

Quickstart — from box to dashboard in ~10 minutes

This gets you capturing live Zigbee traffic with names, a routing map, and diagnostics. For the full tour see the [\[User Guide\]\(user-guide.md\)](#).



What you need

- An **ESP32-C6** dev board (e.g. ESP32-C6-DevKitC-1). **Not** a C61 — that has no 802.15.4 radio.
- A **USB-C cable** and a computer (macOS/Linux/Windows, or a Raspberry Pi / Home Assistant host).
- **[PlatformIO](https://platformio.org/)** (VS Code extension) to flash the firmware, and **Go 1.25+**

to build the host — or use the prebuilt Home Assistant add-on (see step 4b).

- **(Optional but recommended)** a **ZHA backup** from Home Assistant — it carries your network

channel, PAN id, network key, and device names so everything auto-configures.

No hardware yet? Skip straight to `./zbsniff --demo` (step 3) — a fully simulated network

(a home mesh, a Philips Hue network, and a foreign neighbor network on another channel) lets you

explore every tab with no C6 required. A DEMO MODE badge stays visible the whole time so it's

never mistaken for a real capture.

1 · Flash the firmware onto the C6

Plug the C6 into USB, then:

```
cd firmware
pio run -e tethered -t upload
```

Use the board's **USB port** (the native-USB connector, not the UART one) if it has two. This same command reflashes later — the on-board OTA partition table is installed on this first flash.

2 · (Optional) Grab your ZHA backup

In Home Assistant: **Settings → Devices & Services → Zigbee Home Automation → : → Download backup.**

Save the **.json** file. It's the easiest way to get names + the network key + channel.

3 · Build & run the host

```
cd tether
go build -o zbsniff ./cmd/zbsniff
./zbsniff --zha-backup "ZHA backup ....json"      # auto: channel, PAN, key, names
```

No backup? Point it at your network manually:

```
./zbsniff --channel 20 --key 0123...ef            # channel + network key (hex, optional)
```

zbsniff **auto-detects** the C6's USB port. It also **remembers** your settings in **zbsniff.yaml**

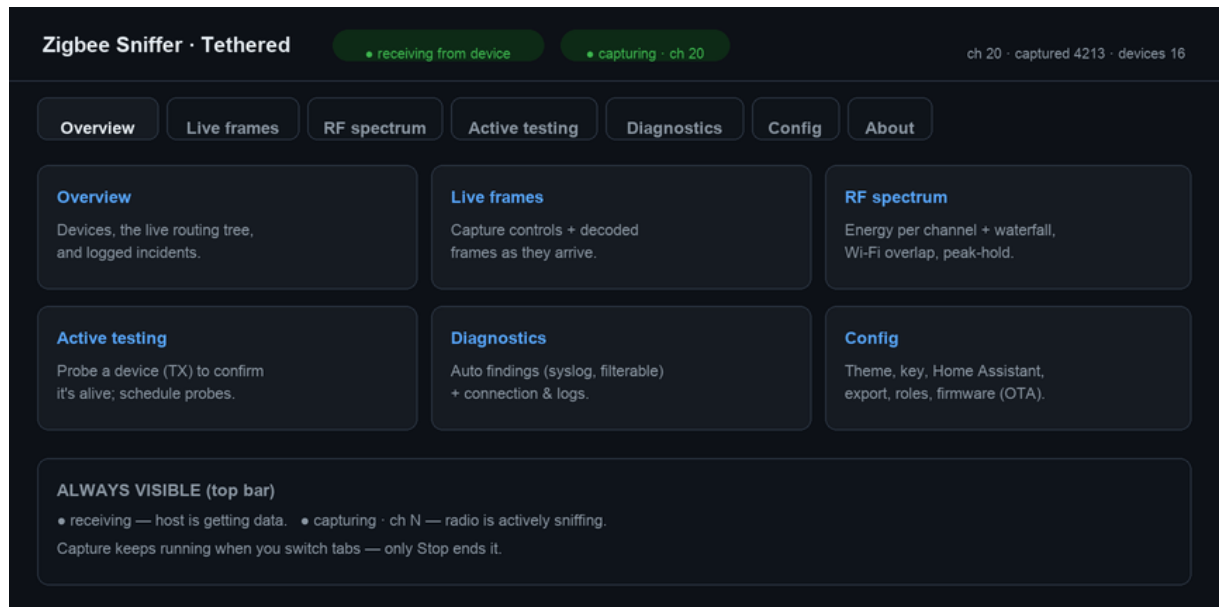
(secrets encrypted), so next time just run **./zbsniff**.

No C6 at all? Run **./zbsniff --demo** instead — no port, backup, or key needed. The same thing happens automatically (after asking first) if zbsniff can't find a C6 within a few seconds; a banner offers **Retry** or **Enter Demo Mode**, so a slow-to-enumerate real device is never silently mistaken for "no hardware."

4 · Open the dashboard

Browse to <http://localhost:8080>. Within a few seconds you should see:

- top bar: ● **receiving from device** and ● **capturing · ch N**
- **Overview** tab: devices filling in, and a routing tree growing toward the coordinator



4b · Prefer to run it inside Home Assistant?

Install it as an add-on instead of running the binary — see [\[deployment-addon.md\]\(deployment-addon.md\)](#).

The UI then appears as a sidebar panel; the C6 plugs into the HA host's USB.

5 · First things to try

1. **Overview** — click a device to open its detail drawer (signal, messages, a "Probe now" button).
2. **RF spectrum** → **Scan once** — see energy per channel and where your Wi-Fi sits.
3. **Diagnostics** — automatic findings; the **Silent device** category is your dropout radar.
4. **Config** → **Home Assistant** — enter your HA host + a long-lived token to get friendly names and

the network's own **Net LQI** (see Two signal readings below).

Two signal readings (important!)



The **RSSI/Qual** † columns are what **your sniffer** hears from one spot — **not** the device's link to its router. A device can look "weak" to the sniffer yet be perfectly healthy on the mesh. Connect

Home Assistant to see **Net LQI** (the coordinator's real view), or use **Active testing** to probe from a radio placed near the device.

If you see "host up — no device data"

If a **no hardware found** banner offers Retry/Demo Mode instead, your C6 just hasn't enumerated

yet — try **Retry** first. Otherwise, open **Diagnostics → Connection & logs**:

- **0 bytes** → wrong USB connector (use the native-USB port), or another **zbsniff** is holding the port. Hit ↺ **Reconnect**.

- **bytes but 0 frames** → firmware/protocol mismatch — reflash (step 1).
- **captured stops climbing** → reflash; older firmware had an RX-buffer stall (fixed).

More in [\[troubleshooting.md\]\(troubleshooting.md\)](#) and the [\[User Guide\]\(user-guide.md\)](#).