mbs_ridgeback

Ridgeback R100 — unit `r100_0160`
Current Software Manual (robot_webserver framework)

MYBOTSHOP GmbH · ROS 2 Jazzy · source: github.com/MYBOTSHOP/mbs_ridgeback

mbs_ridgeback

ROS 2 Jazzy workspace for the Clearpath Ridgeback R100 (unit `r100_0160`), running the shared `robot_webserver` framework (same base as the A2/G1 fleet) instead of the older bespoke `mmp_webserver` this unit shipped with. This is the current, actively-developed repo for this unit — see Migration note for how it relates to the older Ridgeback repos in the org.

Table of Contents

Connection

Network Table

Device	Address	Username	Password
MMP Computer (SSH)	`192.168.131.1` (LAN) / `100.64.0.32` (Tailscale)	`robot`	`clearpath`
MMP Computer (Webserver)	`192.168.131.1:9000` / `100.64.0.32:9000`	`admin`	`mybotshop`
MMP Computer (Cockpit / systemd web UI)	`192.168.131.1:9090`	`robot`	`clearpath`
MMP Ridgeback MCU	`192.168.131.2`	—	—
Hokuyo Lidar Front	`192.168.131.20`	—	—
Hokuyo Lidar Rear	`192.168.131.21`	—	—
Ouster Lidar	`192.168.131.22`	—	—
Left xArm850 (unused on this unit)	`192.168.131.5:18333`	—	—
Right xArm850 (unused on this unit)	`192.168.131.6:18333`	—	—
Access Point	`192.168.131.200`	SSID	`mybotshop`

> **NOTE**: This unit is enrolled directly in the fleet Tailscale VPN (`fleet.eurobotix.de`) as `100.64.0.32` — reachable from anywhere without a jump host, in addition to the local `192.168.131.1` LAN address.

SSH

```
ssh robot@192.168.131.1     # LAN
ssh robot@100.64.0.32       # Tailscale, from anywhere
```

Password `clearpath` for both login and `sudo`.

ROS 2 environment:

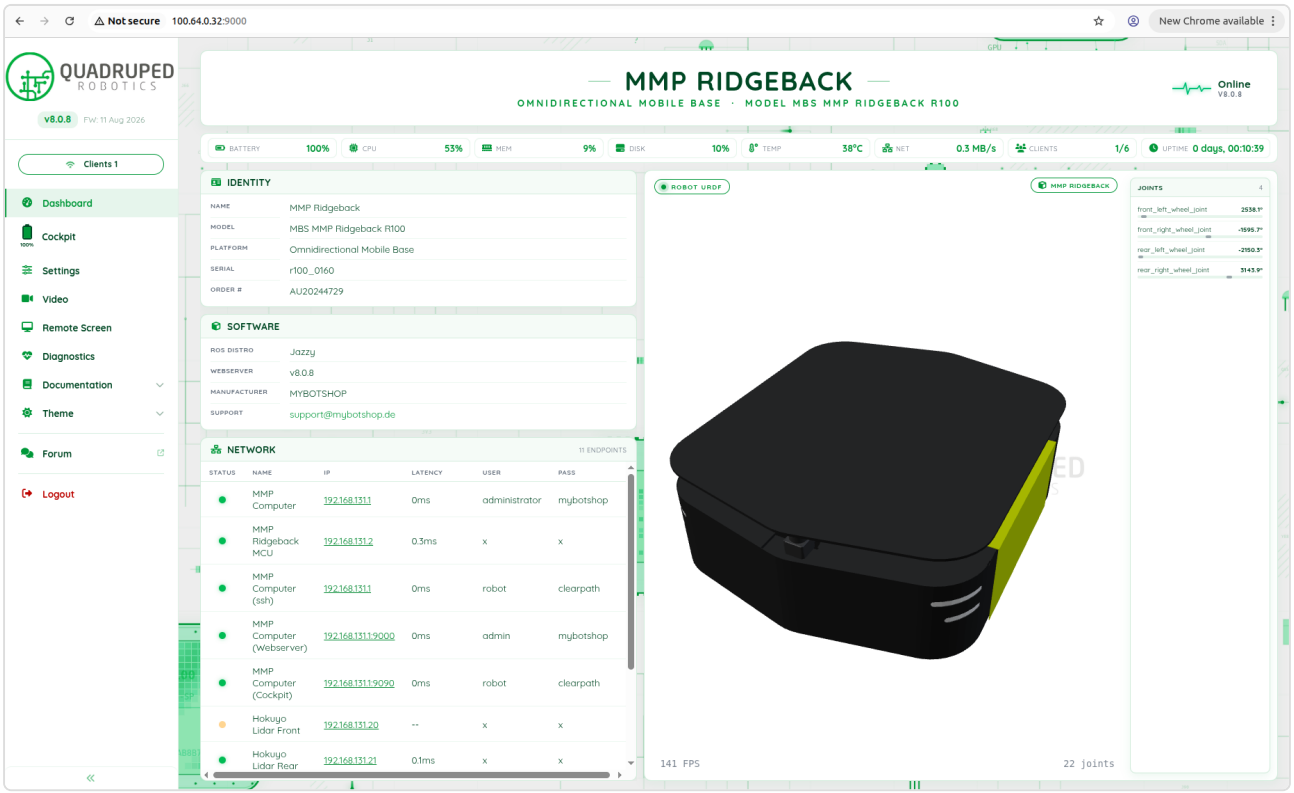
```
source /opt/ros/jazzy/setup.bash
source /opt/mybotshop/install/setup.bash
export ROS_DOMAIN_ID=0
```

Web Interface

Browse to `http://192.168.131.1:9000` (LAN) or `http://100.64.0.32:9000` (Tailscale), login `admin` / `mybotshop`.

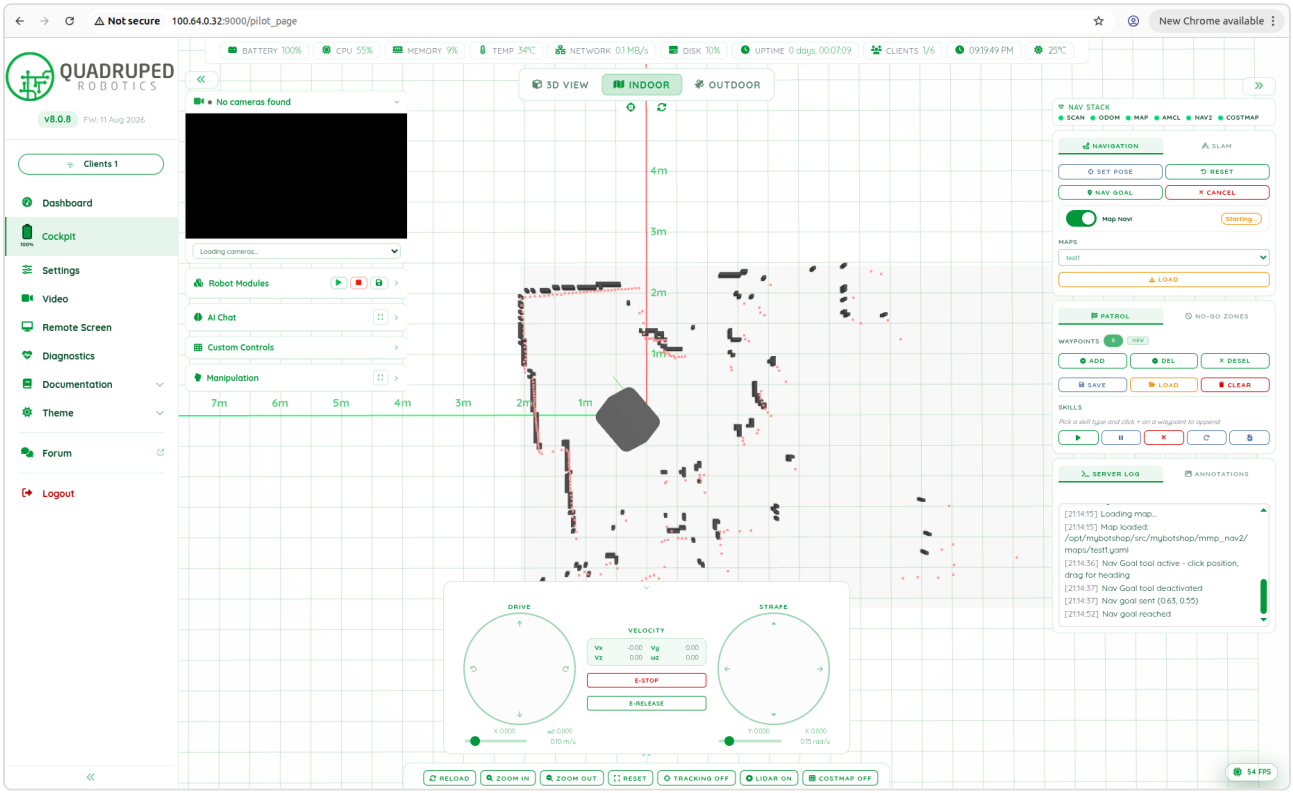
Dashboard

Identity, software versions, live URDF preview with joint angles, and the full network table shown above (also editable from here).



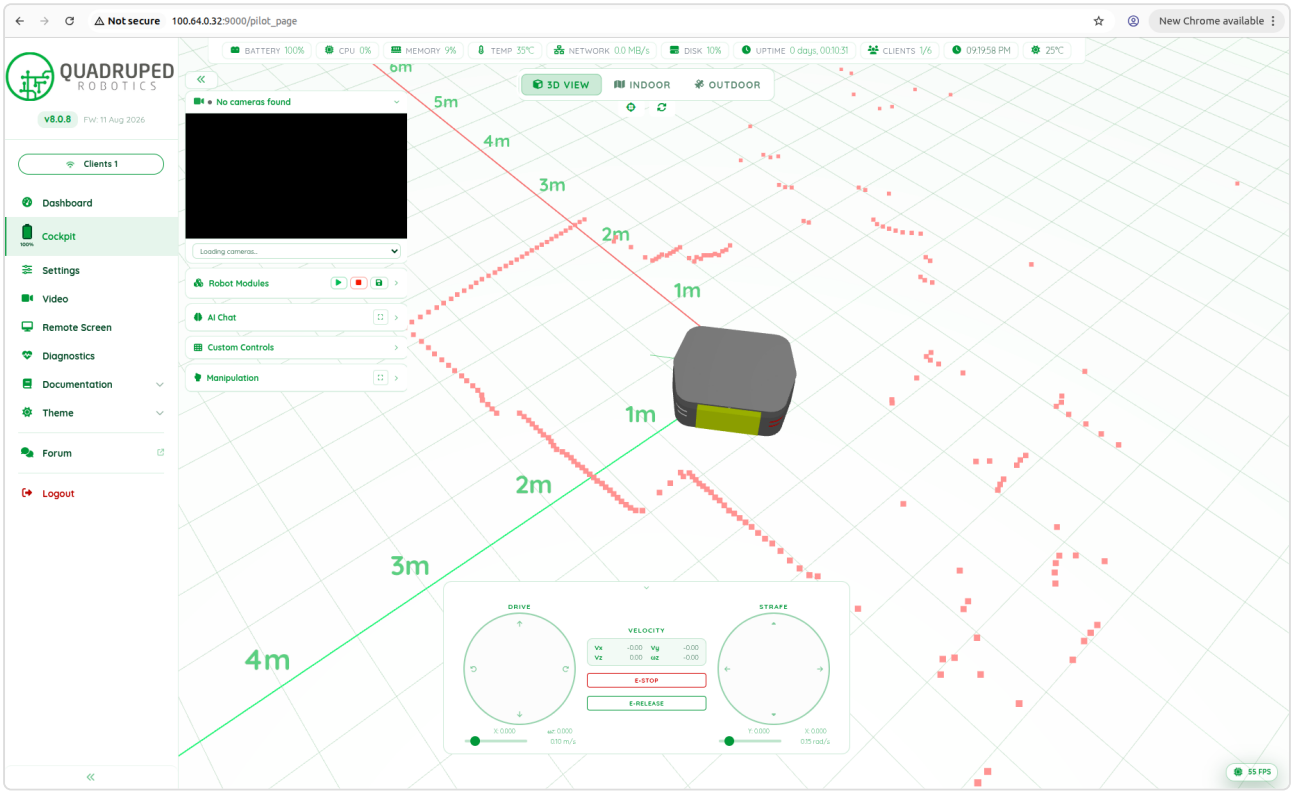
Cockpit — Indoor Navigation

2D map view with the Nav Stack sidebar: live SLAM/AMCL/Nav2/costmap status indicators, Navigation vs. SLAM mode toggle, map load/save, patrol waypoints, no-go zones, and a live server log. The screenshot below shows a loaded map (`test1`) and a nav goal that was sent and successfully reached.



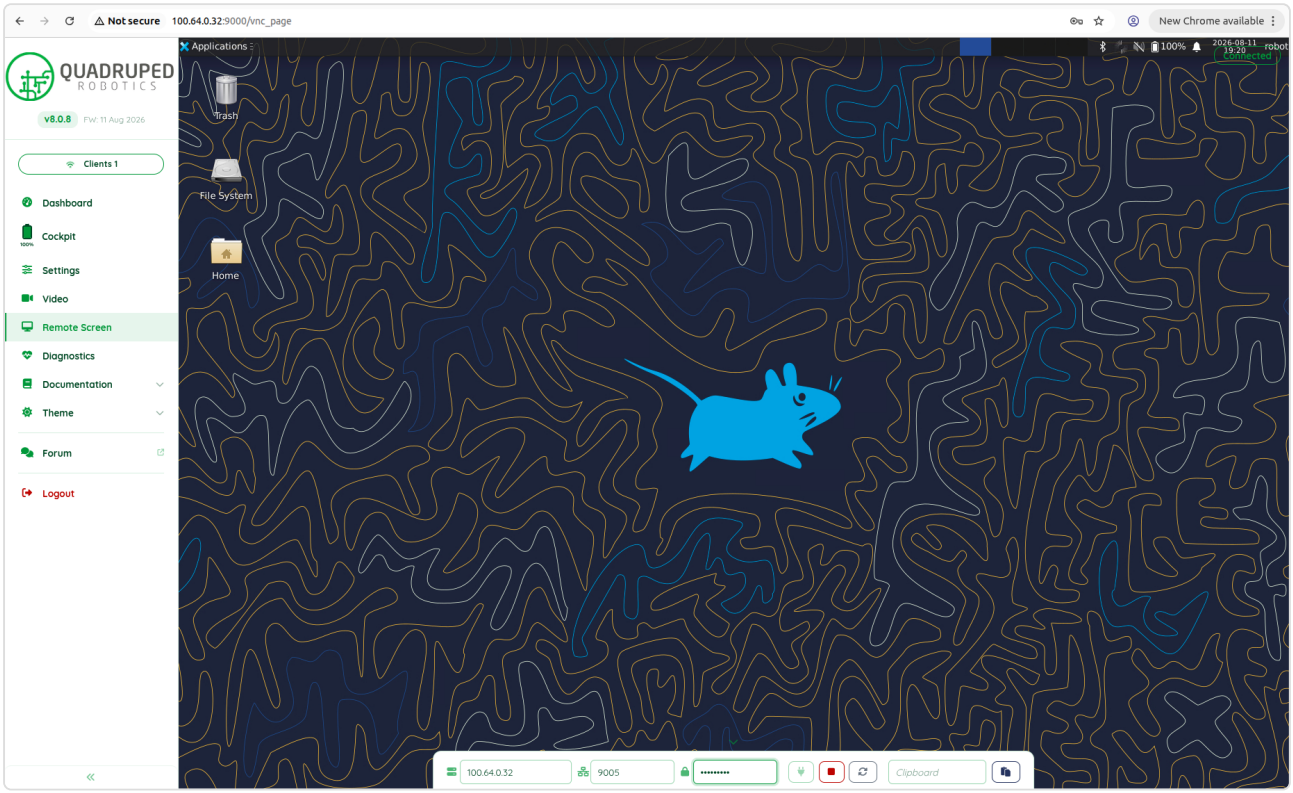
Cockpit — 3D View + Lidar

Live 3D view of the robot with the merged front+rear lidar scan overlaid, drive/strafe joystick widgets, and live velocity readout.



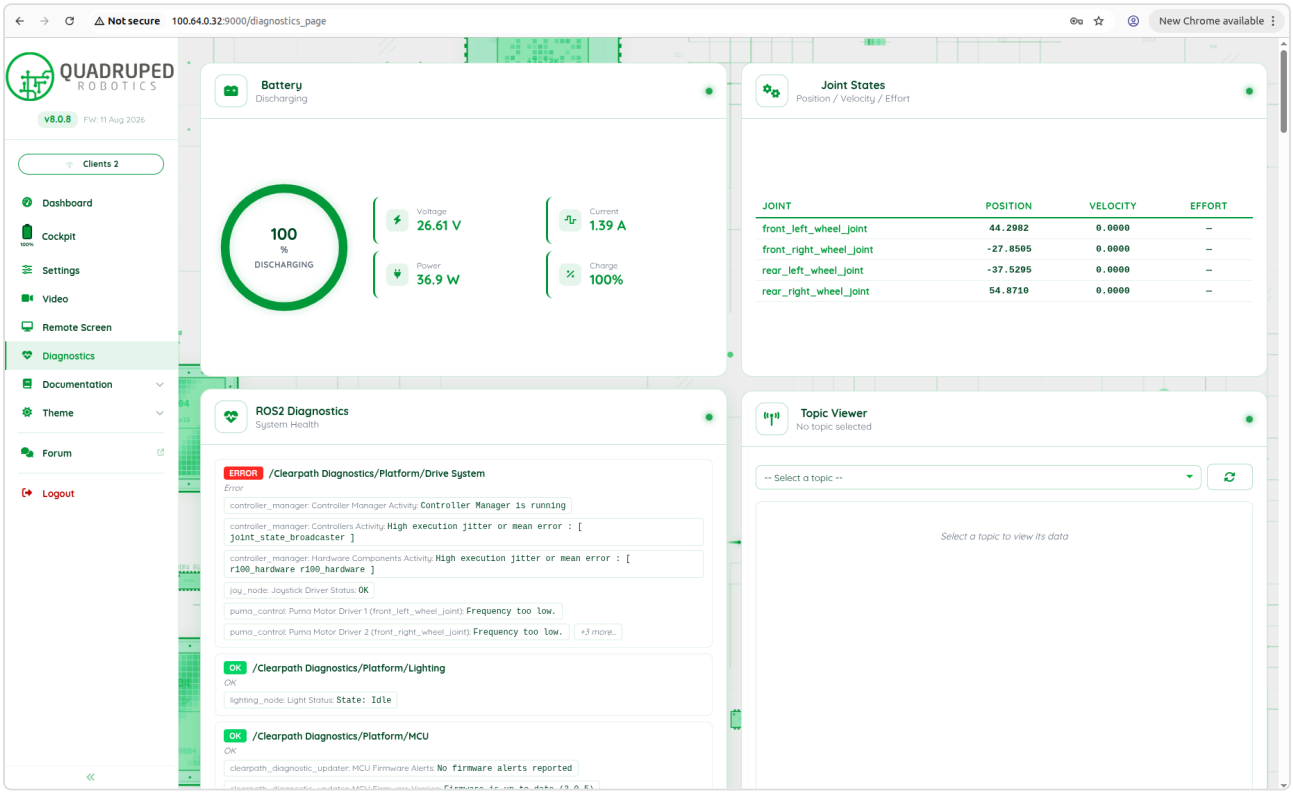
Remote Screen (VNC)

Embedded noVNC session into the onboard XFCE desktop (`websockify` bridging port `9005` to the internal VNC display `:1`) — full host GUI access without a second keyboard/monitor.



Diagnostics

Battery state, per-wheel joint states (position/velocity/effort), the full Clearpath ROS2 diagnostics aggregator tree, and a live topic viewer.



> **NOTE**: The Drive System entry here has shown intermittent `ERROR`/"High execution jitter" / "Frequency too low" readings on the Puma motor drivers and CAN bus — see [Known operational notes](#known-operational-notes).

Other pages

- **Settings** — per-module start/stop, config editor for `robot_webserver.yaml` / `robot_info.yaml`
- **Video** — camera tile selector (no camera mounted on this unit yet)
- **Documentation** — embedded PDF manual viewer, served from the robot itself
- **Theme** — light/dark/accent color switcher

Services

Beyond Clearpath's own stock units (`clearpath-robot` , `clearpath-platform` , `clearpath-sensors` , `clearpath-vcan`):

Service	Purpose	Autostart
`mbs-webserver`	`robot_webserver` dashboard (this repo's `src/mybotshop/robot_webserver`)	yes
`clearpath-scan-merger`	merges front+rear 2D Hokuyo lidars into `sensors/scan`	yes
`clearpath-joy-x-combiner`	combines F710 dual-stick input into a single `linear.x` (see Known operational notes)	yes
`robot-slam`	SLAM mapping mode (`slam_toolbox`)	no — pilot page toggle
`robot-map-navi`	Nav2 + AMCL against a saved map	no — pilot page toggle

Startup order

```
clearpath-robot.service                    (ExecStartPre generates /etc/clearpath
config, then bringup check)
  ├─ clearpath-vcan.service                (virtual CAN bridge)
  ├─ clearpath-platform.service            (ros2_control, twist_mux, joy_node,
teleop, ekf_node, ...)
  │    └─ [drop-in] forces F710 device path before every start (see below)
  ├─ clearpath-sensors.service             (lidars, IMU)
  │    └─ clearpath-scan-merger.service    (needs the raw lidar topics from
clearpath-sensors)
  │         ├─ robot-slam.service          (manual — needs merged scan)
  │         └─ robot-map-navi.service      (manual — needs merged scan)
  └─ clearpath-joy-x-combiner.service      (needs joy_teleop/joy from clearpath-
platform)

mbs-webserver.service                      (independent — only needs clearpath-robot
+ network)
```

`robot-slam` and `robot-map-navi` both depend on `clearpath-scan-merger` being up, but not on each other — see below for why only one should run at a time.

Nav modes are mutually exclusive

`robot-slam` (mapping) and `robot-map-navi` (navigating a saved map) are alternative operating modes for the same lidar data, not something to run together. The pilot page's Navigation/SLAM sidebar tabs enforce this: toggling one on stops the other first (client-side mutex in `navi_indoor_script.js` / `NAV_EXCLUSIVE_SERVICES`), calling the webserver's `/reset_component` and `/deactivate_component` endpoints, which map 1:1 to `systemctl start|stop <service>`. Starting either service directly via `systemctl` on the robot works the same way but bypasses the mutex, so stop the other one manually first.

`robot-map-navi` needs AMCL to have a pose estimate before its Nav2 stack activates — `nav2_map.yaml` hardcodes a default initial pose at the map origin (`set_initial_pose: true`) so this doesn't require a manual "Set Pose" click after every fresh start, but a real "Set Pose" click for accuracy is still a good idea once the robot is actually near the map's origin.

Repo layout

- `src/mybotshop/robot_webserver` — web dashboard, REST/websocket API, pilot controls
- `src/mybotshop/robot_interface` / `a2_interface` — message/service/action definitions the webserver depends on (inherited from the shared framework; robot-specific services like `SetLocomotion` / `PlayEmoji` are unused on a wheeled base)
- `src/mybotshop/mmp_description` — bridges Clearpath's generated `/etc/clearpath/robot.urdf.xacro` into an ament package share dir for `/api/urdf`
- `src/mybotshop/mmp_nav2` — SLAM / map navigation / odom navigation launch files and params
- `src/mybotshop/mmp_interface` — legacy interface package
- `src/third_party/tools` — vendored third-party tools (`ros2_laser_scan_merger`)

Known operational notes

- **F710 device path reverts at boot.** Clearpath's own config tooling regenerates `teleop_joy.yaml` at every boot, reverting `dev:` back to `/dev/input/ps4`. A systemd drop-in (`/etc/systemd/system/clearpath-platform.service.d/override.conf`) forces it back to `/dev/input/f710` via `ExecStartPre` before every start of `clearpath-platform.service`, regardless of what generated the file.
- **`teleop_twist_joy` can only read one physical axis per twist field**, so it can't combine both sticks' vertical axes into `linear.x`. The stock node is left running (harmless, just unused) and `clearpath-joy-x-combiner` (`src/mybotshop/robot_webserver`-adjacent script at `/opt/mybotshop/scripts/joy_x_combiner.py`, not yet folded into a proper colcon package) reads `/joy` directly, sums both sticks' vertical axes into `linear.x`, and publishes to `joy_teleop/cmd_vel_combined` — `twist_mux`'s `joy` input is repointed at that topic instead of the stock node's output. It only publishes while the deadman button is held, so `twist_mux`'s priority slot correctly times out (0.5s) and yields to lower-priority sources (like Nav2) when nobody is actively driving.
- **`/nav2/scan` (lidar overlay) falls back to the `odom` frame.** It's normally transformed into the `map` frame, which only exists once SLAM or Map Navi is actively publishing it. `scan.py` now falls back to `odom` (always available via `clearpath-platform`'s `ekf_node`) so the lidar overlay works with both nav modes off.
- **Recurring CAN/drive-system fault.** The Puma motor drivers and `r100_hardware` ros2_control interface have intermittently faulted (`CAN Receive Timeout` on `vcan0`, drive system stuck) several times, usually clearing with a full reboot but not always. This looks like a physical connection issue (a `clearpath-vcan.service` restart alone doesn't always clear it) rather than anything software here can reliably fix — if it recurs often, check the CAN wiring/connectors on the platform.

- Config edits under `/etc/clearpath/platform/config/` (e.g. `twist_mux.yaml`, `teleop_joy.yaml`) can be wiped by Clearpath's own regeneration at boot — check for `.bak-*` backups left alongside them before assuming a change is still in effect after a reboot.

Migration note

The org already has several older Ridgeback repos (`mbs_mmp_ridgeback`, `mbs_ridgeback_r100`, `mmp_ridgeback_xarm`, `ridgeback_xarm_twi`, `mmp_ridgeback`), all built around the older bespoke `mmp_webserver` and not actively maintained. This repo represents the current, in-use setup on `r100_0160` after migrating to the shared `robot_webserver` framework — it does not include `mmp_webserver` itself, which remains on disk on the robot (`/opt/mybotshop/src/mybotshop/mmp_webserver`, not started) as a rollback path but isn't part of this repo.