Mini ROS Rover Bot

This is my first step into learning ROS — and no, it's not just another toy car. Building a mini rover is one of the most practical starting points for robotics and ROS because it brings together hardware control, movement planning, and real-world measurements in a tangible way.

Components Used

- **ESP32 Development Board**
- **MX1508 Dual Motor Driver**
- **2 × N20 DC Gear Motors (with gearbox)**
- **Lithium-ion battery pack** (with BMS)
- **3.3V voltage regulator** (for powering ESP32)
- **TP4056 charging module**
- **On/Off switch**
- **LED indicator**
- **Wires, connectors, and chassis**

What I Did

Instead of hardcoding guesses for movement, I measured actual motor RPM, wheel diameter, and gear ratio to calculate the distance to time relationship for my rover. That way, when I say "move forward 400 mm," it's based on practical measurements, not just theory.

The bot runs in specified directions using programmed sequences forward, left turn, right turn with durations tuned from these real-world measurements.

This gives me a repeatable and predictable movement system, which is essential when I move into ROS for tasks like mapping, localization, and autonomous navigation.

Why This Matters for ROS

Learning ROS isn't just about installing packages and running simulations it's about understanding how robots move in the real world. This project is my foundation:

- **Hardware understanding** – controlling motors, power management, pin mapping.
- **Movement calibration** – translating physical parameters into commands.
- **Structured coding** – preparing for ROS topics, services, and action clients.

From here, I'll start integrating sensors, odometry, and eventually ROS nodes to control the rover.

Future Plans

I plan to add a **mini display** for live status output, replace the current ESP32 with an **ESP32-CAM** to work with **OpenCV** for computer vision, integrate an **ultrasonic sensor** for obstacle detection, and continue expanding capabilities as my ROS learning progresses.

CONNECTIONS:

ESP32 → MX1508

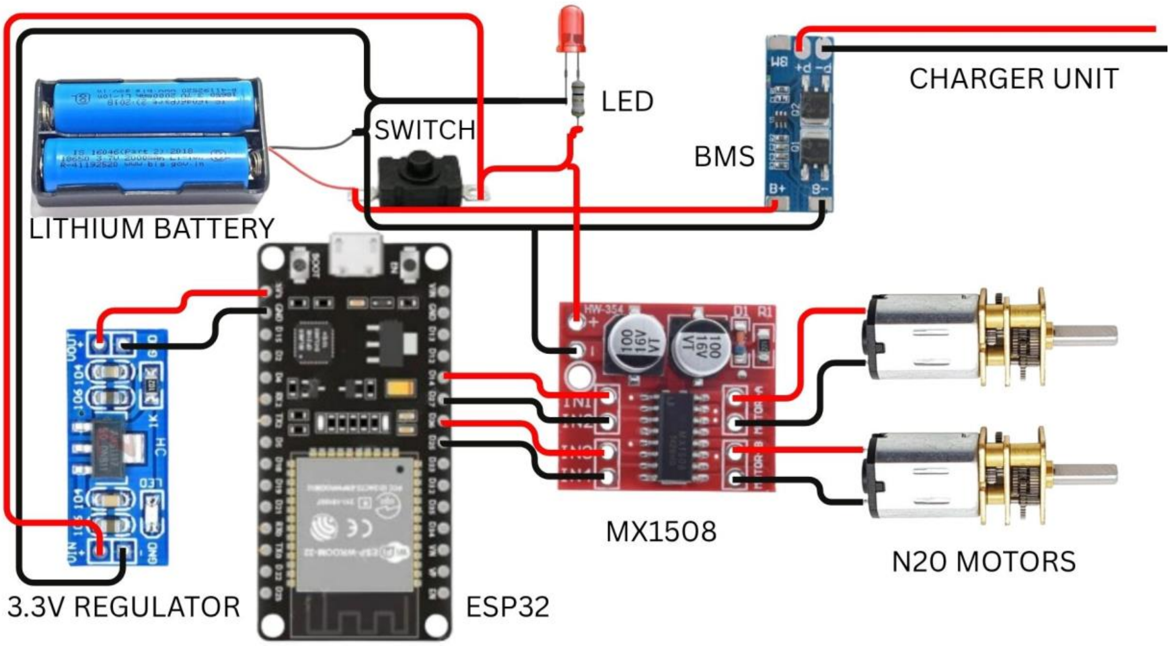
- GPIO14 → IN1
- GPIO27 → IN2
- GPIO26 → IN3
- GPIO25 → IN4
- GND → GND

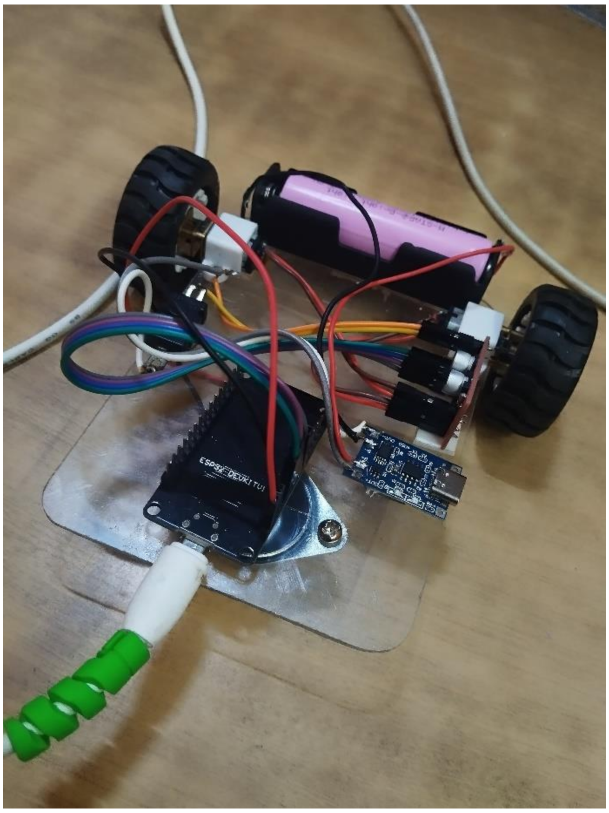
MX1508 → Motors

- OUT1 → Motor A terminal 1
- OUT2 → Motor A terminal 2
- OUT3 → Motor B terminal 1
- OUT4 → Motor B terminal 2

Power

- Battery + (through switch & BMS) → MX1508 VCC
- Battery – → MX1508 GND
- Battery + (through switch & BMS) → 3.3V regulator VIN
- Battery – → 3.3V regulator GND
- Regulator 3.3V → ESP32 3V3 pin
- Regulator GND → ESP32 GND




ESP32_DEVKITV1