** KPI; logs ship to Grafana Loki. [Live in production.](#)

Free triage

ONLINE in htracker + OFFLINE in Wialon = **backend victim**, not a dead radio.

GNSS log findings

What we analyzed

7 swap-matrix bench captures (RTK board $\times$ NTRIP master) + one healthy baseline.

Three views per unit: u-blox .ubx (receiver), 4G-NTRIP console .log, server retranslator tcpdump.

Decoded the NTRIP-caster and ZED-F9P configuration files.

Built local tooling so this is repeatable (explorer + analyzer + dashboard).

Headline: receivers are fine — RTK is not being achieved

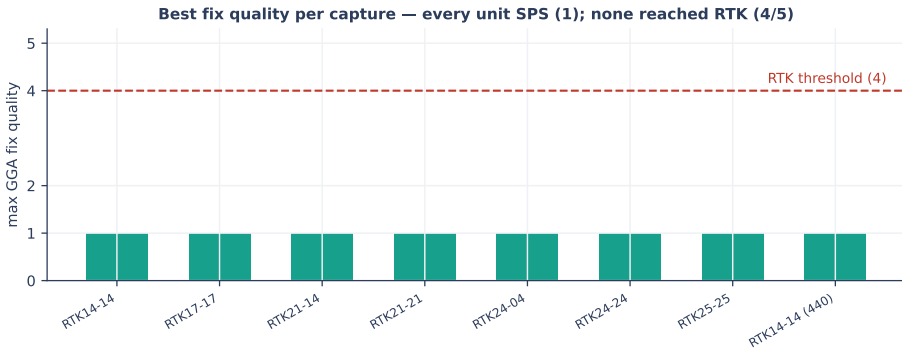
Every board emits **valid NMEA** (6–12 sats, good HDOP). The ZED-F9P is not the failure.

But **every epoch is SPS (quality 1)** – zero RTK (4/5). Centimetre position never delivered.

Why: free public **rtk2go** caster (no SLA) + F9P config saved to **RAM only** + indoor/bench sky.

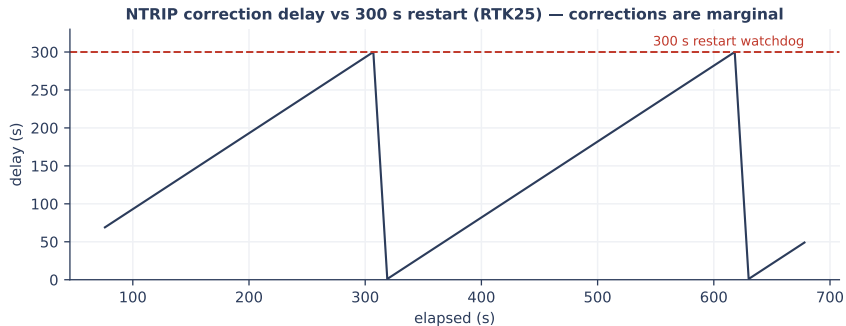
The F9P message config itself is **correct** – GGA on UART2 @ 1 Hz.

Every unit is SPS — none reached RTK



Max GGA fix quality per capture (7 bench + baseline) – all flat at 1 (SPS); the RTK line (4) is never reached.

Why: the corrections never settle



Even on a healthy unit the NTRIP correction delay sawtooths up to the 300 s restart watchdog – the free rtk2go feed is intermittent.

Healthy baseline — RTK14_440 (three views agree)

Receiver: 341 epochs, 100% valid, 12 sats, HDOP 0.75 – SPS.

Server: 984 packets, steady 1 Hz, **zero dropouts**, live in Wialon.

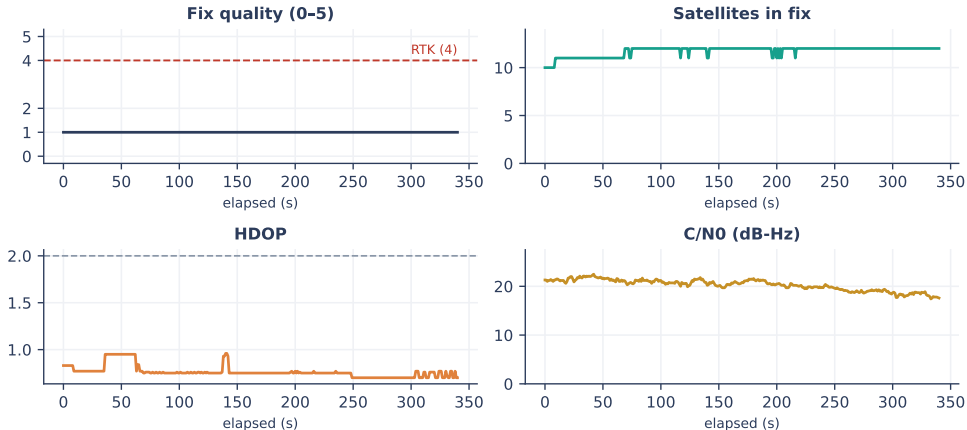
Console: “registered” only – no forwarding lines.

The trap

Healthy **delivery** $\neq$ **RTK**. Baseline = healthy SPS + reliable delivery.

Baseline receiver — healthy geometry, stuck at SPS

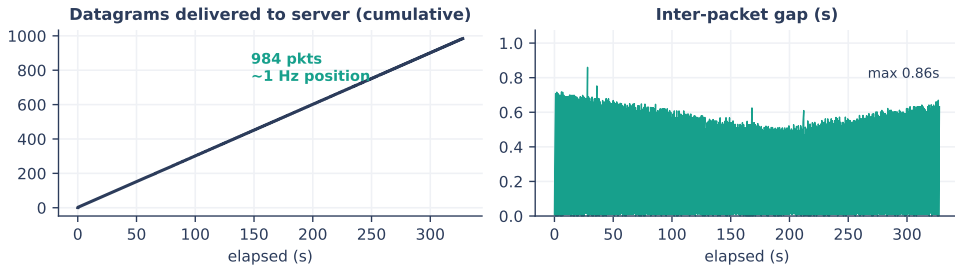
Healthy baseline RTK14_440 — receiver is fine (12 sats, HDOP 0.75) but stuck at SPS, never RTK



RTK14_440 u-blox capture: 12 sats, HDOP 0.75, C/N0 ~20 dB-Hz – but the fix never leaves 1 (SPS).

Baseline delivery — healthy end-to-end

Retranslator capture — steady, zero dropouts (healthy end-to-end delivery)



Server-side tcpdump: 984 datagrams, steady ~ 1 Hz, max gap 0.86 s – the unit is genuinely reporting (live in Wialon).

The console blind spot — and a retraction

The provably-healthy unit logs the **same sparse console** as units that looked “failed”.

⇒ the 4G-NTRIP console **cannot prove forwarding**. Absence of lines $\neq$ failure.

Retraction: the earlier “master-14 ingest fault” is **withdrawn**.

Only **master 21** is a confirmed fault: modem won't register (0xFFFFF + 511 AT-parser errors).

Per-master verdict

NTRIP master				Result
#04	#17	#24	#25	Healthy – registered + forwarding
#14				Indeterminate – console can't confirm; baseline proves it forwards
#21				Failed – modem won't register (0xFFFF + 511 AT errors)

Confirm a unit only from a .ubx (fix), the Blackpill, or a server delivery capture – never the console alone.

Heat — not implicated (by these logs)

Room-temperature bench captures, **no thermal channel** (firmware v1).

The hard faults reproduce **cold** $\Rightarrow$ heat is not the cause of what we see here.

The **field** thermal hypothesis stays open – needs a thermocouple cabinet soak and/or v2 MCU-temp logging.

Changes & tooling

The Blackpill watchdog already computes fix, packets, state, battery, MCU temp.

v2.1 logging spec: replace the self-overwriting ANSI dashboard with a **structured, timestamped, parseable** stream (+ decoded reset cause).

`hbox-log-tools` repo: in-browser Log Explorer (parse + plot + compare to baseline, offline), batch analyzer, dashboard, manual.

The new per-unit log line (v2.1)

```
BOOT,32,fw=2.1,reset=POR  
COLS,ms,state,link,fix,sat,rx,rate,vbat,v5v,veh,ign,tc,tmax,der,rec,limp,nmea_age  
TLM,190000,RUN,OK,4,12,340,1HZ,5.01,1,1,48,49,0,0,0,310  
EVT,490000,RUN,DEGR,RTK lost -> recycle 4G/NTRIP
```

Fix + packets + state + reset cause $\Rightarrow$ per-unit health is unambiguous from the unit's own log – no server tcpdump needed.

Next steps

Next steps

1. Save the F9P config to **FLASH** + verify (RAM-only is a power-cycle risk); check the simpleRTK2B Pro XBEE↔UART2 switch.
2. Move corrections off rtk2go → a **dedicated caster** (HelgenSRV) with uptime.
3. **Open-sky RTK retest** with a verified caster → confirm fix ≥ 4 .
4. Connect the **Blackpill** with v2.1 logging; modem-triage master 21; thermal soak.

**The radios work.
We deliver SPS reliably — not RTK.**

The blockers are corrections and observability — both now addressed.