RB2

Designer: Roméo St-Cyr

— Technical Summary

The **RB2**, or **RomBot2**, is a mobile inspection robot designed to explore confined spaces, particularly crawl spaces. Its main goal is reliable teleoperation, with camera, LIDAR, odometry, manual mapping, and the ability to return or recover in case of lost communication.

1. General Architecture

The RB2 is built on a multi-controller architecture:

Component	Role
Main Raspberry Pi 5	Web server, control interface, ROS 2, LIDAR, mapping, main logic
Raspberry Pi Zero 2 W	Independent camera and fallback mode
ESP32	Low-level control of motors, relays, pan/tilt servos, and serial commands
BNO055	RB2's current IMU
LD19	360° LIDAR for distance and mapping
Laptop	Piloting station via web browser

The RB2 uses a coordinated set: **Pi 5 + Zero 2 W + ESP32 + sensors**.

2. Main Computer

The main controller is a **Raspberry Pi 5**

It manages:

- the RB2's web interface;
- the `/terrain4` pages;
- the LIDAR data streams;
- communication with the ESP32;
- systemd services;
- ROS 2 Jazzy;
- odometry;
- mapping;
- automatic recovery on startup.

System used:

```
Ubuntu 24.04 LTS
ROS 2 Jazzy
Network name: rb2-ros
```

Main AP address:

```
10.42.0.1
```

Main web interface:

```
http://10.42.0.1:5051/terrain4
```

---

3. Independent Camera

The RB2 uses a **Raspberry Pi Zero 2 W** as a separate camera module.

Camera:

```
Raspberry Pi Camera Module 3 Wide / imx708_wide
```

The Zero 2 W is connected to the Pi 5 via direct Ethernet.

Addresses:

Component	Address
Pi 5, Ethernet side	`10.50.0.1`
Zero 2 W camera	`10.50.0.20`

Direct camera stream:

```
http://10.50.0.20:8080/stream.mjpg
```

Stream relayed via the Pi 5:

```
/zero_cam_stream.mjpg
```

The Zero 2 W also serves as the basis for the **fallback mode**, with its own access point in case the normal access methods are lost.

On July 17, the image (previously 4:3 and noisy) was fixed to 16:9, clear. On July 24, video lag/delay was also reduced.

---

4. Fallback Mode

The RB2 has an independent recovery mode via the Zero 2 W.

Fallback access point:

`RB2-ZERO-SECOURS`

Address:

`10.60.0.1`

Fallback mode functions:

- camera access;
- fallback web page;
- route to the Pi 5 via Ethernet;
- recovery possible if the main Wi-Fi or the normal interface no longer responds.

Intended principle: the Zero should not normally reconnect to the local Wi-Fi. It should stay neutral, then activate the fallback only if needed.

---

5. Onboard ESP32 Controller

The ESP32 handles low-level control.

It controls:

- the motors;
- the BTS7960 bridges;
- the pan/tilt servos;
- the relays;
- certain direct safety functions;
- serial commands from the Pi 5.

Observed serial port:

`CH340 → /dev/ttyUSB0`

The Pi 5 communicates with the ESP32 via a USB serial link.

---

6. Motorization

The RB2 uses four DC motors with encoders.

Known characteristics:

Item	Value
Motors	4 × JGA25
Rated speed	35 rpm
Encoders	yes
Motor bridges	2 × BTS7960
Wheels	about 85 mm
Track width	about 225 mm
Ticks/rev used	10529

Motors are controlled by usable speed levels:

Speed button	Effective command
25%	`d15`
50%	`d17`
75%	`d19`
100%	`d20`

The RB2 has a **central STOP**, essential for safety.

---

7. Power Supply

Main power is based on a battery:

```
DeWalt 20V / 5Ah
```

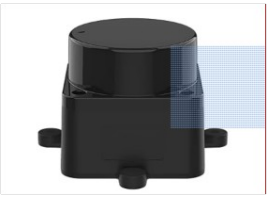
Typical distribution:

Voltage	Use
20 V	main battery
12 V	motors
5 V	Raspberry Pi, servos, electronics, light

The RB2 therefore uses separate distribution to prevent the motors from strongly disturbing the computers and sensors.

---

8. LIDAR



The main distance sensor is an `LD19`

Usage characteristics:

Item	Value
Type	360° LIDAR
Frequency	about 10 Hz
Installed height	about 24 cm
Front position from robot center	about 28 cm
Rear position from robot center	about 8 cm
Lateral width used	about 10 cm on each side

Data produced:

```
/tmp/rb2_lidar.json
/tmp/rb2_lidar_map.json
```

Related services:

```
rb2-lidar-json.service
rb2-lidar-map-json.service
rb2-lidar-watchdog.service
```

The LIDAR is used to:

- display front/back/left/right distances;
- produce a local map;
- protect the motors;
- assist with return or replay;
- contribute to SLAM.

The LIDAR watchdog, added at the end of June, caused reboot loops that cut off wheel response; it was made more patient before reacting. A bug related to its internal restarts was also fixed on July 21.

---

9. Current IMU

The current IMU is a `BNO055`

I²C address:

`0x28`

Service used:

`rb2-imu-json.service`

File produced:

`/tmp/rb2_imu.json`

---

10. ROS 2

The RB2 uses `ROS 2 Jazzy`

Main ROS elements:

```
/scan
/odom
/tf
/map
/map_metadata
/rb2_odometry
/rb2_wheel_ticks_json
/ldlidar_publisher_ld19
```

---

11. Mapping and SLAM

The RB2 uses SLAM Toolbox for manual mapping.

Service:

`rb2-slam.service`

Available functions:

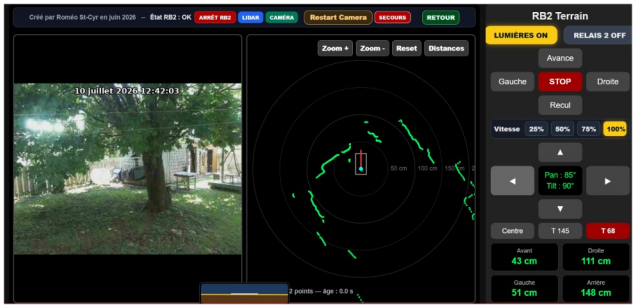
```
/slam_start
/slam_stop
/slam_status
```

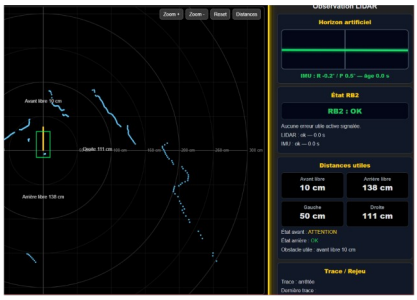
Data used:

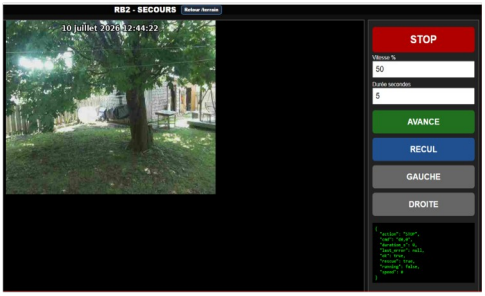
- LIDAR `/scan`;
- odometry `/odom`;
- transforms `/tf`;
- map `/map`.

Maps can be saved with:

```
ros2 run nav2_map_server map_saver_cli -f ~/rb2_maps/map_name -t /map
```







12. Web Interface

The RB2 has several web pages.

Page	Role
`/terrain`	old terrain page, replaced (do not use anymore)
`/terrain4`	current main driving page
`/secours`	direct recovery page
`/lidar_drive2`	LIDAR, SLAM, and trace tracking/return (button back to /terrain4)
`/zero_cam_stream.mjpg`	Zero camera relay
`/captures`	gallery of photos and videos

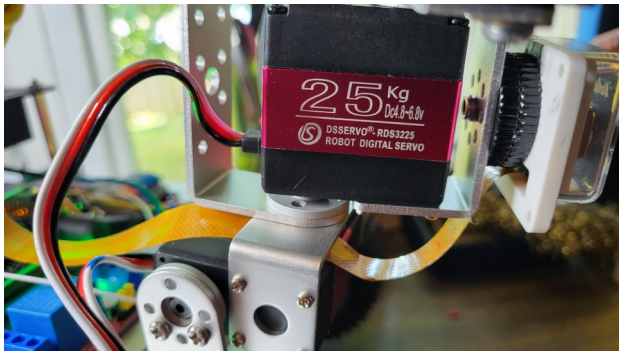
Page	Role
	taken

The most recent and functional page is:

`/terrain4`

`It includes:`

- camera on the left;
- LIDAR map on the right;
- motor controls;
- pan/tilt;
- 25/50/75/100% speeds;
- front/right/back/left distance buttons;
- light;
- relay;
- second relay;
- Wi-Fi 5G toggle (the 2.4G button was removed on July 24);
- SSID and DISCONNECT buttons, to join then leave an on-site client's Wi-Fi (since July 25);
- full robot SHUTDOWN (Pi 5 + Zero);
- photo/video commands;
- motor STOP.



13. Pan/Tilt

The mobile camera system uses two servos `DS3225`

Control:

Function	ESP32 GPIO
PAN	GPIO 19
TILT	GPIO 18

Limits used:

Axis	Limits
TILT	about 68° to 145°
PAN	about 60° to 135°, depending on TILT position

These limits are important to avoid mechanically forcing the servos or camera.

14. Network

The RB2 is designed to operate on a standalone network.

Recommended field mode:

```
Laptop connected directly to the RB2's AP
```

Main access point:

```
RB2-ROS
10.42.0.1
```

The local Wi-Fi can be disabled to avoid losses caused by the home network.

Important AP interface:

```
wlx00c0cabaa826
```

---

15. Important systemd Services

Main services:

```
rb2-bringup.service
rb2-esp32-panel.service
rb2-lidar-json.service
rb2-lidar-map-json.service
rb2-lidar-watchdog.service
rb2-imu-json.service
rb2-slam.service
rb2-startup-healthfix.service
```

The `rb2-startup-healthfix.service` service corrects the startup order and avoids having to manually restart the panel before each use.

---

16. Safety and Recovery

The RB2 has several levels of safety:

- software motor STOP;
- STOP button in the interface;
- page `/secours`;
- LIDAR watchdog;
- cleanup of old JSON files;
- ability to restart services individually;
- standalone AP mode;
- Zero 2 W fallback mode.

Known ongoing issues: RB2 can occasionally reconnect to a weaker Wi-Fi access point during a mission (a more thorough fix was tried, then paused); a wheel can loosen (lock-washer mitigation applied in late July, not yet confirmed effective long-term).

Very Short Summary

The **RB2** is a mobile inspection robot based on a **Raspberry Pi 5**, a **Raspberry Pi Zero 2 W**, an **ESP32**, a **LIDAR LD19**, a **BNO055 IMU**, four encoder motors, and a teleoperation web interface. It operates on a standalone access-point network, with an independent camera, 360° LIDAR, odometry, ROS 2 Jazzy mapping, fallback mode, and automatic recovery mechanisms.